Jörg Cassens

Explanation Awareness and Ambient Intelligence as Social Technologies

Thesis for the degree doctor scientiarum

Trondheim, May 2008

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Computer and Information Science

NTNU

NTNU
Norwegian University of Science and Technology

Thesis for the degree doctor scientiarum

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Computer and Information Science

ISBN 978-82-471-8396-0 (printed version)
ISBN 978-82-471-8401-1 (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, 2008:119

Printed by NTNU-trykk

For Johann Engelbart Cassens

Abstract

This work focuses on the socio-technical aspects of artificial intelligence, namely how (specific types of) intelligent systems function in human workplace environments. The goal is first to get a better understanding of human needs and expectations when it comes to interaction with intelligent systems, and then to make use of the understanding gained in the process of designing and implementing such systems.

The work presented focusses on a specific problem in developing intelligent systems, namely how the artefacts to be developed can fit smoothly into existing socio-cultural settings. To achieve this, we make use of theories from the fields of organisational psychology, sociology, and linguistics. This is in line with approaches commonly found in AI. However, most of the existing work deals with individual aspects, like how to mimic the behaviour or emulate methods of reasoning found in humans, whereas our work centers around the social aspect. Therefore, we base our work on theories that have not yet gained much attention in intelligent systems design. To be able to make them fruitful for intelligent systems research and development, we have to adapt them to the specific settings, and we have to transform them to suit the practical problems at hand.

The specific theoretical frameworks we draw on are first and foremost activity theory and to a lesser degree semiotics. Activity theory builds on the works of Leont'ev. It is a descriptive tool to help understand the unity of consciousness and activity. Its focus lies on individual and collective work practise. One of its strengths, and the primary reason for its value in AI development, is the ability to identify the role of material artefacts in the work process. Halliday's systemic functional theory of language (SFL) is a social semiotic theory that sets out from the assumption that humans are social beings that are inclined to interact and that this interaction is inherently multimodal. We interact not just with each other, but with our own constructions and with our natural world. These are all different forms of interaction, but they are all sign processes.

Due to the obvious time and spatial constraints, we cannot address all of the challenges that we face when building intelligent artefacts. In reducing the scope of the thesis, we have focused on the problem of explanation, and here in particular the problem of explanation from a user perspective. By putting social theories to work in the field of artificial intelligence, we show that results from other fields can be beneficial in understanding what explanatory capabilities are needed for a given intelligent system, and to ascertain in which situations an explanation should be delivered. Besides lessons learned in knowledge based system development, the most important input comes from activity theory.

The second focus is the challenge of contextualisation. Here we show that work in other scientific fields can be put to use in the development of context aware or ambient intelligent systems. Again, we draw on results from activity theory and combine this with insights from semiotics.

Explanations are themselves contextual, so the third challenge is to explore the space spanned by the two dimensions *ability to explain* and *contextualisation*. Again, activity theory is beneficial in resolving this issue.

The different theoretical considerations have also led to some practical approaches. Working with activity theory helps to better understand what the relevant contextual aspects of a given application are and helps to develop models of context which are both grounded in the tradition of context aware systems design and are plausible from a cognitive point of view.

Insights from an analysis of research in the knowledge based system area and activity theory have further lead to the amendment of a toolbox for requirements engineering, so called problem frames. New problem frames that target explanation aware ambient intelligent systems are presented. This is supplemented with work looking at the design of an actual system after the requirements have been elicited and specified. Thus, the socio-technical perspective on explanations is coupled with work that addresses knowledge representation issues, namely how to model sufficient knowledge to be able to deliver explanations.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in partial fulfilment of the requirements for a *Doctor Scientiarum*. It is organised as a collection of papers, with a research overview, consisting of an introduction, a research description, and conclusions, given in the first part. The articles, following the originally published text, can be found in the second part. The work has been conducted at the Department of Computer and Information Science (IDI), under supervision of Professor Agnar Aamodt.

Thanks

> "Since every new finding is formed on the basis of the potential of the already accumulated common knowledge, a merit which can be separated and alone accredited to a particular individual and thereafter even translated into special gratification is not determinable." [Ruschig, 1995, p. 10]

Reseach is not the solitary process of a single person. It always builds on the existing body of knowledge, and ideally it evolves in close collaboration with other researchers. I have had the pleasure over the last couple of years to meet some brilliant researchers, and I would like to thank all of them for their inspiration.

In person, I would first and foremost like to thank my closest collaborators, my supervisor and the co-authors of the articles presented in this collection. Agnar Aamodt for his supervision and having the patience to let me find my own way. Anders Kofod-Petersen for targeting my interest towards ambient intelligence, sending some "fill in the blanks", and spending numerous hours in front of different whiteboards. Thomas Roth-Berghofer for our discussions about yet to be written papers on explanation. Frode Sørmo for starting the work on explanation goals with me. Rebekah Wegener for teaching me the foundations of systemic-functional linguistics and relating it to my own work. And all of them for numerous hours of inspiring discussions and lots of coffee and tea.

I would also like to thank my evaluation committee, Michael M. Richter, William Cheetham, and Helge Langseth, for their valuable feedback and smooth handling of the evaluation process.

Some friends and colleagues I have met at NTNU have had a more indirect influence on this work, and I would in particular like to thank Richard Blake, Zoran Constantinescu-Fülöp, Diego Federici, Arne Dag Fidjestøl, Mingyang Gu, Trond Schjelderup Hegdahl, Peter H. Hughes, Wacław Kuśnierczyk, Per Kristian Lehre, Annette Lykknes, Eric Monteiro, Knut Harald Hassel Nielsen, Sobah Abbas Petersen, Jahnavi Phalkey, Martin Thorsen Ranang, Aslak Raanes, Rolv Seehuus, Rune Sætre, Axel Tidemann, John Walmsley, and Roland Wittje for endless discussions about research and the food quality in the NTNU canteens, moral support, conference trips, sharing offices, ski trips, making music, or just being at the right place at some important point in time.

I would like to say thanks also to the administrative and technical staff at the department and the faculty who was always very helpful and friendly, even if forms etc. were delivered very late.

A very special thank you goes to my family, my parents Johann and Margarete for laying the foundations for everything, and to my wife Anne Swoboda for her encouragement and support and for knowing when to push me and when to leave me alone.

Contents

Part I

Research Overview

1 Introduction

This thesis targets research in the field of artificial intelligence (AI) research. It focuses on the socio-technical aspects of artificial intelligence, namely how (specific types of) intelligent systems function in human workplace environments. The goal is first to get a better understanding of human needs and expectations when it comes to interaction with intelligent systems, and then to make use of the understanding gained in the process of designing and implementing such systems.

AI is a broad and diverse field. It is usually considered to be a subfield of computer science, but it draws it's foundations from many different sources such the cognitive sciences, psychology, and biology. Definitions of AI found in textbooks or introductory articles are often similar to the following one given by John McCarthy:

> "Q. What is artificial intelligence? A. It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable." [McCarthy, 2007]

This preliminary definition poses new questions, the most important of these arguably being the question of what intelligence is. Although, as with many areas, there is no universally accepted answer, it is generally understood that intelligence is a capacity displayed by humans (at least as a potential ability). Gottfredson offers the following definition:

> "Intelligence is a very general mental capability that, among other things, involves the ability to reason, plan, solve problems, think abstractly, comprehend complex ideas, learn quickly and learn from experience. It is not merely book learning, a narrow academic skill, or test-taking smarts. Rather, it reflects a broader and deeper capability for comprehending our

> surroundings – 'catching on,' 'making sense' of things, or 'figuring out' what to do." [Gottfredson, 1997]

Gottfredson further contends the accurate measurability of intelligence. Neisser et al. [1996], on the other hand, point out that no generally agreed definition of intelligence exists and give an overview of different concepts of intelligence. Theories highlighting the developmental aspects of intelligence are for example proposed by Vygotsky [1978] and Piaget [1955] who both acknowledge the importance of acting in the world for the development of intelligence.

To avoid the pitfalls of trying to attempt an explicit definition of intelligence, we can instead look at intrinsic aspects of humans as a species exhibiting intelligence. A classical definition of human beings is Aristotle's definition of humans as an animal rationale, [Aristotle, 1994]. Although this definition leads to several problems, these being some of the the reasons for which Descartes [1641] discarded it, we will, for the moment, accept it as a working definition.

One important aspect of rationality is the ability to express the reasons behind ones own behaviour. Kant, in his categorical imperative – "Act so that the maxim of thy will can always at the same time hold good as a principle of universal legislation" [Kant, 1788, p. 30], couples the ability to act morally (and thus freely) with the ability to give a rational explanation of behaviour. Accepting this, we can thus ascribe the ability of explaining ones behaviour and motives to every rational being, that means to every intelligent entity. We can therefore count explanatory capabilities, in particular the ability to explain ones own understanding of the world and ones own behaviour, as a necessary precondition for appearing intelligent.

Not only is explanatory capacity central to the appearance of intelligence, it is also an important vehicle in conveying information in everyday human-human interaction. Humans are social beings, and smooth and ongoing interaction relies on trust. Explanations are an important way of building this trust, they help us to understand one another and enhance the knowledge of the communication partners, increasing the liklihood that they will accept certain statements [Pu and Chen, 2006]. By each partner understanding more, they are better positioned to make informed decisions.

The importance of explanations also holds for artificial intelligence. When we look at an intelligent system as an artefact which mimics (some parts of) human behaviour, it is clear that the ability of the system to explain itself is an important capability and something that needs to be incorporated in any system if it is to appear as if being (human-like) intelligent. This need for explanations to be provided by knowledge-based systems is well documented [Swartout, 1983; Buchanan and Shortliffe, 1984; Swartout and Smoliar, 1987]. The adequacy of explanations and justifications, is dependent on background knowledge derived pragmatically. Thus, what counts as a good explanation in a certain situation is determined by context-dependent criteria [Leake, 1995].

Intelligent (or knowledge based) systems in todays research scenarios are no longer considered as black boxes that provide a full solution to a problem. Instead, problem solving is seen as an interactive process (a socio-technical process). Problem descriptions, as well as other input, can be incomplete and changing. As a consequence, there has to be communication between human and software agents. Communication requires mutual understanding that can be essentially supported by explanations. Such explanations can improve the problem solving process to a large degree. Thus, explanations are not only important in creating intelligent systems which mimic human interaction, they are also important in assisting in the problem solving process, a process that is essentially a social process.

We return then to the definition of human intelligence that we raised earlier. The description of humans as animal rationale focuses on the individual. Human activities, however, are rarely reclusive. If humans are animal rationale, we are also social animals. Halliday [1978, p. 14] highlights the fact that the human indivdual "is destined to become one of a group". We collaborate directly with other human beings, we build our own ideas on the ideas of other human beings, and we make use of artefacts designed, built, and used by other human beings. Leont'ev [1978] therefore stresses the social aspect in the development of cognitive abilities.

So far, for the purpose of this work, the following important facets of intelligent systems can be identified:

- **Rationality:** Instead of exploring the whole range of what intelligence means, we will focus on the display of rational behaviour by intelligent systems.

- **Ability to explain:** This ability is a prerequisite for an entity to appear rational and has been identified as a core ability intelligent systems should exhibit.

- **Sociality:** Human activities take place in social settings. Intelligent systems should be able to integrate themselves into such settings.

- **Contextualisation:** Both what constitutes appropriate behaviour and

> what is considered a good explanation is context dependant, therefore intelligent systems should be contextualised.

Artificial intelligence then, in order to create intelligent artefacts, needs to take account of both the social and the technical. McCarthy's definition of AI cited earlier gives some additional pointers to what should be considered within its domain. He sees AI as being both a science and an engineering discipline. The science part opens up for theoretical inquiries about the nature of human intelligence. The questions asked here are not unlike those asked in theoretical philosophy and psychology, for example. But the engineering part ties the field of AI to the construction of artefacts and poses questions about the design and implementation of intelligent systems.

We consider our work to be based on the theoretical side. That is, we will present arguments based on theories from other fields, like organisational psychology, and apply them to specific problems in artificial intelligence. However, the aim of this work is to make these theories beneficial for application development, and to this extent it has an engineering focus. The goal is to examine the theoretical value of social theories and to develop a framework which can help to design and build intelligent systems. This dual focus is reflected in the structure of the discussion of theory in Section 1.3 below.

1.1 Definitions

As we now have laid out the backdrop for the work we are presenting in this thesis, we can take a closer look at the title – Explanation Awareness and Ambient Intelligence as Social Technologies – and give a first definition of some of the concepts involved. These concepts will not be defined formally, e.g. like the definitions of context-free or context-sensitive grammars in the Chomsky hierarchy, but are meant to guide the reader through the text.

Mate and Silva [2005] define a socio-technical system as a "complex inter-relationship of people and technology, including hardware, software, data, physical surroundings, people, procedures, laws, and regulations". We use the term *social technologies* for technologies which facilitate interaction, support socialised cognition and activity, and generally enhance the socio-technical system. Intelligent social technologies alleviate the above mentioned *sociality* facet of intelligent systems.

Ducatel et al. [2001] give a definition of *ambient intelligence*. At the core of an ambient intelligent system lies the ability to appreciate the system's environment, be aware of persons in this environment, and respond intelligently to their needs. Ambient intelligent systems support the *contextualisation* issue described above.

The term *explanation awareness*, although used as a title for a whole series of workshops, see for example Roth-Berghofer et al. [2007], does not appear as yet to be well defined. Our working definition describes explanation awareness as the ability of an agent to 1. give explanations about its reasoning and actions and use explanation during the reasoning process as well and to 2. understand explanations given by others and to incorporate these into its own reasoning process.

For the purpose of this thesis, we will mainly deal with the ability of an actant to explain its own behaviour and reasoning since this enhances an intelligent system's *ability to explain*.

1.2 Research Questions

This thesis takes the form of collected papers. As such, each paper has its own specific research question. These papers do however form a related set, and while the structure of this thesis negates the need for formal research questions, it nevertheless follows that in presenting a cohesive discussion, certain questions were addressed. These questions, which reflect the research process in general, are enumerated below:

1. What are some of the socio-technical issues which have to be addressed when embedding intelligent systems into workplace environments and to what extent do they differ from the problems that traditional, non intelligent systems face?

2. Can theories from the social sciences, psychology, or linguistics be useful in tackling some of these problems, particularly those with a special focus on intelligent systems?

3. If so, what particular theories can help us in understanding the socio-technical settings?

4. How can theoretical results from answering the questions above be made useful for the problem of designing such systems, in particular how can the design process of intelligent systems be improved?

1.3 Theory

The work presented in this thesis is typical for research in artificial intelligence in the sense that it draws from different sources. It focusses on a specific problem in developing intelligent systems, namely how the artefacts to be developed can fit smoothly into existing socio-cultural settings. Special attention is paid to the peculiarities of intelligent systems, namely the change from seeing the artefact as a pure tool to a setting where the technical system becomes more of a partner, being pro-active and generally displaying abilities which traditionally are ascribed to humans.

To achieve this, we make use of theories from the fields of organisational psychology, sociology, and linguistics. This is in line with approaches commonly found in AI. However, most of the existing work deals with individual aspects, like how to mimic the behaviour or emulate methods of reasoning found in humans, whereas our work centers around the social aspect. Therefore, we base our work on theories which have not yet gained much attention in intelligent systems design, but they have been used in other subfields of computer science like software engineering and human-computer interaction (HCI). To be able to make them fruitful for intelligent systems research and development, we have to adapt them to the specific settings, and we have to transform them to suit the practical problems at hand. This includes the necessity to strip away some aspects which are important in the descriptive setting in which they are developed, but are a hinderance in the formative processes to which they will be put. This must, amongst other things, be done to avoid the trap of the last instance Althusser [1962]: to say that particular situations are systematically determined can be helpful in an analytical setting, but it does not explain every peculiarity of the situation. A dialectic relation between system and instance is never active in the pure state, but a reflection of the unity of systematic account and specific act. For example, the notion of the historicity of human activity is a valuable insight of activity theory, but the attempt to ground every aspect of human activity which is to be modelled with the help of computer systems in its historicity is futile.

The specific theoretical frameworks we draw on are first and foremost activity theory [Nardi, 1996; Leont'ev, 1978], to a lesser degree semiotics [de Souza, 2005; Halliday and Matthiessen, 2004], and superficially actor network theory [Monteiro, 2000; Latour, 1988]. A major problem with research of this nature, research which attempts to integrate theories from diverse areas of practice, is the perception that there are very different underlying philosophies. This is of particular importance when trying to integrate the strengths of theories for the purposes of solving real world problems. It is necessary to always have in mind the motivations and assumptions that hold, sometimes rather opaquely, behind a theory. Take the theories that are examined in this thesis for example. It can be considered, that there are very different underlying perspectives behind socio-technical theories such as activity theory or actor network theory, requirements engineering, and intelligent systems. It is necessary that any work that sets out to bring these potentially collaborative perspectives together needs to justify this coalition and argue for why they do not represent a problem.

In bringing out the unifying aspects of these approaches, we need to consider the underlying theoretical similarities. In Section 1.1 above, we discussed the dual structure of this thesis in terms of its theoretical and engineering goals. Let us begin by examining the theoretical approach. The thesis sets out from the assumption that humans are social by nature. Thus, intelligence is seen as a social rather than individual aspect. All the theoretical frameworks we propose to make use of strengthen the idea that human activities are social in nature and that humans do interact with artefacts in different manners.

Turning to the engineering aspect of the thesis. This work takes an approach that seeks to find engineering solutions that mimic social aspects of human behaviour and cognition rather than the biologically inspired solutions found in approaches such as neural networks or evolutionary algorithms. So we can see that both the theoretical and engineering aspects focus on the social rather than the biological.

1.4 Methodology

With this thesis, we are following a mixed methods approach [Johnson and Onwuegbuzie, 2004] where different theories are brought together for the explicit purpose of solving particular practical problems that intelligent systems design is faced with. The thesis is largely a theoretical examination of the issues at hand, but it proposes practical consequences as well. Some qualitative evaluation of particular aspects is also performed.

Due to the obvious time and spatial constraints, we cannot address all of the challenges that we face when building intelligent artefacts. In reducing the scope of the thesis, we have focused on the problem of explanation, and here in particular the problem of explanation from a user perspective. By putting social theories to work in the field of artificial intelligence, we show that results from other fields can be beneficial in understanding what explanatory capabilities are needed for a given intelligent system, and to ascertain in which situations an explanation should be delivered. Besides lessons learned in knowledge based system development as outlined above, the most important input comes from activity theory [Nardi, 1996].

The second focus is the challenge of contextualisation, and here we show that work in other scientific fields can be put to use in the development of context aware or ambient intelligent systems. Again, we draw on results from activity theory and combine this with insights from semiotics [de Souza, 2005].

We have seen that explanations are themselves contextual, so the third challenge is to explore the space spanned by the two dimensions *ability to explain* and *contextualisation*. Again, activity theory is beneficial in resolving this issue.

As discussed previously, the different theoretical considerations have also led to some practical approaches. Working with activity theory helps to better understand what the relevant contextual aspects of a given application are and helps to develop models of context which are both grounded in the tradition of context aware systems design and are plausible from a cognitive point of view.

Insights from an analysis of research in the knowledge based system area and activity theory have lead to the amendment of a toolbox for requirements engineering, so called problem frames [Jackson, 2001]. New problem frames which target explanation aware ambient intelligent systems are presented. This perspective looks out from the system design process into the world.

By comparison with this outward focus, the final part ties in with other work looking inwards, or at the design of an actual system after the requirements have been elicited and specified. Thus, the socio-technical perspective on explanations is coupled with work that addresses knowledge representation issues, namely how to model sufficient knowledge to be able to deliver explanations.

We do not aim in this thesis to contribute deeply to the further development of the social theories used, such as activity theory, actor network theory, or semiotics. Having said this, in any instance where theories are put to work in a practical way, there is a value for the theories. There is always a methodological problem turning research which gives *a posteriori* insights in socio-technical processes into methods which help in *a priori* design issues.

On the other hand, such a turn tests the ability of social and semiotic theories to provide useful and valuable information with predictive power, e.g. whether these theories do remain descriptive or have formative aspects as well. The questions asked from artificial intelligence are along the lines of what is required from a theory of context for it to be useful, what is needed from a social semiotic theory of language in order for it to be adequate, what are the challenges, what are the demands. These questions can ultimately lead to better theories and the potential for better outcomes for end users.

1.5 Thesis Outline

The thesis has the form of a paper collection. This initial section includes the introduction, a research description in Chapter 2, and the conclusions in Chapter 3. All publications are included in Part II in chronological order.

For the most part, the articles included in this thesis have been or will be published in different workshop or conference proceedings, and one in a journal. Where the majority of papers has been peer reviewed and accepted, the last article has been submitted for review but not accepted at the time of completion. The following papers can be found in Part II of the thesis:

A: Jörg Cassens: **A Work Context Perspective on Mixed-Initiative Intelligent Systems.** In: *Proceedings of the IJCAI 2003 Workshop on Mixed-Initiative Intelligent Systems*, pages 30–35, Acapulco, 2003.

B: Jörg Cassens: **Knowing What to Explain and When.** In: Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, number 142-04 in Technical Report of the Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, pages 97–104, Madrid, 2004.

C: Thomas R. Roth-Berghofer and Jörg Cassens: **Mapping Goals and Kinds of Explanations to the Knowledge Containers of Case-Based Reasoning Systems.** In: Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *LNAI*, pages 451–464, Chicago, 2005. Springer.

D: Jörg Cassens: **User Aspects of Explanation Aware CBR Systems.** In: Maria Francesca Costabile and Fabio Paternó, editors, *Human-Computer Interaction – INTERACT 2005*, volume 3585 of *LNCS*, pages 1087–1090, Rome, 2005. Springer.

E: Frode Sørmo, Jörg Cassens, and Agnar Aamodt: **Explanation in Case-Based Reasoning – Perspectives and Goals.** In: *Artificial Intelligence Review*, 24(2):109–143, October 2005.

F: Anders Kofod-Petersen and Jörg Cassens: **Using Activity Theory to Model Context Awareness.** In: Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers*, volume 3946 of *LNCS*, pages 1–17, Edinburgh, 2006. Springer.

G: Jörg Cassens and Anders Kofod-Petersen: **Using Activity Theory to Model Context Awareness: a Qualitative Case Study.** In: Geoff C. J. Sutcliffe and Randy G. Goebel, editors, *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, pages 619–624, Melbourne Beach, 2006. AAAI Press.

H: Jörg Cassens and Anders Kofod-Petersen: **Designing Explanation Aware Systems: The Quest for Explanation Patterns.** In: Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Explanation-Aware Computing – Papers from the 2007 AAAI Workshop*, number WS-07-06 in Technical Report, pages 20–27, Vancouver, BC, 2007. AAAI Press.

I: Anders Kofod-Petersen and Jörg Cassens: **Explanations and Context in Ambient Intelligent Systems.** In: Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors, *Modeling and Using Context – CONTEXT 2007*, volume 4635 of *LNCS*, pages 303–316, Roskilde, Denmark, 2007. Springer.

J: Jörg Cassens and Anders Kofod-Petersen: **Explanations and Case-Based Reasoning in Ambient Intelligent Systems.** In: David C. Wilson and Deepak Khemani, editors, *ICCBR-07 Workshop Proceedings*, pages 167–176, Belfast, Northern Ireland, 2007.

K: Jörg Cassens and Rebekah Wegener: **Making Use of Abstract Concepts – Systemic-Functional Linguistics and Ambient Intelligence.** Accepted for: IFIP AI 2008 – The Second IFIP International Conference on Artificial Intelligence in Theory and Practice. 7-10 September, 2008. Milan, Italy.

L: Jörg Cassens and Anders Kofod-Petersen: **Modelling with Problem Frames: The Knowledge Needed for Explanation-Aware Ambient Intelligent Systems.** Submitted to: ECAI 2008 – The 18th European Conference on Artificial Intelligence. 21-25 July 2008. Patras, Greece.

Although the whole thesis can be read chronologically, there are a number of other paths through this thesis. In the section below we outline these paths and the ways in which the articles are related to each other. This is also visualised in a graph depicted in Figure 1.1.

Readers wishing to set out from the socio-technical background should begin with Paper A as an appropriate point of departure. This paper introduces three views on intelligent systems in workplace environments; 1. Work process view (using actor network theory), 2. HCI interface view (using semiotics), and 3. HCI system view (using activity theory). It is further exemplified how these theories can help to tackle different issues in mixed-initiative intelligent systems.

These three views are generalised and further developed in Paper D. This paper addresses some issues of explanations in intelligent systems. This path through the thesis explores the relation between socio-technical theories and explanations. This leads further to a discussion of one particular view, namely the HCI system view and activity theory, in Paper B. The question tackled here is how an intelligent system can decide when to deliver an explanation about its own behaviour.

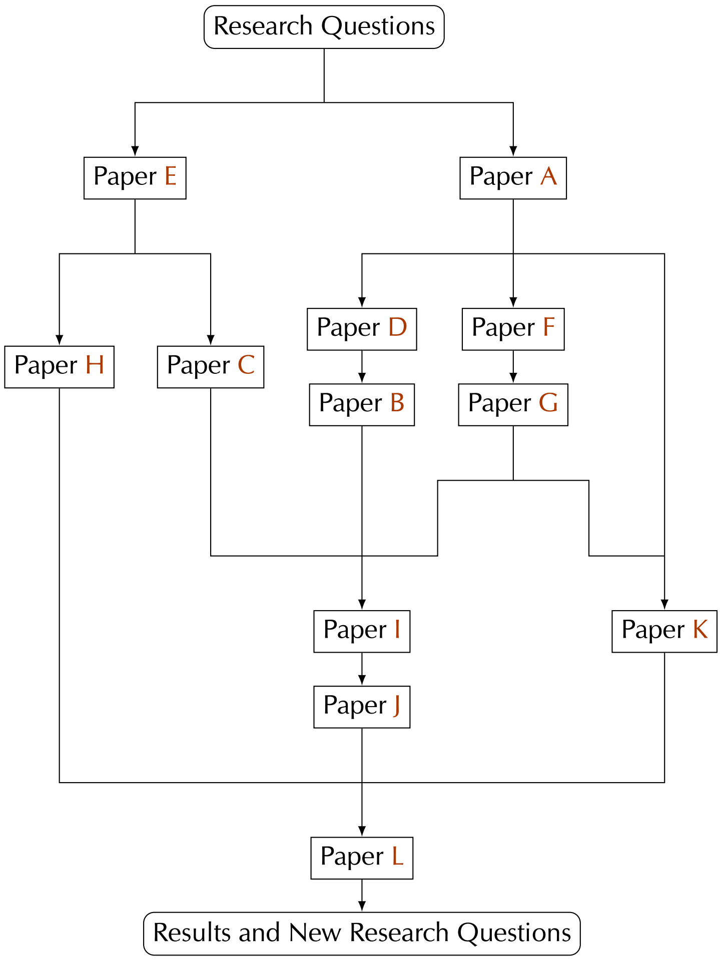

Figure 1.1.: Relations between the different papers included in the thesis.

The other path starting from the mixed initiative setting from Paper A focuses also on activity theory, but in the setting of context aware or ambient intelligent systems. Paper F takes a knowledge level perspective on context modelling and establishes a relation between cognitive sciences and context in intelligent systems. Different concepts from activity theory are mapped to different categories of context well established in context aware computing, and a psychologically plausible context model is proposed. This line of research is further explored by implementing the proposed structure in a context aware system for a hospital ward domain. The results of this work and a qualitative empirical study of its performance in a simulated environment is given in Paper G.

Another way to start exploring the contributions starts not with a dedicated socio-technical perspective, but with explanations in intelligent systems from a general user perspective. In Paper E, a systematic overview on explanation in philosophy and cognitive sciences and a historic overview of the use of explanations in artificial intelligence are given. Five goals a user can have with explanations are introduced, namely 1. *Transparency* (explain how the system reached the answer), 2. *Justification* (explain why the answer is a good answer), 3. *Relevance* (explain why a question asked is relevant), 4. *Conceptualisation* (clarify the meaning of concepts), and 5. *Learning* (teach the user about the domain). The use of explanations in case-based reasoning is reviewed and challenges are identified.

In order to tie the high level user goals in with a view on explanations from a systematic classification, Paper C explores the relation of the notion of user goals with work on different kinds of explanation conducted by Roth-Berghofer [2004]. A process model for the design activity is proposed, and several mappings from user goals over explanation kinds to the different knowledge containers of case-based reasoning systems are identified, opening up for moving towards the implementation of explanatory capabilities from a user goal perspective.

These three paths, coming from the user goal notion of Paper C, the work on context awareness in Paper G, and research in activity theory and explanations in Paper B, come together in Papers I and J.

Paper I details the relationships between *context awareness* and *context sensitivity* on the one hand and explanations as a *means of reasoning* and a *means of communication* with the user on the other hand. It is proposed how concepts from activity theory can be used to address the different goals a user can have towards explanations, and it is discussed how these goals can be satisfied in the different phases of the use of the system.

Looking more into the implementation issues in the form of an ambient intelligent case-based reasoning system is Paper J. The context aware system for a hospital ward environment introduced in G is explored further, and it is shown how explanation needs in different phases of the use of a system are addressed.

Starting from the general overview of explanations in Paper E, Paper H takes a step towards software engineering issues of designing explanation aware systems. The five explanation goals introduced in Paper E are revisited and a problem frame diagram suitable for requirements engineering processes for each of the goals is presented. These goals are intended to enable software engineers to explicitly model explanatory capabilities.

Going back to the socio-technical start in Paper A, another socio-technical approach is explored in Paper K. Semiotics is one of the three views on intelligent systems in workplace environments introduced in Paper D, and Paper K explores a specific semiotic theory, systemic functional linguistics, and shows its use in ambient intelligence. The main focus is on abstract concepts.

All of the paths outlined until now, but especially the results of work on explanations in ambient intelligent (Paper I) and the requirements engineering method outlined in Paper H, are merged in Paper L. The problem frames for ambient intelligent and explanation aware system introduced earlier are further developed. An example of how to use the newly developed frames in requirements engineering is given. To this end, a step towards the re-design of an existing application from the hospital ward domain (Paper G) is taken and it is shown how explanation patterns can help to model requirements towards the explanatory capabilities of the system.

Bibliography

Louis Althusser. Contradiction and overdetermination. In *For Marx*. Penguin Press, 1962. URL http://www.marxists.org/reference/archive/althusser/1962/overdetermination.htm.

Aristotle. *Metaphysik*. rowohlts enzyklopädie. Rowohlt, Hamburg, 1994.

Bruce G. Buchanan and Edward H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison Wesley, Reading, 1984.

Clarisse Sieckenius de Souza. *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press, 2005. ISBN 0262042207. QA76.9.H85 .D465 2005.

René Descartes. *Meditationen über die Grundlagen der Philosophie*. Number 27 in Philosophische Bibliothek. Felix Meiner Verlag, Hamburg, 1641.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Linda S. Gottfredson. Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography. *Intelligence*, 24(1):13–23, 1997.

Michael A.K. Halliday. *Language as a Social Semiotic: the social interpretation of language and meaning*. University Park Press, 1978.

Michael A.K. Halliday and Christian M.I.M. Matthiessen. *An Introduction to Functional Grammar, Third edition*. Arnold, London, UK, 2004.

Michael Jackson. *Problem Frames – Analysing and Structuring Software Development Problems*. Addison-Wesley, Boston, MA, 2001.

R. Burke Johnson and Anthony J. Onwuegbuzie. Mixed methods research: A research paradigm whose time has come. *EDUCATIONAL RESEARCHER*, 33(7):14–26, October 2004. doi: 10.3102/0013189X033007014. URL http://dx.doi.org/10.3102/0013189X033007014.

Immanuel Kant. *Kritik der praktischen Vernunft*. Akademie, 1788.

Bruno Latour. *Science in Action: How to Follow Scientists and Engineers Through Society*. Harvard University Press, 1988. ISBN 0674792912.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

Aleksei N. Leont'ev. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

Jose Luis Mate and Andres Silva, editors. *Requirements Engineering for Sociotechnical Systems*. Idea Group Publishing, 2005.

John McCarthy. What is ai? Internet, 2007. URL http://www-formal.stanford.edu/jmc/whatisai/. Last visited 2007-12-09.

Eric Monteiro. Actor-network theory. In Claudio U. Ciborra, editor, *From Control to Drift*, pages 71–83. Oxford University Press, 2000.

Bonnie A. Nardi, editor. *Context and Consciousness*. MIT Press, Cambridge, MA, 1996.

Ulric Neisser, Gwyneth Boodoo, Thomas J. Bouchard Jr., A. Wade Boykin, Nathan Brody, Stephen J. Ceci, Diane F. Halpern, John C. Loehlin, Robert Perloff, Robert J. Sternberg, and Susana Urbina. Intelligence: Knowns and unknowns. *American Psychologist*, 51(2):77–101, 1996.

Jean Piaget. *The Construction of Reality in the Child*. Routledge and Kegan Paul, London, 1955.

Pearl Pu and Li Chen. Trust building with explanation interfaces. In *IUI '06: Proceedings of the 11^{th} international conference on Intelligent user interfaces*, pages 93–100, New York, NY, USA, 2006. ACM. ISBN 1-59593-287-9. doi: http://doi.acm.org/10.1145/1111449.1111475.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Peter Funk and Pedro A. González Calero, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, pages 389–403, Berlin Heidelberg, 2004. Springer.

Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors. *Explanation-Aware Computing – Papers from the 2007 AAAI Workshop*, number WS-07-06 in Technical Report, Vancouver, BC, 2007. AAAI Press. ISBN 978-1-57735-333-1. URL http://www.aaai.org/Library/Workshops/ws07-06.php.

William R. Swartout. What kind of expert should a system be? xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

William R. Swartout and Stephen W. Smoliar. On making expert systems more like experts. *Expert Systems*, 4(3):196–207, 1987.

Lew S. Vygotsky. *Mind in Society*. Harvard University Press, Cambridge, MA, 1978.

Research Description 2

The purpose of this chapter is to give a detailed overview of the research results presented in the papers in Part II and their relation to established fields of research. Sections 2.1 through 2.3 will deliver some background information. We start in Section 2.1 by giving an overview of the socio-technical framework employed. Section 2.2 gives some background on explanation in intelligent systems. The final section of the background material is covered in Section 2.3 where we look at context.

The following Sections 2.4 through 2.7 will describe our own research contributions. Section 2.4 will investigate the relation between context and explanation. We then turn to modelling issues, beginning in Section 2.5 with work on context and following up in Section 2.6 with modelling explanations. Section 2.7 concludes this chapter by considering explanations and context combined.

2.1 Socio-Technical Framework

In this section, we describe the socio-technical theories that have formed the basis of this work. When an intelligent system is considered not as a replacement of, but a supplement to human work, the question of an adequate form of interaction arises. Such systems are to a certain degree trespassing the boundary of the computer system being a tool, and increasingly, are acting as pro-active partners in a work environment.

In the light of these changes, the human computer interaction should be revisited. Traditional interface engineering methods focusing on the computer as a tool do not seem to be appropriate in the design of intelligent systems. Furthermore, the integration of these kinds of systems into work processes is dynamic, and is thus likely to vary in the future.

At first sight, this has the consequence that the whole work situation must be taken into account when developing AI systems. It is more than likely that Traditional software engineering techniques, mainly focusing

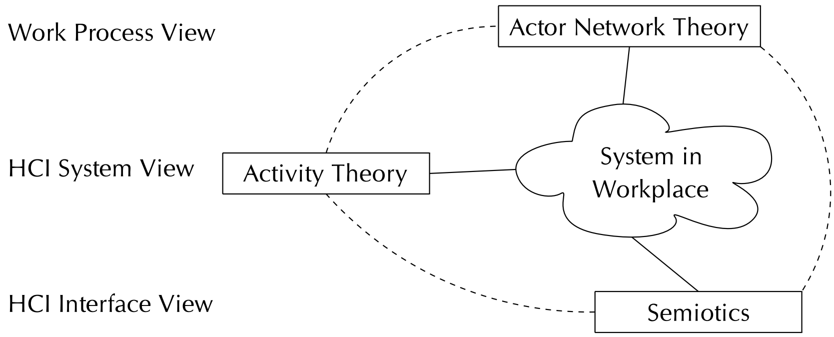

Figure 2.1.: Overview: Different views on the work context.

on the artefact itself, would not give adequate results. Therefore, in order to produce better results, the software production process must integrate work analysis methodologies.

In order to understand how the system fits into a work place situation, we have in Papers A and D proposed a theoretical framework that focuses on three different perspectives (see Figure 2.1):

- Work process view: Actor Network Theory,
- HCI interface view: Semiotics,
- HCI system view: Activity Theory.

This theoretical framework, it is argued, is helpful for understanding how AI systems in general and case-based reasoning (CBR) systems in particular fit into a work process, including the way in which they interact with the user.

2.1.1 *Activity Theory*

In this subsection, we will begin by giving a short summary of aspects of activity theory (AT) that are important for the present work. The theoretical foundations of AT in general can be found in the works of Vygotsky [1978, 1985] amd Leont'ev [1978]. For the relation between AT and computer science see Nardi [2003] for a short introduction and Bødker [1991] or Nardi [1996a] for deeper coverage.

Activity theory is a descriptive tool to help understand the unity of consciousness and activity. Its focus lies on individual and collective work practise. One of its strengths, and the primary reason for its value in AI development, is the ability to identify the role of material artefacts in the work process. An activity (Figure 2.2) is composed of a subject, an object,

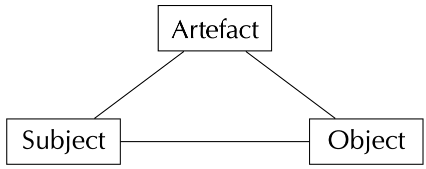

Figure 2.2.: Activity Theory: The basic triangle of Mediation

and a mediating artefact or tool. A subject is a person or a group engaged in an activity. An object is held by the subject, and the subject has a goal directed towards the object he wants to achieve, motivating the activity and giving it a specific direction.

Some basic properties of activity theory are:

- **Hierarchical structure of activity:** Activities (the topmost category) are composed of goal-directed actions. These actions are performed consciously. Actions, in turn, consist of non-conscious operations.
- **Object-orientedness:** Activity theory takes account of objective and socially or culturally defined properties. Our way of doing work is grounded in a praxis which is shared by our co-workers and determined by tradition. The division of labour and the way an artefact is used influences the design. Hence, artefacts pass on the specific praxis they are designed for.
- **Mediation:** Human activity is mediated by tools, such as for example language. The artefacts as such are not the object of our activities, but appear already as socio-cultural entities.
- **Continuous Development:** Both the tools used and the activity itself are constantly reshaped. Tools reflect accumulated social knowledge, hence they transport social history back into the activity and to the user.
- **Distinction between internal and external activities:** Traditional cognitive psychology focuses on what is denoted internal activities in activity theory, but it is emphasised that these mental processes cannot be properly understood when separated from external activities, that is the interaction with the outside world.

Taking a closer look on the hierarchical structure of activity, we can find the following levels:

- **Activity:** An individual activity is for example to check into a hotel, or to travel to another city to participate in a conference. Individual activities can be part of collective activities, e.g. when someone organises a workshop with some co-workers.

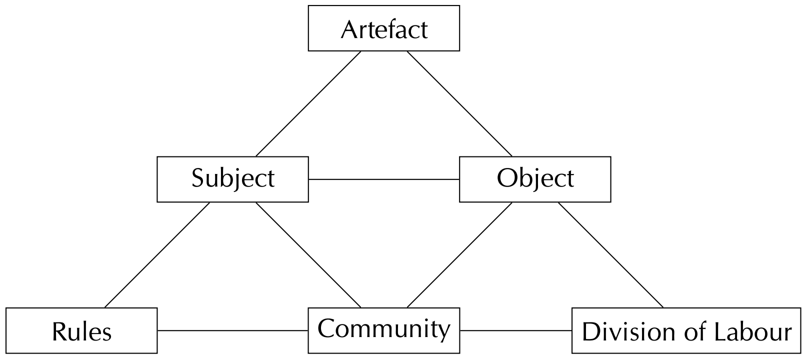

Figure 2.3.: Cultural Historical Activity Theory: Expanded triangle, incorporating the community and other mediators.

- **Actions:** Activities consist of collections of actions. An action is performed consciously, the hotel check-in, for example, consists of actions like presenting the reservation, confirmation of room types, and handover of keys.
- **Operations:** Actions consist themselves of collections of non-conscious operations. To stay with our hotel example, writing your name on a sheet of paper or taking the keys are operations. That operations happen non-consciously does not mean that they are not accessible.

It is important to note that this hierarchical composition is not fixed over time. If an action fails, the operations comprising the action can get conceptualised, they become conscious operations and might become actions in the next attempt to reach the overall goal. This is referred to as a breakdown situation. In the same manner, actions can become automated when done many times and thus become operations. In this way, we gain the ability to model a change over time.

An expanded model of activity theory, the cultural historical activity theory (CHAT), covers the fact that human work is done in a social and cultural context (compare e.g. Kutti [1996]; Mwanza [2000]). The expanded model (depicted in Figure 2.3) takes this socio-cultural aspect into account by adding a *community* component and other mediators, especially *rules* (an accumulation of knowledge about how to do something) and the *division of labour*.

In order to be able to model the fact that several subjects can share the same object, we add the community to represent a subject as being embedded in a social context. We now have relationships between subject and community and between object and community, respectively. These relationships are themselves mediated, with rules relating to the subject and the division of labour relating to the object.

2.1.2 *Semiotics*

Human work processes take place with and through sign systems. Understanding these sign systems is crucial for making intelligent artefacts. Semiotics can be described as the study of sign systems and their historicity. For an introduction to semiotics we refer to Peirce [1904] and Halliday and Matthiessen [2004]. For a comprehensive account of semiotics as it is applied to computing we recommend the works of Gudwin and Queiroz [2006], in particular Andersen and Brynskov [2006]; Clarke et al. [2006], as well as de Souza [2005]. The current state of knowledge is never divorced from the past that has affected it. Thus, the positive contributions of semioticians such as Saussure, Pierce and Voloshinov to the social semiotics of Halliday and those working in this tradition are briefly outlined below.

Saussure [1966] was central to the development of linguistics and semiotics as fields of study. A major contribution was the focus on the relational nature of signs. For Saussure, the meaning of a sign was not simply understood as an intrinsic property of the sign itself, that is, by its substance, but its relation to other signs within a whole system of signs. The differentiating or contrastive nature of signs was an outstanding concept in Saussurian semiotics, and he strengthened the view that relationships between signs were important when trying to understand their meaning.

Peirce [1904] and his triadic representation of the sign process has had a major impact on semiotics. The semiotic triangle in its different variations (see Figure 2.4) is probably the most used concept of semiotics in computer science. Pierce emphasises that it is not possible to isolate signs and strip them of their relations. Being a sign is a way of being in relation to something, not some abstract property of being in itself. Pierce is also well known for the explicit distinction of different sign types, distinguishing between indexical, iconic, and symbolic. He was also particularly focused on strengthening the role of the interpreting entity. Social aspects of the material setting are coded through the (indexical) nature of signs.

One of the main contributions to semiotics of Voloshinov [1973] is the emphasis he placed on the fact that social processes cannot be taken out of any analysis. It is not fruitful to look at the sign decoupled from the social process in which it occurs. Voloshinov externalises the sign process and makes its roots in the social process explicit. This acknowledges

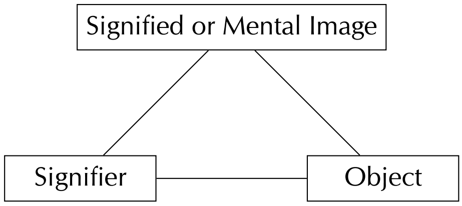

Figure 2.4.: The semiotic triangle.

the importance of contextualising the sign process. Because he treated the sign process as inherently a social phenomenon, he was able to see that "...the form of signs is conditioned above all by the social organisation of the participants and also by the immediate conditions of their interaction..." [Voloshinov, 1973, p. 21].

In this research, the social semiotics outlined by Halliday (see for example Halliday and Matthiessen [2004]) has been used. Halliday combines the strengths of the approaches of Saussure, Pierce, and Voloshinov. He brings together the tradition of relational thinking from Saussure, the understanding that different modalities have consequences for the structure of meanings from Pierce and, from Voloshinov, the insistence that the sign is social and can never be divorced from the social process.

Halliday's systemic functional theory of language (SFL) is a social semiotic theory that sets out from the assumption that humans are social beings that are inclined to interact and that this interaction is inherently multimodal [Halliday, 1978]. We interact not just with each other, but with our own constructions and with our natural world. These are all different forms of interaction, but they are all sign processes.

2.1.3 Actor Network Theory

Actor network theory, ANT, is an integrated model that maps relations between entities that are at the same time material (between artefacts or natural things) and non-material or semiotic (between concepts). The main assumption is that many relations are at the same time material and semiotic, involving both ideas and motivations of actors involved and their expression in material artefacts or technologies. All these aspects, being material or not, are captured in the same network. We recommend Latour [1991] for a general introduction to the underlying ideas and Monteiro [2000] for a computer science point of view.

The underlying concept of ANT might be said to be reasonably uncomplicated: whenever you do something, many influences on *how* you do it exist. For instance, if you attend a conference, it is likely that you will stay in a hotel. How you behave at that hotel is influenced by your previous experience with hotels, regulations for check-in and check-out, and the facilites the hotel offers you (breakfast room, elevators, etc.).

Actions are not performed anew each time, but are influenced by a wide range of factors. The aim of ANT is to provide a unified view on these factors and your own action. An actor network in this notion is "... the act linked together with all of its influencing factors (which again are linked), producing a network" (see Monteiro [2000, p. 4]).

In this network, you find both technical and non-technical elements. By this, ANT avoids the trap of either overstating the role of technological artefacts in a socio-technological system or underestimating their normative power by applying the same framework to both human actors and technological artefacts. This makes it possible for us to understand how technological artefacts influence the doings of human actors in much the same way as other human actors.

Some key concepts of the theory are (compare e.g. Monteiro [2000]):

- **Actors:** Humans and technological artefacts,
- **Actor-network:** The totality of actors, interests, organisations, rules, standards, and their interaction,
- **Translation:** Actors interests translated into technical or social arrangements,
- **Inscription:** Result of the translation of one's interest into material form,
- **Subscription:** Acceptance of the inscribed interests by other actors.

In actor network theory, technological artefacts can stand for human goals and praxis. Hotel keys, for example, are often not very conveniently designed, because the hotel owner has *inscribed* his intention (that the keys do not leave the hotel) into metal tags (which is why the guests *subscribe* to the owner's intention: they do not want to carry this weight). A software system for workflow management is a representation of organisational standards in the company where it is used (and makes human users follow these standards).

One advantage of ANT in the setting of intelligent systems is that it already comprises technical artefacts and humans in the same model. Humans and artefacts are exchangeable and can play the same role in the network. But, in contrast to traditional artefacts, which are merely passive (black boxes in which human interests are subscribed) or whose active role is restricted to translating intentions of the designer into changes of the praxis of the user, AI systems play a more active role: they have to *act -as-if* they had human capabilities.

This has for example been utilised by Pieters [2001], who has argued that intelligent systems have to show certain capabilities usually ascribed to humans in order to interact with the user in a meaningful way. On the other hand, since at least some of these capabilities rely on transcendental concepts, it is not possible to *design* machines that display them.

ANT opens up the potential to abstract away this problem. Since artefactual and human actors can play the same role in a network, we use the notion of *as-if* in our approach: in roughly the same way as humans can never be sure that human counterparts have the capabilities they expect them to have, but ascribe it to them, our goal is to design intelligent systems which act in a way that makes humans ascribe human characteristics to them. Pieters [2001] further argues that some properties of knowledge-intensive case-based reasoning systems make them well suited for exposing this *as-if* capability.

2.2 Explanations

Explanations are an important vehicle to convey information between communicating people in everyday human to human interaction. Explanations enhance the knowledge of the participants in such a way that they accept certain statements and gain a better understanding of the actions of the other persons involved and their motivations. They understand more, allowing them to make better informed decisions themselves. According to Schank [1986], explanations are the most common method used by humans to support their decision making.

This is supported by Spieker [1991] and his investigation into natural language explanations in expert systems. We identify some typical reactions of humans as soon as we cannot follow a conversation:

- we ask our conversation partner about concepts that we did not understand,
- we request justifications for some fact or we ask for the cause of an event,
- we want to know about functions of concepts,
- we want to know about purposes of concepts, and
- we ask questions about his or her behaviour and how he or she reached a conclusion.

All those questions and answers are used to understand what has been said and meant during a simple conversation. An important effect of explanations is that the process of explaining certainly has some effect on one's trust in the competence of a person or machine: We keep our trust, we increase or decrease it. At least, providing explanations makes decisions more transparent, and motivates users to further use the system.

The need for explanations provided by intelligent systems to its users is well-known and was well addressed in expert systems research. For knowledge-based systems, explanations and knowledge acquisition are the only two communications channels through which they interact with their environment.

The more complex intelligent systems get, the more explanation capabilities the users expect when using such systems. This requirement was recognised early on in expert systems research and development, for example by Swartout [1983]; Buchanan and Shortliffe [1984]; Swartout and Smoliar [1987]. Considerable results were produced, but research activity decreased together with the general decline of expert systems research in the 1990s. The major problems in connection with classical expert systems seemed to be solved.

In the mid 1990's, we can see an increasing interest in this topic in case-based reasoning, in particular by Leake [1996] and Schank et al. [1994]. At the turn of the century, we find the issue discussed again in the context of knowledge-based systems as seen by Gregor and Benbasat [1999] or Swartout and Moore [1993]. Recently, we have seen a renewed focus in CBR on this track of research. The European Conference on Case-Based Reasoning (ECCBR) 2004 featured, for example, a workshop on Explanation in case-based reasoning [Gervás and Gupta, 2004] as well as a couple of papers on explanation at the main conference [Funk and Calero, 2004], and the journal Artificial Intelligence Review had a special issue on explanation in case-based reasoning [Leake and McSherry, 2005].

It is important to note that the term explanation can be used in different ways. Leake [2004] identifies three different facets of explanations within the context of case-based reasoning which can be generalized for all intelligent systems:

- Using explanations to support the case-based reasoning process
- Generating explanations by case-based reasoning
- Using cases for explaining system results to an external user

If we consider this from a functional perspective, it is possible to subsume the last two facets under the heading of user oriented aspects, since both are targeted towards the user of the system. In our understanding, showing the case to the user is a special case of "generating explanations by case-based reasoning", making use of the case-based reasoning assumption that similar problems have similar solutions. Provided that the user has some knowledge about the similarity function and that the case structure is understandable by the user, the displayed case acts as an explanation to the user [Cunningham et al., 2003]. We are left with two functions of an explanation, as described by Aamodt [1991]: first, enhancing and promoting the reasoning process. Second, delivering some knowledge about the reasoning process, its results, or implication to the user. We call the first aspect the *system centric view* on explanation and the second one the *user centric view* on explanation:

- Explanation as **part of the reasoning process** itself.
  Example: a knowledge intensive case-based reasoning system can use its domain knowledge to explain the absence/variation of feature values.
- Giving explanations of the found solution, its application, or the reasoning process **to the user**.
  Example: in an engine failure diagnosis system, the user gets an explanation on why a particular case was matched.

The work presented in this thesis generally focuses on the user centric aspects of explanations. For explanations to be suited to the user they need to be contextually appropriate.

2.3 Context

We have so far introduced the idea that for an artefact to be considered intelligent it must exhibit intelligent behaviour. What we generally refer to when we say this is behaviour that is contextually appropriate. An ability to accurately read context is important for any animal if it is to survive, but it is especially important to social animals and of these perhaps humans have made the most out of being able to read context, where such an ability is tightly linked to reasoning and cognition [Cohnitz, 2000; Leake, 1995].

In understanding human cognition and reasoning, disciplines such as neuroscience, psychology, sociology, linguistics, and philosophy have had to take a stance on context as a concept. Setting aside the more mechanistic views taken on reasoning, which typically need not consider context at all, positions on context tend to fall into two broad domains: those who see context as vast and unable to be coded and those who see it as vast but able to be coded. This divides roughly along the same lines as the relativism debate: those who believe in an ultimate reality and those who believe reality is relative. For most this debate can remain a largely theoretical debate impinging little on day to day research, for the field of artificial intelligence however, this debate has very real consequences. Because of the need to study reasoning in the real world, AI has, like fields such as anthropology, been forced to work with context however underelaborated the models.

For social and practical reasons, historically, AI has drawn heavily from formal logic. For example, one of the benefits of such models was that they were comparably easy to implement. Formal logic is concerned with the explicit representation of knowledge and places great emphasis on the need to codify all facts that could be of importance. This focus on knowledge as an objective truth can be traced back to e.g. the logic of Aristotle [1998] who believed that at least a particular subset of knowledge had an objective existence (Episteme). This view contrasts with that of, for example, Polanyi [1964], who argues that no such objective truth exists and all knowledge is at some point personal and hidden (tacit).

The total denial of the existence of an objective truth is problematic, since consequently there can exist no criterion to value any representation of knowledge. We can contrast this with the view of Kant, who regards the accordance of the cognition with its object as being presupposed in the definition of truth [Kant, 1787, p. 52]. Going further, he makes clear that a purely formal and universal criterion of truth cannot exist. He foregrounds the dialectic relation between the formal logic and the objects to which this logic may be applied and which are given through intuition. Such a dialectic approach overcomes the conceptual difficulties outlined above, but the consequences for computational models are not easily accounted for.

The denial of objective truth can be seen as one core component in many approaches often referred to as postmodern or deconstructivist. Sokal and Bricmont [2001] eloquently present a discussion of many postmodern theoreticians with regard to modern scientific knowledge. Of particular interest for the topic at hand is an intermezzo on epistemic relativism in philosophy of science. The authors acknowledge that we never have direct access to the world, but only through our senses. In contrast to Kant who argues for a dialectical process of apperception, Sokal and Bricmont take a pragmaticist approach and reject a relativist epistemology.

Context does not fit very well with the strict logical view on how to model the world. However, an extremely personal and unique account of context serves little purpose in attempting generality. Context is, after all, a shared and very elusive type of knowledge. Despite the fact that humans can quite easily read context, context is hard to quantify in any formal way, and it is difficult to establish the type of knowledge that is useful in any given situation. Ekbia and Maguitman [2001] argue that this has led to context being largely ignored by the AI community. Neither the relativist nor the formal logic approach to context has been very useful at producing accounts of context which resonate with the AI community, and, except for some earlier work on context and AI, like Doug Lenat [1998] and the others that we discuss later in this section, Ekbia and Maguitman's observation still holds.

Like many who write about context, Ekbia and Maguitman's paper is not a recipe on how to incorporate contextual reasoning into logistic systems, but rather an attempt to point out the deficiencies and suggest possible directions AI could take to include context. Their work builds on the work by the American philosopher John Dewey. According to Ekbia and Maguitman, Dewey distinguishes between two main categories of context: background context (spatial and temporal context); and selective interest. The spatial context covers all contemporary parameters. The temporal context consists of both intellectual and existential context. The intellectual context is what we would normally label as background knowledge, such as tradition, mental habits, and science. The notion of situation comes out of a combination of Existential context with selective interest. A situation in Dewey's account, is a confused, obscure, and conflicting thing, that a human reasoner attempts to make sense of through the use of context. The pragmatist approach, as exposed by Dewey, leads to the following statements about context [Ekbia and Maguitman, 2001, p. 5]:

1. Context, most often, is not explicitly identifiable.
2. There are no sharp boundaries among contexts.
3. The logical aspects of thinking cannot be isolated from material considerations.
4. Behaviour and context are jointly recognisable.

Although it is by no means unique to him, Dewey's work foregrounds the absolute inseparability of mind and nature. Ekbia and Maguitman suggest that AI, by focussing on the the logical approach to (artificial) reasoning, has not dealt with context in any consistent way. This has led to many of the problems associated with the use of context because it works with a separation between mind and nature.

By considering the weaknesses of different logic-based AI methods and systems, Ekbia and Maguitman argue that, where context is concerned, AI has not yet parted company with the limitations of logic driven approaches. Furthermore, they stress, based on the notion of situations described above, intelligence should be seen as being action-oriented.

The conception of intelligence as action-oriented, an approach which makes context a tool for selecting the correct action, is shared by many within the computer science community. Most notably the works by Strat [1993], where context is applied to select the most suitable algorithm for recognition in computer vision, and by Öztürk and Aamodt [1998] who utilised context to improve the quality and efficiency of case-based reasoning.

Strat [1993] reports on the work done in computer vision to use contextual information in guiding the selection of algorithms in image understanding. When humans observe a scene they utilise a large amount of information (context) not captured in the particular image. At the same time, all image understanding algorithms use some assumptions in order to function, creating an epistemic bias. Examples are algorithms that only work on binary images, or that are not able to handle occlusions.

Strat defines three main categories of context: *physical*, being general information about the visual world independent of the conditions under which the image was taken; *photogrammetric*, which is the information related the acquisition of the image; and *computational*, being information about the internal state of the processing. The main idea in this work is to use context to guide the selection of the image-processing algorithms to use on particular images. This is very much in line with the ideas proposed by Ekbia and Maguitman, where intelligence is action-oriented, and context can be used to bring order to diffuse and unclear situations.

This action-orientated view on reasoning and use of context is also advocated by Öztürk and Aamodt [1998]. They argue that the essential aspects of context are the notions of *relevance* and *focus*. To facilitate improvements to case-based reasoning a context model is constructed. This model builds on the work by Hewitt, where the notions of *intrinsic* and *extrinsic* context types are central. Tiberghien [1986] states that according to Hewitt, intrinsic context is information related to the target item in a reasoning process, and extrinsic is the information not directly related to the target item. This distinction is closely related to the concepts of *selective interest* and *background context* as described by Dewey. The authors refine this view by focusing on the intertwined relationship between the *agent* doing the reasoning, and the *characteristics* of the problem to be solved. This is exactly the approach recognised as being missing in AI by Ekbia and Maguitman.

Öztürk and Aamodt build a taxonomy of context categories based on this merger of the two different worlds of information (internal vs. external). Beside this categorisation, the authors impose the action, or task, oriented view on knowledge in general, and contextual knowledge in particular. The goal of an agent *focuses* the attention, and thereby the knowledge needed to execute tasks associated with the goal. The example domain in their paper is from medical diagnostics, where a physician attempts to diagnose a patient by the hypothesise-and-test strategy. The particular method of diagnostics in this case-based reasoning system is related to the strategy used by Strat. They differ insofar as Strat used contextual information to select the algorithms to be used, whereas Öztürk and Aamodt have, prior to run-time, defined the main structure of a diagnostic situation, and only use context to guide the sub-tasks in this process.

Zibetti et al. [2001] focus on the problem of how agents understand situations based on the information they can perceive. To our knowledge, this work is the only one that does not attempt to build an explicit ontology on contextual information prior to run-time. The idea is to build a (subjective) taxonomy of ever-more-complex situations solely based on what a particular agent gathers from the environment in general, and the behaviour of other agents in particular.

The implementation used to exemplify this approach contains a number of agents “living” in a two-dimensional world, where they try to make sense of the world by assessing the spatial changes to the environment. Obviously the acquisition of knowledge starting with a *tabula rasa* is a long and tedious task for any entity. To speed up the process the authors predefined some categories with which the system is instantiated.

Research and development in intelligent systems taking context into account is often labelled as *ambient intelligence*. Following the account of Ducatel et al. [2001], we would like to remind that this means that ambient intelligent systems should have the ability to appreciate the system’s environment, be aware of persons in this environment, and respond intelligently to their needs. To realise the abilities of an ambient intelligent system, three main areas of responsibility can be identified [Kofod-Petersen and Aamodt, 2006]: first, the initial responsibility of *perceiving* the world that the system inhabits; second, the responsibility of being aware of the environment and reason about ongoing situations, which traditionally has been labelled as *context awareness*; and third, exhibit appropriate behaviour in ongoing situations by being *context sensitive* [Kofod-Petersen and Aamodt, 2006; Yau et al., 2002].

In Kofod-Petersen and Aamodt’s architecture, context serves two purposes. Initially it is used as a focussing lens on the part of the world that can be perceived. Here the context limits the parts of the knowledge that the system uses to classify the situation. The second use of context is in the context sensitivity layer, where context is viewed as a lens that focuses the part of the system’s knowledge that is to be used to satisfy the goal of the situation.

For the purpose of this work we will disregard the perception layer of the architecture as the perception layer demonstrates no reasoning capabilities, and only structures perceived data syntactically. Following arguments earlier introduced by Kofod-Petersen and Aamodt [2006], we identify these two aspects as two distinct steps in the reasoning process:

- **Context Awareness:** Trying to detect the situation the system is in. *Example:* An ambient intelligent system for supporting health personnel figures out that the user is on a ward-round because of the time of the day, the location, and the other persons present.

Table 2.1.: Explanation and context

	Context Awareness	**Context Sensitivity**
System Centric	Generate an explanation to **recognise** the situation	**Identify** the behaviour the system should expose
User Centric	**Elucidate** why the system identifies a particular situation	**Explicate** why a certain behaviour was chosen

- **Context Sensitivity:** Acting according to the situation the system thinks it is in.
  Example: the same system fetches the newest versions of electronic patient records of all patients in the room from the hospital systems. When the user stands close to the bed of a patient, the system automatically displays them.

2.4 Explanations and Context

The adequacy of explanations and justifications is dependent on pragmatically given background knowledge. What counts as a good explanation in a certain situation is determined by context dependent criteria [Cohnitz, 2000; Leake, 1995]. So how are context and explanations related? We follow the distinctions introduced earlier in this chapter. This means we look at 1. explanation as *part of the reasoning process* itself or 2. at giving explanations of the found solution, its application, or the reasoning process *to the user*. At the same time, we have the distinction of 1. *context awareness*, trying to figure out which situation the system is in, and 2. *context sensitivity*, acting according to the situation the system thinks it is in. Combing these views on explanation and on context, we end up with two dimensions of inquiry as depicted in Table 2.1.

We will further investigate the relationship between context and explanations by examining an example from a hospital (cardiology) ward domain. A case-based reasoning system is used to identify the different situations. The system's main purpose is to identify ongoing situations and proactively acquire digital information required by the persons present.

2.4.1 *Recognise*

In this step, we are *using explanations to recognise the current ongoing situation*. The system uses all available resources in its reasoning process.

Let us assume that the ongoing situation is a *ward round*. Normally ward rounds take place in a patient room, however the current situation is occurring in the hallway. This discrepancy can be explained away by the system generalising that both locations can indeed contain patient beds. When our case-based systems retrieves a matching case, the system has no explicit knowledge stating that a hallway can contain hospital beds. The initial match is of a syntactical nature only. However, it can use its general knowledge and the reasoning mechanism of plausible inheritance to generate an explanation supporting the hypothesis that beds can be located in the hallway, for example because they are both some kind of room, and beds are some kind of object that have a room as a location. Therefore, as all other parameters are consistent with a ward round, the system assumes that it is indeed a ward round situation.

The explanation used by the system in this example states that a hallway is a room and can therefore contain a hospital bed.

2.4.2 Elucidate

We now want to *generate an explanation for the user that tells the user why the system assumes a certain situation*. The system will make use of all available sources of knowledge in order to gain the user's confidence in its capabilities. It will also have to consider the user's goals when choosing a specific explanation. It has been shown that simply presenting the reasoning trace is not always sufficient (it can even be counter productive) [Majchrzak and Gasser, 1991; Gregor and Benbasat, 1999]. The system might therefore generate an after-the-fact explanation, which for example justifies its assumption. Since the ward round situation is occurring in an unusual place, the system will point out the time of the day, the availability of the other expected participants, and the fact that hallways might contain beds, as the reason for its assumption instead of only displaying its generalisation of the location.

The explanation shown to the user is a justification of the system's belief that it is on a ward round.

2.4.3 Identify

After the system has successfully identified the context, it is *using explanations to generate a plan for a reasonable course of action*. Now, it is using only the knowledge sources important for the situation at hand (the context is acting as a focus lense [Kofod-Petersen and Aamodt, 2006]). When we now assume that the system has recognised that we are on a ward round, discussing medical conditions and treatments with several patients, it will try to prepare all the relevant information to be presented to the user. This includes all test results. The system can now ask other available artefacts for test results on the user, and the medical images database can offer a MR image whereas the patient record offers a textual description of the MRI. Because of limitations of handheld devices, the system will for example not be able to display high resolution MR images. When choosing which of the artefacts to query, the system will reject the medical image database and only query the electronic patient record database.

The explanation used by the system is based on the knowledge that a high resolution image displaying device is not available on a ward round.

2.4.4 *Explicate*

Looking at the user centric part again, we are now in need of *generating an explanation for why the systems takes a specific action*. The system will take into account which situation it assumes it is in and the possible goals the user might have for an explanation. In executing its plan, the system proposes its user visit the isolation room with patients who should be kept separate. The user is surprised since he is not aware that any of the patients he should see on the ward round are in the isolation room, and no information on this was exchanged in the morning briefing. The system can then generate an explanation that shows the relevance of the proposal by pointing out that one particular patient had to be moved to the isolation room for medical reasons since the time of the morning meeting, and this information was available via the patient information system. This explanation would not be useful if it had not been established already that we are on a ward round and the aim was to visit the patients. Vice versa, if the system generates a justification for its assumption of being on a ward round, this would still not satisfy the need of the user to know why he should go to the isolation room.

The explanation shown to the user is pointing out the relevance of performing a particular action, namely visiting the isolation room.

2.5 Modelling Context

In this section, we describe the research results in terms of context modelling. We will look at an activity theory inspired context model and see how it integrates with existing approaches in the fields of context aware and pervasive computing. The last subsection will outline how an ethnographic study of work place situations can be used to populate such a context model with situation and domain specific data.

2.5.1 Activity Theory

Our aim is now to identify which aspects of an activity theory based analysis can help us to capture a knowledge level view of contextual knowledge that should be incorporated into an ambient intelligent system. This contextual knowledge should include knowledge about the acting subjects, the objects towards which activities are directed and the community as well as knowledge about the mediating components, like rules or tools.

Traditional Context Model

The context model used in this work draws on a subjective view of situations. That is, even though the model is general, any instance of the model belongs to one user only. Thus, as in Zibetti et al. [2001], any situation will be described from the personal perspective, leading to the possibility of many instances describing the "same" situation. This is in contrast to the leading perspective, where a system will describe *objective* situations.

In the extreme consequence the model used by any subject could also be subjective and unique. However, to avoid the problem of a *tabula rasa* and based on our acceptance of objective descriptions of at least some aspects, we have chosen a pragmatic view on how to model context. The model is based on the definition of context given by Dey [2001], applying the following definition:

> Context is the set of suitable environmental states and settings concerning a user, which are relevant for a situation sensitive application in the process of adapting the services and information offered to the user.

This definition from Dey does not explicitly state that context is viewed as knowledge. However, we believe that the knowledge intensive approach is required if we wish a system to display many of the characteristics mentioned in the introduction. At the same time we also adhere to the view advocated by Brézillon and Pomerol [1999] that context is not a special kind of knowledge. They argue that context is in the eye of the beholder: "...knowledge that can be qualified as 'contextual' depends on the context!" [Brézillon and Pomerol, 1999, p.7]

Even though we argue for a context model where context is not a special type of information, we also believe that only a pragmatical view on context will enable us to construct actually working systems. Following this pragmatic view we impose a meronomy on the context model in the design phase (see Figure 2.5). This meronomy is inherited from the con-

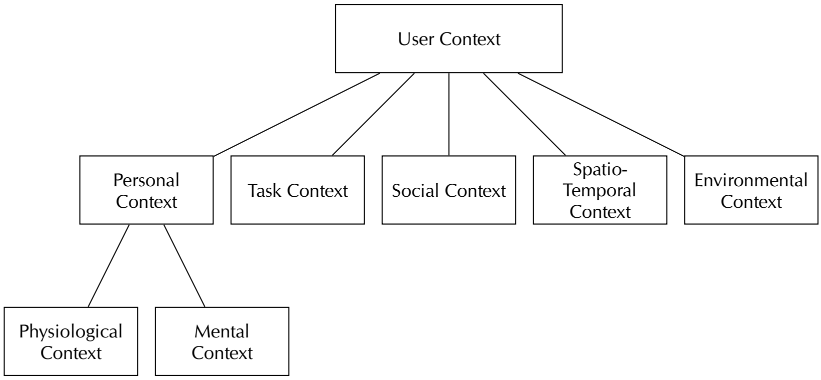

Figure 2.5.: Context meronomy

text aware tradition and adapted to make use of the general concepts we find in activity theory.

The context is divided into five sub-categories (a more thorough discussion can be found in Göker and Myrhaug [2002] or Kofod-Petersen and Mikalsen [2005]):

1. **Environmental context:** This part captures the users surroundings, such as things, services, people, and information accessed by the user.
2. **Personal context:** This part describes the mental and physical information about the user, such as mood, expertise and disabilities.
3. **Social context:** This describes the social aspects of the user, such as information about the different roles a user can assume.
4. **Task context:** the task context describe what the user is doing, it can describe the user's goals, tasks and activities.
5. **Spatio-temporal context:** This type of context is concerned with attributes like: time, location and the community present.

The model depicted in Figure 2.5 shows the top-level ontology. To enable the reasoning in the system this top-level structure is integrated with a more general domain ontology, which describes concepts of the domain (*e.g.*, Operating Theatre, Ward, Nurse, Journal) as well as more generic concepts (Task, Goal, Action, Physical Object) in a multi-relational semantic network. The model enables the system to infer relationships between concepts by constructing context dependent paths between them. We are approaching the situation assessment by applying knowledge-intensive case-based reasoning [Aamodt, 2004]. One of the important aspects of

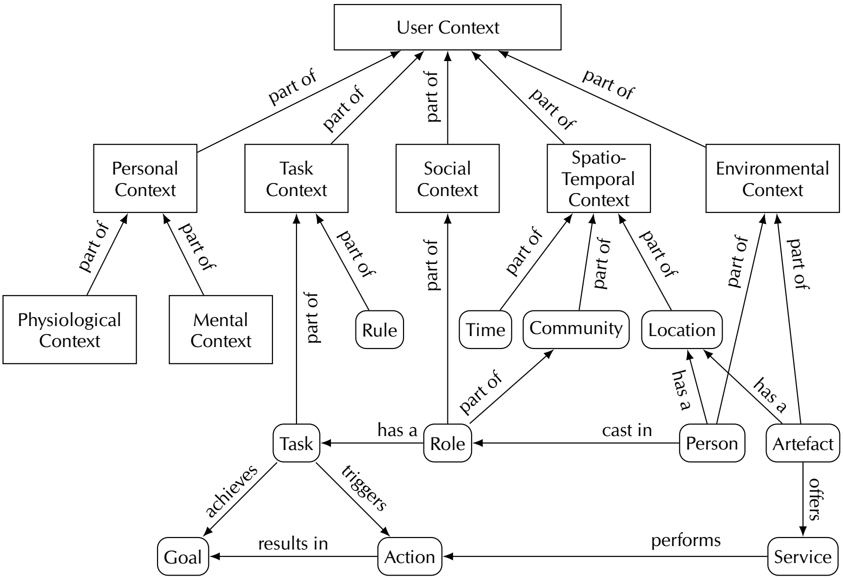

Figure 2.6.: Populated context structure

knowledge-intensive case-based reasoning is the ability to match two case features that are syntactically different, by explaining why they are similar [Aamodt, 1994; Jære et al., 2002].

Activity Theory for the Identification of Context Components

The generic concepts are partly gathered through the use of activity theoretic analysis. These concepts include the six aspects shown in Figure 2.7. The top-level meronomy including the concepts acquired from AT is depicted in Figure 2.6. The context model is now primed to model situations and the activities occurring within them.

If we look at the model we can see how each of the AT aspects is modelled. The *artefact* exists within the *environmental context*, where it can offer *services* that can perform *actions*, which assist the *subject* (described in the *personal context*) in achieving the *goals* of the *role* (in the *social context*) played by the subject. Other *persons*, being part of the situation through the environmental context, can also affect the *outcome* (goal) of the situation. They are cast in different roles that are part of the *community* existing in the *spatio-temporal context*. The roles also implicitly define the *division of*

Table 2.2.: Basic aspects of an activity and their relation to a meronomy of contextual knowledge

CHAT aspect	**Category**
Subject	Personal Context
Object	Task Context
Community	Spatio-Temporal Context
Mediating Artefact	Environmental Context
Mediating Rules	Task Context
Mediating Division of Labour	Social Context

labour in the community. The *rules* governing the subject are found in the *task context*.

As an example, we want the contextual knowledge to contain both information about the acting subject itself (like the weight or size) and the tools (like a particular software used in a software development process). To this end, we propose a mapping from the basic structure of an activity into the meronomy of contextual knowledge as depicted in Table 2.2. We can see that the personal context contains information we would associate with the acting subject itself.

We would like to point out that we do not think that a strict one to one mapping exists or is desirable at all. Our view on contextual knowledge is contextualised itself in the sense that different interpretations exist, and what is to be considered contextual information in one setting is part of the general knowledge model in another one. Likewise, the same piece of knowledge can be part of different categories based on the task at hand.

The same holds for the AT based analysis itself: the same thing can be an object and a mediating artefact from different perspectives and in different task settings. The mapping suggested here should lead the development process and allow the designer to focus on knowledge-level aspects instead of being lost in the modelling of details without being able to see the relationship between different aspects on a socio-technical system level.

As an example, let us consider a software development setting where a team is programming a piece of software for a client. The members of the team are all *subjects* in the development process. They form a *community* together with representatives of the client and other stake-holders. Each member of the team and personnel from other divisions of the software

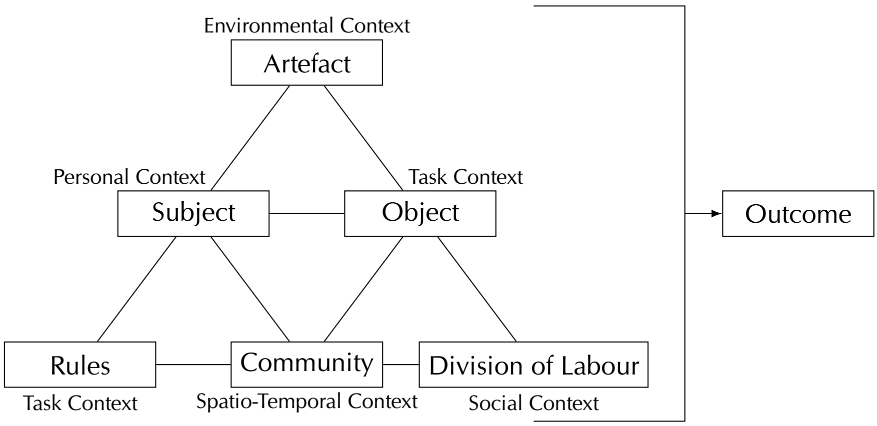

Figure 2.7.: Mapping from activity theory to context model

company work together in a *division of labour*. The *object* at hand is the unfinished prototype, which has to be transformed into something that can be handed over to the client. The task is governed by a set of *rules*, some explicit, like coding standards, some implicit, like what is often referred to as a working culture. The programmers use a set of *mediating artefacts* (tools), like methods for analysis and design, programming tools, and documentation.

When we design a context aware system for the support of this task, we include information about the user of the system (*subject*) in the *personal context* and about the other team members in the *environmental context*. Aspects regarding the special application a developer is working on (*objects*) are part of the *task context*, it will change when the same user engages in a different task (lets say he is looking for a restaurant). The *rules* are part of the *task context* since they are closely related to the task at hand – coding standards will not be helpful when trying to find a restaurant. We find the tool aspects (*artefacts*) in the *environmental context* since access to the different tools is important for the ability of the user to use them. Knowledge about his co-workers and other stake-holders (*community*) are modelled in the *spatio-temporal context*. Finally, his interaction with other team members (*division of labour*) is described as part of the *social context*.

In the design process, we can also make use of the hierarchical structure of activities. On the topmost level, we can identify the *activities* the context aware system should support. By this, we can restrict the world view of the system and make the task of developing a context model manageable.

Furthermore, we can make use of the notion of *actions* to identify the different situations the system can encounter. This helps us to asses the different knowledge sources and artefacts involved in different contexts, thereby guiding the knowledge acquisition task. Finally, since *operations* are performed subconsciously, we get hints on which processes should be supported by automatic and proactive behaviour of the system.

Let us consider our example again. We know that the *activity* we want to support is the development of an IT system. Therefore, we can restrict ourselves to facets of the world which are related to the design process, and we do not (necessarily) have to take care of supporting e.g. meetings some of the team members have as players at the company's football team. On the other hand, the system has to be concerned with meetings with the customer. Furthermore, different *actions* which are also part of the activity should be supported, like e.g. team meetings or programming sessions, and the different *actions* involved can lead to the definition of different situations or contexts.

A context aware application therefore should know at all times in which *action* the user is engaged. This is, in fact, the main aspect of our understanding of the term *context awareness*. At last, to support the *operations* of the user, it might be necessary to proactively query different knowledge sources or request other resources the user might need without being explicitly told to do so by the user. This is at the core of what we refer to as *context sensitivity* in order to distinguish between these two different aspects of context.

It is important to keep in mind that the hierarchical structure of activities is in a constant state of flux. Activity theory is also capable of capturing changing contexts in break-down situations. Lets consider that a tool used in the development process, such as a compiler, stops working. The operation of evoking the compiler now becomes a conscious action for the debugging process. The focus of the developer shifts away from the client software to the compiler. He will now be involved in a different task where he probably will have to work together with the system administrators of his work-station. In this sense other aspects of the activity, such as the community, change as well. It is clear that the contextual model should reflect these changes. The ability of activity theory to identify possible break-down situations makes it possible for the system designer to identify these possible shifts in situation and model the anticipated behaviour of the system.

Populating the context model

We now have a well defined semantic network serving as a knowledge model that is sound both from an activity theory viewpoint and from the tradition of context aware computing. The next step is to populate the model with data from real world situations.

To gather data about work processes, we have designed forms for a study which allow us to focus on different parts of an activity theoretic analysis of the work process. The forms had to meet certain requirements:

- It should be possible to clearly identify the different activities the users were involved in. Furthermore, the goal for each situation should be identified, even if the users did not explicitly state these goals. This would enable us to identify the different outcomes anticipated by the users, and eventually could help us in building a model capturing the hierarchical structure of activities.
- The artefacts used should be identified, and different forms of use for these artefacts should be recorded. This would give us hints about the mediating role of artefacts. Special interest should be given to the use of infomation sources.
- The different entities involved in the activity as depicted in the basic triangle (see Figure 2.3) should, if observable, be described in order to be able to directly connect the data collected to the knowledge level model.
- By observing the praxis of using artefacts, deeper insight on externalisation of cognitive processes can be gained. Although this is not in the scope of our current work, a study design which takes this aspect into consideration could help us in evaluating the capabilities an intelligent system would have to provide to its users in order to be seen as an intelligent partner.
- Although a truly intelligent system would be able to adapt itself to completely new situations, we consider the usage situation, e.g. with regard to the governing rules and the capabilities of the tools used, as being relatively constant. Therefore, our study design did not particularly deal with issues of continuous development.

At the same time, the resulting form could not be too extensive since it should be able to be filled in by a single person observing the activities. The end result was a form which captured essentially the following aspects:

- **Location:** The room where the situation occurred
- **User:** The user of the system
- **Role:** The role of the user
- **Present:** Other persons present

- **Role:** The role of each of the persons present
- **Patient:** The ID of the patient in question
- **Time:** The time of day
- **Source:** Information sources and targets
- **I/O:** The direction of the information flow
- **Information:** Type of information

An empirical study which allowed us to collect both domain specific and situation specific data was performed and is described in more detail in Paper G. This serves as a proof of concept that activity theory is not only fruitful for developing a context model, but can also aid in populating the model with specific data.

2.5.2 Semiotics

In this section, we will give our basic understanding of how semiotics can be used to understand the peculiarities of user interaction with ambient intelligent systems. The basic concept of the chosen interpretion of semiotics is the sign, a triadic relation of a signifier, a signified, and object. We look at the process of sense-making, where a representation *(signifier)* and its mental image *(signified)* refer to an entity *(object)* (the meaning of a sign is not contained within a symbol, it needs its interpretation).

On the background of semiotics, meaningful human communication is a sign process. It is a process of exchanging and interpreting symbols referring to objects. The user of a computer systems sees his interaction with this system on this background. When typing a letter, he does not send mere symbols, but signs to the computer, and the feedback from the machine, the pixels on the screen, are interpreted as signs: to the user, the computer is a "semiotic machine". The question that arises is whether a computer is actually itself taking part in the sense making process.

On one hand, following for example Kant, human understanding has as a necessary constituent the ability to conceptualise perceived phenomena through an active, discursive process of making sense of the intuitive perception [Kant, 1787, p. 58]. Following this understanding, computer systems are only processing signals, lacking the necessary interpreting capabilities humans have. They only manipulate symbols without conceptualising them. However, intelligence is in the eye of the beholder, and it can be argued that even mere signal processing units can appear as sign processors to the human if they sufficiently mimic human behaviour.

On the other hand, we can take a pragmaticist approach, following for example Peirce and Dewey, and focus not on whether the machine is itself a sense maker, but on how its use changes the ongoing socio-technical process, and whether it can mediate the sense making process. From this point of view, the computer can be a sense making agent if its actions are appropriate in terms of the user's expectations.

Both approaches lead to a change in the issues we deal with when constructing an ambient intelligent system. The problem is transformed from one where the issue is to build a machine which itself realises a sense making process to one in which the issue is to build a computer thats actions are appropriate for the situation it is in and which exhibits sufficient sign processing behaviour.

We argue that, in order to make a pervasive, ambient intelligent system that behaves intelligently in a situation, it must be able to execute actions that make a difference to the overall sense making process in a given context. This differs from the interaction with traditional systems in which case the sense-making falls wholly on the side of the human user: You do not expect a text processor to understand your letter, but you expect an ambient intelligent system to display behaviour suggesting that it understands relevant parts of the situation you are in. When interacting with ambient intelligent systems, the user should be facilitated to subscribe to the sense making abilities of the artefacts.

One important challenge here is the features that allow the system to show its capabilities. This can be described as a communication problem: the system has to interpret the actions of the user and information perceived about the context in a meaningful way and itself present results that make sense for the user. This process of sense-making is highly interactive: an intelligent partner in a communication process asks (meaningful) questions if an unclear situation occurs and is able to explain its own actions. Therefore, it is desirable that the artefacts mimic some abilities usually ascribed to humans, like this explanatory capacity. The semiotic approach is useful to analyse this sense-making process with the help of transferring knowledge about similar processes from other semiotic domains.

In Paper K, we have outlined how semiotics can be put to use when dealing with problems of *abstract concepts* when dealing with ambient intelligent systems.

2.6 Modelling Explanations

In this section, we first highlight how *explanation goals* focus on user needs and expectations towards explanations and help to understand *what* the system has to be able to explain and *when* to explain something. In the second step, we introduce some patterns which can help to model explanatory needs in the requirements engineering process for an intelligent system.

2.6.1 User Goals

We suggest several explanation goals for case-based reasoning systems which are valid for knowledge-based systems, in general (see Paper E). The user of an intelligent system has certain expectations towards explanations, he has an interest depending on his own historicity and the state of the socio-technical system. Different explanations are of different utility for the user. The task the system is faced with is to decide which explanations will be most useful for the user, and then to decide upon the amount of information which has to be presented. We argue that the goals we present are indeed reachable because the systems we consider are mostly designed to perform limited tasks for a limited audience, thus making it possible to make reasonable assumptions about the user's goals and the explanation context. The identified explanation goals are:

Transparency:

Explain how the system reached the answer
"I had the same problem with my car yesterday, and charging the battery fixed it."

The goal of an explanation of this kind is to impart an understanding of how the system found an answer. This allows the users to check the system by examining the way it reasons and to look for explanations for why the system has reached a surprising or anomalous result. If transparency is the primary goal, the system should not try to oversell a conclusion it is uncertain of. In other words, fidelity is the primary criterion even though such explanations may place a heavy cognitive load on the user. The original *how* and *why* explanations of the MYCIN system [Clancey, 1983] are good examples.

This goal is most important with knowledge engineers seeking to debug the system and possibly domain experts seeking to verify the reasoning process [Gregor and Benbasat, 1999]. It is also reasonable to expect that in domains with a high cost of failure it can be supposed that the user wishes to examine the reasoning process more thoroughly.

Justification:

Explain why the answer is a good answer

"You should eat more fish - your heart needs it!"
"My predictions have been 80% correct up until now."

This has the goal of increasing confidence in the advice or solution offered by the system by giving some kind of support for the conclusion suggested by the system. This goal allows for a simplification of the explanation compared to the actual process the system goes through to find a solution. Potentially, this kind of explanation can be completely decoupled from the reasoning process, but it may also be achieved by using additional background knowledge or reformulation and simplification of knowledge that is used in the reasoning process.

Empirical research suggests that this goal is most prevalent in systems with novice users [Mao and Benbasat, 2000], in domains where the cost of failure is relatively low, and in domains where the system represents a party that has an interest in the user accepting the solution.

Relevance:

Explain why a question asked is relevant
"I ask about the more common failures first, and many users do forget to connect the power cable."

An explanation of this type would have to justify the strategy pursued by the system. This is in contrast to the previous two goals that focus on the solution. The reasoning trace type of explanations may display the strategy of the system implicitly, but it does not argue why it is a good strategy. In conversational systems, the user may wish to know why a question asked by the system is relevant to the task at hand. It can also be relevant in other kinds of systems where a user would like to verify that the approach used by the system is valid. In expert systems, this kind of explanation was introduced by NEOMYCIN [Clancey, 1983].

Conceptualisation:

Clarify the Meaning of Concepts
"By 'conceptualisation' we mean the process of forming concepts and relations between concepts."

One of the lessons learned after the first wave of expert systems had been analysed was that the users did not always understand the terms used by a system. This may be because the user is a novice in the domain, but also because different people can use terms differently or organise the knowledge in different ways. It may not be clear, even to an expert, what the system means when using a specific term, and he may want to get an explanation of what the system means when using it. This requirement for providing explanations for the vocabulary was first identified by Swartout and Smoliar [1987].

Learning:

Teach the user about the domain
"When the headlights won't work, the battery may be flat as it is supposed to deliver power to the lights."

All the previous explanation goals involve learning – about the problem domain, about the system, about the reasoning process or the vocabulary of the system. Educational systems, however, have learning as the primary goal of the whole system. In these systems, we cannot assume that the user will understand even definitions of terms, and may need to provide different explanations for people at different levels of expertise. The goal of the system is typically not only to find a good solution to a problem, but to explain the solution process to the user in a way that will increase his understanding of the domain. The goal can be to teach more general domain theory or to train the user in solving problems similar to those solved by the system. This means that the explanation is often more important than the answer itself. Systems that fulfil the relevance and transparency goals may have some capabilities in this area, but a true tutoring system must take into account how humans solve problems. It cannot attempt to teach the user a problem solving strategy that works well in a computer but that is very hard to reproduce for people.

2.6.2 Problem Frames

The use of patterns [Alexander et al., 1977] is common for different software engineering approaches. Patterns can be used in different software development phases and they can have different foci. We can also identify knowledge engineering approaches making use of patterns.

In the initial phases of the requirements engineering process, the use of *problem frames* as proposed by Jackson [2001] is a method to classify software development problems. Problem frames are focus on the world and attempt to describe the problem and its solution in the real world. Problem frames introduce concepts like 'Information Display' and 'Commanded Behaviour'.

Jackson's set of basic problem frames can be extended to be better able to model domain specific aspects. For example, Hatebur and Heisel [2005] introduce new problem frames for security problems. Their proposal includes problem frames for issues like 'Accept Authentication' and 'Secure Data Transmission'. They also provide architectural patterns connected to these problem frames.

The main purpose of any problem frame is to propose a machine that improves the combined performance of itself and its environment by describing the machine's behaviour in a specification. To explain ones behaviour a problem frame must be constructed that relates the behaviour the system shows to different parts of knowledge used by the system to support the chosen course of action in a specification.

Jackson [2001] originally described five different basic frames, each of which comes in different *flavours* and *variants*: 'required behaviour', 'commanded behaviour', 'information display', 'simple workpieces' and 'transformation'. Each basic frame has its own requirements, domain characteristics, domain involvement, and frame concern.

In general, a problem frame assumes a user driven perspective. Except for the 'required behaviour' basic frame, each frame assumes that the user is in control and dictates the behaviour of the machine. Since intelligent systems (ideally) take a much more proactive approach and mixed-initiative issues become relevant, new problem frames addressing these topics have to be developed. For the purposes of this work, we will focus exclusively on frames targeting explanatory aspects and will not discuss other types of problem frames.

Problem frames have a standardised form of representation that are known as problem frame diagrams. Diagramatic representation takes the form of dashed ovals, representing the requirements, plain rectangles, denoting application domains, and a rectangle with a double vertical stripe, standing for the machine (or software machine) domain to be developed. These entities become the nodes of the frame diagram. They are connected by edges, representing shared phenomena and denoting an interface. Dashed edges refer to requirement references. Dashed arrows designate constraining requirement references.

The domains can be of different types, indicated by a letter in the lower right corner. Here, a 'C' stands for a *causal* domain whose properties include predictable causal relationships among its phenomena. A 'B' denotes a *biddable* domain that lacks positive predictable internal behaviour. Biddable domains are usually associated with user actions. Finally, an 'X' marks a *lexical* domain. Such a domain is a physical representation of data and combines causal and symbolic phenomena.

In the software development process, problem frames are used in the

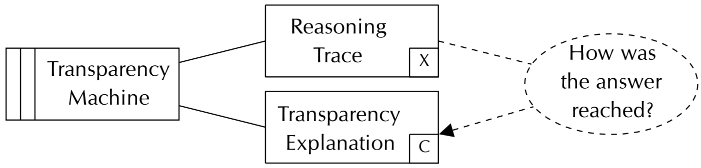

Figure 2.8.: Transparency Explanation. An explanation supporting this goal gives the user some insight into the inner working of the system. To this end, the system inspects its own reasoning trace when formulating the explanation.

following way. First, we start with a *context diagram*, which consists of domain nodes and their relations, but without the requirements. Afterwards, the context diagram is divided into subproblems. The resulting subproblems should, whenever possible, relate to existing generic *problem frames*. These generic problem frames are hereby instantiated to describe the particular subproblem at hand.

In the remainder of this subsection, we propose a set of new generic problem frames to capture aspects of explanations connected to the aforementioned different user goals identified in the previous subsection.

Transparency Explanation

The goal of an explanation of this kind is to impart an understanding of how the system found an answer, allowing the user to test the system by querying the reasoning and when unusual conclusions are reached to search for explanations.

The frame diagram depicted in Figure 2.8 highlights that in order to support the transparency goal, the software system has to inspect its reasoning trace and represent the relevant facts of its reasoning process to the user. We expect a transparency explanation usually to be given after the reasoning process has terminated.

Justification Explanation

This is primarily a confidence raising goal, aimed at increasing the confidence of the user in the advice or solution offered by the system. It allows for a simplification of the explanation.

An explanation supporting the justification goal, as shown in Figure 2.9, has not only to take the reasoning of the machine into account, but it will

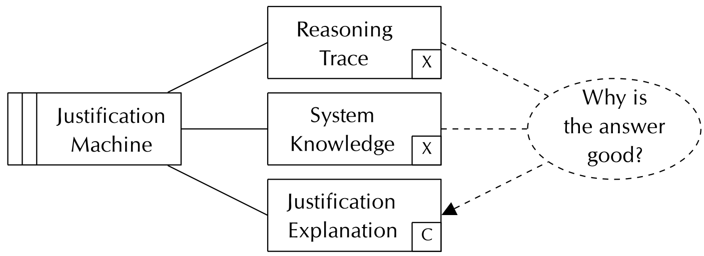

Figure 2.9.: Justification Explanation. In contrast to the transparency explanation, in justification explanations, the user is not as interested in why the system exposes a particular behaviour, but is more interested in having evidence supporting the veracity of this behaviour. Therefore, other knowledge has to be taken into account besides the reasoning trace.

also make use of other parts of the system's knowledge in order to generate after the fact explanations supporting its actions or decisions. Since justification explanations complement transparency explanations, we expect it to typically be given after the reasoning process has terminated.

Relevance Explanation

In contrast to the previous two goals that focus on the solution, a relevance explanation justifies the strategy pursued by the system.

Since this goal, depicted by the frame diagram in Figure 2.10, is of particular interest for mixed-initiative systems, the explaining machine has to relate its explanation both to its own dialogue with the user (and here in particular the questions asked by the system or the actions performed), the reasoning trace (in oder to relate to the situation the system assumes it is in) and the system knowledge relevant. In contrast to the first two goals, an explanation supporting this goal is important to be given during the reasoning process of the system.

Conceptualisation Explanation

The goal of conceptualization explanations is to ensure that there is a sigular understanding of terminology.

This explanation machine, represented with the frame diagram depicted in Figure 2.11, builds on its own system knowledge. This high-

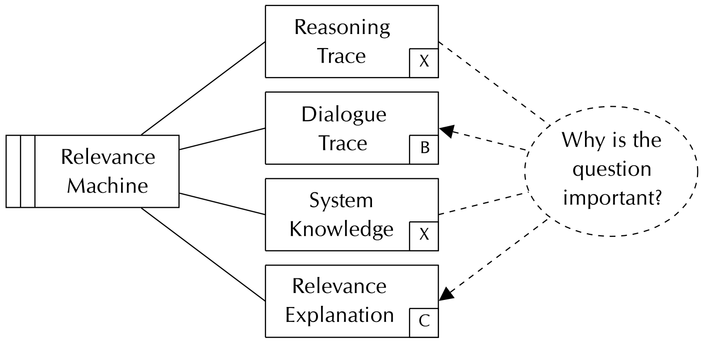

Figure 2.10.: Relevance Explanation. An explanation supporting this goal should instil confidence by indicating that the system's behaviour is connected to the task at hand. Consequently, the reasoning and dialogue traces should be taken into account as well as other (domain) knowledge.

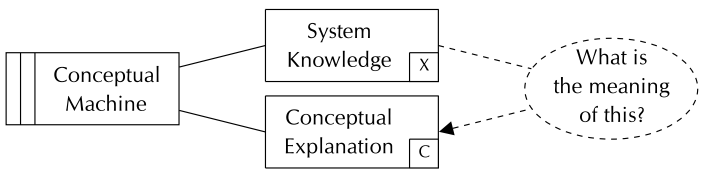

Figure 2.11.: Conceptualisation Explanation. By giving a conceptualisation explanation, the system explicates its own conceptualisation of the domain or the task at hand to the user. Hence, it will connect the concept to be explained with its own knowledge components.

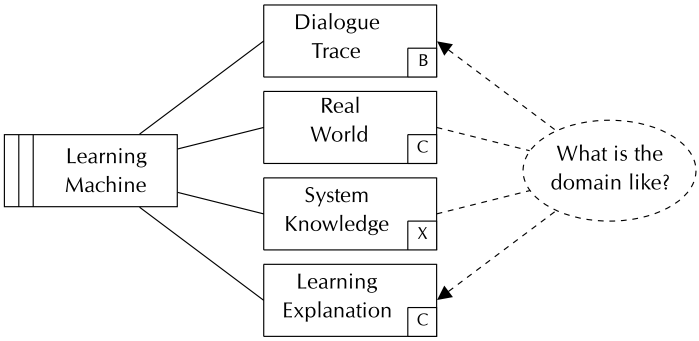

Figure 2.12.: Learning Explanation. This goal is special, since it focuses on the user's interest in the application domain (hence the real world), and not on some particular behaviour of the system.

lights the fact that explanations supporting this goal should set unknown concepts in the context of the other knowledge the system has, and which is expected to be shared with the user already. Conceptualisation explanations are important both during the reasoning process (e.g. in addition to a relevance explanation) and after the reasoning process has terminated (e.g. in addition to a justification explanation).

Learning Explanation

This goal is of specific interest for educational applications, which have learning as the primary goal of the whole system. Here, typically the goal of the system is not only to find a good solution to a problem, but to explain the solution process to the user in a way that will increase his understanding of the domain.

The Figure 2.12 highlights this fact by pointing out that the explanatory machine has to connect its own system knowledge with the real world (representing the application domain) in order to generate explanations supporting the user in gaining a better understanding of the application domain. The explanation given should also relate to the system's assumptions about its user, which is influenced by the dialogue trace. Because of the nature of this goal, it will usually be important during the system's reasoning process.

This section has introduced explanation problem frames on a conceptual level. In Paper L, we show how these frames can be used to (re-)

design an explanation aware ambient intelligent system by using our existing hospital ward system described e.g. in Paper G as an example.

2.7 Modelling Explanations in Context

This section gives an overview about how the pieces outlined in the previous sections come together. It is shown how a socio-technical approach, exemplified through the use of activity theory, is put to use to model explanatory capabilities in context. It is further outlined how this can be used in a workflow from an ethnoraphical analysis of existing workplace situations down to a specification of the knowledge necessary in the different knowledge holding components of a case-based reasoning system.

2.7.1 *Activity Theory for Explanations in Context*

As described above, activity theory has been used to recognise contextual facets of a work situation. By integrating the knowledge necessary for supporting the different explanatory goals of the user with this contextual information, the explanatory capabilities of the system are coupled with the different contexts. Hence, the hypothesis is that this explanatory knowledge will indeed primarily be used in the appropriate context.

We will now explore the relations between the basic properties of activity theory and explanation goals.

Hierarchical structure of activity: The fact that activities are hierarchically structured, and that changes in these structures occur, facilitates certain explanation goals. Actions that are performed often will be transformed into operations. Vice versa, if an anticipated outcome of an operation does not occur, non-conscious operation will become conscious actions. This is called a breakdown situation. The explanatory capabilities of a system should support this. In fact two goals are relevant in these situations:

- **Transparency:** If parts of the non-conscious operations are carried out by artefacts, the system might need sufficient knowledge to explain the artefacts inner working in case of a breakdown.
- **Relevance:** If an artefact involved in an action can behave differently than expected, it should be made clear why the unexpected behaviour occurred.

Object-orientedness: In the activity theoretical sense of the term object-oriented, the meaning of this term is twofold. On one hand, it highlights that all human activities have an objective, a goal, and therefore points towards the mental part of an activity. On the other hand, it refers to the fact that this mental objectives are directed towards the physical world. This holds for automated processes insofar as the automation already assumes a goal, and is supposed to support this goal:

- **Transparency:** It should be possible for a system to explain its relation to the physical processes.

- **Justification:** An intelligent system should be able to explain its goals to the user.

Mediation: Every activity will incorporate some tools, be it physical (like machinery) or psychological artefacts (like language). If parts of the activity are carried out by an intelligent artefact, this artefact both acts as a mediator in the physical world and as a mediator of the psychological processes of the user:

- **Justification:** The system should be able to explain the connection between its actions and the reasoning process.

Continuous development: The aspect of continuous development deals with the continuous change in the way we interact with the world. Both the user's activities and the artefacts used are changing. It should be noted that this includes the necessity for an intelligent system to adopt to changes over time:

- **Learning:** The system should be able to support the user's learning processes. If the system is extended, or new capabilities are included, the system should be able to act as teacher. It should therefore incorporate knowledge about how the new component facilitates the problem solving process.

Distinction between internal and external activities: Activity theory tries to overcome the dichotomy of mental processes and the outside world by focussing on the relation between internal and external activities. It is therefore crucial that the system supports the user in building an understanding of the artefacts used.

- **Conceptualisation:** The system should support the user's understanding of it by providing means of explaining its own world model to him.

Not all explanation goals can be satisfied by an activity theoretical perspective alone. Some goals can only be satisfied by inspecting other parts of the knowledge model, either in all cases or for certain situations. As an example, when recognising a situation the transparency goal can be satisfied by supplying a trace of the reasoning process used for classification. The different sources of knowledge required to satisfy the different goals will be further discussed in the following section.

2.7.2 *Goals, Kinds, and Knowledge Containers*

Roth-Berghofer [2004] has explored some fundamental issues with different useful kinds of explanations and their connections to the different knowledge containers of a case-based reasoning system. Based on earlier findings from natural language explanations in expert systems, five different kinds of explanation are identified: *conceptual explanations*, which map unknown new concepts to known ones, *why-explanations* describing causes or justifications, *how-explanations* depicting causal chains for an event, *purpose-explanations* describing the purpose or use of something, and *cognitive explanations* predicting the behaviour of intelligent systems. Roth-Berghofer, further on, ties these different kinds of explanation to the different knowledge containers of case-based reasoning systems [Richter, 1995], namely case base, similarity measure, adaptation knowledge, and vocabulary.

Building on these two works, we are investigating a combined framework of user goals and explanation kinds, see Paper C. The goal of this work is to outline a design methodology that starts from an analysis of usage scenarios in order to be able to identify possible expectations a user might have towards the explanatory capabilities of an intelligent system. The requirements recognised can further on be used to identify which kind of knowledge has to be represented in the system, and which knowledge containers are best suited for this task.

In Paper I, we have revisited this mapping from goals to kinds and further on to knowledge containers in the light of our work on explanations in context. We now have a workflow which allows us to start with a socio-technical analysis of workplace environments and end up with specific requirements on the knowledge to be contained in the different components of a case-based reasoning system and which can be generalised to other types of intelligent systems. We can 1. start from an ethnographic analysis of workplace situations, 2. identify both contextual and explanation related aspects, 3. model these in the requirements engineering process with the help of problem frames, 4. break down the outward looking user goals into system specific kinds of explanation, and finally 5. describe what knowledge has to be represented in the different knowledge containers of case-based reasoning systems.

Bibliography

Agnar Aamodt. Knowledge-intensive case-based reasoning in CREEK. In Funk and Calero [2004], pages 1–15.

Agnar Aamodt. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. PhD thesis, Norwegian Institute of Technology, Department of Computer Science, Trondheim, May 1991. URL http://www.idi.ntnu.no/grupper/su/publ/phd/aamodt-thesis.pdf.

Agnar Aamodt. Explanation-driven case-based reasoning. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-Based Reasoning: Proceedings of the 1st European Workshop EWCBR 1993 – Selected Papers*, volume 837 of *Lecture Notes in Computer Science*, pages 274–288, Berlin Heidelberg, 1994. Springer. ISBN 3-540-58330-0.

Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. Oxford University Press, New York, 1977.

Peter Bøgh Andersen and Martin Brynskov. The semiotics of smart appliances and pervasive computing. In *Semiotics and Intelligent Systems Development* Gudwin and Queiroz [2006], pages 211–255. ISBN 1599040638.

Aristotle. *Analytica Posteriora*. Akademie, 1998. URL http://classics.mit.edu/Aristotle/posterior.html.

Susanne Bødker. Activity theory as a challenge to systems design. In Hans-Erik Nissen, Heinz K. Klein, and Rudy Hirschheim, editors, *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pages 551–564. North Holland, 1991.

Patrick Brézillon and Jean-Charles Pomerol. Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62(3):223–246, 1999.

Bruce G. Buchanan and Edward H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison Wesley, Reading, 1984.

William J. Clancey. The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence*, 20(3):215–251, 1983.

Rodney Clarke, Aditya Ghose, and Aneesh Krishna. Systemic semiotics as a basis for an agent oriented conceptual modeling methodology. In *Semiotics and Intelligent Systems Development* Gudwin and Queiroz [2006], pages 256–285. ISBN 1599040638.

Daniel Cohnitz. Explanations are like salted peanuts. In Ansgar Beckermann and Christian Nimtz, editors, *Proceedings of the Fourth International Congress of the Society for Analytic Philosophy*, 2000. `http://www.gap-im-netz.de/gap4Konf/Proceedings4/titel.htm` [Last access: 2004-08-11].

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An evaluation of the usefulness of case-based reasoning explanation. In Kevin D. Ashley and Derek Bridge, editors, *Case-Based Reasoning Research and Development: Proceedings ICCBR 2003*, number 2689 in Lecture Notes in Artificial Intelligence, pages 122–130, Berlin Heidelberg, 2003. Springer.

Clarisse Sieckenius de Souza. *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press, 2005. ISBN 0262042207. QA76.9.H85 .D465 2005.

Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Hamid R. Ekbia and Ana G. Maguitman. Context and relevance: A pragmatic approach. *Lecture Notes in Computer Science*, 2116:156–169, 2001.

Peter Funk and Pedro A. González Calero, editors. *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2004. Springer.

Pablo Gervás and Kalyan May Gupta, editors. *Proceedings of the ECCBR 2004 Workshops*, Technical Report 142-04, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Ayse Göker and Hans Inge Myrhaug. User context and personalisation. In *Workshop proceedings for the sixth European Conference on Case Based Reasoning*, 2002.

Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, 23 (4):497–530, 1999.

Ricardo Gudwin and Joao Queiroz. *Semiotics and Intelligent Systems Development*. IGI Publishing, Hershey, PA, USA, 2006. ISBN 1599040638.

Michael A.K. Halliday. *Language as a Social Semiotic: the social interpretation of language and meaning*. University Park Press, 1978.

Michael A.K. Halliday and Christian M.I.M. Matthiessen. *An Introduction to Functional Grammar, Third edition*. Arnold, London, UK, 2004.

Denis Hatebur and Maritta Heisel. Problem frames and architectures for security problems. In Rune Winther, Bjoern Axel Gran, and Gustav Dahll, editors, *Proceedings of the 24th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, number 3688 in Lecture Notes in Computer Science, pages 390–404, Berlin, 2005. Springer.

Michael Jackson. *Problem Frames – Analysing and Structuring Software Development Problems*. Addison-Wesley, Boston, MA, 2001.

Martha Dørum Jære, Agnar Aamodt, and Pål Skalle. Representing temporal knowledge for case-based prediction. In Susan Craw and Alun Preece, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2002*, number 2416 in Lecture Notes in Artificial Intelligence, pages 174–188, Aberdeen, 2002. Springer.

Immanuel Kant. *Kritik der reinen Vernunft (2. Auflage)*. Akademie, 1787.

Anders Kofod-Petersen and Agnar Aamodt. Contextualised ambient intelligence through case-based reasoning. In Thomas R. Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *Proceedings of the Eighth European Conference on Case-Based Reasoning (ECCBR 2006)*, volume 4106 of *Lecture Notes in Computer Science*, pages 211–225, Berlin, September 2006. Springer.

Anders Kofod-Petersen and Marius Mikalsen. Context: Representation and reasoning – representing and reasoning about context in a mobile environment. *Revue d'Intelligence Artificielle*, 19(3):479–498, 2005.

Kari Kutti. Activity theory as a potential framework for human-computer interaction research. In Nardi [1996b], pages 17–44.

Bruno Latour. Technology is society made durable. In John Law, editor, *A Sociology of Monsters*, pages 103–131. Routledge, 1991.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

David B. Leake, editor. *Case-Based Reasoning: Experiences, Lessons, & Future Directions*. AAAI Press/MIT Press, Menlo Park, 1996.

David B. Leake. Workshop introduction: ECCBR-04 workshop on explanation in CBR. Slides, last access 22.02.2007, 2004. URL https://www.cs.tcd.ie/research_groups/mlg/ecbrws2004/.

David B. Leake and David McSherry. Special issue on explanation in case-based reasoning. *Artificial Intelligence Review*, 24(2), October 2005. ISSN 0269-2821.

Doug Lenat. The dimensions of context-space. Technical report, Cycorp, Austin, TX, October 28 1998. URL http://www.cyc.com/doc/context-space.pdf.

Aleksei N. Leont'ev. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

Ann Majchrzak and Les Gasser. On using artificial intelligence to integrate the design of organizational and process change in us manufacturing. *AI and Society*, 5:321–338, 1991.

Ji-Ye Mao and Izak Benbasat. The use of explanations in knowledge-based systems: Cognitive perspectives and a process-tracing analysis. *Journal of Managment Information Systems*, 17(2):153–179, 2000.

Eric Monteiro. Actor-network theory. In Claudio U. Ciborra, editor, *From Control to Drift*, pages 71–83. Oxford University Press, 2000.

Daisy Mwanza. Mind the gap: Activity theory and design. Technical Report KMI-TR-95, Knowledge Media Institute, The Open University, Milton Keynes, 2000.

Bonnie A. Nardi. A brief introduction to activity theory. *KI – Künstliche Intelligenz*, 1:35–36, 2003.

Bonnie A. Nardi. Activity theory and human computer interaction. In *Context and Consciousness* Nardi [1996b], pages 7–16.

Bonnie A. Nardi, editor. *Context and Consciousness*. MIT Press, Cambridge, MA, 1996b.

Pinar Öztürk and Agnar Aamodt. A context model for knowledge-intensive case-based reasoning. *International Journal of Human Computer Studies*, 48:331–355, 1998.

Charles Sanders Peirce. New elements (kaina stoicheia). In Carolyn Eisele, editor, *The New Elements of Mathematics by Charles S. Peirce*, volume 4, Mathematical Philosophy, pages 235–263. 1904. URL http://www.cspeirce.com/menu/library/bycsp/stoicheia/stoicheia.htm.

Wolter Pieters. Free will and intelligent machines. Project Report, NTNU Trondheim, 2001.

Michael Polanyi. *Personal Knowledge: Towards a Post-critical Philosophy*. N.Y.: Harper & Row, 1964.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Funk and Calero [2004], pages 389–403.

Ferdinand de Saussure. *Course in General Linguistics*. McGraw-Hill, 1966.

Roger C. Schank. *Explanation Patterns – Understanding Mechanically and Creatively*. Lawrence Erlbaum, New York, 1986.

Roger C. Schank, Alex Kass, and Christopher K. Riesbeck, editors. *Inside Case-Based Explanation*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1994.

Alan Sokal and Jean Bricmont. *Eleganter Unsinn*. dtv, München, 2001.

Peter Spieker. *Natürlichsprachliche Erklärungen in technischen Expertensystemen*. Dissertation, University of Kaiserslautern, 1991.

Thomas M. Strat. Employing contextual information in computer vision. In *DARPA93*, pages 217–229, 1993.

William R. Swartout. What kind of expert should a system be? xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

William R. Swartout and Johanna D. Moore. Explanation in second generation expert systems. In Jean-Marc David, Jean-Paul Krivine, and Reid Simmons, editors, *Second Generation Expert Systems*, pages 543–585. Springer, Berlin, 1993.

William R. Swartout and Stephen W. Smoliar. On making expert systems more like experts. *Expert Systems*, 4(3):196–207, 1987.

Guy Tiberghien. Context and cognition: Introduction. *European Bulletin of Cognitive Psychology*, 6(2):105–119, 1986.

Valentin N. Voloshinov. *Marxism and the Philosophy of language*. Seminar Press, New York, 1973.

Lew S. Vygotsky. *Mind in Society*. Harvard University Press, Cambridge, MA, 1978.

Lew S. Vygotsky. *Ausgewählte Schriften Bd. 1: Arbeiten zu theoretischen und methodologischen Problemen der Psychologie*. Pahl-Rugenstein, Köln, 1985.

Stephen S. Yau, Yu Wang, and Fariaz Karim. Development of situation-aware application software for ubiquitous computing environments. In *29th Annual International Computer Software and Applications Conference (COMPSAC'02)*, volume 1, pages 107–112, Edinburgh, Scotland, July 2002. IEEE Computer Society.

Elisabetta Zibetti, Vicenc Quera, Francesc Salvador Beltran, and Charles Tijus. Contextual categorization: A mechanism linking perception and knowledge in modeling and simulating perceived events as actions. *Modeling and using context*, pages 395–408, 2001.

3 Conclusions

In this chapter, we will highlight the main contributions of our research as well as address some limitations. A short evaluation will be given, before we conclude with an outlook on further work.

3.1 Contributions

The main research contributions of this thesis are:

- Adding a socio-technical perspective to the theoretical foundations of ambient intelligent and explanation aware systems, therewith broadening the scope of the theoretical foundations of artificial intelligence which traditionally focuses on individual aspects of intelligence.
- Improving the requirements engineering process of intelligent systems by introducing patterns for modelling requirements for explanations as a means of communication with the user and coupling these patterns with 1. a socio-technical analysis, 2. traditions in expert systems research, and 3. knowledge level system specification models.

In the following section, we will revisit the research questions we have identified in the introductory chapter and see how this thesis has dealt with them.

1. What are some of the socio-technical issues that have to be addressed when embedding intelligent systems into workplace environments and to what extent do they differ from the problems that traditional, non intelligent systems face?

In the construction of intelligent artefacts, problems that are hard to tackle with traditional computer systems are made solvable or more tractable through the implementation of abilities traditionally ascribed to humans. Systems which use heuristic problem solving strategies can deal with problems that have a large search space. Intelligent systems can handle problem areas that are ill defined. At the same time, humans should be able to interact with intelligent systems in a more intuitive and natural way.

The challenge when trying to achieve the goal of better interaction is to design the artefact in ways that make it easier for the human to subscribe to the intelligent capabilities of the artefact. We have focused on social aspects of human cognition and behaviour and have identified the ability to act and think contextually together with the ability to give explanations about ones reasoning and actions as core attributes of human cognition and reasoning.

When it comes to contextualisation, the definition of ambient intelligence highlights the fact that the artefacts should be aware of the environment [Ducatel et al., 2001]. The use of the word awareness is motivated. By using this word, it is stressed that something more than pure reactiveness is required. Awareness implies the ability to reason about the state as a necessary precondition. Within the notion of awareness is the implicit precondition of reasoning about the state. A simple stimulus-response system, while suitable in many cases, would not qualify as being ambient intelligent.

While the exact type of model and mechanism can be disputed, the symbolic AI community agrees that some kind of knowledge model and symbolic reasoning mechanism is required for achieving such awareness. Such a knowledge model must be structurally defined and populated with knowledge about the world. This is not a matter with which non-intelligent artefacts are faced.

Intelligent artefacts are also distinguished by the need and ability to give contextually appropriate explanations. While it is also possible to integrate information about the functioning of an artefact in traditional computer systems, for example by means of help texts or user manuals, such explanations lack dynamism and it remains largely up to the user to make use of this information, for example by looking through structured frequently asked questions, and no real machine processing takes place. This static, highly structured and generally ad hoc approach is insufficient for the demands of AI systems.

Intelligence, and hence artificial intelligence, must be understood as ensemble effects. The general "problem" is that "intelligence" is not something which can be understood or modelled as an individual, solitary property. It has to be understood in its social setting. If it is to mimic human intelligence in any way, artificial intelligence has to take aspects of sociability into account, side by side with results from the cognitive sciences or philosophy which form a large part of the backdrop for AI. This will have consequences for the design of artefacts as well, since, as McCarthy [2007] has put it, AI "is the science and engineering of making intelligent machines".

2. Can theories from the social sciences, psychology, or linguistics be useful in tackling some of these problems, particularly those with a special focus on intelligent systems?

While there is a plethora of theories for modelling social aspects of human behaviour and reasoning, to date, artificial intelligence has made little use of this research. The reasons for this shortfall can be manifold. It may be that these theories or frameworks are not suited to the tasks at hand in AI. It may also be that the general problem of making *a posteriori* theories of analysis work for *a priori* design tasks has proved too challenging. In response to the latter suggestion, however, general AI research as well as the application of socio-technical theories in other subfields of computer science has shown that this is possible.

Saying that socio-technical theories and models have not attracted wide spread attention does not mean they have been totally neglected. Besides the examples discussed in Chapter 2 (see for example, work on activity theory and context by Kaenampornpan and O'Neill [2004] or research on semiotics and smart appliances by Andersen and Brynskov [2006]), we have the recently reported results from Walton [2007] on speech act theory and explanations, the work of Potter [2007] on integrating discourse theory and knowledge representation, and work on contextual graphs by Brézillon [2007].

Thus, we would dispute the suggestion posited above that such theories cannot be put to work in AI research.

Within the current work, in addition to having identified explanation awareness and ambient intelligence as examples for issues to be tackled, we have also developed a stratified view of the problems faced when introducing AI systems into pre existing socio-technical situations, including the suggestion of potentially beneficial theories. Contextuality and explainability play a crucial role, from socio-technical systems at large down to the individual interaction. We have exemplified the usefulness of this layered approach by pointing to some theories that work at the different levels.

While our results also give us evidence for the claim that such theories are indeed useful, we will defer examination of our own findings until the discussion of the next few research questions since it cannot be meaningful substantiated without looking at the specific theories we have used.

3. If so, what particular theories can help us in understanding the socio-technical settings?

We initially argued for taking several socio-technical theories into account, each taking a different perspective on the problem domain. The choice made after a preliminary literature survey was to take a closer look at actor network theory, activity theory, and semiotics. During the course of the research, however, we have focused primarily on one such theory, namely activity theory. This model has shown its usefulness in several areas described below.

We have also gained some additional insights which were not clear on the outset. First, activity theory has proven its value on more questions than what we had initially envisioned. In fact, we have effectively ceased the use of actor network theory beyond the most general tool set, since we could already handle many of the same topics through activity theory. Second, the problem domain is not only layered as we originally modeled, but there are other dimensions as well. This ties in with other results, for example with Bødker and Andersen [2005] who make a distinction between instrumental and semiotic processes.

One of the strongest results to emerge from this research is the validation that socio-technical theories in general, and activity theory and semiotics in particular, can be used to tackle problems specific to AI systems. While we do not dismiss the potential that other theories may well be equally suited, we have indications that activity theory is particularly useful for a large array of problems. Our initial research on semiotic theories has yielded similar results, and we are convinced that it will prove fruitful to combine aspects of semiotics and activity theory in both the theoretic and engineering part of AI research.

4. How can theoretical results from answering the questions above be made useful for the problem of designing such systems, in particular how can the design process of intelligent systems be improved?

Our work with activity theory resulted in several improvements for dealing with the problem of embedding intelligent artefacts into socio-technical settings: 1. The theoretical concepts of human activities in AT resulted in an improved general context model that is both in line with best practices in engineering and is psychologically plausible. 2. The theory allowed us to design an ethnographic study that assessed both static, domain specific components that have to be integrated into such a knowledge model and dynamic, situation specific information that became the basis for our initial case base. 3. AT allowed us to theoretically define possible situation candidates where need for explanation arose, and 4. the ethnographic study delivered situation specific knowledge about such incidents. 5. Finally, it was possible to tie activity theory to the different user goals towards an explanation, thereby tying it in with both the tradition of knowledge based or expert system research and a design methodology which covers the design process from requirements engineering down to the specification of the contents of the different knowledge containers.

Although our research on semiotics has not been as extensive as our work with activity theory, we have seen promising results 1. for the modelling of abstract concepts in context aware applications and 2. for integrating a concept of multimodality that is of special importance for working with behavioural interfaces as found in ambient intelligent systems. Additionally, although not discussed in any depth in this thesis, we have found strong connections between our cultural historical approach to activity theory and social-semiotics of systemic functional linguistics (SFL). On initial research, it seems likely that the main concepts of SFL can be mapped onto basic properties of knowledge intensive case-based reasoning applications.

3.2 Limitations

Every transformation of theory from one domain to another has numerous potential pitfalls. When theories of human cognition and acting are utilised in artificial intelligence, the boundaries of our understanding of natural and cultural processes are accentuated by the limits of computerised implementations. Every such transformation has the further problem of potential misunderstanding or misrepresentation due to the diversity of cultural backgrounds within the various disciplines involved. It is quite possible that we have fallen into such traps in this instance, however errors such as this are intrinsic to interdisciplinary research, and the benefits of interdisciplinarity outweigh the costs in this respect.

There are further limitations resulting from the constraints of the research situation, either conscious choices to limit analysis, or limitations resulting from time constraints. Certainly, we did not have time to explore all the interesting aspects (yet). For example, we make use of cultural historical activity theory, yet we have a very restricted view on the historical dimension that every artefact carries. We have two reasons for this: firstly, while we find the concept of considering the cultural dimension of every single artefact compelling, trying to take this into account could lead to overdetermination [Althusser, 1962]. Secondly, it is simply a question of resource – how deep need the questioning of this historic dimension go when our ultimate goal is "only" the production of a working artefact?

A typical example of a concept we are interested in but did not have the time to investigate yet is the notion of functional organs and the limited plane of actions [Kaptelinin, 1996].

Finally, it should be remembered that our work here is focusing on making socio-technical theories useful in the field of artificial intelligence, and not on further development of these theories themselves. The process of putting a theory to use can expose weaknesses or deficits in the theory that are beneficial for the development of the theory, but it is the task of the theoreticians themselves to transfer these findings back into the original domain of research.

3.3 Evaluation

This thesis is primarily focused on a theoretical discussion. So the arguments given must be evaluated on their own merits. But we also claim to advance the engineering part of artificial intelligence, that is we aim to improve the design of intelligent systems. There is no consensus on how to evaluate this, however, Cohen and Howe [1988] have described a set of criteria for evaluating, amongst others things, models, research problems, and implementations.

In Cohen [1989], another set of six questions is given that should be considered during the evaluation of intelligent systems:

1. What are the metrics for evaluating the method?
2. How is the method an improvement, an alternative, or a complement to existing technologies?
3. What are the underlying architectural assumptions?
4. What is the scope of the method?
5. Why does the method work (or not work)?
6. What is the relationship between the class of tasks, of which the current task is an example, and the method?

Let us take a look at theses questions one by one.

1. What are the metrics for evaluating the method?

Results from psychology, cognitive sciences, and philosophy are commonly used in artificial intelligence. From a theoretical point of view, the cognitive plausibility of AI theories and implementations with regard to such theories is an appropriate measure. By cognitive plausibility, we mean an assessment of the extent to which the model or method reflects a psychological model, and whether an implementation of the model or method retains its characteristics. We extend the theoretical backdrop towards social aspects of cognition and behaviour, so the question of cognitive plausibility is a measure of choice.

When looking at the design process, one of the primary goals of our approach is to ensure that social aspects of intelligence can be modeled. It is not within the scope of this thesis to address the question of whether the theory itself has merit. We do however draw attention to the fact that all theories considered in our work have an established tradition in fields like pedagogy, human-computer interaction, or software engineering.

2. How is the method an improvement, an alternative, or a complement to existing technologies?

Our approach adds the ability to take social aspects of intelligence into account when developing intelligent systems. In that sense, it is an improvement on existing technologies. Moreover, the explanation problem frames tie in with existing requirements engineering methods, thereby complementing the existing toolset and extending it towards a different class of requirements, namely the modelling of explanatory capabilities. As a final point, we complement existing approaches in the specification of aspects of explanations in intelligent systems by connecting the socio-technical theories to traditions in expert systems design on one hand and (via the notion of explanation types) to the knowledge container metaphor for case-based reasoning systems on the other hand.

3. What are the underlying architectural assumptions?

Typical concepts we deal with are situations, artefacts, persons, and roles. The methods developed help to incorporate such concepts into a structural knowledge model and populate the model with episodical knowledge. With such a strong focus on a knowledge level perspective, the methods outlined do not cater to sub-symbolic approaches to intelligent systems. In addition, some aspects of our work target (knowledge intensive) case-based reasoning methods specifically, but most parts of the work can be generalised to other symbolic reasoning and modelling paradigms.

4. What is the scope of the method?

The methods developed are suited for any problem domain where socio-technical analysis, for example in the form of ethnographic studies, can be performed, and where enough information is available to extract requirements for context and/or explanations.

5. Why does the method work (or not work)?

This question and the following only make sense when related to specific instances of application of the methods, and not the socio-technical approach in general. When we look at the design of a context model for an ambient intelligent hospital ward information system, which we have developed, then we can state that the explicit focus on a psychologically plausible knowledge model combined with the relatively structured problem domain were the main reasons for succeeding.

6. What is the relationship between the class of tasks, of which the current task is an example, and the method?

We also address this question with regard to the hospital ward information system design. Two main tasks can be identified: first, the design of a context model that was both related to best practices in ambient intelligence and cognitive plausible, and second the population of this context model. The first task can be seen as the instantiation of a general model for human activities with regard to a specific problem domain. For the second task, we have used some aspects of the socio-technical theory as a template for an ethnographic study design.

3.4 Future Work

Starting our research outlook with activity theory, there are still blind spots in our framework which we would like to address. The most blatant of these is perhaps the lack of integration of the concept of functional organs discussed by Kaptelinin [1996] and mentioned in the previous section. Kaptelinin states that one of the discriminating features of computer tools is that they do not have one fixed function and can easily be used as an extension of the internal plane of actions, that is the ability to perform manipulations with an internal representation before acting on external objects, see e.g. Marx [1867, p. 193]. We expect intelligent systems to significantly enhance this abilities compared to non-intelligent tools.

On the semiotics front, we would like to do further research into sense making processes as negotiated processes. It is not simply one meaner that has to be considered. In any exchange there are always at least two meaners, and more typically more than two. Multiparticipant communication represents a challenge to modelling. We have to keep in mind that others may share our conceptualisations and meanings only to a certain extent. When intelligent systems link different people this is an important thing to remember. The closer a person is in our social network the more likely they are to share our meanings, while the further out in our social network the less likely they are to share meanings. In the hospital environment, ambient intelligent devices can belong to different groups of users. Should we model them in a way that the assistant of a nurse is more likely to share concepts with the assistant of another nurse than that of a physician?

Another issue we would like to explore further is the extent to which it is possible to relate a semiotic approach to intelligent systems design to our work on activity theory. Bødker and Andersen [2005] have outlined some properties of a socio-technical approach taking advantage of ideas from both theoretical frameworks, and we would like to extend this to cover specific aspects of SFL and cultural-historical activity theory. This will potentially extend the number of projects from which findings may be borrowed, meaning the potential for a richer description of the hospital environment.

An additional point we have not yet fully explored is the relation of concepts from SFL with specific methods from the field of artificial intelligence. For example, the notion of genres in SFL seems to be a likely candidate for knowledge poor lazy learning mechanisms, while the descriptive power of the register might be exploitable in knowledge intensive or ontology based approaches. A promising candidate to combine these aspects is knowledge-intensive case-based reasoning.

Looking at the requirements engineering processes, we have to deepen our understanding of the relation between the design documents and the actual implementation. Our results show that problem frames are beneficial in identifying which explanatory knowledge and mechanisms should be provided, but the methods for identifying the missing "knowledge containers" and suggesting remedies have to be extended beyond the existing work on the relation between explanation goals, explanation kinds, and knowledge containers. We are considering the potential of coupling problem frames with design patterns to give system designers information about the implementation issues.

Another interesting aspect that needs further exploration is the extent to which (explanation) problem frames can be useful in the requirements elicitation phase. Our current approach of using observational ethnographic studies focuses primarily on a socio-technical analysis of workplace environments, but it is possible that problem frames be used to communicate requirements in focus groups or workshop settings that could append these empirical studies.

Bibliography

Louis Althusser. Contradiction and overdetermination. In *For Marx*. Penguin Press, 1962. URL http://www.marxists.org/reference/archive/althusser/1962/overdetermination.htm.

Peter Bøgh Andersen and Martin Brynskov. The semiotics of smart appliances and pervasive computing. In *Semiotics and Intelligent Systems Development*, pages 211–255. IGI Publishing, Hershey, PA, USA, 2006. ISBN 1599040638.

Susanne Bødker and Peter Bøgh Andersen. Complex mediation. *Journal of Human Computer Interaction*, 20(4):353–402, 2005.

Patrick Brézillon. Context modeling: Task model and practice model. In Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors, *Modeling and Using Context – CONTEXT 2007*, volume 4635 of *Lecture Notes in Computer Science*, pages 122–135, Roskilde, Denmark, 2007. Springer. ISBN 978-3-540-74254-8. URL http://dx.doi.org/10.1007/978-3-540-74255-5_10.

Paul Cohen. Evaluation and case-based reasoning. In Kristian J. Hammond, editor, *Proceedings of a Workshop on Case-Based Reasoning*, pages 168–172, San Mateo, CA, USA, 1989. DARPA, Morgan Kaufmann. ISBN 1-55860-068-X.

Paul Cohen and Adele Howe. How evaluation guides AI research: The message still counts more than the medium. *AI Magazine*, 9(4):35–43, 1988.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Manasawee Kaenampornpan and Eamonn O'Neill. Modelling context: an activity theory approach. In Panos Markopoulos, Berry Eggen, Emilie Aarts, and James L. Croeley, editors, *Ambient Intelligence: Second European Symposium on Ambient Intelligence, EUSAI 2004*, volume 3295 of *Lecture Notes in Computer Science*, pages 367–374. Springer, November 2004.

Viktor Kaptelinin. Computer-mediated activity: Functional organs in social and developmental contexts. In Bonnie A. Nardi, editor, *Context and Consciousness*, pages 45–68. MIT Press, Cambridge, MA, 1996.

Karl Marx. *Das Kapital. Kritik der politischen Ökonomie. Band 1: Der Produktionsprozeß des Kapitals*, volume 23 of *Marx Engels Werke (MEW)*. Dietz-Verlag, Berlin, 1867.

John McCarthy. What is ai? Internet, 2007. URL http://www-formal.stanford.edu/jmc/whatisai/. Last visited 2007-12-09.

Andrew Potter. A discourse approach to explanation aware knowledge representation. In Roth-Berghofer et al. [2007], pages 56–63. ISBN 978-1-57735-333-1. URL http://www.aaai.org/Library/Workshops/ws07-06.php.

Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors. *Explanation-Aware Computing – Papers from the 2007 AAAI Workshop*, number WS-07-06 in Technical Report, Vancouver, BC, 2007. AAAI Press. ISBN 978-1-57735-333-1. URL http://www.aaai.org/Library/Workshops/ws07-06.php.

Douglas Walton. Dialogical models of explanation. In Roth-Berghofer et al. [2007], pages 1–9. ISBN 978-1-57735-333-1. URL http://www.aaai.org/Library/Workshops/ws07-06.php.

Part II

Publications

A Work Context Perspective on Mixed-Initiative Intelligent Systems

A

Author:

Jörg Cassens

Abstract:

The issue of mixed-initiative intelligent systems has gained increasing interest in recent years. In particular, much attention has been paid on sharing the initiative between the user and the system on the tool level. In this paper, we are focusing on the problem of embedding the system into a workplace. We are proposing a framework for the analysis of how intelligent systems fit into a work context. We outline an approach with three different perspectives, focusing on the work process as a whole as well as human computer interaction on the interface and system level. The theoretical background consists of the Actor Network Theory, Semiotics, and the Activity Theory. We describe some challenges for the design of mixed initiative intelligent systems and outline how our framework might help to deal with these challenges.

Main Result:

This paper introduces three views on intelligent systems in workplace environments; 1. Work process view (using actor network theory), 2. HCI interface view (using semiotics), and 3. HCI system view (using activity theory). It is exemplified how these theories can help tackle different issues in mixed-initiative intelligent systems, namely 1. the control issue, 2. the communication issue, and 3. the evolvement issue.

Published in:

Proceedings of the IJCAI 2003 Workshop on Mixed-Initiative Intelligent Systems, pages 30–35, Acapulco, 2003.

A.1 Introduction

Case Based Reasoning (CBR) is a research area in the field of AI. Its aim is to understand and build systems which are able to use former experience in order to solve new problems. A CBR system is able to learn by taking care of experience in the form of so called cases, which describe problems and their solutions. When a new problem arises, one sufficiently similar previous problem has to be identified and the former solution has to be adapted to the new problem. The new solution might also be based on more than one previous case.

Being capable of learning during its use, CBR systems are one way to overcome the knowledge acquisition bottleneck. But it might be useful not to build the whole knowledge abductive, but to include given domain knowledge. The Division of Intelligent Systems in Trondheim is focusing on CBR systems which do not only learn from experience, but also incorporate given general domain knowledge to solve the problems (see e.g. Aamodt [1995]). This is referred to as knowledge-intensive CBR.

The group is aiming towards building a framework for such CBR systems. This involves identifying usable knowledge and reasoning structures as well as questioning how to embed the system in user tasks.

When an AI system is considered not as a replacement of, but a supplement to human work, the question of an adequate form of interaction arises. An AI system is to a certain degree trespassing the boundary of viewing the computer system as a tool, and extending this to as to act as a partner in a work flow.

The notion of mixed initiative takes these role change into account. It is made explicit that both the human user and the machine can take the initiative in the interaction. The system might proactively request information from the user which is needed to solve a given problem. The control might either lie in the hands of the user when entering data, or the system can guide her through a dialogue.

In the light of this changes also the human computer interaction should be revisited. Traditional interface engineering methods focusing on the computer as a tool seem not to be appropriate to design intelligent systems. Further on, the integration of this kind of systems into work processes is likely to change.

This has in first sight the consequence that an AI systems must definitely be developed by taking the whole work situation into account. Traditional software engineering techniques, mainly focusing on the artifact itself, might possibly not give adequate results. Therefore, the software production process must integrate methodologies of work analysis.

In order to understand how the system fits into a work place situation,

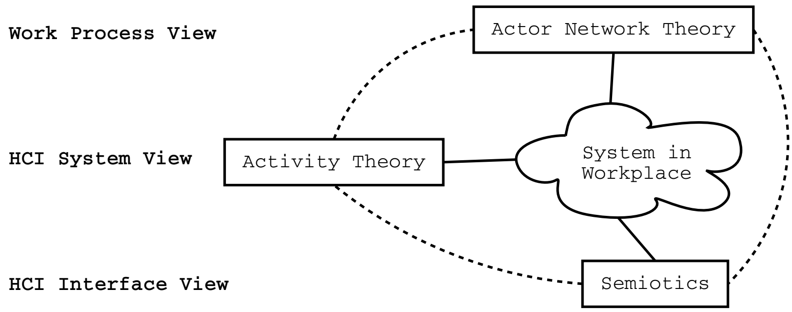

Figure A.1.: Overview: Different views on the work context.

we propose a theoretical framework which is focussing on three different perspectives (see figure A.1):

- Work process view: Actor Network Theory,
- HCI interface view: Semiotics,
- HCI system view: Activity Theory.

We are arguing that this theoretical framework is helpful for understanding how AI systems in general and especially CBR systems fit into a work process, and how they interact with the user.

A.1.1 Challenges

Mixed initiative intelligent systems face a couple of interesting challenges. We will shortly mention some of them:

- **The control issue:** How can we deal with the shift of initiative and control between different actors, both human and non-human?
- **The communication issue:** How can we facilitate the exchange of knowledge and information between actors involved?
- **The evolvement issue:** It is unlikely that the form of interaction remains unchanged over time. How can we assure a sufficient flexibility in communication abilities?

In order to illustrate how the different views in our framework can be used to cope with these challenges, we will now introduce a short example. We will later on look at some aspects of this system from our different viewpoints. The example is a diagnostic system in the oil drilling industries. It is used to monitor the drilling process in order to identify situations where the oil drill can get stuck. To this end, it collaborates with human users. The system is a a knowledge-intensive CBR system.

A.2 Work Process View: Actor Network Theory

We model the context in which the system is implemented with the help of the Actor Network Theory, ANT (see e.g. Latour [1991] and Monteiro [2000]). The basic idea here is fairly simple: whenever you do something, many influences on *how* you do it exist. For instance, if you visit this conference, it is likely that you stay at a hotel. How you behave at the hotel is influenced by your own previous experience with hotels, regulations for check-in and check-out, the capabilities the hotel offers you (breakfast room, elevators).

So, you are not performing from scratch, but are influenced by a wide range of factors. The aim of the ANT is to provide an unified view on these factors and your own acting. An actor network in this notion is 'the act linked together with all of its influencing factors (which again are linked), producing a network' (see [Monteiro, 2000, p. 4]).

In this network, you find both technical and non-technical elements. By this, the ANT avoids the trap of either overstating the role of technological artifacts in a socio-technological system or underestimating their normative power by applying the same framework to both human actors and technological artifacts. This makes it possible for us to understand how technological artifacts influence the doing of human actors in much the same way as other human actors.

Some key concepts of the theory are (compare e.g. Monteiro [2000]):

- **Actors:** Humans and technological artifacts,
- **Actor-network:** The totality of actors, interests, organizations, rules, standards, and their interaction,
- **Translation:** Actors interests translated into technical or social arrangements,
- **Inscription:** Result of the translation of one's interest into material form,
- **Subscription:** Acceptance of the inscribed interests by other actors.

In the ANT, technological artifacts can stand for human goals and praxis. Hotel keys, for example, are often not very handy, because the hotel owner has *inscribed* his intention (that the keys do not leave the hotel) into metal tags (which is why the guest *subscribe* to the owners intention: they do not want to carry this weight). A software system for workflow management is a representation of organizational standards in the company where it is used (and makes human users follow these standards).

One advantage of the ANT in the setting of intelligent systems is that it already comprises technical artifacts and humans in the same model. Humans and artifacts are exchangeable and can play the same role in the network. But in contrast to traditional artifacts, which are merely passive (black boxes in which human interests are subscribed) or which active role is restricted to translating intentions of the designer into changes of the praxis of the user, AI systems play a more active role: they have to *act-if* they had human capabilities.

In previous work in our group (see Pieters [2001]), we have argued that intelligent systems have to show certain capabilities usually ascribed to humans in order to interact with the user in a meaningfull way. On the other hand, since at least some of these capabilities rely on transcendental concepts, it is not possible to *design* machines which expose them.

In contrast to e.g. Edmonds [2000], who proposes a system which opens for the evolvement of certain properties, we use the notion of *as-if* in our approach: in roughly the same way as humans can never be sure that human counterparts have the capabilities they expect them to have, but ascribe it to them, our goal is to design intelligent systems which act in a way that makes humans ascribe human characteristics also to them. Also in Pieters [2001], it is argued that some properties of knowledge-intensive Case-Based Reasoning systems make them well suited for exposing this *as-if* capability. We will not focus on this.

A.2.1 Example

For the design of mixed-initiative systems, it is important to notice that the border between human and artifical actors is weakened in the Actor Network Theory. This makes it for example easier to include the notion of alternating the control between human and machine actors, thereby making the *control issue* explicit. By understanding how the initiative for a task is shared between different human actors, we get hints for how a technical artifact should behave in the same situation.

In our drilling problem example, we can with the help of the Actor Network Theory describe the organizational standards for dealing with critical conditions and identify situations where the diagnostic system should intervene.

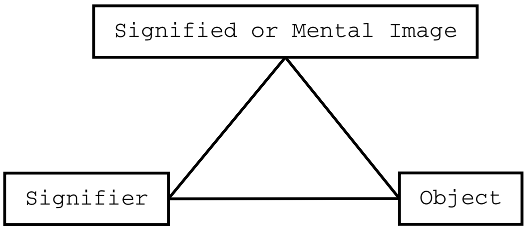

Figure A.2.: The semiotic triangle.

A.3 HCI Interface View: Semiotics

As seen in discussions between Ben Shneiderman, long time proponent of direct manipulation interfaces, and Pattie Maes, proponent of an agent oriented view of user interaction, at IUI-97 and CHI-97[1], the underlying metaphors for both views make a combination rather difficult.

Whereas Shneiderman strengthens the '[. . .] goal to create environments where users comprehend the display, where the system is predictable, and where they are willing to take responsibility for their actions' [Alty et al., 1997, p. 44], Maes clarifies that giving up some control is very common in every day tasks, but that this does not mean that the overall process is not controlled at all [Alty et al., 1997, p. 54].

It is very important to notice basic differences between direct manipulation and agent based interfaces as illustrated by this control example, which can be generalized for the whole interaction process of human and AI actors.

When focusing on the interaction of a particular user with the system, we use the semiotics approach (see e.g. Nake [1994] and Andersen [2001]) to understand the peculiarities of interaction with intelligent systems. The basic concept of the chosen interpretion of semiotics is the sign, a triadic relation of a signifier, a signified, and object (see figure A.2). It is the process of sense-making, where a representation *(signifier)* and its mental image *(signified)* refer to an entity *(object)* (the meaning of a sign is not contained within a symbol, it needs its interpretation).

On the background of semiotics, meaningfull human communication is a sign process. It is a process of exchanging and interpreting symbols referring to objects. The user of an informatics systems sees her interaction with this system on this background. When typing a letter, she does not send mere symbols, but signs to the computer, and the feedback from the

[1] As documented in Alty et al. [1997]

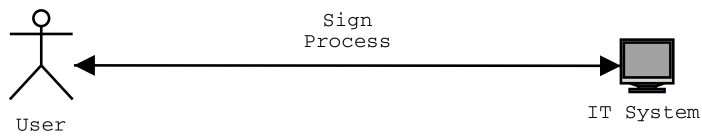

Figure A.3.: Semiotics: The human user sees the system as-if it was a partner in communication; the interaction appears to be a sign process.

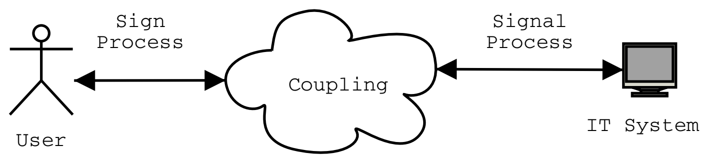

Figure A.4.: Semiotics: In human computer interaction, a sign and a signal process have to be coupled. The human sign process is reduced to an algorithmic signal process, which in turn is interpreted by the human user.

machine, the pixels on the screen, are interpreted as signs: to the user, the computer is a 'semiotic machine' (Wolfgang Coy), see figure A.3.

In contrast, computer systems are only processing signals, lacking the necessary interpreting capabilities humans have. They only manipulate symbols without *making-sense* out of them. The human sign process and the machine signal process have to be coupled (see figure A.4). This holds both for traditional informatics systems and AI systems.

We argue that, in order to make intelligent systems work not merely as a tool or a media, but as actants to whose (decision) abilities a human user can subscribe, the system must appear *as-if* it was capable of a meaningfull interaction.[2] We use again the *as-if* notion: an intelligent systems behaves in such a way that the user ascribes to the system the ability of participation in a sign process. The upper-level analysis of the work process helps in defining the aspects of user interaction where this ascription has to succeed in order to make the user believe in the system capabilities.

[2]Which differs from the interaction with traditional systems in which case the sense-making falls wholly on the side of the human user: You do not expect a text processor to understand your letter, but you expect a decision support system to understand the information you deliver.

On important challenge here is the ability of the system to show off its abilities. This can be described as a communication problem: the system has to interpret the actions of the user in a meaningfull way and itself present results that make sense for the user. This process of sense-making is highly interactive: an intelligent partner in a communication process asks (meaningfull) questions if an unclear situation occurs and is able to explain its own actions. The semiotic approach is useful to analyse this sense-making process with the help of transferring knowledge about similar processes from other semiotic domains.

A.3.1 Example

In our drilling problem example, it is due to time constraints important that new knowledge can easily be incorporated both into the system and presented to human users. For a knowledge-intensive CBR system, this can either be done in the form of cases or by enhancing the domain knowledge of the system. Given the latter, the system can monitor its reasoning processes and identify areas where it has insufficient knowledge to find causal relations. By means of plausible inheritance (compare e.g. Sørmo [2000]), it can find probable candidates for new causal explanations.

The semiotic approach can be used to model how the system could represent this probably new knowledge to the user in a way that strengthens the users believe in the sign-processing capabilities of the system. Therefore, semiotics can be helpfull to find solutions for the *communication issue*.

A.4 HCI System View: Activity Theory

The semiotics perspective is helpful to understand medial aspects of Human Computer Interaction, e.g. how knowledge is communicated. It is, however, not as helpful to analyze their use as instruments for achieving a predefined (by the human) goal in the work process and especially to understand the transformation of the artifact itself or the the socio-technical system during this process.

In our research, we found the Activity Theory (AT, see e.g. Bødker [1991], Nardi [2003]) suitable to cover this aspects in our framework. Its focus lies on individual and collective work practice. One of its strength is the ability to identify the role of material artifacts in the work process. An activity (see figure A.5) is composed of a subject, an object, and a mediating artifact or tool. A subject is a person or a group engaged in an activity. An object is held by the subject and motivates activity, giving it a specific direction.

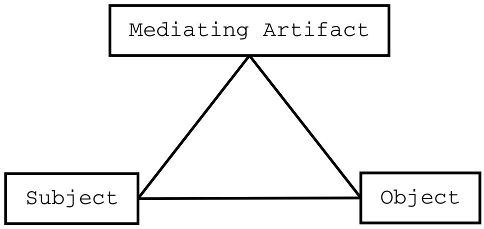

Figure A.5.: Activity Theory: The basic triangle of Mediation.

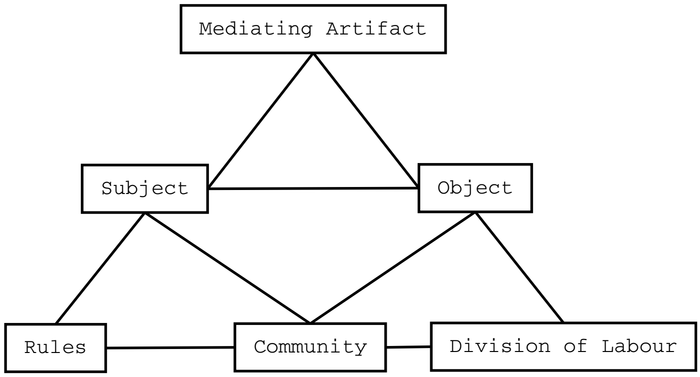

Figure A.6.: Activity Theory: Expanded triangle, incorporating the community and other mediators.

Later, the Activity Theory was extended to cover the fact that human work is done in a social and cultural context (compare e.g. Mwanza [2000]). The expanded model takes this aspect into account by adding a community component and other mediators, especially rules (an accumulation of knowledge about how to do something) and the division of labour (see figure A.6).

Some basic properties of the AT are:

- **Hierarchical structure of activity:** Activities (the topmost category) are composed of goal-directed actions. These actions are performed consciously. Activities, in turn, consist of non-conscious operations.
- **Object-orientedness:** Objective and socially or culturally defined properties. Our way of doing work is grounded in a praxis which is shared by our co-workers and determined by tradition. Praxis

forms the look of artifacts, and by these the artifacts are passing on a specific praxis.

- **Mediation:** Human activity is mediated by tools, language, etc. The artifacts as such are not the object of our activities, but appear already as socio-cultural entities.

Taking a closer look on the hierarchical structure of activity, we can find the following levels:

- **Activity:** This is the topmost level. An individual activity is for example to check into a hotel, or to travel to the conference city. Individual activities can be part of collective activities, e.g. when you organize a workshop with some co-workers.
- **Actions:** Activities consists of a collections of actions. An action is performed consciously, the hotel check-in, for example, consists of actions like presenting the reservation, confirmation of roomtypes, and handover of keys.
- **Operations:** Actions consist themselves of collections of non-conscious operations. To stay with our hotel example, writing your name on a sheet of paper or taking the keys are operations. That operations happen non-consciously does not mean that they are not accessable.

It is important to note that this hierarchical composition is not fixed over time. If an action fails, the operations comprising the action can get conceptualized, they become conscious operations and might become actions in the next try to reach the overall goal. This is referred to as a breakdown situation. In the same manner, actions can get automated when done many times and thus become operations. By this, we gain the ability to model a change over time.

Since an AI system is more a partner in a work process than a tool, its role in the user interaction changes. Whereas a classical informatics system is a passive translator and memory of praxis, the intelligent system is constantly re-shaping the praxis through its use. The usage of a tool might change, but the tool itself will not change. If you look at an decision support system, so is the decision making process itself transformed by the ability of the system to react differently, e.g. through accumulated experience and usage context.

But since the AT itself models artifacts as being preformed as socio-cultural entities, we can describe the artifacts in a way which takes this modification into account. Again, our upper-level model helps us to identify the mediation process and the role of both human and non-human actors in the usage process.

As described before is the ability of an intelligent system to adapt to the user very important. In the process of re-shaping the praxis, a user expects from an (*as-if*) intelligent system that it is adopting to the changed praxis.

A.4.1 Example

Whereas in the beginning of the use of our example Case-Based diagnostic system, it will be important to explain the user in detail why a particular case (former stuck pipe situation) was matched to a new problem, the user expects from an intelligent partner that the same match will be explained in less detail when occurring very frequently (since the artifact should be changed by the changed praxis, that is here the accumulated knowledge on both sides).

This change of interaction over time is related to the *evolvement issue*: the shift between different modi operandi. The AT is suitable for capturing this kind of change over time (transforming of actions into operations and vice versa) and can therefore be helpfull in modeling a change of behavior over time.

A.5 Related Work

Our group is developing the Creek[3] framework for knowledge-intensive CBR. Creek makes extensive use of general domain knowledge and knowledge of the reasoning process itself. This knowledge has to be acquired and engineered at least partly before the system starts learning from cases. In Tecuci et al. [1999], a mixed-initiative approach for the development of a knowledge base is proposed and evaluated. The type of the acquired knowledge differs from the semantic net structure of Creek. The concept of having a concept of competence in the building process of a knowledge base and the use of different knowledge acquisition strategies is nevertheless intersting for the Creek toolchain, but lies for the time being outside of the scope of our tools. This work is located at the Work Process View since it deals with the competence of the newly defined system.

Compared to our own views, a very different approach to design issues on the User Interaction System View is pursued by Hartrum and DeLoach [1999]. In their multi-agent approach, they unify the interaction view

[3] Case-Based Reasoning through Extensive Explicit Knowledge.

between different types of actants, being it humans or intelligent agents. They use Z specifications for formally defining the the agents, including structural and behavioral aspects. This approach is complementary to our use of the Activity Theory.

Langley [1999] deals with adaptive interfaces. The importance of personalized presentation of information is pointed out. This is not restricted to the form of the presentation, but also the contents. This is a very important point. The challenge of giving a transparent impression of the systems capabilities is directly dependant on the users own knowledge, and on the ability of the system to change its behavior towards a learning user over time. In our framework, this issue is addressed by the ability of the Activity Theory to reflect changes of the involved artifacts over time.

Also looking at personalization issues, Blanzieri [2002] proposes a four level analysis of situated intelligent systems. He focuses on the need of a particular user instead of the social stance taking by the Activity Theory. In this sense, his approach is complementary to our framework.

On the User Interaction Interface Level, Eggleston [1999] describes a cognitive engineering approach to the modeling of user interface agents. A unified view on human-human and human-agent network communications is taken and design principles from the cognitive engineering stance are stated. Whilst our focus lies on different aspects (the communication aspect of the semiotic theory), his statement on the importance of coupling human thinking and automated reasoning so that joint cognitive work is enhanced can also be found in our notion of a Case-Based Reasoner as enhancing the human capabilities.

In McSherry [2002], a taxonomy for mixed initiative dialogue is given. The focus lies on the Interface Level and deals mainly with tool aspects and differs in that sense from our communication oriented approach. Features like the need for the explanation of reasoning and the control issue are nevertheless challenges we have to deal with as well.

An example of a theoretical and empirical validation of the usefullness of an mixed initiative approach to Conversational CBR can be found in Gupta et al. [2002]. This differs from the Creek framework we use as we do not focus on text conversation. On the other hand, it might be very interesting to apply the semiotic framework to this approach, since the semitics of written language is a well researched subject.

A.6 Conclusion

We have proposed a theoretical framework for a consistent model of intelligent systems in work process. Our model includes an upper-level analysis of the work process as a whole as well as means to understand the interaction between user and system.

We have further on outlined that the proposed framework supplies theoretical tools for the analysis of mixed initiative system with different perspectives. We have shown that the different theories in our framework can deal with important issues of mixed initiative intelligent systems.

Equally important, but not topic of this paper, is a translation of this a posteriori analysis into an a priori design methodologies.

Bibliography

Agnar Aamodt. Knowledge acquisition and learning by experience – the role of case-specific knowledge. In Gheorghe Tecuci and Yves Kodratoff, editors, *Machine Learning and Knowledge Acquisition – Integrated Approaches*, chapter 8, pages 197–245. Academic Press, 1995.

Jim Alty, Ben Shneiderman, and Pattie Maes. Direct manipulation vs. interface agents. *Interactions*, pages 42–61, 11+12 1997.

Peter Bøgh Andersen. What semiotics can and cannot do for HCI. *Knowledge-Based Systems*, 14:419–424, 2001.

Enrico Blanzieri. A cognitive framework for personalization of the cbr cycle. In *Workshop Proceedings ECCBR-2002*, Aberdeen, 2002.

Susanne Bødker. Activity theory as a challenge to systems design. In Hans-Erik Nissen, Heinz K. Klein, and Rudy Hirschheim, editors, *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pages 551–564. North Holland, 1991.

Michael T. Cox, editor. *Mixed Initiative Intelligence (MII): Proceedings of the AAAI Workshop*, 1999.

Bruce Edmonds. Towards implementing free will. In *AISB 2000 symposium on How to Design a Functioning Mind*, Birmingham, April 2000.

Robert G. Eggleston. Mixed-initiative transactions: A cognitive engineering approach to interface agent modeling. In Cox [1999].

Kalyan Moy Gupta, David W. Aha, and Nabil Sandhu. Exploiting taxonomic and causal relations in conversational case retrieval. In Susan Craw and Alun Preece, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2002*, number 2416 in Lecture Notes in Artificial Intelligence, Aberdeen, 2002. Springer.

Thomas C. Hartrum and Scott A. DeLoach. Design issues for mixed-initiative agent systems. In Cox [1999].

Pat Langley. User modeling in adaptive interfaces. In Judy Kay, editor, *User Modeling: Proceedings of the seventh International Conference, UM-99*, pages 357–370, Banff, Alberta, 1999. Springer.

Bruno Latour. Technology is society made durable. In John Law, editor, *A Sociology of Monsters*, pages 103–131. Routledge, 1991.

David McSherry. Mixed-initative dialogue in case-based reasoning. In *Workshop Proceedings ECCBR-2002*, Aberdeen, 2002.

Eric Monteiro. Actor-network theory. In Claudio U. Ciborra, editor, *From Control to Drift*, pages 71–83. Oxford University Press, 2000.

Daisy Mwanza. Mind the gap: Activity theory and design. Technical Report KMI-TR-95, Knowledge Media Institute, The Open University, Milton Keynes, 2000.

Frieder Nake. Human-computer interaction – signs and signals interfacing. *Languages of Design*, 2:193–205, 1994.

Bonnie A. Nardi. A brief introduction to activity theory. *KI – Künstliche Intelligenz*, 1:35–36, 2003.

Wolter Pieters. Free will and intelligent machines. Project Report, NTNU Trondheim, 2001.

Frode Sørmo. Plausible inheritance semantic network inference for case-based reasoning. M.sc. thesis, Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science, Trondheim, 2000.

Gheorge Tecuci, Mihai Boicu, Kathryn Wright, and Seok Won Lee. Mixed-initiative development of knowledge bases. In Cox [1999].

Knowing What to Explain and When

B

Author:
Jörg Cassens

Abstract:
We have argued elsewhere that user goals should be taken into account when deciding what kind of explanation of its results a CBR system should give. In this paper, we propose the use of an Activity Theory based methodology for identifying different user goals and expectations towards explanations given by a system supporting a work process.

Main Result:
The paper proposes to utilise the notion of an action cycle for goal-directed pragmatic action (based on activity theory) to identify situations where an intelligent system should provide explanations of its behaviour.

Published in:
Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, number 142-04 in Technical Report of the Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, pages 97–104, Madrid, 2004.

B.1 Introduction

Customized IT Systems are usually designed for specific purposes and tasks which the system has to support and in settings where comparable work was done also before the system was introduced. It is used by people with specific needs and qualifications, and it should preferably adapt to changes in these needs over time [McSherry, 2002; Totterdell and Rautenbach, 1990]. Althoff and Wilke [1997] have introduced an organizational view of the CBR cycle for the purpose of business process modeling. For the purpose of this paper, we are looking at CBR systems embedded in such a work context, but on a more general level.

B.2 Problems with Explanations in CBR

The term explanation can be interpreted in two different ways in AI [Aamodt, 1991, p. 59]. One interpretation deals with explanations as part of the reasoning process itself, for example used in the search for a diagnostic result in order to support certain hypotheses. The other interpretation deals with usage aspects: making the reasoning process, its results, or the usage of the result transparent to the user. Both interpretations can be found in CBR research. The ability to explain its results is often considered as one of the main advantages of CBR systems [Leake, 1996; McSherry, 2001; Cunningham et al., 2003]. A knowledge-intensive CBR system may use explanations to guide the CBR process itself [Aamodt, 1993, 1994].

One problem is that the question of what makes up a good explanation depends on the goals of the user [Leake, 1995]. This also means that we cannot be sure that we will match the user's needs by presenting the case alone as it is [Sørmo and Cassens, 2004]. So it might not be as straightforward as it sounds to provide the user with an adequate explanation. Where explanations are used to support the CBR process, this problem reappears when the explanation is used to asses the users needs or wishes, e.g. in an adaptive CBR system.

Another important point is that it might not suffice to purely present the best matched case(s) to the user to give an explanation even when his goals are matched. McSherry [2003] points out that the presented case(s) might contain both features supporting the given results and features opposing it. Smyth and McClave [2001] strengthen the importance of giving a set of results with sufficient diversity for certain types of problems. McGinty and Smyth [2003]; Smyth and McGinty [2003] propose an adaptive way of presenting a set of cases adapted to the user's changing needs for diversity. All these works deal with the shortcomings of presenting a single (or to narrow set of) case(s).

Using only cases as explanations means further on to rely on the implicit assumption that by presenting the case to the user he will be able to do a similarity comparison himself. This may often be true, but is by no means guaranteed, especially when the case structure is complex or the similarity measure more convoluted. The problem increases when we start incorporating other AI technologies into the CBR process (as suggested by Watson [1999]), e.g. when using a neural network in similarity assessment.

B.3 Activity Theory

In this paper, we propose the use of Activity Theory (AT) to support the design of CBR systems which take these problems into account. We can use AT to analyze the use of intelligent systems as instruments for achieving a predefined (by the human) goal in the work process and especially to understand the transformation of the artifact itself or the socio-technical system during this process. This could help us understand which types of explanations are expected. On the other hand, our knowledge about the work process can help us understand problems showing up in the use of the CBR system, supporting our (implicit or explicit) user model.

B.3.1 Basic Properties of AT

In this section, we give a short summary of aspects of AT that are important for this work. See Nardi [2003] for a short introduction to AT and Bødker [1991]; Nardi [1996] for deeper coverage. The theoretical foundations of AT in general can be found in the works of Vygotsky [1978, 1985]; Leont'ev [1978].

Activity Theory is a descriptive tool to help understand the unity of consciousness and activity. Its focus lies on individual and collective work practice. One of its strengths is the ability to identify the role of material artifacts in the work process. An activity (Fig. B.1) is composed of a subject, an object, and a mediating artifact or tool. A subject is a person or a group engaged in an activity. An object is held by the subject and motivates activity, giving it a specific direction.

Some basic properties of the AT are:

- **Hierarchical structure of activity:** Activities (the topmost category) are composed of goal-directed actions. These actions are performed consciously. Activities, in turn, consist of non-conscious operations.

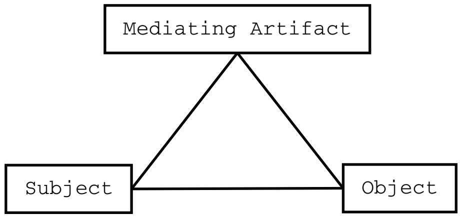

Figure B.1.: Activity Theory: The basic triangle of Mediation.

- **Object-orientedness:** Objective and socially or culturally defined properties. Our way of doing work is grounded in a praxis which is shared by our co-workers and determined by tradition. Praxis forms the look of artifacts, and by these the artifacts are passing on a specific praxis.

- **Mediation:** Human activity is mediated by tools, language, etc. The artifacts as such are not the object of our activities, but appear already as socio-cultural entities.

Taking a closer look on the hierarchical structure of activity, we can find the following levels:

- **Activity:** This is the topmost level. An individual activity is for example to check into a hotel, or to travel to another city to participate at a conference. Individual activities can be part of collective activities, e.g. when someone organizes a workshop with some coworkers.

- **Actions:** Activities consist of a collections of actions. An action is performed consciously, the hotel check-in, for example, consists of actions like presenting the reservation, confirmation of roomtypes, and handover of keys.

- **Operations:** Actions consist themselves of collections of non-conscious operations. To stay with our hotel example, writing your name on a sheet of paper or taking the keys are operations. That operations happen non-consciously does not mean that they are not accessable.

It is important to note that this hierarchical composition is not fixed over time. If an action fails, the operations comprising the action can get conceptualized, they become conscious operations and might become

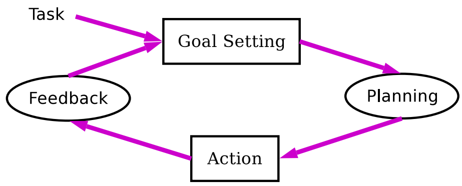

Figure B.2.: The Action Cycle (according to Fjeld et al. [2002]).

actions in the next attempt to reach the overall goal. This is referred to as a breakdown situation. In the same manner, actions can become automated when done many times and thus become operations. In this way, we gain the ability to model a change over time.

B.3.2 Action Cycle

Fjeld et al. [2002] describe the notion of an action cycle for goal-directed pragmatic action. Their model is based on Hacker's work on Activity Theory and occupational psychology [Hacker, 1998]. An action cycle (Fig. B.2) consists of:

1. **Goal setting:** The initial goal for performing a task is set.
2. **Planning:** Plan how to achieve the goal set, including the selection of tools and choice of actions required.
3. **Action:** Consciously performed mental or physical actions.
4. **Feedback:** Controlling whether the anticipated goal was achieved, and if not, identifying the reasons for failure.

The starting point for this cycle is the activity, here identified as a task to perform.

We have a twofold interest in the notion of an action cycle when designing CBR systems. First, it is useful for identifying and modeling parts of the workplace activity where the CBR system is performing a task "on its own" with basically no human interaction. Here we try to translate the human activity as a whole into the artifact. Second, the CBR system can act as a supportive agent for the human performing the task. This is of special importance in situations where both the CBR system acquires knowledge and the human user gains an insight into the working of the artifact. For example, when the CBR system is introduced into a new workplace situation and has to acquire case knowledge at the same time as the user is gaining confidence in using it.

This second aspect is important for the issue at hand: identifying different explanatory needs. Given enough data, an analysis of the workplace situation based on the notion of an action cycle will explicitely model the goal settings (thereby identifying different user goals) as well as typical problems in the execution of the cycle (thereby identifying different needs for explanation in breakdown situations).

B.4 Using AT in Knowledge-Intensive CBR

We are considering knowledge-intensive CBR systems which incorporate both case knowledge, task knowledge, and domain knowledge in one single system. Later in this section we will show how AT helps us in defining what to include in the design of the system's various knowledge containers [Leake et al., 1997; Richter, 1995].

B.4.1 *Choosing a Helpful Explanation*

As pointed out before, the presentation of a single case may not suffice to give the user an explanation of the solution found. What constitutes a good explanation is to a large degree dependent on the goals of the user [Sørmo and Cassens, 2004]. We therefore have to identify the possible goals of the user to the largest extent possible. This might be based on stereotypes of users, [Rich, 1979], for example in recommender systems, or on a survey of the concrete work situation the system is going to be embedded into [Tautz et al., 2000], for example for experience management.

When using Activity Theory to model the work process, [Korpela et al., 2002; Fjeld et al., 2002], we can identify different types of activities the users are involved in. This helps us to understand the goals the user has when accessing the system, thereby also identifying the type of explanations necessary. We can use this analysis to guide us when defining the knowledge model of the system. This makes sure that a useful explanation can be given when requested by the user. For example, if in an AT based analysis of a learning situation we see that students tend to relate the knowledge to be acquired to a different domain, then the system should be capable of using these analogies as a basis for the explanation given.

B.4.2 *Example Application*

As mentioned in the introduction, an Intelligent System should adapt to its usage over time. Lets consider a CBR system for decision support.

The user gets filled-out applications for credit cards and has to rate the creditability of the applicants. In the beginning, the user is likely to be interested in a detailed description of the results found. This is both due to the fact that the user has to learn to trust the system's capabilities, and that working with the system is quite new. In the language of Activity Theory, the user will perform mostly conscious actions and has not yet operationalized parts of the work process and/or the interaction with the system.

Over time, when the system offers correct or useful solutions, the user will both trust the system more, thereby eliminating his need to assess the reasoning process, and operationalize the work with the system. A lengthy presentation of the results by our CBR system will disrupt this process of operationalization. A mixed-initiative system will probably decrease its own activity: it no longer has to remind the user to do certain actions since they have become part of automated operations.

Let us now consider what happens when a breakdown situation occurs. For example, the user is now involved with free-form applications, e.g. by telephone. The problem viewed from the CBR system remains the same at first, but the work-flow of the user is disturbed so that he conceptualizes the operations he performs again. The CBR system must recognize this problem and change its behavior.

If we include relevant parts of the AT dealing with breakdown situations explicitly in our (general) user model, we can discover that such a situation has occurred (e.g. because the user is increasingly requesting explanations in situations where he has not done so before). The system can now adapt itself to give more detailed explanations and be more proactive again. Likewise, the breakdown situation is a hint that the system's knowledge may no longer be adequate, and the search for solutions might have to be broadened until enough new (case) knowledge is acquired.

B.5 Ongoing and Future Work

The integration of an a posteriori method of analysis with design methodologies is always challenging. One advantage AT has is that it is process oriented, which fits nicely to a view on systems design where the deployed system itself is not static and where the system is able to incorporate new knowledge over time [Aamodt, 1995]. Activity Theory has its blind spots, and our goal is therefore to combine AT with other theories into a framework of different methods supporting the systems design process [Cassens, 2003].

Focussing on AT, the relationship between the action cycle and the CBR process has to be examined further. Likewise, a methodological approach to integrate the findings of a work process analysis into the different knowledge containers of a CBR system has to be developed. Further on, a real world application of the outlined approach is necessary to asses its practicability. The method of choice is a qualitative study where the methodology has to be co-evolved with the ongoing project.

B.6 Conclusion

We have pointed out the importance of modeling the user's potential goals when defining which types of explanation an intelligent system can give. We have further introduced Activity Theory as a means of achieving this objective. Likewise we have suggested that the hierarchichal model of activity can be modeled in such systems to enable it to adapt to changes in usage over time. The action cycle as a model for goal directed pragmatic action can help identifying possible breakdown situations and resulting needs for specific types of explanation from the supporting intelligent system.

In our opinion, CBR system design methodologies will in the long run benefit from the integration of theories from occupational psychology and information systems design. They offer a supplement to cognitive science based approaches and integrate an understanding of organizational issues into the CBR process itself. What is and what is not a good explanation is dependent both on the individual user and her capabilities and on the organizational context. Therefore, we think it is necessary to achieve an understanding of the workplace situation the CBR system is going to be embedded into to deliver explanations of results which satisfy the user's needs.

Bibliography

Agnar Aamodt. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. PhD thesis, Norwegian Institute of Technology, Department of Computer Science, Trondheim, May 1991. URL http://www.idi.ntnu.no/grupper/su/publ/phd/aamodt-thesis.pdf.

Agnar Aamodt. Explanation-driven retrieval, reuse, and learning of cases. In Michael M. Richter, Stefan Wess, Klaus-Dieter Althoff, and Frank Maurer, editors, *EWCBR-93: First European Workshop on Case-Based Reasoning*, number SR-93-12 (Technical Report), pages 279–284, Kaiserslautern, 1993. University of Kaiserslautern.

Agnar Aamodt. Explanation-driven case-based reasoning. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-Based Reasoning: Proceedings of the 1st European Workshop EWCBR 1993 – Selected Papers*, volume 837 of *Lecture Notes in Computer Science*, pages 274–288, Berlin Heidelberg, 1994. Springer. ISBN 3-540-58330-0.

Agnar Aamodt. Knowledge acquisition and learning by experience – the role of case-specific knowledge. In Gheorghe Tecuci and Yves Kodratoff, editors, *Machine Learning and Knowledge Acquisition – Integrated Approaches*, chapter 8, pages 197–245. Academic Press, 1995.

David W. Aha and Ian Watson, editors. *Case-Based Reasoning Research and Development: Proceedings ICCBR 2001*, number 2080 in Lecture Notes in Artificial Intelligence, Vancouver, 2001. Springer.

Klaus-Dieter Althoff and Wolfgang Wilke. Potential uses of case-based reasoning in experience based construction of software systems and business process support. In Ralph Bergmann and Wolfgang Wilke, editors, *Proc. of the Fifth German Workshop on Case-Based Reasoning, Centre for Learning Systems and Applications*, number LSA-97-01E (Technical Report), pages 31–38. University of Kaiserslautern, 1997.

Kevin D. Ashley and Derek Bridge, editors. *Case-Based Reasoning Research and Development: Proceedings ICCBR 2003*, number 2689 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2003. Springer.

Susanne Bødker. Activity theory as a challenge to systems design. In Hans-Erik Nissen, Heinz K. Klein, and Rudy Hirschheim, editors, *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pages 551–564. North Holland, 1991.

Jörg Cassens. A work context perspective on mixed-initiative intelligent systems. In *Proceedings of the IJCAI 2003 Workshop on Mixed-Initiative Intelligent Systems*, pages 30–35, Acapulco, 2003.

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An evaluation of the usefulness of case-based reasoning explanation. In Ashley and Bridge [2003], pages 122–130.

Morten Fjeld, Kristina Lauche, Martin Bichsel, Fred Voorhoorst, Helmut Krueger, and Mathias Rauterberg. Physcial and virtual tools: Activity theory applied to the design of groupware. *CSCW*, 11(1-2):153–180, 2002.

Georg Gottlob and Toby Walsh, editors. *Proceedings of the 18th International Conference on Artificial Intelligence (IJCAI 2003)*, Acapulco, 2003. Morgan Kaufmann.

Winfried Hacker. *Allgemeine Arbeitspsychologie*. Huber, 1998.

Mikko Korpela, Anja Mursu, and Hettie A. Soriyan. Information systems development as an activity. *CSCW*, 11(1-2):111–128, 2002.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

David B. Leake. CBR in context: The present and future. In David B. Leake, editor, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*, pages 3–30. AAAI Press/MIT Press, Menlo Park, 1996.

David B. Leake, Andrew Kinley, and David C. Wilson. Learning to integrate multiple knowledge sources for case-based reasoning. In Pollack [1997], pages 246–251. URL `citeseer.nj.nec.com/leake97learning.html`.

Aleksei N. Leont'ev. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

Lorraine McGinty and Barry Smyth. On the role of diversity in conversational recommender systems. In Ashley and Bridge [2003], pages 276–290.

David McSherry. Interactive case-based reasoning in sequential diagnosis. *Applied Intelligence*, 14:65–76, 2001.

David McSherry. Mixed-initative dialogue in case-based reasoning. In *Workshop Proceedings ECCBR-2002*, Aberdeen, 2002.

David McSherry. Explanation in case-based reasoning: an evidential approach. In Brian Lees, editor, *Proceedings of the eighth UK Workshop on Case-Based Reasoning*, pages 47–55, Cambridge, 2003.

Bonnie A. Nardi. A brief introduction to activity theory. *KI – Künstliche Intelligenz*, 1:35–36, 2003.

Bonnie A. Nardi, editor. *Context and Consciousness*. MIT Press, Cambridge, MA, 1996.

Martha Pollack, editor. *Proceedings of the 15th International Conference on Artificial Intelligence (IJCAI 1997)*, Nagoya, 1997. Morgan Kaufmann.

Elaine Rich. User modeling via stereotypes. *Cognitive Science*, (3):329–354, 1979.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Barry Smyth and Paul McClave. Similarity vs. diversity. In Aha and Watson [2001], pages 347–361.

Barry Smyth and Lorraine McGinty. The power of suggestion. In Gottlob and Walsh [2003], pages 127–132.

Frode Sørmo and Jörg Cassens. Explanation goals in case-based reasoning. In Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, number 142-04 in Technical Report of the Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, pages 165–174, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid. URL `http://www.idi.ntnu.no/~cassens/work/publications/download/2004-ECCBR-WS-goals.pdf`.

Carsten Tautz, Klaus-Dieter Althoff, and Markus Nick. Learning from project experience – a case study on experience factory. In *Proc. GI-Fachgruppentreffen Maschinelles Lernen (FGML'00)*, Sankt Augustin, 2000. GMD.

Peter Totterdell and Paul Rautenbach. Adaptation as a design problem. In *Adaptive User Interfaces*, pages 59–84. Academic Pres, 1990.

Lew S. Vygotsky. *Mind in Society*. Harvard University Press, Cambridge, MA, 1978.

Lew S. Vygotsky. *Ausgewählte Schriften Bd. 1: Arbeiten zu theoretischen und methodologischen Problemen der Psychologie*. Pahl-Rugenstein, Köln, 1985.

Ian Watson. Case-based reasoning is a methodology, not a technology. *Knowledge-Based Systems*, 12(5-6):303–308, 1999.

Mapping Goals and Kinds of Explanations to the Knowledge Containers of Case-Based Reasoning Systems

C

Authors:
Thomas R. Roth-Berghofer and Jörg Cassens

Abstract:
Research on explanation in Case-Based Reasoning (CBR) is a topic that gains momentum. In this context, fundamental issues on what are and to which end do we use explanations have to be reconsidered. This article presents a prelimenary outline of the combination of two recently proposed classifications of explanations based on the type of the explanation itself and user goals which should be fulfilled. Further on, the contribution of the different knowledge containers for modeling the necessary knowledge is examined.

Main Result:
This paper ties together the work of Roth-Berghofer on different kinds of explanation with the work of Sørmo, Cassens, and Aamodt on user goals for explanations. A process model for the design activity is proposed, and several mappings from user goals over explanation kinds to the different knowledge containers of case-based reasoning systems are identified.

Published in:
Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *LNAI*, pages 451–464, Chicago, 2005. Springer.

My main contributions to the paper:

- Motivation for socio-technical point of view
- The user goals for explanations were contributed from previous joint work with other authors
- The overall process model was proposed
- Development of the example

The following aspects were jointly developed by the authors:

- Motivation for explanations
- Relations between goals and kinds

C.1 Why Bother to Explain?

In everyday human-human interactions explanations are an important vehicle to convey information in order to understand one another. Explanations enhance the knowledge of the communication partners in such a way that they accept certain statements. They understand more, allowing them to make informed decisions. According to Schank [1986] explanations are the most common method used by humans to support their decision making.

This is supported by Spieker's investigation into natural language explanations in expert systems [Spieker, 1991]. We identify some typical reactions of humans as soon as we cannot follow a conversation:

- we ask our conversation partner about concepts that we did not understand,
- we request justifications for some fact or we ask for the cause of an event,
- we want to know about functions of concepts,
- we want to know about purposes of concepts, and
- we ask questions about his or her behavior and how he or she reached a conclusion.

All those questions and answers are used to understand what has been said and meant during a simple conversation. An important effect of explanations is that the process of explaining certainly has some effect on one's trust in the competence of a person or machine: We keep our trust, we increase or decrease it. At least, providing explanations makes decisions more transparent, and motivates the use to further use the system.

The need for explanations provided by knowledge-based systems is well-known and was addressed by such fields as expert systems. For knowledge-based systems, explanations and knowledge acquisition are the only two communications channels with which they interact with their environment.

The adequacy of explanations as well as of justifications is dependent on pragmatically given background knowledge. What counts as a good explanation in a certain situation is determined by context-dependent criteria [Cohnitz, 2000; Leake, 1995].

The more complex knowledge-based systems get, the more explanation capabilities the users expect when using such systems. This requirement was recognized early on in expert systems research and development [Swartout, 1983; Buchanan and Shortliffe, 1984; Swartout and Smoliar, 1987]. Considerable results were produced, but research activity decreased together with the general decline of expert systems research in the 1990s. The major problems in connection with classical expert systems seemed to be solved.

At the same time there was an increasing interest on this topic in Case-Based Reasoning (CBR) [Leake, 1996; Schank et al., 1994]. At the turn of the century, we find the issue discussed again in the context of knowledge-based systems [Gregor and Benbasat, 1999; Swartout and Moore, 1993]. Recently, we can see a renewed focus in CBR on this track of research. ECCBR 2004 featured, for example, a workshop on Explanation in Case-Based Reasoning as well as a couple of papers on explanation at the main conference [Gervás and Gupta, 2004; Funk and Calero, 2004].

Research on explanation is of interest today because it can be argued that the whole scenario on research on knowledge-based systems has changed [Richter, 2005]: knowledge-based systems are no longer considered as boxes that provide a full solution to a problem. Problem solving is seen as an interactive process (a socio-technical process). Problem description as well as the special input can be incomplete and changing. As a consequence, there has to be communication between human and software agents. Communication requires mutual understanding that can be essentially supported by explanations. Such explanations can improve the problem solving process to a large degree.

It is important to note here that the term explanation can be interpreted in two different ways. One interpretation deals with explanations as part of the reasoning process itself. The other interpretation deals with usage aspects: making the reasoning process, its results, or the usage of the result transparent to the user. In this paper, we will focus on the second interpretation.

The remainder of this paper is organized as follows: In the next section, we describe the setting for explanation-aware CBR systems as being a component of socio-technical systems. In section C.3, we present two perspectives on explanation that can help understand and organize what to explain and when. The subsequent section focusses on knowledge containers and their contribution to the explanation capabilities of CBR systems. In Section C.5, we propose a system design process achitecture. We explore further on the relations of explanation goals, explanation kinds, and knowledge containers in a simplified example. We conclude our paper with an outlook on further research.

C.2 Explanation in Socio-Technical Systems

Whenever one talks about a 'system' one has to clarify what is meant by that term. In decision- support scenarios, the human and the computer are the decision system. Such socio-technical systems can for example be modelled with the help of the Actor Network Theory, ANT [Latour, 1991; Monteiro, 2000]. The basic idea here is fairly simple: whenever you do something, many influences on *how* you do it exist. For instance, if you visit a conference, it is likely that you stay at a hotel. How you behave at the hotel is influenced by your own previous experience with hotels, regulations for check-in and check-out, the capabilities the hotel offers you (breakfast room, elevators).

So, you are not performing from scratch, but are influenced by a wide range of factors. The aim of the ANT is to provide a unified view on these factors and your own acting. An actor network in this notion is *the act linked together with all of its influencing factors (which again are linked), producing a network* [see Monteiro, 2000, p. 4].

In this network, you find both technical and non-technical elements. In the ANT, technological artifacts can stand for human goals and praxis. Hotel keys, for example, are often not very handy, because the hotel owner has *inscribed* his intention (that the keys do not leave the hotel) into metal tags (which is why the guests *subscribe* to the owners intention: they do not want to carry this weight). A software system for workflow management is a representation of organizational standards in the company where it is used (and makes human users follow these standards).

One advantage of the ANT in the setting of intelligent systems is that it already comprises technical artifacts and humans in the same model. Humans and artifacts are to a certain degree exchangeable and can play the same role in the network. But in contrast to traditional artifacts, which are merely passive (black boxes in which human interests are subscribed) or which active role is restricted to translating intentions of the designer into changes of the praxis of the user, AI systems play a more active role. It has also been argued that intelligent systems have to show certain capabilities usually ascribed to humans in order to interact with the user in a meaningful way [Pieters, 2001], and we would include the ability to give good explanations.

Moreover, the issue of 'trust' is generally important for socio-technical systems. 'Trust' can be defined in different ways, for the purpose of this paper it is sufficient to describe the problem as to whether and to which degree a human is willing to accept proposals from technical components, and to which degree he is willing to give up control. For a detailed survey on different definitions of trust in the context of automation systems, see e.g. [Lee and See, 2004]. In the context of expert systems, it has been shown that explanation capabilities have a large effect on the user's acceptance of advices given by the system [Ye and Johnson, 1995].

To summarize, the ability of an IT system to give good explanations is important for the functioning of a socio-technical system. Good explanations depend on the context, it would therefore be helpful to be able to include an analysis into the system design process.

C.3 Views on Explanations

In this section, we outline two perspectives on explanation: The *Explanation Goals* focus on user needs and expectations towards explanations and help to understand *what* the system has to be able to explain and *when* to explain something. The *Kinds of Explanations* focus on different *types* of explanations, their *usefulness* for the user, and how they can be represented in the different *knowledge-containers* [Richter, 1995].

Any kind of interactivity implies that one has some kind of user model that provides answers based on what the user knows and what he or she does not know [Richter, 1992]. The user (probably) knows about the used vocabulary, about general strategies, policies, or procedures to follow, and about (most of) the standard situations in the given problem domain. But he or she may not know all the details and data, about rare cases and exceptions, and about consequences of combinatorial number of interactions of different alternatives. Then, a basic approach to explanation would be to not comment on routine measures (without being asked), to emphasize on exceptional cases (e.g., exceptions from defaults and standards, exceptions from plausible hypotheses), and to allow for further questions.

It is hard to anticipate user needs due to two main reasons [Richter, 1992]: First, not all of the needs must be met, but those important to the user. Second, all deficits and their estimated importance depend on the specific user. Thus, personalization is a basic requirement, not only some added value.

C.3.1 Explanation Goals

Sørmo and Cassens [2004]; Sørmo et al. [2005] suggest several explanation goals for Case-Based Reasoning systems (which are valid for knowledge-based systems, in general). They also argue that those goals are indeed reachable because case-based reasoners are mostly made to perform limited tasks for a limited audience, thus allowing to make reasonable assumptions about the user's goals and the explanation context. The identified explanation goals are:

Transparency:

Explain how the system reached the answer
"I had the same problem with my car yesterday, and charging the battery fixed it."

The goal of an explanation of this kind is to impart an understanding of how the system found an answer. This allows the users to check the system by examining the way it reasons and allows them to look for explanations for why the system has reached a surprising or anomalous result. If transparency is the primary goal, the system should not try to oversell a conclusion it is uncertain of. In other words, fidelity is the primary criterion, even though such explanations may place a heavy cognitive load on the user. The original *how* and *why* explanations of the MYCIN system [Clancey, 1983] would be good examples.

This goal is most important with knowledge engineers seeking to debug the system and possibly domain experts seeking to verify the reasoning process [Gregor and Benbasat, 1999]. It is also reasonable to think that in domains with a high cost of failure it can be expected that the user wishes to examine the reasoning process more thoroughly.

Justification:

Explain why the answer is a good answer
"You should eat more fish - your heart needs it!"
"My predictions have been 80% correct up until now."

This is the goal of increasing the confidence in the advice or solution offered by the system by giving some kind of support for the conclusion suggested by the system. This goal allows for a simplification of the explanation compared to the actual process the system goes through to find a solution. Potentially, this kind of explanation can be completely decoupled from the reasoning process, but it may also be achieved by using additional background knowledge or reformulation and simplification of knowledge that is used in the reasoning process.

Empirical research suggests that this goal is most prevalent in systems with novice users [Mao and Benbasat, 2000], in domains where the cost of failure is relatively low, and in domains where the system represents a party that has an interest in the user accepting the solution.

Relevance:

Explain why a question asked is relevant

"I ask about the more common failures first, and many users do forget to connect the power cable."

An explanation of this type would have to justify the strategy pursued by the system. This is in contrast to the previous two goals that focus on the solution. The reasoning trace type of explanations may display the strategy of the system implicitly, but it does not argue why it is a good strategy. In conversational systems, the user may wish to know why a question asked by the system is relevant to the task at hand. It can also be relevant in other kinds of systems where a user would like to verify that the approach used by the system is valid. In expert systems, this kind of explanations was introduced by NEOMYCIN [Clancey, 1983].

Conceptualization:

Clarify the Meaning of Concepts
"By 'conceptualization' we mean the process of forming concepts and relations between concepts."

One of the lessons learned after the first wave of expert systems had been analyzed was that the users did not always understand the terms used by a system. This may be because the user is a novice in the domain, but also because different people can use terms differently or organize the knowledge in different ways. It may not be clear, even to an expert, what the system means when using a specific term, and he may want to get an explanation of what the system means when using it. This requirement for providing explanations for the vocabulary was first identified by Swartout and Smoliar [1987].

Learning:

Teach the user about the domain
"When the headlights won't work, the battery may be flat as it is supposed to deliver power to the lights."

All the previous explanation goals involve learning – about the problem domain, about the system, about the reasoning process or the vocabulary of the system. Educational systems, however, have learning as the primary goal of the whole system. In these systems, we cannot assume that the user will understand even definitions of terms, and may need to provide explanations at different levels of expertise. The goal of the system is typically not only to find a good solution to a problem, but to explain the solution process to the user in a way that will increase his understanding of the domain. The goal can be to teach more general domain theory or to train the user in solving problems similar to those solved by the system. In other words, the explanation is often more important than the answer itself. Systems that fulfill the relevance and transparency goals may have some capabilities in this area, but a true tutoring system must take into account how humans solve problems. It cannot attempt to teach the user a problem solving strategy that works well in a computer but that is very hard to reproduce for people.

For the remainder of this paper we will not focus on the learning goal since it is specifically targeted towards educational systems.

C.3.2 Kinds of Explanations

Roth-Berghofer [2004] looks at explanations from a knowledge-container perspective. He addresses the issue of what can naturally be explained by the four containers (see Section C.4).

One starting point is the work of Spieker [1991] on the usefulness of explanations. According to Spieker, there are five useful kinds of explanations he discusses in the context of expert systems:

Conceptual Explanations:

They are of the form 'What is ...?' or 'What is the meaning of ...?'. The goal of conceptual explanations is to build links between unknown and known concepts. Conceptual explanations can take different forms:

- Definition: "What is a bicycle?" "A bicycle is a land vehicle with two wheels in line. Pedal cycles are powered by a seated human rider. A bicycle is a form of human powered vehicle."
- Theoretical proposition: "What is force?" "Force is Mass times Acceleration."
- Prototypical example: "What is a bicycle?" "The thing, the man there crashed with."
- Functional description: "What is a bicycle?" "A bicycle serves as a means of transport."

Conceptual explanations are answers to extensional or descriptional questions.

Why-explanations:

Why-explanations provide causes or justifications for facts or the occurrence of events. Whereas the first concept is causal in nature and not symmetrical, the latter only provides evidence for what has been asked for. For example:

- Justification: "Why is it believed that the universe expands?" "Because we can observe a red shift of the light emitted by other galaxies."
- Cause: "Why is it believed that the universe expands?" "Because, according to the Big Bang theory, the whole matter was concentrated at one point of the universe and the whole matter moves away from each other."

Why-explanations explain single events or general laws and can consist of single causes/justifications (among others) or a complete list of causes/justifications.

How-explanations:

How-explanations are a special case of why-explanations, describing processes that lead to an event by providing a causal chain. They are similar to action explanations (see below) that answer how-questions. How-questions ask for an explanation of the function of a device, for example:

- "How does a combustion engine work?" "A combustion engine is an engine that operates by burning its fuel."

Purpose-explanations:

The goal of *Purpose-explanations* is to describe the purpose of a fact or object. Typical questions are of the form 'What is ... for?' or 'What is the purpose of ...?', for example:

- "What is a valve for?" "The valve is used to seal the intake and exhaust ports."

Cognitive Explanations:

Cognitive Explanations explain or predict the behavior of 'intelligent systems' on the basis of known goals, beliefs, constraints, and rationality assumptions. There are action and negative explanations:

- Action explanation: "Why was this seat post selected?" "For the given price, only one other seat post for this bicycle is currently available. But that seat post is too short."

- Negative explanation: "Why was no carrier chosen?" "A carrier is only available for touring bikes. The user did not choose a touring bike."

C.4 Knowledge Containers

Knowledge containers, according to Richter [1995]; Lenz et al. [1998], contain and structure the knowledge of a knowledge-based system. A knowledge container is a collection of knowledge that is relevant to many tasks. For rule-based systems, for instance, one can easily identify facts and rules as important knowledge containers. For CBR systems, Richter describes four knowledge containers: *vocabulary*, *similarity measures*, *adaptation knowledge*, and *case base*. They are depicted in Fig. C.1.

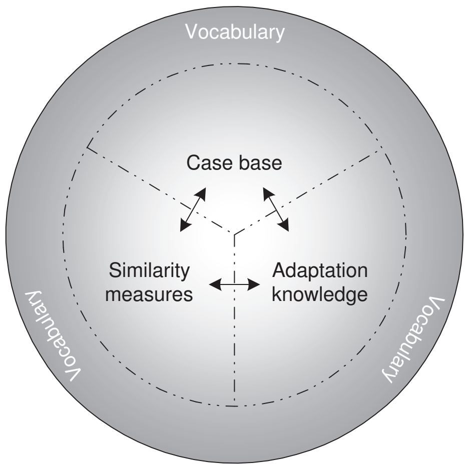

Figure C.1.: The four knowledge containers of a CBR system

The *vocabulary* defines attributes, predicates, and the structure of the domain schema. Thus the vocabulary forms the basis for all of the other three containers. Hierarchies, if available, can be used to order domain concepts.In object-oriented models, inheritance (*is-a*) and decomposition (*part-of*) induce hierarchical orderings quite naturally. Additional ontological relations can further add hierarchical information. Those hierarchies can be exploited for conceptual and (partly) for purpose explanations (because the ordering often is inferred from specialization/generalization). Other easily available information is information on the kind of attribute. *Input attributes* may be used to infer information for *retrieval attributes* as well as for filling *output attributes* of a query or a case. For example, imagine a CBR system for PC configuration in an electronic commerce scenario. The request for a multimedia PC triggers completion rules for filling such retrieval attributes as `processor` and `graphic card` accordingly. Not specified attributes of the query automatically become output attributes. The CBR system now could use the information for cognitive explanations based on why it filled the retrieval attributes etc.

Table C.1.: Knowledge containers and their contribution to explanations [Roth-Berghofer, 2004]

Knowledge container	**contributes to**
Vocabulary	conceptual explanations, why-explanations, how-explanations, and purpose explanations
Similarity measures	why-explanations, how-explanations, purpose explanations, and cognitive explanations
Adaptation knowledge	why-explanations, how-explanations, and cognitive explanations
Case base	why-explanations, how-explanations, and context

The knowledge that determines how the most useful case is retrieved and by what means the similarity is calculated, is held by the *similarity measures* container, which can be further divided into the sub-containers for local similarity measures and amalgamation functions. Each local measure compares values of one attribute of a case. It contains domain knowledge, e.g., about different processor speeds or graphic cards. Amalgamation functions are task oriented and contain utility knowledge (relevances for the task, e.g., the importance of the graphic card vs. the importance of the processor speed when selecting a multimedia PC). The already mentioned completion rules provide knowledge about dependencies between attributes.

The *adaptation knowledge* container covers the knowledge for translating a prior solution to fit a given query and the *case base* stores the experience of the CBR system, i.e., the cases. Knowledge about the types of cases used by the case-based reasoner, such as *homogeneous* vs. *heterogeneous* and *episodic* vs. *prototypical* cases [Watson, 1999] as well as cases of *rule* vs. *constraint* type [Richter, 1997], structures this knowledge container further.

Table C.1 shows an overview of which knowledge container contributes to which kind of explanation [see Roth-Berghofer, 2004, for details].

C.5 Exploring the Relations of Goals and Kinds

As we have outlined before, there is a need to take the context of explanations as well as different goals with and types of explanation into account. A methodology for the development of explanation-aware CBR systems should therefore comprise components for the workplace analysis (like ANT described in section C.2 or activity theory [Cassens, 2004]) as well as methods to translate the analytical findings into system synthesis. Further on, this process has to be integrated with methods for the continuous maintenance of the CBR system [Roth-Berghofer, 2003]. We propose therefore a overall process architecture as depicted in figure C.2.

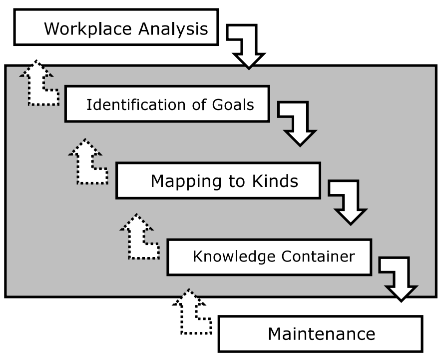

Figure C.2.: The overall process architecture.

During the remainder of this article, we will propose a 3-step process to identify which explanations a CBR system should be able to give and to understand how to make the necessary knowledge accessible in the different knowledge containers (see the grey box in figure C.2):

1. Use the *Explanation Goals* perspective to identify user needs for explanations from a user model and system view which takes the usage situation into account.

2. Use the *Explanation Kinds* view to find useful prototypical explanations and assess the requirements for contents that have to be modeled into the system.

3. Use the different *Knowledge Containers* to store the necessary knowledge to support the different kinds of explanation identified.

The mapping of goals to kinds and kinds to containers, respectively, is not necessarily a one to one relation which can be followed mechanically. The mapping proposed in this paper gives rather hints for the modeling task by focusing the work of the system designer on probable solutions.

As a simplified example, we look at a case-based diagnostic system for engine failures. We have a mixed initiative dialogue system where the system can ask questions about the engine status and the user can voluntarily provide information he deems important.[1] The system can give detailed explanations on possible causes for the problems as well as advice on how to avoid future occurrences. It is supportive, e.g., the user should be enabled to understand similar situations in the future without having to rely on the system.

There is no adaptation of cases since we are purely interested in the possible cause of a failure and not a solution to solve this problem. Further on, we assume the system to be capable of generating plausible and justified explanations itself without going into details about the underlying mechanism.

Conceptualization goal fullfilled by a conceptual explanation (definition):

During the symptom assessment, the system asks the user to fill in the specific gravity of the fuel. The user is not familiar with the term specific gravity so he asks the system to explain this. The system gives this explanation in the form of a *conceptual explanation*, in our example as a *definition*:
User: "What is the specific gravity?"
System: "It is a measure of the density of a liquid relative to the density of water, with water having a specific gravity of 1.0."

[1]Please, keep in mind that there are other possible human computer interfaces than natural language dialogue systems as depicted here for exemplary reasons. For example, one possibility is to provide the user with pre-formulated queries from which he or she can choose the most appropriate one.

Following our argument on the contribution of different Knowledge Containers to explanation kinds, the necessary knowledge for this kind of answers would come from the *vocabulary* of the system. Undoubtedly, conceptual explanations need much more knowledge than provided in most CBR systems, even in knowledge rich systems such as CREEK [Aamodt, 1994, 2004], where general (domain-dependent) knowledge is represented as semantic network.[2]

Relevance goal fullfilled by a causal why-explanation:

The user can further on ask to know why the system asked this specific question. The system should give explanations which set the question asked into context with the symptoms already known. This can be described in terms of a *causal why-explanation:*
User: "Why is the density of the fuel interesting?"
System: "Air/Fuel ratio is based on pounds of AIR to pounds of FUEL. Changing the specific gravity of your fuel can have a profound affect on the A/F ratio of your engine since carburetors are metered by volume."

Again, the necessary knowledge to answer this question can be found in the *vocabulary*. This causal why-explanation will typically rely on the domain knowledge modelled in the vocabulary of the system.

Justification goal fullfilled by a theoretical conceptual explanation:

The system finally presents a solution to the user which requires him to change the spark lead time. In this situation, the user might want to know why this is a solution. This goal can be satisfied by presenting a *theoretical conceptual explanation*:
User: "How is this adjustment related to the problem of too much CO_2 in the exhaust gases?"
System: "Different fuels and engine setups require different spark lead times. When using a wrong timing, the combustion process is suboptimal, and this will lead to high CO_2 values."

Also here, the necessary knowledge can best be stored in the *vocabulary* container. In order to answer this question we need some explicitly modeled domain knowledge. A knowledge poor CBR system could only point to cases and its own similarity measure and leave it to the user to find the explanation himself.

[2] In CREEK, explanations are generated to explain reasoning steps or to justify conclusions to the user, but mainly for the internal use of the reasoner.

Transparency goal fullfilled by a cognitive explanation:

The user might not be fully satisfied and tries to understand how the system came up with this answer. The system can explain the reasoning process using a *how explanation*:
User: "How did you come up with this answer?"
System: "Looking at my case base we can see that the most relevant cases had problems with the spark lead time as well, with only the exception of those cases where the problem can be explained by the wrong type of fuel."

The explanatory knowledge for this question can be found in the *similarity measure* of the system. The system needs to be able to explain why it delivered a certain case in terms of its similarity assessment. The *case base* container provides the context for the explanation by restricting the problem space to the available cases. Please note that a knowledge rich CBR system might be able to explain the absence of certain features in the solution case by referring to its domain knowledge, stored in the *vocabulary*.

C.6 Conclusions and Future Research Directions

We have outlined a unified view on explanations in Case-Based Reasoning, which takes both the goals of the user and the type of an explanation into account. Both perspectives are to a certain degree independent from each other.

The next step in our fellow work is to integrate an explanation goals view with methods for the analysis of workplace situations like ANT and activity theory (as proposed, e.g., by Cassens [2004]) and integrate the explanation kind perspective with existing design and maintenance methodologies (such as Inreca [Bergmann et al., 2003] and Siam [Roth-Berghofer, 2003]).

We want to develop further our structural view on explanations and supporting knowledge available in CBR systems, with the ultimate goal of providing a methodology on how to develop explanation-aware CBR systems in the future.

Bibliography

Agnar Aamodt. Knowledge-intensive case-based reasoning in CREEK. In Funk and Calero [2004], pages 1–15.

Agnar Aamodt. Explanation-driven case-based reasoning. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-*

Based Reasoning: Proceedings of the 1st European Workshop EWCBR 1993 – Selected Papers, volume 837 of *Lecture Notes in Computer Science*, pages 274–288, Berlin Heidelberg, 1994. Springer. ISBN 3-540-58330-0.

Ralph Bergmann, Klaus-Dieter Althoff, Sean Breen, Mehmet Göker, Michel Manago, Ralph Traphöner, and Stefan Wess. *Developing Industrial Case-Based Resoning Applications: The INRECA Methodology*, volume 1612 of *Lecture Notes in Computer Science*. Springer, Berlin, second edition, 2003.

Bruce G. Buchanan and Edward H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison Wesley, Reading, 1984.

Jörg Cassens. Knowing what to explain and when. In Gervás and Gupta [2004], pages 97–104.

William J. Clancey. The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence*, 20(3):215–251, 1983.

Daniel Cohnitz. Explanations are like salted peanuts. In Ansgar Beckermann and Christian Nimtz, editors, *Proceedings of the Fourth International Congress of the Society for Analytic Philosophy*, 2000. http://www.gap-im-netz.de/gap4Konf/Proceedings4/titel.htm [Last access: 2004-08-11].

Peter Funk and Pedro A. González Calero, editors. *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2004. Springer.

Pablo Gervás and Kalyan May Gupta, editors. *Proceedings of the ECCBR 2004 Workshops*, Technical Report 142-04, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, 23 (4):497–530, 1999.

Bruno Latour. Technology is society made durable. In John Law, editor, *A Sociology of Monsters*, pages 103–131. Routledge, 1991.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

David B. Leake, editor. *Case-Based Reasoning: Experiences, Lessons, & Future Directions*. AAAI Press/MIT Press, Menlo Park, 1996.

John D. Lee and Katrina A. See. Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80, 2004.

Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, and Stefan Wess, editors. *Case-Based Reasoning Technology: From Foundations to Applications*, volume 1400 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin, 1998.

Ji-Ye Mao and Izak Benbasat. The use of explanations in knowledge-based systems: Cognitive perspectives and a process-tracing analysis. *Journal of Managment Information Systems*, 17(2):153–179, 2000.

Eric Monteiro. Actor-network theory. In Claudio U. Ciborra, editor, *From Control to Drift*, pages 71–83. Oxford University Press, 2000.

Wolter Pieters. Free will and intelligent machines. Project Report, NTNU Trondheim, 2001.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Michael M. Richter. Remarks on current explanation research in artificial intelligence, 2005. Personal notes by Thomas R. Roth-Berghofer.

Michael M. Richter. *Prinzipien der Künstlichen Intelligenz*. B. G. Teubner, Stuttgart, 2. edition, 1992.

Michael M. Richter. Generalized planning and information retrieval. Technical report, University of Kaiserslautern, Artificial Intelligence – Knowledge-based Systems Group, 1997.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Funk and Calero [2004], pages 389–403.

Thomas R. Roth-Berghofer. *Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology*, volume 262 of *Dissertationen zur Künstlichen Intelligenz*. Akademische Verlagsgesellschaft Aka GmbH/IOS Press, Berlin, 2003.

Roger C. Schank. *Explanation Patterns – Understanding Mechanically and Creatively*. Lawrence Erlbaum, New York, 1986.

Roger C. Schank, Alex Kass, and Christopher K. Riesbeck, editors. *Inside Case-Based Explanation*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1994.

Frode Sørmo and Jörg Cassens. Explanation goals in case-based reasoning. In Gervás and Gupta [2004], pages 165–174. URL http://www.idi.ntnu.no/~cassens/work/publications/download/2004-ECCBR-WS-goals.pdf.

Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in case-based reasoning – perspectives and goals. *Artificial Intelligence Review*, 24(2):109–143, October 2005. ISSN 0269-2821. URL http://dx.doi.org/10.1007/s10462-005-4607-7.

Peter Spieker. *Natürlichsprachliche Erklärungen in technischen Expertensystemen*. Dissertation, University of Kaiserslautern, 1991.

William R. Swartout. What kind of expert should a system be? xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

William R. Swartout and Johanna D. Moore. Explanation in second generation expert systems. In Jean-Marc David, Jean-Paul Krivine, and Reid Simmons, editors, *Second Generation Expert Systems*, pages 543–585. Springer, Berlin, 1993.

William R. Swartout and Stephen W. Smoliar. On making expert systems more like experts. *Expert Systems*, 4(3):196–207, 1987.

Ian Watson. Survey of CBR application areas, 1999. Invited Talk at the 3rd International Conference on Case-Based Reasoning ICCBR.

L. Richard Ye and Paul E. Johnson. The impact of explanation facilities on user acceptance of expert systems advice. *MIS Q.*, 19(2):157–172, 1995. ISSN 0276-7783.

User Aspects of Explanation Aware CBR Systems

D

Author:
Jörg Cassens

Abstract:
This paper addresses the problem of embedding explanation-aware intelligent systems into a workplace environment. We outline an approach with three different perspectives, focusing on the work process as a whole as well as user interaction from an interface and a system view. The theoretical background consists of Actor Network Theory, Semiotics, and Activity Theory. We further propose to integrate this workplace analysis into a design process for knowledge-intensive and explanation-aware Case-Based Reasoning systems.

Main Result:
This paper applies three views on intelligent systems in workplace environments; 1. Work process view (using actor network theory), 2. HCI interface view (using semiotics), and 3. HCI system view (using activity theory); to address issues of explanation in intelligent systems.

Published in:
Maria Francesca Costabile and Fabio Paternó, editors, *Human-Computer Interaction – INTERACT 2005*, volume 3585 of *LNCS*, pages 1087–1090, Rome, 2005. Springer.

D.1 Introduction

Explanations are an important vehicle to convey information in everyday human-human interaction. They help us to understand one another and enhance the knowledge of the communication partners in such a way that they accept certain statements. The partners understand more, allowing them to make informed decisions. The need for explanations provided by knowledge-based systems is well documented [Swartout, 1983; Buchanan and Shortliffe, 1984; Swartout and Smoliar, 1987]. The adequacy of explanations is dependent on pragmatically given background knowledge. What counts as a good explanation in a certain situation is determined by context-dependent criteria [Leake, 1995].

Research on explanation is of interest today because it can be argued that the whole scenario on research on Knowledge-Based Systems (KBS) has changed: KBS are no longer considered as black boxes that provide a full solution to a problem. Instead, problem solving is seen as an interactive process (a socio-technical process). Problem descriptions as well as other input can be incomplete and changing. As a consequence, there has to be communication between human and software agents. Communication requires mutual understanding that can be essentially supported by explanations. Such explanations can improve the problem solving process to a large degree.

Case-Based Reasoning (CBR, [Aamodt and Plaza, 1994]) is a research area in the field of AI. Its aim is to understand and build systems which are able to use previous experience in order to solve new problems. A CBR system is able to learn by storing experience in the form of so called cases, which describe problems and their solutions. When a new problem arises, a sufficiently similar previous problem has to be identified and the former solution has to be adapted to the new problem. The new solution might also be based on more than one previous case.

We are suggesting a framework for the design of explanation-aware CBR systems which takes the usage of the system into account. We are therefore in need of methodologies which can describe the workplace environment in which the system is going to be used on the human-computer interaction level. This analysis is then integrated into the design process as a whole, as described in Roth-Berghofer and Cassens [2005].

In order to understand how the system fits into a workplace situation, we propose a theoretical framework which is focusing on three different perspectives:

- Work process view: Actor Network Theory,
- HCI interface view: Semiotics, and

- HCI system view: Activity Theory.

D.2 Work Process View: Actor Network Theory

We model the context in which the system is implemented with the help of the Actor Network Theory, ANT [Latour, 1991; Monteiro, 2000]. The basic idea here is fairly simple: whenever you do something, many influences on *how* you do it exist. For instance, if you visit a conference, it is likely that you stay at a hotel. How you behave at the hotel is influenced by your own previous experience with hotels, regulations for check-in and check-out, the capabilities the hotel offers you (breakfast room, lifts), amongst others.

In effect, you are not performing from scratch, but are influenced by a wide range of factors. The aim of ANT is to provide an unified view on these factors and your own acting. According to Monteiro, an actor network in this notion is 'the act linked together with all of its influencing factors (which again are linked), producing a network' [see Monteiro, 2000, p. 4].

In this network, you find both technical and non-technical elements. By this, the ANT avoids the trap of either overstating the role of technological artifacts in a socio-technological system or underestimating their normative power by applying the same framework to both human actors and technological artifacts.

This makes it possible for us to understand how technological artifacts influence the doing of human actors in much the same way as other human actors.

D.3 HCI Interface View: Semiotics

When focusing on the interaction of a particular user with the system, we use the semiotics approach [Nake, 1994; Andersen, 2001] to understand the peculiarities of interaction with intelligent systems. In the terminology of semiotics, human communication is a sign process. In contrast, conventional computer systems are only processing signals, lacking the necessary interpreting capabilities humans have.

We argue that in order to make intelligent systems work not merely as tools or media, but as actors to whose decision making abilities a human user can subscribe, the system must appear to the user as if it was capable of a meaningful interaction. Since both processes, sign and signal, have to be coupled, the goal is to make an intelligent system behave in such a way that the user ascribes to the system the ability to participate in a sign process. The upper-level analysis of the work process helps in defining the aspects of user interaction where this ascription has to succeed in order to make the user believe in the system's capabilities.

One important challenge here is the ability of the system to show its capabilities. This can be described as a communication problem: the system has to interpret the actions of the user in a meaningful way and itself present results that make sense for the user. This process of sense-making is highly interactive: an intelligent partner in a communication process asks (meaningful) questions if an unclear situation occurs and is able to explain its own actions. The semiotic approach is useful to analyse this sense-making process with the help of transferring knowledge about similar processes from other semiotic domains.

D.4 HCI System View: Activity Theory

In our framework, we use Activity Theory (AT, [Bødker, 1991]) to analyse the use of artifacts as instruments for achieving a predefined goal in the work process and especially to understand the transformation of the artifact itself and the individual and collective work practice during this process.

Since an AI system is more a partner in a work process than a tool, its role in the user interaction changes. Whereas a classical informatics system is a passive translator and memory of praxis, the intelligent system is constantly re-shaping praxis through its use. Looking at a decision support system, the decision making process itself is transformed by the ability of the system to react differently, e.g. through accumulated experience and usage context.

But since AT itself models artifacts as being preformed as socio-cultural entities, we can describe the artifacts in a way which takes this modification into account. Again, our upper-level model helps us to identify the mediation process and the role of both human and non-human actors in the usage process.

The ability of an intelligent system to adapt to the user is very important. In the process of re-shaping praxis, a user expects from an (as-if) intelligent system that it adapts to the changed situation. In the beginning of the usage of a Case-Based diagnostic system, it will be important to explain to the user in detail why a particular case was matched to a new problem, but the user expects from an intelligent partner that the same match will be explained in less detail when occurring very frequently (since the artifact should be changed by the changed praxis, that is here the accumulated knowledge on both parts). On the other hand, in the event of a breakdown situation, the level of detail in explanations given by the system should be increased again.

In addition, the notion of action cycles [Fjeld et al., 2002] is helpful for mapping the CBR system model to existing work processes. For example, identifying those situations where feedback to the system is both required and fits into the existing work process helps in avoiding obtrusive system behaviour.

D.5 Conclusions

The three perspectives presented allow system designers to analyse different socio-technical aspects of the targeted workplace environment. They can be used together to get a more complete model, but this is not always possible or even necessary. For example, in recent work we have investigated how Activity Theory alone can be used to model the knowledge needed in context-aware systems [Kofod-Petersen and Cassens, 2005].

On the other hand, combining these three approaches allows modelling different aspects of human-computer interaction ranging from the socio-technical network to the design of the user interface itself on the knowledge level. Our goal is therefore to integrate these perspectives further and combine them with other steps of the CBR system lifecycle [Roth-Berghofer and Cassens, 2005].

Bibliography

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994.

Peter Bøgh Andersen. What semiotics can and cannot do for HCI. *Knowledge-Based Systems*, 14:419–424, 2001.

Susanne Bødker. Activity theory as a challenge to systems design. In Hans-Erik Nissen, Heinz K. Klein, and Rudy Hirschheim, editors, *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pages 551–564. North Holland, 1991.

Bruce G. Buchanan and Edward H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison Wesley, Reading, 1984.

Morten Fjeld, Kristina Lauche, Martin Bichsel, Fred Voorhoorst, Helmut Krueger, and Mathias Rauterberg. Physcial and virtual tools: Activity theory applied to the design of groupware. *CSCW*, 11(1-2):153–180, 2002.

Anders Kofod-Petersen and Jörg Cassens. Activity theory and context-awareness. In Stefan Schulz, David B. Leake, and Thomas R. Roth-Berghofer, editors, *IJCAI-05 Workshop on Modelling and Retrieval of Context – Working Notes*, pages 1–12, Edinburgh, 2005. IJCAI. URL http://ceur-ws.org/Vol-146/paper1.pdf.

Bruno Latour. Technology is society made durable. In John Law, editor, *A Sociology of Monsters*, pages 103–131. Routledge, 1991.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

Eric Monteiro. Actor-network theory. In Claudio U. Ciborra, editor, *From Control to Drift*, pages 71–83. Oxford University Press, 2000.

Frieder Nake. Human-computer interaction – signs and signals interfacing. *Languages of Design*, 2:193–205, 1994.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, pages 451–464, Chicago, 2005. Springer. URL http://dx.doi.org/10.1007/11536406_35.

William R. Swartout. What kind of expert should a system be? xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

William R. Swartout and Stephen W. Smoliar. On making expert systems more like experts. *Expert Systems*, 4(3):196–207, 1987.

Explanation in Case-Based Reasoning – Perspectives and Goals

E

Authors:

Frode Sørmo, Jörg Cassens, and Agnar Aamodt

Abstract:

We present an overview of different theories of explanation from the philosophy and cognitive science communities. Based on these theories, as well as models of explanation from the knowledge-based systems area, a framework for explanation in case-based reasoning (CBR) is presented, based on explanation goals. We propose ways that the goals of the user and system designer should be taken into account when deciding what is a good explanation for a given CBR system. Some general types of goals relevant to many CBR systems are identified, and used to survey existing methods of explanation in CBR. Finally, we identify some future challenges.

Main Result:

A systematic overview on explanation in philosophy and cognitive sciences and a historic overview of the use of explanations in artificial intelligence are given. Five goals a user can have with explanations are introduced, namely 1. *Transparency* (explain how the system reached the answer), 2. *Justification* (explain why the answer is a good answer), 3. *Relevance* (explain why a question asked is relevant), 4. *Conceptualization* (clarify the meaning of concepts), and 5. *Learning* (teach the user about the domain). The use of explanations in case-based reasoning is reviewed and challenges are identified.

Published in:

Artificial Intelligence Review, 24(2):109–143, October 2005.

Copyright:

My main contributions to the paper:

- Outline of explanations in expert systems
- Survey of explanations in CBR

The following aspects were jointly developed by the authors:

- Goals were defined in close collaboration between Frode Sørmo and myself
- Identification of challenges

E.1 Introduction

The term explanation can be interpreted in two different ways in AI [Aamodt, 1991, p. 59]. One interpretation deals with explanation as part of the reasoning process itself, for example used in the search for a diagnostic result in order to support a particular hypothesis. The other interpretation deals with usage aspects: attempting to make the reasoning process, its result, or the usage of the result understandable to the user. This paper primarily deals with the latter interpretation, but explanation as part of the reasoning process is also addressed where appropiate.

In our daily lives we experience explanations every day, and they seem to exist in an unlimited number of forms. Everything from "I didn't wash the dishes because there was no more detergent", to "I hate shopping", and even "Because I said so!" can serve as satisfactory explanations in certain circumstances. Explanation is one of the concepts that everyone has a good intuition of, but which are very hard to explicitly define.

In this paper, we will attempt to characterize important aspects of an explanation, and relate them to explanations in and from CBR systems.

When reviewing the literature we find that many accounts of explanation explicitly recognize that the context of an explanation situation, and the goal of the user in that situation, influence what is and what is not a good explanation. While goal situations may vary a lot among domains, systems, and users, some goal situations are common. We will present a framework of explanation based on important explanation goals, and discuss how they place limitations on each other and how different kinds of systems may be better suited to fulfill different goals.

We will begin by looking at foundational and theoretical issues of explanation, as developed within philosophy and cognitive science (Section 2). This is followed, in Section 3, by views and models of explanation from within the expert systems and intelligent tutoring communities. In Section 4 we review current accounts of explanation in CBR and present a set of explanation goals for CBR systems. A brief survey on explanation in different CBR systems follows in Section 5. In Section 6 we highlight some challenges for the future before concluding with Section 7.

E.2 Philosophical and Cognitive Accounts of Explanation

People tend to think of explanation as something identifying the cause for a particular event or state, as for example in the sentence "the train is late because of a faulty stop light". This is also the case in many philosophical theories of explanation [see for instance Salmon, 1984]. However, in daily life we also use explanations that are functional ("there is rubber on the end of the pencil so you can erase mistakes") and intentional ("I turned off the light because I want to sleep" [Brewer et al., 1998]). This is further complicated by the fact that both the sender and recipient of an explanation have goals in the exchange, and their goals influence what candidate explanations are and are not acceptable [Leake, 1995a]. Thus it may be very hard to form a complete theory of explanation. We will characterize some accounts of explanations discussed in the philosophical and cognitive science communities.

E.2.1 Basic Philosophical Accounts

The nature of explanation has been studied extensively by philosophers, particularly by researchers in the philosophy of science. Here the targets for explanation are specific observations, predicted outcomes, or scientific theories themselves. Explanations are sought based on observations and existing knowledge. Two different approaches, or rather classes of approaches, emerged throughout the 50's and 60's. The logical deductive approach, suggested by Carl Hempel and Paul Oppenheim [1948; 1965] was linked with a positivistic view on science. This approach was severely criticized by several people, resulting in several suggestions of different, and more pragmatic, approaches to explanation. Important early contributions were given by Gilbert Harman [1965], Sylvain Bromberger [1965], and Wesley Salmon [1971].

The positivist approach takes a scientific theory to be an axiomatic formalization of a set of sentences in a logic system. Hempel and Oppenheim refers to it as a "deductive-nomological" (deduction from laws) explanation, also referred to as the "covering law model", reflecting that the theory subsumes or covers the things that are explained. This work was subsequently extended with a formal model of probabilistic inference as well, the "inductive-statistical" model [Hempel, 1965]. In order to analyze explanations formally, an explanation structure in both these models is defined to consist of two parts; the part that is to be explained, called the *explanandum*, and the explanatory expression, called the *explanans*. For example: The patient died (explanandum); The patient had cancer (explanans); The patient died because he had cancer (explanation).

While the pragmatic aspects of explanation are acknowledged by all philosophers of science (including Hempel), a characterization of the non-positivist tradition is that the pragmatics of an explanation situation, in terms of context, purpose, etc., is at the very basis of the nature of explanation. Pragmatics becomes the starting point for the understanding of explanation, rather than an additional challenge for axiomatic formalization.

Early advocates of pragmatic approaches criticized the deductive-nomological account for being too syntax-oriented, in that semantical interpretations (i.e. the content of theories) started out from the interpretation of the logical syntax of expressions, rather than from the needs of the real world. Pragmatic approaches attempt to offer a semantic that starts out from the real world, with the necessary or suitable syntax following from pragmatic needs. While deductive inference certainly is an important inference type, several philosophers have shown the importance of abductive inference – and particularly the form referred to as "inference to the best explanation" – as a frequently occurring inference type in hypothesis formation and evaluation [Josephson and Josephson, 1994]. The strict requirement of truth-preserving inference underlying logical deduction is relaxed here. Out of a set of hypotheses, the hypothesis that can best explain the facts is chosen. Originating from Charles Sanders Peirce, an early account in philosophy of science was suggested by Gilbert Harman [1965]. While other researchers have proposed abductive models of scientific discovery, Harman's model concentrated on justification. The basic idea behind his model was to argue that an inference from some data to the best explanation is a justified mode of inference and leads to true hypotheses.

Case-based reasoning is concerned with problems that are open-ended, and often changing, and uncertainty as well as incompleteness of theories and input descriptions are typically assumed. Viewing explanations as deductive proofs will be too severe a limitation for our purpose, and hence less relevant for the type of explanations CBR systems need to generate. A pragmatic view of explanation will therefore be accounted for in the following, while the Hempel-Oppenheim account sometimes will be used for comparison.

Philosophers who study linguistics and everyday speech have also made significant contributions to the nature of explanations. An early influential example is Sylvain Bromberger [1965], who in particular criticized two weaknesses of Hempel's and Oppenheim's theory. Through a series of examples he showed that perfectly valid deductive-nomological explanations can be made with true but irrelevant premises.

The second problem was related to the symmetrical properties of logical inference, particularly when the explanatory law has a functional form. The equations can be rewritten so that any of the variables becomes the value to explain, i.e. the explanandum. One of his famous examples is the flagpole example. When the line of sight of the sun across the top of a flagpole is at a given angle with the ground, the height of the flagpole and the length of the shadow it casts are related. Under the deductive-nomological model, it can be explained why the length of the shadow takes a given value by citing this law and the height of the pole. So far so good. But the equation and the length of the shadow can equally well be used to explain the height of the flagpole, i.e. to explain why the flagpole has the height it has, which seems entirely inappropriate in all but very peculiar situations.

Bromberger analysed explanation triggering questions in the form of why-questions, and suggested that an important type of question arises "when one believes that the presupposition is true, views it as a departure from a general rule, and thinks that the conditions under which departures from the general rule occur can be generalized" [Bromberger, 1966, p. 100]. Asking this type of why-question would then imply that the person asking it is in some way surprised about the fact implied in the why-question (the presupposition) while still believing its truth.

An early and influential approach to the treatment of causality in explanations was presented by Wesley Salmon [1971]. Salmon characterizes explanation as the pursuit of understanding, and to explain as to attribute a cause. As opposed to Hempel's experimentalist position, Salmon worked in the realist tradition. Salmon's "causal realism" theory of explanation started out from Bayesian probability, viewing an explanation basically as a set of statistically relevant factors, but he later found that theory inadequate in accounting for how explanations produce scientific understanding (see the following subsection).

E.2.2 Later Philosophical Accounts

Later accounts include continued work on scientific explanation by Bas van Fraassen [1980], Wesley Salmon [1984], and Paul Thagard [1988], explanation in natural language by Peter Achinstein [1983], as well as cognitive models of explanation, by Roger Schank [1986], Robert Keil and Frank Wilson [2000], and David Leake [1995b]. Some of these theories are also applicable to everyday explanations.

One of these is formulated by Bas van Fraassen in his book *The Scientific Image* [van Fraassen, 1980]. Van Fraassen takes a strictly empiricist approach (often referred to as "constructive empiricism"), and claims that an explanation is always an answer to an implicit or explicit contrastive why-question. By 'contrastive', he means a question of the form "Why S_0 rather than $S_1 \ldots S_n$?" where one state or event is preferred over a set of alternatives. For example, the explanation "The train is late because of a faulty stop light" is an answer to the question "Why is the train somewhere else rather than here?" According to van Fraassen, an acceptable explanation must favor the observed state S_0 over the other states. By this, he means that the answer or explanation must increase the probability of the observed state S_0 relative to $S_1 \ldots S_n$. He suggests that this can be calculated by applying Bayes' Rule to each candidate answer. As long as each candidate satisfies the previous criteria of favoring the observed state, van Fraassen claims there are no objective criteria for preferring one over another, but that the context of the question implicitly contains information about which answer the receiver would prefer. Perhaps the most useful feature of van Fraassen's theory for application in knowledge based systems is that it suggests a minimum criterion an explanation must fulfill (it must favor the observed state) as well as a framework for understanding explanations (as answers to contrastive why-questions).

Salmon's later account on causal explanation was triggered by problems of causal relevance and causal asymmetry in his early account, and by the distinction between true causal processes and pseudoprocesses. An example illustrating the latter difference is the beam of a torch as the torch is moved by hand so the light describes an arc through the sky. The movement of the beam is a pseudoprocess, since later stages of the beam are not caused by earlier stages, while the hand movement of the torch itself is a true causal process - as is the electrical production of light within the torch. A central idea in his "causal mechanical" model of explanation is that a causal process is a physical process that is characterized by being able to transmit a "mark" in a continuous manner. A mark is a local modification to the physical structure involved, such as a scratch in its surface. True causal processes have marks, pseudoprocesses not. A second element in his theory is the notion of causal interaction, through which marks are transmitted between causal processes. According to the causal-mechanical model, an explanation of some phenomenon will trace the causal processes – including interactions – which lead up to the phenomenon, and describe the processes and interactions of the phenomenon itself. If successful, the explanation will show how the phenomenon to be explained fits into a causal structure. Salmon developed a detailed and complex theory, resulting in a set of instructions for how to produce an explanation by creating a causal model for a given phenomenon.

An influential follower of Bromberger in the philosophy of natural language is Peter Achinstein [1983]. He follows the tradition that a request for explanation is a request for understanding of something. He addresses questions such as: Why have the standard models of scientific explanation been unsuccessful? What is causal explanation, and must explanation in the sciences be causal? What is a functional explanation? He emphasizes the role of the explanation process – the explaining act in which someone writes or utters something to someone else. What is written or uttered in this process is called the explanation product. Achinstein's view is that an explanation (product) can not be understood or evaluated without reference to the explaining act, which leads to his "illocutionary" theory of explanation. The explaining act defines some aspect of the context and purpose behind the explanation which is needed for a correct and meaningful interpretation of the explanation product.

He believes this request can take many forms, not just the why-questions of Bromberger and van Fraassen but any number of questions (why, what, where, how, etc.). Achinstein says that an explanation is the intention of giving someone the knowledge to understand some phenomena from some frame of reference. Like van Fraassen, Achinstein suggests that there is further preference for some explanations over others, and that this preference is defined by the context of the conversation and ultimately in the control of the individual requesting the explanation. For example, an explanation that a train is full because it is the rush hour may be useful for a passenger, but for the train scheduling department a more useful explanation is that too few trains are scheduled at this time of the day.

This view of explanations suggests that a very wide variety of statements can serve as explanations. An explanation need not, for example, be a causal chain of events leading up to the matter to be explained. The explanation may have as a goal facilitating the formation of such a causal chain by the recipient, but it need not contain it explicitly. It is enough to supply the recipient with the knowledge that he needs in order to infer it. This is a case of observing one of the 'rules of communication' often seen in human conversation: Only information that is not obvious should be communicated. If someone asks "Why is Peter not here?" a perfectly good explanation can be "Anne is sick" if the explainer is aware that the recipient knows that Peter has a daughter called Anne and that he has to stay at home and take care of her when she is sick.

On the one hand, this emphasizes the value of knowing the recipient quite well and it suggests that to form efficient explanations, accurate user models may be necessary. On the other hand, it alleviates the requirement of the explainer to put forward a complete explanation if the system can make reasonable assumptions about what the recipient knows and is capable of. For instance, an Artificial Neural Network that is trained to compare two pictures of a certain type can give a similarity measure, e.g. from 0 to 1, but it is difficult to explain how it came up with this score in a way most people can understand. However, presenting the pictures to the user so he can validate the similarity for himself can itself serve as an explanation. For many types of pictures, it is a reasonable assumption for the system to believe that the user is able to compare the pictures quite well on his own. Note that this is only the case if the goal of the receiver is to gain understanding of how good an answer the system has supplied. If the goal is to gain understanding of how the system arrived at the conclusion, the above explanation is far from sufficient.

Like Achinstein [1983], Thagaard [1988] is concerned with the pragmatics of an explanation. He developed what he calls a "computational philosophy of science", based on computational metaphors of epistemology, and by implementing and testing his theories in computer programs. Thagard also views explanation as a process of providing understanding, and understanding is to a large extent achieved through locating and matching. This is a view of reasoning based on retrieval and adaptation of knowledge structures, functionally similar to the mops (memory organization packages) in Schank's [1983] theories, see Section E.2.3. In Thagaard's early model, called PI, the knowledge structures – based on concrete or generalized situations or episodes – are supplemented with more general knowledge in the form of rules. In order for an explanation to be understood, it must activate this 'mental model' in a meaningful way – that is in a way that enables the existing knowledge structure to confirm the explanation without seriously contradicting other parts of the knowledge structure. On this basis, Thagard developed a theory referred to as "explanatory coherence", based on the notion of a coherent body of knowledge.

The notion of knowledge coherence – as a relaxation of the formal notion of consistency – has been adopted by many people, including AI researchers (e.g. Douglas Lenat [1987]). Paul Thagard [1989], however, takes this further into a theory of explanation. Coherence, in this theory, is basically a property over a set of propositions. It only makes sense to talk about coherence of a single proposition if viewed with respect to another set of propositions. The notion of acceptability is introduced to characterize this property of single propositions. Starting out from a model of abductive inference, in the sense of inference to the best explanation, he identifies three important criteria for selecting the best explanation: *conscilience* (favoring explanatory breadth), *simplicity* (favouring explanations with few propositions), and *analogy* (favouring explanations based on analogies). Thagard's work not only presents an approach to scientific explanation, but also defines the role of explanation within a wider theory of coherence-seeking abductive inference. His research has focused on analogy and case-based reasoning [Thagard and Holyok, 1989], as well as other computational models, which include connectionist networks and probabilistic network models.

Additional philosophical accounts of explanation include the "unificationist" accounts of Michael Friedman [1974] and Philip Kitcher [1976] and the information theoretic model of Joseph Hanna [1982]. The basic idea of the former is that a scientific explanation should attempt to unify a range of different phenomena. A successful unification may reveal relationships between phenomena, previously unknown or unaware of, which seems to be something that good explanations are expected to do. Hanna proposed the notion of "transmitted information", coming from information theory, as the basis for evaluating the goodness of an explanation. Hanna responds to a crucial problem with Hempel's inductive-statistical model in that it does not adequately take relevance into account. Transmitted information, according to Hanna, reflects a relevance relation, which in turn is linked with explanatory power.

E.2.3 Cognitive Science Accounts

Thagard's research, as described above, also spans the philosophy of mind, and hence is positioned within the field of cognitive science as well as philosophy. Several other researchers in this community have also addressed the issue of explanation related to cognition.

Roger Schank and colleagues further developed Schank's "dynamic memory" [Schank, 1983] theory of reminding, problem solving, and learning, into a theory of explanation generation and evaluation. As one of the founders of CBR as we know it today, he proposed a case-based approach to explanation, based on storing, indexing, and retrieval of "explanation patterns" [Schank, 1986]. Explanation patterns are specific or generalized cases of explanation events. A particular focus has been the exploration of case-based reasoning as a platform for creativity [Schank and Leake, 1989]. In this model, creativity comes from retrieving explanations related to a situation, but using them in new ways - referred to as "tweaking" of explanations. Depending on the retrieval and adaptation processes used, CBR has the potential to provide solutions to a range of creativity tasks, from close to copying old solutions up to producing novel ideas. The following has been a focusing problem for studying various types of explanations:

In 1984, Swale was the best 3-year-old racehorse, and he was winning all the most important races. A few days after a major victory, he returned from a light morning gallop and collapsed outside his stable. The shocked racing community tried to figure out why. Many hypotheses appeared, but the actual cause was never determined.

The experimental system that implements several of the methods investigated, the SWALE system, attempts to explain the anomaly in Swale's premature death [Leake, 1992]. It generates explanations of why Swale died by retrieving and tweaking remindings of explanation patterns for other cases of death. The approach has demonstrated the generation of a variety of interesting possible explanations of its death, including a heart attack (the "Jim Fixx explanation pattern"), and a drug overdose (the "Janis Joplin explanation pattern").

Abductive inference also has a central position in cognitive accounts of explanations (including Thagard's work, see Section E.2.2). Extending from his earlier research on explanation patterns, David Leake [1995b], in his work about models for everyday abductive explanations, identifies a set of issues related to comparing abductive reasoning methods. One is the issue of *when to explain something*, which links to the central ability of the reasoner itself to decide when an explanation is merited. Leake considers both plausibility criteria and the role of goals. He divides traditional plausibility criteria into the three groups of *structural minimality criteria*, motivated by the principle of Occam's razor, *proof-based approaches*, which are based on an evaluation of the generated proof-like explanations, and *probabilistic and cost-based criteria*, that focus on the costs and probabilities related to the generated explanations.

In contrast to these syntactic-oriented criteria, a set of goal-based criteria are suggested [Leake, 1995a]. Explanations are assumed to have two roles – either as a support of a claim or an argument against it. This work follows the tradition of Lalljee and Abelsen [1983], who suggest that explanations can be either 'constructive' or 'contrastive', and Schank [1983], who specifies that an explanation is required first and foremost in anomalous situations that do not fit a person's internalized model of the world (cf. the "surprises" assumed by Bromberger, Section E.2.1). Leake's view on explanation is related to the natural language philosophy view outlined before, in the sense that it takes the recipient's frame of reference into account. However, Leake has an operational view on explanations and not a purely descriptive one. While Achinstein deals with general communication issues, Leake focuses on the evaluation of given explanations for the actor. In this sense, Leake's theory can be seen as an operationalization of certain aspects of a more general theory of communication.

In the book *Explanation and Cognition*, Keil and Wilson [2000] collect recent research on explanations from a cognitive science point of view. They set out to study a set of questions about explanation, such as: "Are there different kinds of explanation?", "Do explanations correspond to domains of knowledge?", and "How central are causes to explanation?".

These questions are examined by studying for example whether there are fundamental differences between explanations offered and requested by children and those used by scientists [Brewer et al., 1998]. Keil and Wilson describe three broad types of explanation; the scientific, the narrative and the goal-based. The narrative explanation is what we use in daily life to chain together events. An example of this would be to explain that a window is broken because the children playing football in the back yard accidentally kicked the ball through the glass. This kind of explanation contrasts with explanations that explain events from generalized principles, which Keil and Wilson call scientific explanations. The last type, the goal-based explanation, are useful to explain actions in terms of the actors' goals. For instance, the workings of a car may well be described by mechanical laws, but the reasons for building it are better explained in terms of the goals of car manufacturers and consumers.

While the scientific explanation typically can be used to predict events from a set of observations, the narrative explanation can be formed after the fact and has little in the way of predictive power. Keil and Wilson claim that narrative explanations are more intuitive to people. They suggest that these explanations are useful in that they may narrow down the inductive space or help us gather information in a more efficient fashion. For instance, a spectator at a cricket match may ask questions about the rules so that he is better able to understand and gather information about the game in real time. In this scenario, prediction may not be very relevant to the spectator – he is simply trying to understand the game.

From an AI perspective, the difference between the narrative and scientific explanations is interesting. In expert systems, explanations initially focused on how the system made the prediction by showing how it followed from generalized rules. In essence, the system attempted to show how the conclusion must follow from the knowledge contained in the system. Although the process used to do this was not necessarily or typically deductive (at least in expert systems), the explanations produced seem to be closer to the scientific explanations than the narrative. Applying this to case-based reasoning, it seems likely that using a similar case to justify a conclusion is closer to a narrative account than using a rule from a rule-based system. However, the case will not typically contain a narrative account of how a conclusion followed from the findings. Rather, the way a case is used is that it represents a very local "rule" for drawing the conclusion but if the case contains a causal account of how the solution followed from the findings, it is not typically used by the system for explanation. Keil and Wilson suggest that depending on the goals of the users, they may not seek to know how the system found the case, but rather how the case's solution is a product of its findings.

We round off this Section with a final remark about the goodness of an explanation. We have earlier shown that the truth, or correctness, of an explanation is generally not sufficient to make it good. The flagpole height explanation is one example. An overly general explanation is another. What about necessity? Is correctness – or truth – a necessary criterion for a good explanation? One of the counter-arguments is related to the notion of truth. McDermott [1987] argues that an explanation may be good merely by making the observed facts probable, not necessarily proving their truth. Another argument is related to pragmatics. Achinstein [1983, p. 108] expresses it as follows: "The goodness or worth of an explanation is multidimensional; correctness is only one dimension in an evaluation. An explanation is evaluated by considering whether, or to what extent, certain ends are served. The ends may be quite varied. They may concern what are regarded as universal ideals to be achieved, particularly in science, e.g. truth, simplicity, unification, precision. Other ends are more 'pragmatic'."

E.3 Explanations in Expert Systems

In early rule-based expert systems like MYCIN the user could ask *how* the system reached the conclusion presented, and an explanation in the form of a reasoning trace from the system would be presented. This would offer the user a degree of transparency into how the system reached its conclusions. The user could also choose a *why* explanation that would provide a more local explanation that justified why a question was asked.

It was soon found that this capability was insufficient for answering many of the explanation requests from users. For instance, the problem solving strategy of a rule-based expert system is implicitly defined in the system, but was not explicitly encoded in such a way that it was accessible or easily explained to an end user. NEOMYCIN extended MYCIN's capabilities in this respect by explicitly encoding strategic information [Clancey, 1983].

Another notable extension was the XPLAIN system [Swartout, 1983]. This system would record additional domain knowledge associated with each rule, so that the system could produce explanations that gave background information for the rule, and pointers to literature.

The focus of these early extensions was usually to extend the explanation capabilities by adding the type of knowledge required by the user. These explanations could be divided into four types [Swartout and Smoliar, 1987; Chandrasekaran et al., 1989; Gregor and Benbasat, 1999]:

- **Reasoning Trace:** Producing an explanation from the trace of the reasoning process used by the system to find the solution. Examples are MYCIN's *how* and *why* explanations [Clancey, 1983].

- **Justification:** Providing justification for a reasoning step by referring to deeper background knowledge. This type of explanation was first offered by the XPLAIN system [Swartout, 1983].

- **Strategic:** Explaining the reasoning strategy of the system. The NEOMYCIN system first provided this kind of explanation [Clancey, 1983].
- **Terminological:** Defining and explaining terms and concepts in the domain. This type of explanation was identified in [Swartout and Smoliar, 1987].

Although it was found that expert system designers, and to some extent domain experts, appreciated the reasoning trace explanations, many end users did not understand or were not interested in the inner workings of the expert system. Later analysis of failed expert systems suggested that many of the attempts to provide explanations in early systems failed because they were incomprehensible to the user or failed to address the users' goals in demanding an explanation [Majchrzak and Gasser, 1991].

In response to this, further research went into how explanations could better be generated dynamically to fit the user's needs and goals. In Swartout and Moore [1993], five requirements for the explanation capability of expert systems were put forth. The *fidelity* requirement says that the explanation given should mirror the knowledge used by the system in its reasoning. The explanation should also have *low construction overhead* or justify any increased resources spent on it. It must always remain *efficient* and not degrade runtime capability. The explanation produced must also be *understandable* to the user, and must be *sufficient* in that enough knowledge must be represented in the system to answer the question the user may have.

The fidelity criterion mentioned above appears more controversial than the other four. Wick and Thompson [1992] argue that explanation should be viewed as a problem-solving process separate from the process used to determine the conclusion in the first place. They admit that while expert system designers need explanations that accurately represent the reasoning done by the system, this may be inappropriate for an end user. They suggest three major explanation goals. *Verification* is the goal of the knowledge engineer in verifying that the system works as it should. A successful verification explanation would accurately and precisely convey the knowledge of the system on the knowledge level. *Duplication* is to help the domain expert examine the knowledge of the system. The system should not only expose its own knowledge, but help the user learn the methods and knowledge used in the problem solving process. Finally, the goal of *ratification* is to increase the end user's confidence in the system's conclusion.

Wick and Thompson suggest that each of these goals has different audience and focus. As the goal moves away from verification toward ratification, the explanation process should increasingly be decoupled from the reasoning process in order to provide explanations that focus on the solution. This allows the system to convey tailored information about the domain to the user. The higher degree of decoupling from the original reasoning processes will decrease the fidelity of the explanation as defined by Swartout and Moore above, but Wick and Thompson point out that explanations provided by human experts also tend to lack fidelity, although they are nevertheless perceived as useful.

As expert systems have been deployed in production environments, empirical studies have been conducted to identify when different kinds of users ask for explanations, and what they expect to get from them. Results from this research include the observation that novices tend to ask for explanations to learn or clarify preferring justification and terminological explanations [Mao and Benbasat, 2000]. Experts tend to require explanations to verify the reasoning of the system and explain away surprising results. As such, they tend to prefer strategic and reasoning trace explanations. A full survey of the empirical studies on explanations is beyond the scope of this paper, but we recommend Gregor and Benbasat [1999] for a more in-depth review.

A number of educational systems have also been built as extensions of expert systems. These systems have as their goal not only to help the user solve a problem, but also teach the user about the domain. One idea emerging from these systems is that it is often beneficial for learning if the user participates in the formation of explanations. The Cognitive Tutor [Aleven and Koedinger, 2002] system assists students in explaining solutions to geometry problems. They find that this helps the students learn the task better and helps them avoid bad generalization. Ford et al. [1993] use Concept Maps to help the student navigate an expert model to form explanations.

E.4 Explanation in CBR

We have reviewed several attempts to define criteria for explanations and categorizations of different kinds of explanations. Philosophical accounts focus on criteria for scientific explanations, while the cognitive accounts describe how humans use explanations in a wide range of contexts. However, many explanations may be produced that are not perceived as useful in a given context. This happens even if they fulfill criteria of what is considered a good explanation.

The research on explanation within expert systems provides a focus for a situational context that is similar to what we find with most case-based reasoning systems. Although the technology for generating and presenting advice is different from traditional rule-based expert systems, most CBR systems today are computer systems that give decision advice to human users. Because of this similarity in situational context, it is reasonable to believe that the typology of explanations useful in expert systems will be a good fit for CBR. In this section we introduce five explanation goals that are strongly influenced by expert systems.

Below the abstraction level of the explanation goals, we need to look at particular issues in applying these goals to CBR. For instance, traditional rule-based systems paraphrased the rules to form explanations. While CBR systems typically do not have rules, the basic unit of knowledge in CBR – the case – can also be used to produce explanations. It has long been an article of faith in the CBR community that displaying an earlier solved case that represents a situation similar to the present problem situation can serve as a good explanation for adopting the solution of the previous case. After presenting the explanation goals, we will examine this approach further. In addition, we will discuss if cases are really the only source of knowledge that should contribute to explanations in a CBR system.

E.4.1 Explanation Goals

We will designate our explanation categories based on a set of *explanation goals*. We do this in order to recognize that a single explanation technique can serve many of these goals at once, and that not all of these goals are of equal importance in all systems. The goals are based on the four content categories from Gregor and Benbasat [1999] as presented in the Section E.3. In addition, we have a category that focuses on the learning perspective, similar to the Duplication goal of Wick and Thompson [1992]. Our aim is not to provide an exhaustive list of goals – the rationale for introducing them is to discuss how some current explanation criteria, and methods, hold up in the light of these goals which have proved quite universal in expert systems.

Explain How the System Reached the Answer (Transparency)

"I had the same problem with my car yesterday, and charging the battery fixed it."

The goal of an explanation of this kind is to impart an understanding of how the system found an answer. This allows the users to check the system by examining the way it reasons and allows them to look for explanations for why the system has reached a surprising or anomalous result. If transparency is the primary goal, the system should not try to oversell a conclusion it is uncertain of. In other words, fidelity is the primary criterion, even though such explanations may place a heavy cognitive load on the user. The original *how* and *why* explanations of the MYCIN system would be good examples.

This goal is adapted from the reasoning trace type of explanations from Gregor and Benbasat [1999] and the verification goal of Wick and Thompson [1992]. As they suggest, this goal is most important with knowledge engineers seeking to debug the system and possibly domain experts seeking to verify the reasoning process. It is also reasonable to think that in domains with a high cost of failure it can be expected that the user wishes to examine the reasoning process more thoroughly.

Explain Why the Answer is a Good Answer (Justification)

"You should eat more fish - your heart needs it!"
"My predictions have been 80% correct up until now."

This is the goal of increasing the confidence in the advice or solution offered by the system by giving some kind of support for the conclusion suggested by the system. This goal allows for a simplification of the explanation compared to the actual process the system goes through to find a solution. Potentially, this kind of explanation can be completely decoupled from the reasoning process such as advocated by the ratification goal of Wick and Thompson, but it may also be achieved by using additional background knowledge (as in XPLAIN) or reformulation and simplification of knowledge that is used in the reasoning process. As such, this goal also contains the category of justification explanations from Gregor and Benbasat [1999]. Empirical research suggests that this goal is most prevalent in systems with novice users [Mao and Benbasat, 2000], in domains where the cost of failure is relatively low, and in domains where the system represents a party that has an interest in the user accepting the solution. Some e-commerce recommender systems fall into this category, although Herlocker et al. [2000] suggest that in high-cost domains (such as expensive vacation packages compared to relatively cheap books or music) users are unlikely to accept solutions without more in-depth explanations.

Explain Why a Question Asked is Relevant (Relevance)

"I ask about the more common failures first, and many users do forget to connect the power cable."

An explanation of this type would have to justify the strategy pursued by the system. This is in contrast to the previous two goals that focus on the solution. The reasoning trace type of explanations may display the strategy of the system implicitly, but it does not argue why it is a good strategy. In conversational systems, the user may wish to know why a question asked by the system is relevant to the task at hand. It can also be relevant in other kinds of systems where a user would like to verify that the approach used by the system is valid. In expert systems, this kind of explanation was introduced by NEOMYCIN (and was one of the types of explanation discussed in the previous section).

Clarify the Meaning of Concepts (Conceptualization)

"By 'conceptualization' we mean the process of forming concepts and relations between concepts."

One of the lessons learned after the first wave of expert systems had been analyzed was that the users did not always understand the terms used by a system. This may be because the user is a novice in the domain, but also because different people can use terms differently or organize the knowledge in different ways. It may not be clear, even to an expert, what the system means when using a specific term, and he may want to get an explanation of what the system means when using it. This requirement for providing explanations for the vocabulary was first identified by Swartout and Smoliar [1987].

Teach the User About the Domain (Learning)

"When the headlights won't work, the battery may be flat as it is supposed to deliver power to the lights."

All the previous explanation goals involve learning – about the problem domain, about the system, about the reasoning process or the vocabulary of the system. Educational systems, however, have learning as the primary goal of the whole system. In these systems, we cannot assume that the user will understand even definitions of terms, and may need to provide explanations at different levels of expertise. The goal of the system is typically not only to find a good solution to a problem, but to explain the solution process to the user in a way that will increase his understanding of the domain. The goal can be to teach more general domain theory or to train the user in solving problems similar to those solved by the system. In other words, the explanation is often more important than the answer itself. Systems that fulfill the relevance and transparency goals may have some capabilities in this area, but a true tutoring system must take into account how humans solve problems. It should not attempt to teach the user a problem solving strategy that works well in a computer but that is very hard to reproduce for people.

This goal has similarities with the duplication goal of Wick and Thompson [1992], where the system should be able to explain itself on the knowledge level in order to transfer its knowledge to a user. Although Wick and Thompson claim that this goal is primarily for the domain expert to gain an understanding of the system's capabilities, the name and description suggest that the goal is to transfer the knowledge contents and competence of the system to the user. The participatory explanation techniques [Ford et al., 1993; Aleven and Koedinger, 2002], where the system helps students form explanations, are good examples of techniques for achieving this goal.

E.4.2 The Case as Explanation

The case-based reasoning methodology seems quite transparent. It is fairly easy to understand the basic concept of searching for very similar, concrete cases and base the decision-making on them. This understanding has supported the basic approach to explanation in CBR – displaying the case that is most similar to the problem case. In addition to the intuitive feeling and ad hoc reports that this works, there has been research showing that displaying cases along with the solution significantly improved user confidence in the solution compared to only showing the solution, or displaying a rule that was used in finding the solution [Cunningham et al., 2003].

There is also theoretical support for the case-as-explanation method fulfilling the justification goal by looking at it from the viewpoint of Achinstein's theory. It is likely that a previous example with a high degree of similarity would increase the relative probability of the solution from this case compared to other solutions [Faltings, 1997]. However, the underlying assumption of this approach seems to be caught better by van Fraassen's [1980] framework. Displaying the retrieved case to the user is a kind of knowledge communication that allows the user to make his own judgment about the similarity of the old situation compared to the current one.

Both of these views depend on the user's ability to understand the case and to confirm the similarity assessment. In general, for the retrieved case to serve as an explanation to the user, the similarity between the retrieved case and current problem must be obvious to him. The difficulty for the user in comparing cases increases as the case structure becomes more complex and the similarity measures more convoluted. It also increases with the use of more complex adaptation techniques where the retrieved case may not be the most similar but one that fits the adaptation process, as suggested e.g. by Smyth and Keane [1998].

There is another problem. Displaying the case may serve as a window into the methodology of the reasoner. It does, however, not help the user understanding how the symptoms connect with the solution. End users may be less concerned about how the most similar case was found than why the solution in the presented case works. Based on Keil and Wilson's [2000] work presented in Section E.2.2, we suggest that such an account would be required for the case to serve as a narrative episode and explanation for humans.

Schank's research suggests that people do use single cases to explain extraordinary situations where no more general theory covers the situation – they are a sort of index of situations where the general model failed. However, his theory also suggests that these single exceptions are perceived as very tentative in their predictive power compared to general knowledge that has been confirmed again and again. As we will see in Section E.5, some recent research in CBR attempts to address these shortcomings.

E.4.3 Knowledge Containers

The competence of a knowledge-based system depends on the knowledge sources available to it. Richter describes the knowledge sources used in problem solving as knowledge containers [Richter, 1995]. Rule-based systems typically have *facts* and *rules* as knowledge containers, while Richter identifies four such containers for case-based reasoning systems – the *case base*, the *similarity measure*, the *adaptation knowledge* and the *vocabulary*.

The *vocabulary* provides the basis for the other knowledge containers by defining the terms and structure of the domain. The *case base* contains the concrete or prototypical problems previously solved by the system or otherwise provided to it. The *similarity measure* contains knowledge on how to compare cases and compute a similarity ranking of cases relative to a new problem, while the *adaptation knowledge* allows the reasoner to change the solution of a previous case to better fit a new problem.

Richter points out that given a complete case coverage of the problem domain, the similarity measure and adaptation problems become trivial since any problem can simply be looked up. Similarly, if we have perfect adaptation knowledge so that any old solution can be adapted to a perfect new solution, the process only requires a starting position for the adaptation so that there is little need for cases or a similarity measure. Finally, if the system is always able to order the cases so that the cases with the correct solution are ranked highest, a classification system only needs a case representing each solution class, and there is no need for adaptation knowledge. This means that CBR systems may put different weights on these containers depending on what is most convenient for the domain and system.

Roth-Berghofer [2004] points out that this insight by Richter places in doubt the idea that displaying the best case is a sufficient explanation – at least if the system places any weight on the other knowledge containers. If much of the competence of the problem solving emerges through adaptation, it will be hard to explain the reasoning of the system without using the adaptation knowledge. This is certainly true if the goal of the explanation is to provide transparency into the system, but it can also become a problem in learning and justification if a solution is justified by a case that is not obviously similar and has a slightly different solution than that suggested by the system. In addition, conceptualization and relevance explanations cannot be provided by the case base. The vocabulary container seems perfect to provide explanations that serve to help conceptualization, but it is not clear from where strategic explanations can emerge. Possibly this requires a fifth knowledge container in CBR in the same way that it required a different level of representation in rule-based expert systems.

E.5 Survey of Explanation in CBR

In this section, we review explanation techniques in different case-based reasoning systems, with an emphasis on the more recently developed techniques. Many early CBR systems also had explanation capabilities, extensive surveys of which have been published elsewhere, for instance Kolodner [1993].

E.5.1 *Displaying similar cases*

The most common form of explanation in CBR systems amounts to displaying the most similar case. This technique is used by many research systems (e.g. CARES [Ong et al., 1997]) and in commercial CBR tools such as Orenge (developed by Empolis). In the previous Section, we discussed some limitations of this approach and recently some researchers have attempted to address some of these.

Doyle et al. [2004] point out that the most similar case is not necessarily the most convincing case. When trying to convince his parents to let him see the latest 'Harry Potter' movie, a child knows that friends that are younger than him are more convincing examples than his best friend even if he is the closest match in terms of age. Doyle et al. suggest a method for selecting cases of the same solution class as the problem case that is closer to a class boundary than the problem case for explanation purposes. This has the effect of increasing the awareness of class borders in the user. However, it may also provide evidence that is atypical. Any parent knows that a child will choose his examples very carefully, avoiding those children that were not allowed to see the movie.

Recently, research on ensemble classifiers has shown that the aggregated output of a set of classifiers can be more accurate than a single classifier. Such an approach may make it harder to find a proper case to display as an explanation to the user. Zenobi and Cunningham [2002] have addressed this by introducing a meta-layer over the set of case-based classifiers that perform the aggregation step. Since this technique is also case-based, it also produces neighbor cases that can be used in explanation.

We have argued that when emphasis is placed on different knowledge sources than the cases, the nearest case may serve neither the justification nor the transparency goal. One way of dealing with this problem is to introduce explanations on multiple layers in the CBR process. The case may serve as a type of top-level explanation, with more detailed levels of explanations for each case feature. The feature weighting may be explained in probability terms and there may also be ways of illustrating the coverage of cases. In the CREEK system [Aamodt, 2004], the user may ask for explanations at the attribute level, and the generation of this explanation depends on the similarity measure. A simple example is that when the similarity of attributes on an interval scale is explained, the range of all values for this attribute is shown to the user so he can more easily see how similar they are in the context of known cases.

This method may even be used to provide explanations for non-CBR systems, as demonstrated by Nugent and Cunningham [2005]. They use this technique to justify solutions produced by black-box systems such as neural networks and support vector machines. This is done by extracting local feature weights for a given solution from the black-box system, and using these, the most similar case from the training data is retrieved and displayed to the user as a justification.

E.5.2 Visualization

Visualization can make it easier for a user to see whether a solution is correct. In one example, McArdle and Wilson [2003] suggest a technique where the similarity of a set of cases is projected on a two-dimensional surface in such a way that the distance between them roughly corresponds to the similarity. While this is a simplification of the similarity measure, it allows the user to get an overview of the case space.

Good visualization techniques may at the same time increase the understanding of the reasoning process and reduce the cognitive load for the user. As such, visualization techniques may at the same time serve the justification and transparency goals. One example of this is the way the FormuCaseViz system [Massie et al., 2004] visualizes how a number of cases differ on a number of attributes and how this leads to predictions. This is done by drawing a two-dimensional graph, where each attribute is represented by a vertical line and the values of the attributes are placed at intervals along that line. A case is then represented as a line along the horizontal axis that intersects the attribute lines at the points representing the value this case has for that attribute. This technique allows at-a-glance comparisons and makes it very easy for people to spot eventual attributes where the values of a problem case do not match those of the cases it is being compared to.

E.5.3 Explanation Models

Knowledge-intensive systems may contain more generalized knowledge that can be of use to a human user in structuring his own internal model of the domain. This should allow knowledge-intensive systems to produce explanations that help in tying general domain knowledge and cases together. Examples of this are the IBP system [Brüninghaus and Ashley, 2003] and the CATO system [Aleven and Ashley, 1997] where model-based reasoning is combined with CBR to predict the outcome of legal cases. This is done by using both older cases and a weak domain model to produce legal arguments. In these systems the explanation is the solution, and the explanation (or argument) must be complete (fulfilling the transparency goal) in order to give justification to the prediction. This can make the argument complex, but as it uses the same problem-solving method as courts do in solving these cases, the target users (lawyers) are able to make sense of them.

It is possible to use models that are built explicitly for explanation, e.g. models that are not used in the reasoning process and used only to generate explanations. In CREEK [Aamodt, 1991], the model-based reasoner can use a causal model to produce explanations of why observations in a case can cause or imply the solution suggested by the system. These explanations are produced purely through backward chaining of causal relations from a solution already given by the CBR component to find how it may be connected to the observed features. As such, the explanations produced tend to fulfill the justification goal. The downside is that these explanations are produced after the fact and are not an accurate representation of how the system found the solution. It also requires a knowledge acquisition effort in building the causal model, but this model can then be tailored to the typical user's level of expertise.

The Colibri environment [Díaz-Agudo and González-Calero, 2000; Bello-Thomás et al., 2004] assists the development of systems that utilize a task/method ontology to make an explicit model of the system structure. The user can see how the CBR reasoning tasks and problem solving methods are linked to the model of general domain knowledge. Hence a transparency of the reasoning process is achieved.

Bergmann et al. [1994] make use of general domain knowledge for explaining similarity. The mechanism is based on an abstraction method, involving the modeling of domain knowledge at several levels of abstractions. The explanation produced justifies the correctness of the solution, rather than reproducing its trace, and is used both for retrieval and adaptation purposes.

E.5.4 Reasoning Trace

The reasoning trace method is feasible in systems that produce explanations as part of the reasoning process. The LID (Lazy Induction of Descriptions) system [Plaza et al., 2005] is an example of this. LID will attempt to find the categories that are maximally general while still as accurately as possible predicting the solution class of member cases. The induction process is similar to techniques used to induce decision trees, but is lazily applied at problem solving time. This process leaves a hierarchy of general-to-specific categories that may serve as an explanation as to the membership category of the problem case.

The relevance goal can also be fulfilled by offering explanations to the user that increase the understanding of the reasoning process. The Top Case mixed-initiative recommender system pursues a strategy where it selects questions that potentially strengthen the match for its currently selected best hypothesis case [McSherry, 2005]. This strategy is explained to the user by showing how an answer to this question could affect the recommendation. Top Case can for instance ask what region the user would like to take a holiday in. If the user would like to know why this is relevant in recommending a trip, the system can offer an explanation like "Because if the region = Tyrol this will increase the similarity of Case 510 from 0.28 to 0.44 and eliminate 866 cases, including Case 574". Because Top Case always displays the best matching cases found so far, the user can relate to these case labels and see how his answer affects the recommendation process.

E.5.5 Case Space Awareness

In case-based reasoning it is important that the transparency goal is not only applied to the reasoning process but also to the case base itself as much of the competence of the system lies in its collection of cases. The visualization techniques discussed above can help to achieve this, as can displaying similar cases, both opposing and supporting the conclusion.

The Stamping Advisor [Leake et al., 2001a] is a system to support feasibility analysis for the production of sheet metal parts in the automotive industry. For the feasibility analysis, it is important to understand the potential problems of a new design. The Stamping Advisor therefore displays two so called "bracketing cases", one where an identified problem exists and the most similar one without the problem. The user can so more easily identify the limits of the design.

Reilly et al. [2005] suggest that their system's compound critiques can play a similar role in recommender systems. The compound critiques generated by their system identify sets of attribute values that are correlated so that the user can see what kind of trade-offs he must make when deciding on a product. An example is that "higher price" and "bigger screen" may correlate when browsing for a TV. While this may not be an example of explanation in the usual sense, it illustrates that quite a wide range of techniques may have explanatory properties as long as they impart knowledge that increase the user's awareness of the problem domain.

E.5.6 Contrasting Evidence

The goal of transparency demands that the system does not try to hide conflicting evidence to its recommendation. In CBR systems this can be achieved by displaying the most similar case(s) that are not of the proposed solution class to the user. The Stamping Advisor [Leake et al., 2001a] mentioned in Section E.5.5 displays cases which are close to each other but with different findings. Compound confidence measures can also be calculated [as for instance in Cheetham and Price, 2004].

McSherry's ProCon system [McSherry, 2004] will identify which attributes of the input case support the suggested solution and which attributes oppose it. The attributes are identified as opposers or supporters of a solution based on how the attributes affect the probability of the solution. This allows the system to present justifications that are not only simpler to understand than possibly complex case similarity measures, but it also helps the user to identify what attributes are important to the conclusion.

The AHEAD system [Murdock et al., 2003] is an interpretative CBR system [Kolodner and Leake, 1996] designed to detect potential asymmetric threat situations (such as a terrorist attack). A situational interpretation is formed by constructing a trace of events, attempting to match it to prototypical threat situations. When matching this trace, AHEAD attempts to justify its conclusion by forming an argument that lists factors for and against the hypothesis based on what matches and does not match the expected findings in the prototypical threat situation. This allows the user to see evidence both for and against the conclusion. The difference between how AHEAD and ProCon identify contrastive evidence is that AHEAD is a knowledge-rich system where expectations about threat situations are modeled in advance by an expert, while ProCon uses machine learning techniques to generalize such knowledge from the case base.

E.5.7 Simplified Problem Solving Strategy

The conversational CBR community has developed methods that are particular to the relevance explanation goal. One such method is used by the Strategist system [McSherry, 1998], a mixed-initiative conversational diagnosis system where the user may enter a dialog where he is asked a single question at a time. The original Strategist induced a decision tree from a set of instances with the explicit goal that for each question the user is asked, the system would be able to give a good explanation for why this question was important to answer. The extension of Strategist into a CBR system [McSherry, 2001] does not form a decision tree in advance, but the question selection method is the same. As an example, the system prefers questions that could confirm or eliminate possible outcome classes in the domain. This allows it to form simple explanations of the relevance of questions the user is asked. In the computer fault domain, for example, the relevance of the question "Can you hear the fan?" might be explained, in the context of other reported evidence, by telling the user "Because if the fan cannot be heard this will confirm faulty power cord as a possible cause" [McSherry, 2001, Figure 7].

E.5.8 Concept Maps

Semantic network representation of knowledge such as in the CREEK system [Aamodt, 1991; Sørmo and Aamodt, 2002] may provide some explanatory support showing the part of the network around the concept the user is interested in. In particular this method may further the conceptualization goal by showing how the system views concepts in relation to other concepts and thus helps the user understand the system's conceptualization. Methods for sharing conceptualizations through two-dimensional visual-based representations are often referred to as topic or concept maps. There has been some work using these in CBR [e.g. Leake et al., 2001b], although the focus on this work has so far not been on its use for explanation.

E.5.9 Machine Learning Induction

The learning goal seems to have a strong preference for knowledge-intensive methods, but generalization may also be done lazily by a number of machine-learning algorithms. The CBR Strategist [McSherry, 2001] and ProCon [McSherry, 2003, 2004] systems mentioned earlier are examples of this as they do induction when presenting an explanation to the user, but they do so lazily. In the example in Section E.5.7, CBR Strategist observed that all surviving cases with "fan cannot be heard" have the same solution ("faulty power cord") and can inform the user that this feature is enough to confirm the solution. The CBR Strategist system may be fairly effective in training users in the skill of identifying computer faults. A limitation of this approach is that the system cannot introduce higher-order concepts or relate to how generalized concepts are used in the environment outside the system.

E.6 Challenges

Recently, there has been a renewed focus on explanation in case-based reasoning. There are, however, still many challenges that remain to be addressed. In this section we identify four such challenges for the future of explanation research in CBR.

E.6.1 Maintaining Transparency in Complex Systems

Displaying the closest case is quite near the actual reasoning process in simple case-based reasoning systems, but when more advanced methods like feature weighting and complex similarity measures are introduced, it will be necessary to provide additional information in order to fulfill the transparency goal. For example, in a k-nearest neighbor system, the transparency goal is no longer fulfilled by only displaying the best case if $k > 1$. The difficulty for the user in comparing cases increases as the case structure becomes more complex and the similarity measures more convoluted. It also increases with the use of more complex adaptation techniques where the retrieved case may not be the most similar but one which fits the adaptation process [e.g. Smyth and Keane, 1998].

In general, it can be argued that the use of other AI technologies in the CBR cycle (as suggested e.g. by Ian Watson [1999]) increases the difficulty for the user to see the explanative character of the case since it is necessary to have an at least intuitive understanding of the different techniques used in order to understand why the case presented offers a solution to the problem. If we cannot expect such an understanding, the steps taken by the different components also have to be explained. For example, consider a system where the case contain image data, and the similarity of two images is assessed by a neural network. Then the similarity measured through the neural network will have to be explained alongside the presented case – at least if complete transparency is the goal.

One way of dealing with this problem, as suggested in Section E.5.1, is to introduce explanations on multiple layers in the CBR process. The case may serve as a type of top-level explanation, with more detailed levels of explanations for each case feature. One problem with this approach is that although it satisfies the transparency goal, the cognitive load of the user increases as similarity measures increase in complexity. This has the interesting effect that as case-based systems grow more complex and are more able to help with exceedingly hard problems, the value of the case as an explanation may go down.

E.6.2 Providing Justification to Novice Users

As we have mentioned before there is an implicit assumption in presenting the case to the user that he is able to do a similarity comparison himself. Just as an explanation may not be required when the solution offered by a system matches the beliefs of the user, an explanation may not be necessary when the similarity between two cases is obvious. No new knowledge is required from the system in these cases in order for the conclusion to be accepted.

In complex domains with complex similarity measures, the similarity may not be so clear, especially to novice users. This has been seen in other kinds of knowledge-based systems, where explanation methods based on showing in detail how the problem-solver found the answer was deemed too complex to be useful by actual users [Majchrzak and Gasser, 1991]. For the novice users, a multi-level reasoning trace places a high cognitive load on the user and may be too complex or too time consuming to understand. In Section E.5, we have reviewed methods for simplifying this explanation as a means to achieve the justification goal, but many of these come at a cost to the fidelity of the explanation. While this may be acceptable in some domains, it is usually a goal to find simplification methods that preserve as much of the fidelity as possible. If a system uses justification explanations to overstate its confidence in the conclusion, it is likely that the user's confidence in the system will decrease over time.

However, research in the cognitive science and expert systems communities suggest that the goal of the user is not necessarily to gain an understanding of how the system solved the problem. When presented with a similar case, it may not be obvious to the user why the solution of the retrieved case was good even for the retrieved case itself. For these situations, providing justification explanations that do not stem from the reasoning process is not misleading the user but is addressing a different explanation goal.

E.6.3 Connecting Cases to General Knowledge in Tutoring

Cognitive theories of learning [e.g. Schank, 1983] assume that people start learning in a new domain by looking at concrete cases, or episodes. At some point, however, humans start to generalize the concrete episodes. This is in contrast to those approaches to CBR that rely on pure just-in-time induction. These lazy learners are well equipped to provide the student with example cases, and although this can be useful, they are ill equipped for assisting the learner in generalizing from these examples.

Today, most systems that attempt to tutor rely on generalized knowledge in addition to cases. Janet Kolodner's [1997] more recent work takes this approach, as does our own [Sørmo and Aamodt, 2002; Sørmo, 2005]. As mentioned in the previous section, there are knowledge-light techniques that do produce generalizations that may be useful for learning in humans (e.g. CBR Strategist [McSherry, 2001] and ProCon [McSherry, 2004]), but these techniques are currently not applied to tutoring.

E.6.4 Scope of Explanation Efforts

In Section E.4, we noted that the case-as-explanation method uses only one of the four knowledge containers Richter identified in CBR [Richter, 1995] – the case base. Competence arising from the three other containers (similarity measure, adaptation knowledge and vocabulary) is not used for explanation. We have surveyed several innovative methods (e.g. FormuCaseViz [Massie et al., 2004] and ProCon [McSherry, 2004]) that explain and visualize the similarity measure, but after CASEY [Koton, 1988], we have seen few efforts at explaining adaptation or vocabulary.

In our own research, we are working on combining different views on explanation. The goal is to integrate them into the CBR system design process in order to be able to make better use of the explanatory potential of the different knowledge containers [Roth-Berghofer et al., 2005; Roth-Berghofer and Cassens, 2005].

A parallel to the above is seen if we look at explanation efforts in the light of the CBR Cycle [Aamodt and Plaza, 1994]. Explanation efforts seem to focus on the *retrieve* step with little effort used to explain the other three steps (*revise*, *reuse* and *retain*). This is perhaps a natural consequence of the greater focus *retrieve* receives in problem-solving, but a CBR system that does not, for instance, retain all cases should be able to explain why a case is dropped or merged into another.

E.7 Conclusions

We have surveyed theories of explanation from the philosophy of science, linguistic and cognitive science communities, and also attempted to draw on the experiences with explanations from the expert-systems community in AI. From these theories and experiences, we believe it is useful to analyze the explanation requirements in the form of explanation goals. The goals that an explanation is required to achieve vary with the domain, system, and user. It can be hard to model these dynamically for the system itself, but the designer of the system can often make assumptions about the goals and capabilities of prototypical users of the system. We also believe that explicitly formulating such explanation goals facilitates the discussion of possible conflicts between goals and makes clear how different approaches tend to favor different types of goals. Although the goals discussed in this paper are abstract goals made to fit a wide range of CBR systems, they are not an attempt at completeness. There will be some CBR systems that fall outside the situational context we have defined for our explanation goals, and individual systems will also benefit from formulating more specific explanation goals that are tailored to their context.

In knowledge-intensive systems there has been continuous work on explanation, but recently this topic has received wider interest as exemplified by many of the methods we survey in Section E.5 of this paper. However, these have mainly been focused on the *retrieve* step in the CBR cycle. Although we are starting to see explanation methods that address competence arising from the similarity-measure knowledge container in addition to the case base, methods explaining vocabulary and adaptation are still rare.

Bibliography

Agnar Aamodt. Knowledge-intensive case-based reasoning in CREEK. In Funk and Calero [2004], pages 1–15.

Agnar Aamodt. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. PhD thesis, Norwegian Institute of Technology, Department of Computer Science, Trondheim, May 1991. URL `http://www.idi.ntnu.no/grupper/su/publ/phd/aamodt-thesis.pdf`.

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994.

Peter Achinstein. *The Nature of Explanation*. Oxford University Press, Oxford, 1983.

Vincent Aleven and Kevin D. Ashley. Teaching case-based argumentation through a model and examples: Empirical evaluation of an intelligent learning environment. In B. du Boulay and R. Mizoguchi, editors, *Artificial Intelligence in Education, Proceedings of AI-ED 1997 World Conference*, pages 87–94, Amsterdam, 1997. IOS Press.

Vincent Aleven and Kenneth R. Koedinger. An effective metacognitive strategy: Learning by doing and explaining with a computer based cognitive tutor. *Cognitive Science*, 26:147–179, 2002.

Kevin D. Ashley and Derek Bridge, editors. *Case-Based Reasoning Research and Development: Proceedings ICCBR 2003*, number 2689 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2003. Springer.

Juan José Bello-Thomás, Pedro González-Calero, and Belén Díaz-Agudo. JColibri: an object-oriented framework for building CBR systems. In Funk and Calero [2004], pages 32–46.

Ralph Bergmann, Gerhard Pews, and Wolfgang Wilke. Explanation-based similarity: A unifying approach for integrating domain knowledge into case-based reasoning for diagnosis and planning tasks. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-Based Reasoning: Proceedings of the 1st European Workshop EWCBR 1993 –*

Selected Papers, volume 837 of *Lecture Notes in Computer Science*, pages 182–196, Berlin Heidelberg, 1994. Springer. ISBN 3-540-58330-0.

William F. Brewer, Clark A. Chinn, and Ala Samarapungavan. Explanations in scientists and children. *Minds and Machines*, 8:119–136, 1998.

Sylvain Bromberger. An approach to explanation. In Ronald J. Butler, editor, *Analytical Philosophy*, volume 2, pages 72–105. Basil Blackwell, Oxford, 1965.

Sylvain Bromberger. Why questions. In Robert G. Colodny, editor, *Mind and Cosmos*, pages 86–111. Pittsburgh University Press, Pittsburgh, 1966.

Stefanie Brüninghaus and Kevin D. Ashley. Combining case-based and model-based reasoning for predicting the outcome of legal cases. In Ashley and Bridge [2003], pages 65–79.

Balakrishnan Chandrasekaran, Michael C. Tanner Tanner, and John R. Josephson. Explaining control strategies in problem solving. *IEEE Expert*, 4(1):9–15, 1989.

William Cheetham and Joseph Price. Measures of solution accuracy in case-based reasoning. In Funk and Calero [2004], pages 106–118.

William J. Clancey. The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence*, 20(3):215–251, 1983.

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An evaluation of the usefulness of case-based reasoning explanation. In Ashley and Bridge [2003], pages 122–130.

Belén Díaz-Agudo and Pedro González-Calero. An architecture for knowledge-intensive CBR systems. In *Advances in Case-Based Reasoning: Proceedings EWCBR 2000*, number 1898 in Lecture Notes in Artificial Intelligence, Berlin, 2000. Springer.

Dónal Doyle, Pádraig Cunningham, Derek Bridge, and Yusof Rahman. Explanation oriented retrieval. In Funk and Calero [2004], pages 157–168.

Boi Faltings. Probabilistic indexing for case-based prediction. In David B. Leake and Enric Plaza, editors, *Case-Based Reasoning Research and Development: Proceedings ICCBR 1997*, volume 1266 of *Lecture Notes in Artificial Intelligence*, pages 611–622, Berlin Heidelberg, 1997. Springer.

Kenneth M. Ford, Alberto J. Cañas, and John Coffey. Participatory explanation. In *Proceedings FLAIRS 1993*, pages 111–115, 1993. URL http://www.ihmc.us/users/acanas/Publications/ParticipatoryExplanation/ParticipatoryExplanation.htm.

Michael Friedman. Explanation and scientific understanding. *Journal of Philosophy*, 71:5–19, 1974.

Peter Funk and Pedro A. González Calero, editors. *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2004. Springer.

Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, 23 (4):497–530, 1999.

Joseph Hanna. Probabilistic explanation and probabilistic causality. In *Phil. Soc. Am.*, volume 2. 1982.

Gilbert Harman. The inference to the best explanation. *The Philosophical Review*, 74(1):88–95, January 1965.

Carl G. Hempel. *Aspects of Scientific Explanation*. Free Press, New York, 1965.

Carl G. Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, (15):135–175, 1948.

Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work*, pages 241–250, 2000. URL citeseer.ist.psu.edu/herlocker00explaining.html.

John R. Josephson and Susan G. Josephson, editors. *Abductive Inference Computation: Philosophy, Technology*. Cambridge University Press, New York, 1994.

Frank C. Keil and Robert A. Wilson, editors. *Explaining Explanation*. Bradford Books, Boston, MA, 2000.

Philip Kitcher. Explanation, conjunction, and unification. *Journal of Philosophy*, 73:207–12, 1976.

Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, 1993.

Janet L. Kolodner. Educational implications of analogy: A view from case-based reasoning. *American Psychologist*, 52(1):57–66, 1997.

Janet L. Kolodner and David B. Leake. A tutorial introduction to case-based reasoning. In David B. Leake, editor, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*. MIT Press & AAAI Press, Cambridge, MA, 1996.

Phyllis Koton. Reasoning about evidence in causal explanations. In *Proceedings of the AAAI-88*, volume 1, pages 256–261, Cambridge, MA, 1988. AAAI Press/MIT Press.

Mansur Lalljee, Margaret Watson, and Peter White. Attribution theory: Social and functional extensions. In *The Organization of Explanations*. Blackwell, Oxford, 1983.

David B. Leake. *Evaluating Explanations: A Content Theory*. Lawrence Erlbaum Associates, New York, 1992.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995a.

David B. Leake. Abduction, experience, and goals: A model of everyday abductive explanation. *Journal of Experimental and Theoretical Artificial Intelligence*, (7):407–428, 1995b.

David B. Leake and David McSherry. Special issue on explanation in case-based reasoning. *Artificial Intelligence Review*, 24(2), October 2005. ISSN 0269-2821.

David B. Leake, Larry Birnbaum, Kristian Hammond, Cameron Marlow, and Hao Yang. An integrated interface for proactive, experience-based design support. In *Proceedings of the sixth International Conference on Intelligent User Interfaces*, pages 101–108, Santa Fe, 2001a.

David B. Leake, Ana Maguitman, and Alberto Cañas. Assessing conceptual similarity to support concept mapping. In *Proceedings of the 15th International Florida Artificial Intelligence Research Society Conference*, pages 172–186, Menlo Park, 2001b. AAAI Press.

Doug Lenat and Edward Feigenbaum. On the thresholds of knowledge. In *Proceedings IJCAI 1987*, pages 1173–1182, 1987.

Ann Majchrzak and Les Gasser. On using artificial intelligence to integrate the design of organizational and process change in us manufacturing. *AI and Society*, 5:321–338, 1991.

Ji-Ye Mao and Izak Benbasat. The use of explanations in knowledge-based systems: Cognitive perspectives and a process-tracing analysis. *Journal of Managment Information Systems*, 17(2):153–179, 2000.

Stewart Massie, Susan Craw, and Nirmalie Wiratunga. Visualisation of case-based reasoning for explanation. In Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, Technical Report 142-04, pages 135–144, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Gavin P. McArdle and David C. Wilson. Visualising case-base usage. In Lorraine McGinty, editor, *Workshop Proceedings ICCBR 2003*, number 4/2004 in Technical Report, pages 105–114, Trondheim, 2003. Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science (IDI).

Drew McDermott. A critique of pure reason. *Journal of Computational Intelligence*, 3(3):151–160, August 1987.

David McSherry. Interactive case-based reasoning in sequential diagnosis. *Applied Intelligence*, 14:65–76, 2001.

David McSherry. Explanation in case-based reasoning: an evidential approach. In Brian Lees, editor, *Proceedings of the eighth UK Workshop on Case-Based Reasoning*, pages 47–55, Cambridge, 2003.

David McSherry. Explaining the pros and cons of conclusions in CBR. In Funk and Calero [2004], pages 317–330.

David McSherry. Explanation in recommender systems. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 179–197. ISSN 0269-2821.

David McSherry. Strategic induction of decision trees. In Rob Milne and Max A. Bramer, editors, *Proceedings of ES98*, pages 15–26, 1998.

Héctor Muñoz-Avila and Francesco Ricci, editors. *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, Chicago, 2005. Springer.

J. William Murdock, David W. Aha, and Len A. Breslow. Assessing elaborated hypotheses: An interpretive case-based reasoning approach. In Ashley and Bridge [2003], pages 332–346.

Conor Nugent and Pádraig Cunningham. A case-based explanation system for black-box systems. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 163–178. ISSN 0269-2821.

Lean Suan Ong, Barry Shepherd, Loong Cheong Tong, Francis Seow-Choen, Yik Hong Ho, Choong Leong Tang, Yin Seong Ho, and Kelvin Tan. The colorectal cancer recurrence support (CARES) system. *Artificial Intelligence in Medicine*, 11(3):175–188, 1997.

Enric Plaza, Eva Armengol, and Santiago Ontañón. Explanatory power of symbolic similarity in case-based reasoning. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 145–161. ISSN 0269-2821.

James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Explaining compound critiques. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 199–220. ISSN 0269-2821.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Funk and Calero [2004], pages 389–403.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Muñoz-Avila and Ricci [2005], pages 451–464. URL `http://dx.doi.org/10.1007/11536406_35`.

Thomas R. Roth-Berghofer, Jörg Cassens, and Frode Sørmo. Goals and kinds of explanations in case-based reasoning. In Klaus Dieter Althoff, Andreas Dengel, Ralph Bergmann, Markus Nick, and Thomas Roth-Berghofer, editors, *WM 2005: Professional Knowledge Management – Experiences and Visions*, pages 264–268, Kaiserslautern, 2005. DFKI.

Wesley Salmon. Statistical explanation. In Robert G. Colodny, editor, *The Nature and Function of Scientific Theories*, pages 173–231. Pittburgh University Press, Pittburgh, 1971.

Wesley Salmon. *Scientific Explanation and the Causal Structure of the World*. Princeton University Press, Princeton, 1984.

Roger C. Schank. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, Cambridge, 1983.

Roger C. Schank. *Explanation Patterns – Understanding Mechanically and Creatively*. Lawrence Erlbaum, New York, 1986.

Roger C. Schank and David B. Leake. Creativity and learning in a case-based explainer. *Artificial Intelligence*, 40(1–3):353–385, 1989.

Barry Smyth and Mark T. Keane. Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence*, 102(2): 249–293, 1998.

Frode Sørmo. Case-based student modeling using concept maps. In Muñoz-Avila and Ricci [2005], pages 492–506.

Frode Sørmo and Agnar Aamodt. Knowledge communication and CBR. In Pedro González-Calero, editor, *Proceedings of the ECCBR-02 Workshop on Case-Based Reasoning for Education and Training*, pages 47–59, Aberdeen, 2002.

William R. Swartout. What kind of expert should a system be? xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

William R. Swartout and Johanna D. Moore. Explanation in second generation expert systems. In Jean-Marc David, Jean-Paul Krivine, and Reid Simmons, editors, *Second Generation Expert Systems*, pages 543–585. Springer, Berlin, 1993.

William R. Swartout and Stephen W. Smoliar. On making expert systems more like experts. *Expert Systems*, 4(3):196–207, 1987.

Paul Thagard. *Computational Philosophy of Science*. MIT Press/Bradford Books, Boston, 1988.

Paul Thagard. Explanatory coherence. *Behavioral and Brain Sciences*, 12(3): 435–467, 1989.

Paul Thagard and Keith Holyok. Why indexing is the wrong way to think about analog retrieval. In *Proceedings of the DARPA Workshop on Case-Based Reasoning*, pages 36–40, San Mateo, 1989. Morgan Kaufman.

Bas van Fraassen, editor. *The Scientific Image*. Clarendon Press, Oxford, 1980.

Ian Watson. Case-based reasoning is a methodology, not a technology. *Knowledge-Based Systems*, 12(5-6):303–308, 1999.

Michael R. Wick and William B. Thompson. Reconstructive expert system explanation. *Artificial Intelligence*, 54(1–2):33–70, 1992.

Gabriele Zenobi and Pádraig Cunningham. An approach to aggregating ensembles of lazy learners that supports explanation. In Susan Craw and Alun Preece, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2002*, number 2416 in Lecture Notes in Artificial Intelligence, pages 436–447, Aberdeen, 2002. Springer.

Using Activity Theory to Model Context Awareness

F

Authors:

Anders Kofod-Petersen and Jörg Cassens

Abstract:

One of the cornerstones of any intelligent entity is the ability to understand how occurrences in the surrounding world influence its own behaviour. Different states, or situations, in its environment should be taken into account when reasoning or acting. When dealing with different situations, context is the key element used to infer possible actions and information needs. The activities of the perceiving agent and other entities are arguably one of the most important features of a situation; this is equally true whether the agent is artificial or not.

This work proposes the use of Activity Theory to first model context and further on populate the model for assessing situations in a pervasive computing environment. Through the socio-technical perspective given by Activity Theory, the knowledge intensive context model, utilised in our ambient intelligent system, is designed.

Main Result:

The paper takes a knowledge level perspective on context modeling. A relation between cognitive sciences and context in intelligent systems is established. Different concepts from activity theory are mapped to different categories of context well established in context aware computing, and a psychologically plausible context model is proposed.

Published in:

Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers*, volume 3946 of *LNCS*, pages 1–17, Edinburgh, 2006. Springer.

Copyright:

My main contributions to the paper:

- Background on activity theory
- Overview on AT in context

The following aspects were jointly developed by the authors:

- Description of context in cognition
- The mapping between the meronomy and basic aspects of AT is the result of many sessions in front of a whiteboard

F.1 Introduction

The original vision of ubiquitous computing proposed by Weiser [1991] envisioned a world of simple electronic artefacts, which could assist users in their day to day activities. This vision has grown significantly. Today the world of ubiquitous computing, pervasive computing or ambient intelligence uses visions and scenarios that are far more complex. Many of the scenarios of today envision pro-active and intelligent environments, which are capable of making assumptions and selections on their own accord.

Several examples exist in the contemporary literature, such as the help Fred receives from the omnipresent system Aura in Satyanarayanan [2001, p. 3], and the *automagic* way that Maria gets help on her business trip in Ducatel et al. [2001, p. 4]. More examples and comments can be found in Lueg [2002a]. Common to many of these examples are the degree of autonomy, common sense reasoning, and situation understanding the systems involved exhibit.

To be truly pro-active and be able to display even a simple level of common sense reasoning, an entity must be able to appreciate the environment which it inhabits; or to understand the situations that occur around it. When humans interpret situations, the concept of context becomes important. Humans use an abundance of more or less subtle cues as context and thereby understand, or at least assess, situations. The ability to acquire context and thereby fashion an understanding of situations, is equally important for artefacts that wish to interact (intelligently) with the real world. Systems displaying this ability to acquire and react to context are known as *context-aware* systems.

A major shortfall of the research into context-aware systems is the lack of a common understanding of what a context model is, and perhaps more importantly, what it is not. This shortfall is very natural, since this lack of an agreed definition of context also plagues the real world. No common understanding of what context is and how it is used exists. So, it is hardly surprising that it is hard to agree on the artificial world that IT systems represent.

Most of the research today has been focused on the technical issues associated with context, and the syntactic relationships between different concepts. Not so much attention has been given to context from a knowledge level [Newell, 1982] perspective or an analysis of context on the level of socio-technical systems [Lueg, 2002b].

This is the main reason for the approach chosen here. It should be feasible to look at how we can use socio-technical theories to design context-aware systems to supply better services to the user, in a flexible and manageable way. The approach should facilitate modelling at the knowledge level as well and furthermore enable the integration of different knowledge sources and the presentation of knowledge content to the user.

It can be stated that one of the most important context parameters available in many situations is the *activity* performed by an entity present in the environment. We therefore believe that by focusing on activities we will gain a better understanding of context and context awareness; thus bringing us closer to realise truly ambient intelligent systems.

Several approaches to examine activity have been proposed, like e.g. Actor-Network Theory [Latour, 1988], Situated Action [Suchman, 1987] or the Locales Framework [Fitzpatrick, 1998]. One of the most intriguing theories, however, is Activity Theory based on the works of Vygotsky [1978, 1985]; Leont'ev [1978]. This work proposes the use of Activity Theory to model context and to describe situations.

Although our approach is general, in the sense that it is applicable to different domains, we are not trying to define a context model which will empower the system to be universally context aware, meaning it will be able to build its own context model on the fly. Although this would be a prerequisite for truly intelligent systems, IT-systems are usually designed for specific purposes and with specific tasks in mind where the system has to support human users. They are used by people with specific needs and qualifications, and should preferably adapt to changes in these needs over time [McSherry, 2002; Totterdell and Rautenbach, 1990]. The aim of the work presented in this article is to assist the design of such systems which are tailored to support such kind of human work.

This article is organised as follows: first some background work on the use of context in cognition is covered. Secondly, some important concepts of Activity Theory are introduced. This is followed by an explanation of how Activity Theory can be utilised to model contextual information, including an illustrative example. In Section F.5, the knowledge model, including context employed in this work, is described. Finally, some pointers for future work are presented.

F.2 Context in Cognition

The concept of context is closely related to reasoning and cognition in humans. Even though context might be important for reasoning in other animals, it is common knowledge that context is of huge importance in human reasoning.

Beside the more mechanistic view on reasoning advocated by neuroscience, psychology and philosophy play important roles in understanding human cognition. It might not be obvious how computer science is related to knowledge about human cognition. However, many sub-fields in computer science are influenced by our knowledge about humans; and other animals.

The field of Artificial Intelligence has the most obvious relations to the study of reasoning in the real world, most prominently psychology and philosophy. Since AI and psychology are very closely related and context is an important aspect of human reasoning, context also plays an important role in the understanding and implementation of Artificial Intelligence.

AI has historically been closely connected to formal logic. Formal logic is concerned with explicit representation of knowledge. This leads to the need to codify all facts that could be of importance. This strict view on objective truth is also known in certain directions within philosophy, where such a concept of knowledge as an objective truth exists. This can be traced back to e.g. the logic of Aristotle who believed that some subset of knowledge had that characteristic (Episteme). This view stands in stark contrast to the views advocated by people such as Polanyi, who argues that no such objective truth exists and all knowledge is at some point personal and hidden (tacit) [Polanyi, 1964].

Since context is an elusive type of knowledge, where it is hard to quantify what type of knowledge is useful in a certain situation, and possibly why, it is obvious that it does not fit very well with the strict logical view on how to model the world. Ekbia and Maguitman [2001] argue that this has led to the fact that context has largely been ignored by the AI community. This observation still holds some truth, despite some earlier work on context and AI, like Doug Lenat's discussion of context dimensions [Lenat, 1998], and the other work we discuss later in this section.

Ekbia and Maguitman's paper is not a recipe on how to incorporate contextual reasoning into logistic systems, but rather an attempt to point out the deficiencies and suggest possible directions AI could take to include context. Their work builds on the work by the American philosopher John Dewey. According to Ekbia and Maguitman, Dewey distinguishes between two main categories of context: spatial and temporal context, coherently know as background context; and selective interest. The spatial context covers all contemporary parameters. The temporal context consists of both intellectual and existential context. The intellectual context is what we would normally label as background knowledge, such as tradition, mental habits, and science. Existential context is combined with the selective interest related to the notion of situation. A situation is in this work viewed as a confused, obscure, and conflicting thing, where a human reasoner attempts to make sense of this through the use of context.

This view, by Dewey, on human context leads to the following suggestion by the pragmatic approach [Ekbia and Maguitman, 2001, p. 5]:

1. Context, most often, is not explicitly identifiable.
2. There are no sharp boundaries among contexts.
3. The logical aspects of thinking cannot be isolated from material considerations.
4. Behaviour and context are jointly recognisable.

Once these premises have been set, the authors show that the logical approach to (artificial) reasoning has not dealt with context in any consistent way. The underlying argument is that AI has been using an absolute separation between mind and nature, thus leading to the problems associated with the use of context. This view on the inseparability of mind and nature is also based on Dewey's work. This view is not unique for Dewey. In recent years this view has been proposed in robotics as *situatedness* by Brooks [1987, 1991a,b], and in ecological psychology by Gibson [1979].

Through the discussion of different logic-based AI methods and systems, the authors argue that AI has not yet parted company with the limitations of logic with regards to context. Furthermore, they stress the point of intelligence being action-oriented; based on the notion of situations described above.

The notion of intelligence being action-oriented, thus making context a tool for selecting the correct action, is shared by many people within the computer science milieu. Most notably the work by Strat [1993], where context is applied to select the most suitable algorithm for recognition in computer vision, and by Öztürk and Aamodt [1998] who utilised context to improve the quality and efficiency of Case-Based Reasoning.

Strat [1993] reports on the work done in computer vision to use contextual information in guiding the selection of algorithms in image understanding. When humans observe a scene they utilise a large amount of information (context) not captured in the particular image. At the same time, all image understanding algorithms use some assumptions in order to function, creating an epistemic bias. Examples are algorithms that only work on binary images, or that are not able to handle occlusions.

Strat defines three main categories of context: *physical*, being general information about the visual world independent of the conditions under which the image was taken; *photogrammetric*, which is the information related the acquisition of the image; and *computational*, being information about the internal state of the processing. The main idea in this work is to use context to guide the selection of the image-processing algorithms to use on particular images. This is very much in line with the ideas proposed by Ekbia and Maguitman, where intelligence is action-oriented, and context can be used to bring order to diffuse and unclear situations.

This action-orientated view on reasoning and use of context is also advocated by Öztürk and Aamodt [1998]. They argue that the essential aspects of context are the notion of *relevance* and *focus*. To facilitate improvements to Case-Based Reasoning a context model is constructed. This model builds on the work by Hewitt, where the notion of *intrinsic* and *extrinsic* context types are central. According to Hewitt, intrinsic context is information related to the target item in a reasoning process, and extrinsic is the information not directly related to the target item. This distinction is closely related to the concepts of *selective interest* and *background context* as described by Dewey. The authors refine this view by focusing on the intertwined relationship between the *agent* doing the reasoning, and the *characteristics* of the problem to be solved. This is exactly the approach recognised as being missing in AI by Ekbia and Maguitman.

Öztürk and Aamodt build a taxonomy of context categories based on this merger of the two different worlds of information (internal vs. external). Beside this categorisation, the authors impose the action, or task, oriented view on knowledge in general, and contextual knowledge in particular. The goal of an agent *focuses* the attention, and thereby the knowledge needed to execute tasks associated with the goal. The example domain in their paper is from medical diagnostics, where a physician attempts to diagnose a patient by the hypothesise-and-test strategy. The particular method of diagnostics in this Case-Based Reasoning system is related to the strategy used by Strat. They differ insofar that Strat used contextual information to select the algorithms to be used, whereas Öztürk and Aamodt have, prior to run-time, defined the main structure of a diagnostic situation, and only use context to guide the sub-tasks in this process.

Zibetti et al. [2001] focus on the problem of how agents understand situations based on the information they can perceive. To our knowledge, this work is the only one that does not attempt to build an explicit ontology on contextual information prior to run-time. The idea is to build a (subjective) taxonomy of ever-complex situations solely based on what a particular agent gathers from the environment in general, and the behaviour of other agents in particular.

The implementation used to exemplify this approach contains a number of agents "living" in a two-dimensional world, where they try to make sense of the world by assessing the spatial changes to the environment. Obviously the acquisition of knowledge starting with a *tabula rasa* is a long and tedious task for any entity. To speed up the process the authors predefined some categories with which the system is instantiated.

All in all, this approach lies in between a complete bottom-up and the top-down approaches described earlier.

F.3 Activity Theory

In this section, we concentrate on the use of Activity Theory (AT) to support the modelling of context. Our aim is to use AT to analyse the use of technical artefacts as instruments for achieving a predefined goal in the work process as well as the role of social components, like the division of labour and community rules. This helps us to understand what pieces of knowledge are involved and the social and technological context used when solving a given problem.

First in this section, we will give a short summary of aspects of AT that are important for this work. See Nardi [2003] for a short introduction to AT and Bødker [1991]; Nardi [1996a] for deeper coverage. The theoretical foundations of AT in general can be found in the works of Vygotsky [1978, 1985]; Leont'ev [1978].

Activity Theory is a descriptive tool to help understand the unity of consciousness and activity. Its focus lies on individual and collective work practise. One of its strengths is the ability to identify the role of material artefacts in the work process. An activity (Fig. F.1) is composed of a subject, an object, and a mediating artefact or tool. A subject is a person or a group engaged in an activity. An object is held by the subject, and the subject has a goal directed towards the object he wants to achieve, motivating the activity and giving it a specific direction.

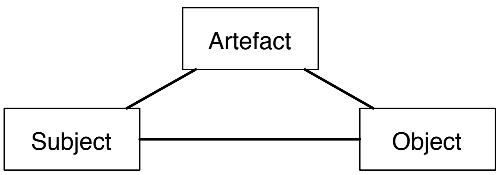

Figure F.1.: Activity Theory: The basic triangle of Mediation

Some basic properties of Activity Theory are:

- **Hierarchical structure of activity:** Activities (the topmost category) are composed of goal-directed actions. These actions are performed consciously. Actions, in turn, consist of non-conscious operations.

- **Object-orientedness:** Objective and socially or culturally defined properties. Our way of doing work is grounded in a praxis which is shared by our co-workers and determined by tradition. The way an artefact is used and the division of labour influences the design. Hence, artefacts pass on the specific praxis they are designed for.
- **Mediation:** Human activity is mediated by tools, language, etc. The artefacts as such are not the object of our activities, but appear already as socio-cultural entities.
- **Continuous Development:** Both the tools used and the activity itself are constantly reshaped. Tools reflects accumulated social knowledge, hence they transport social history back into the activity and to the user.
- **Distinction between internal and external activities:** Traditional cognitive psychology focuses on what is denoted internal activities in Activity Theory, but it is emphasized that these mental processes cannot be properly understood when separated from external activities, that is the interaction with the outside world.

Taking a closer look on the hierarchical structure of activity, we can find the following levels:

- **Activity:** An individual activity is for example to check into a hotel, or to travel to another city to participate at a conference. Individual activities can be part of collective activities, e.g. when someone organises a workshop with some co-workers.
- **Actions:** Activities consist of a collections of actions. An action is performed consciously, the hotel check-in, for example, consists of actions like presenting the reservation, confirmation of room types, and handover of keys.
- **Operations:** Actions consist themselves of collections of non-conscious operations. To stay with our hotel example, writing your name on a sheet of paper or taking the keys are operations. That operations happen non-consciously does not mean that they are not accessible.

It is important to note that this hierarchical composition is not fixed over time. If an action fails, the operations comprising the action can get conceptualised, they become conscious operations and might become actions in the next attempt to reach the overall goal. This is referred to as a

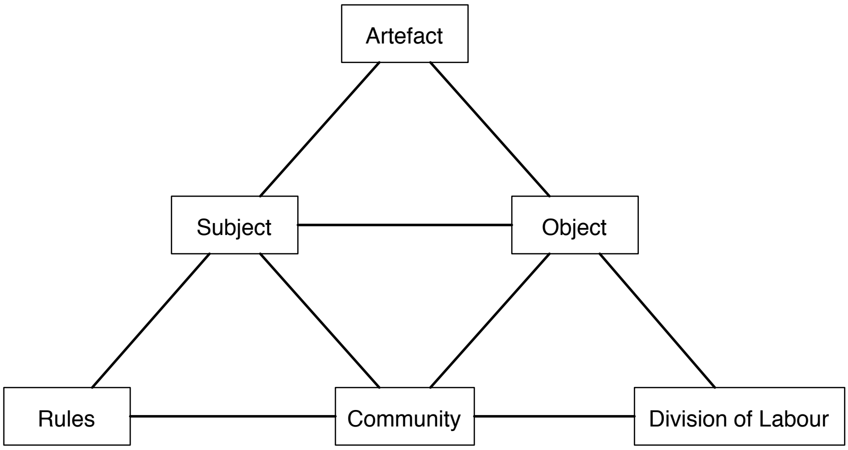

Figure F.2.: Cultural Historical Activity Theory: Expanded triangle, incorporating the community and other mediators.

breakdown situation. In the same manner, actions can become automated when done many times and thus become operations. In this way, we gain the ability to model a change over time.

An expanded model of Activity Theory, Cultural Historical Activity Theory (CHAT), covers the fact that human work is done in a social and cultural context [compare e.g. Kutti, 1996; Mwanza, 2000]. The expanded model (depicted in Fig. F.2) takes this aspect into account by adding a *community* component and other mediators, especially *rules* (an accumulation of knowledge about how to do something) and the *division of labour*.

In order to be able to model that several subjects can share the same object, we add the community to represent that a subject is embedded in a social context. Now we have relationships between subject and community and between object and community, respectively. These relationships are themselves mediated, with rules regarding to the subject and the division of labour regarding to the object.

This expanded model of AT is the starting point for our use of AT in the modelling of context for intelligent systems.

F.4 Activity Theory and Context Awareness

The next step is to identify which aspects of an Activity Theory based analysis can help us to capture a knowledge level view of contextual knowledge that should be incorporated into an intelligent system. This contextual knowledge should include knowledge about the acting subjects, the objects towards which activities are directed and the community as well as knowledge about the mediating components, like rules or tools.

F.4.1 Activity Theory for the Identification of Context Components

As an example, we want the contextual knowledge to contain both information about the acting subject itself (like the weight or size) and the tools (like a particular software used in a software development process). To this end, we propose a mapping from the basic structure of an activity into a taxonomy of contextual knowledge as depicted in Table F.1 (the taxonomy is described in more detail in Section F.5). We can see that the personal context contains information we would associate with the acting subject itself.

Table F.1.: Basic aspects of an activity and their relation to a taxonomy of contextual knowledge

CHAT aspect	**Category**
Subject	Personal Context
Object	Task Context
Community	Spatio-Temporal Context
Mediating Artefact	Environmental Context
Mediating Rules	Task Context
Mediating Division of Labour	Social Context

We would like to point out that we do not think that a strict one to one mapping exists or is desirable at all. Our view on contextual knowledge is contextualised itself in the sense that different interpretations exist, and what is to be considered contextual information in one setting is part of the general knowledge model in another one. Likewise, the same piece of knowledge can be part of different categories based on the task at hand.

The same holds for the AT based analysis itself: the same thing can be an object and a mediating artefact from different perspectives and in different task settings. The mapping suggested here should lead the development process and allow the designer to focus on knowledge-level aspects instead of being lost in the modelling of details without being able to see the relationship between different aspects on a socio-technical system level.

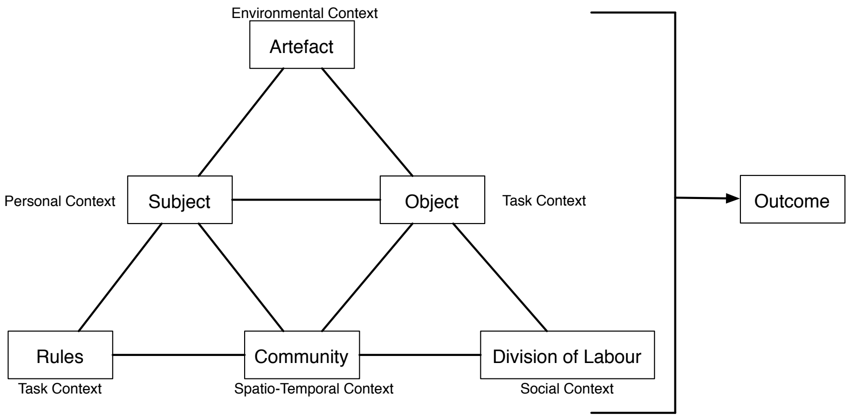

Figure F.3.: Mapping from Activity Theory to context model

As an example, let us consider a software development setting where a team is programming a piece of software for a client. The members of the team are all *subjects* in the development process. They form a *community* together with representatives of the client and other stake-holders. Each member of the team and personnel from other divisions of the software company work together in a *division of labour*. The *object* at hand is the unfinished prototype, which has to be transformed into something that can be handed over to the client. The task is governed by a set of *rules*, some explicit like coding standards some implicit like what is often referred to as a working culture. The programmers use a set of *mediating artefacts* (tools), like methods for analysis and design, programming tools, and documentation.

When we design a context-aware system for the support of this task, we include information about the user of the system (*subject*) in the *personal context* and about the other team members in the *environmental context*. Aspects regarding the special application a developer is working on (*objects*) are part of the *task context*, it will change when the same user engages in a different task (lets say he is looking for a restaurant). The *rules* are part of the *task context* since they are closely related to the task at hand – coding standards will not be helpful when trying to find a restaurant. We find the tool aspects (*artefacts*) in the *environmental context* since access to the different tools is important for the ability of the user to use them. Knowledge about his co-workers and other stake-holders (*community*) are modelled in the *spatio-temporal context*. Finally, his interaction with other team members (*division of labour*) is described as part of the *social context*.

In the design process, we can also make use of the hierarchical structure of activities. On the topmost level, we can identify the *activities* the context-aware system should support. By this, we can restrict the world view of the system and make the task of developing a context model manageable. Further on, we can make use of the notion of *actions* to identify the different situations the system can encounter. This helps us to asses the different knowledge sources and artefacts involved in different contexts, thereby guiding the knowledge acquisition task. Finally, since *operations* are performed subconsciously, we get hints on which processes should be supported by automatic and proactive behaviour of the system.

Let us consider our example again. We know that the *activity* we want to support is the development of an IT system. Therefore, we can restrict ourselves to facets of the world which are related to the design process, and we do not (necessarily) have to take care of supporting e.g. meetings some of the team members have as players at the company's football team. On the other hand, the system has to be concerned with meetings with the customer. Further on, different *actions* which are also part of the activity should be supported, like e.g. team meetings or programming sessions, and the different *actions* involved can lead to the definition of different situations or contexts.

A context-aware application should therefore at all times know in which *action* the user is engaged. This is, in fact, the main aspect of our understanding of the term *context awareness*. At last, to support the *operations* of the user, it might be necessary to proactively query different knowledge sources or request other resources the user might need without being explicitly told to do so by the user. This is at the core of what we refer to as *context sensitivity* in order to distinguish between these two different aspects of context.

It is important to keep in mind that the hierarchical structure of activities is in a constant state of flux. Activity Theory is also capable of capturing changing contexts in break-down situations. Lets consider that a tool used in the development process, such as a compiler, stops working. The operation of evoking the compiler now becomes a conscious action for the debugging process. The focus of the developer shifts away from the client software to the compiler. He will now be involved in a different task where he probably will have to work together with the system administrators of his work-station. In this sense other aspects of the activity, such as the community, change as well. It is clear that the contextual model should reflect these changes. The ability of Activity Theory to identify possible break-down situations makes it possible for the system designer to identify these possible shifts in situation and model the anticipated behaviour of the system.

F.4.2 Other Aspects of Activity Theory and Context

Other work on the use of AT in modelling context has been conducted e.g. by Kaenampornpan and O'Neill [2004]. This work is focusing on modelling features of the world according to an activity theoretic model. However, the authors do not carry out a knowledge level analysis of the activities. We argue that our knowledge intensive approach has the advantage of giving the system the ability to reason about context so that it does not have to rely on pattern matching only. This is helpful especially in situations where not all the necessary features are accessible by the system, for example because of limits of sensory input in mobile applications. On the other hand, Kaenampornpan and O'Neill further on develop a notion of history of context in order to elicit a users goals [Kaenampornpan and O'Neill, 2005]. This work deals with the interesting problem of representing the user's history in context models which we have not addressed explicitly in this article.

Li et al. [2004] propose an activity based design tool for context aware applications. The authors' focus lies not on the use of Activity Theory in the context model itself but on supporting the designer of context aware applications with a rapid-prototyping tool. An interesting idea is the proposed integration of temporal probabilistic models.

Wiberg and Olsson [1999] make also use of Activity Theory, but their focus lies on the design of context aware tangible artefacts. The usage situation is well defined upfront and no reasoning about the context has to be done.

When we look at the design of IT-systems in general and not only the issue of context-awareness, we find that Activity Theory has been applied to many different areas of system development. For example, AT was used in health care settings as a tool to support development of information systems [Korpela et al., 2001]. It has also been used in the design of augmented reality systems, as reported in Fjeld et al. [2002] and for a posteriori analysis of computer systems in use [Bødker, 1996]. A comparative survey of five different AT based methods for information systems development with pointers to additional examples was conducted by Quek and Shah [2004].

In our own work, we are also using Activity Theory to support modelling other, not context depending aspects of intelligent systems. For example are we focusing on breakdown situations in order to enhance the explanatory capabilities of knowledge-rich Case-Based Reasoning systems [Cassens, 2004].

F.5 Context model

The context model used in this work draws on a subjective view on situations. That is, even though the model is general, any instance of the model belongs to one user only. Thus, as in Zibetti et al. [2001], any situation will be described form the personal perspective, leading to the possibility of many instances describing the "same" situation. This is in contrast to the leading perspective, where a system will describe *objective* situations, and leans towards Polanyi's perspective of all knowledge being personal [Polanyi, 1964].

In the extreme consequence the model used by any subject could also be personal and unique. However, to avoid the problem of a *tabula rasa* we have chosen a pragmatical view on how to model context. The model is based on the definition of context given by Dey [2001], applying the following definition:

> Context is the set of suitable environmental states and settings concerning a user, which are relevant for a situation sensitive application in the process of adapting the services and information offered to the user.

This definition from Dey does not explicitly state that context is viewed as knowledge. However, we believe that the knowledge intensive approach is required if we wish a system to display many of the characteristics mentioned in the introduction. At the same time we also adhere to the view advocated by Brézillon and Pomerol [1999] that context is not a special kind of knowledge. They argue that context is in the eye of the beholder: "...knowledge that can be qualified as 'contextual' depends on the context!" [Brézillon and Pomerol, 1999, p.7]

Even though we argue for a context model where context is not a special type of information, we also believe that only a pragmatical view on context will enable us to construct actually working systems. Following this pragmatic view we impose a taxonomy on the context model in the design phase (see Fig. F.4). This taxonomy is inherited from the context-aware tradition and adopted to make use of the general concepts we find in Activity Theory.

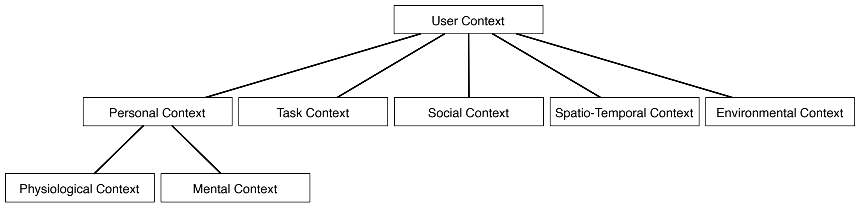

Figure F.4.: Context taxonomy

The context is divided into five sub-categories (a more thorough discussion can be found in Göker and Myrhaug [2002] or Kofod-Petersen and Mikalsen [2005b]):

1. **Environmental context:** This part captures the users surroundings, such as things, services, people, and information accessed by the user.

2. **Personal context:** This part describes the mental and physical information about the user, such as mood, expertise and disabilities.

3. **Social context:** This describes the social aspects of the user, such as information about the different roles a user can assume.

4. **Task context:** the task context describe what the user is doing, it can describe the user's goals, tasks and activities.

5. **Spatio-temporal context:** This type of context is concerned with attributes like: time, location and the community present.

The model depicted in Fig. F.4 shows the top-level ontology. To enable the reasoning in the system this top-level structure is integrated with a more general domain ontology, which describes concepts of the domain (*e.g.*, Operating Theatre, Ward, Nurse, Journal) as well as more generic concepts (Task, Goal, Action, Physical Object) in a multi-relational semantic network. The model enables the system to infer relationships between concepts by constructing context-dependent paths between them. We are approaching the situation assessment by applying knowledge-intensive Case-Based Reasoning [Aamodt, 2004]. One of the important aspects of knowledge-intensive Case-Based Reasoning is the ability to match two case features that are syntactically different, by explaining why they are similar [Aamodt, 1994; Jære et al., 2002].

The generic concepts are partly gathered through the use of activity theoretic analysis. These concepts include the six aspects shown in Fig. F.3. The top-level taxonomy including the concepts acquired from AT is depicted in Fig. F.5. The context model is now primed to model situations and the activities occurring within them.

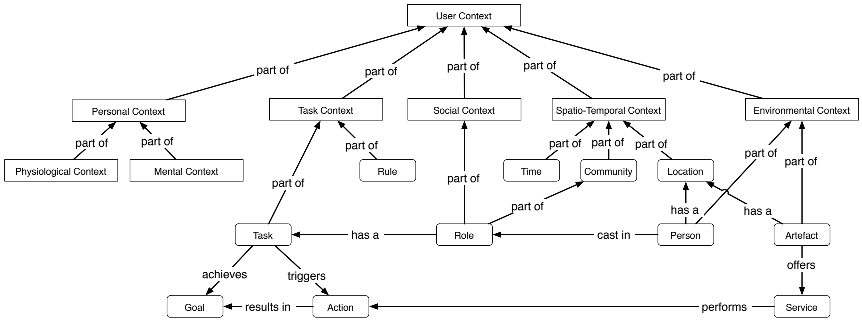

Figure F.5.: Populated context structure

If we look at the model we can see how each of the AT aspects are modelled. The *artefact* exists within the *environmental context*, where it can offer *services* that can perform *actions*, which assist the *subject* (described in the *personal context*) in achieving the *goals* of the *role* (in the *social context*) played by the subject. Other *persons*, being part of the situation through the environmental context, can also affect the *outcome* (goal) of the situation. They are cast in different roles that are part of the *community* existing in the *spatio-temporal context*. The roles also implicitly define the *division of labour* in the community. The *rules* governing the subject are found in the *task context*.

F.6 Ongoing and Future Work

We have outlined how the design of context-aware systems can benefit from an analysis of the underlying socio-technical system. We have introduced a knowledge-level perspective on the modelling task, which makes it possible to identify aspects of knowledge that should be modelled into the system in order to support the user with contextual information. We have furthermore proposed a first mapping from an Activity Theory based analysis to different knowledge components of a context model. The basic aspects of our socio-technical model fit nicely to the taxonomy of context categories we have introduced before, thus making AT a prime candidate for further research.

The use of Activity Theory allows for system designers to develop the general models of activities and situations. General models are necessary to support the initial usage of the system. They are an important prerequisite for the Case-Based Reasoning system to integrate new situations; thereby adapting to the personal and subjective perspective of the individual user.

In Section F.5 we have formulated the problem of identifying the tasks connected to a particular situation, the goals of the user, and the artefacts and information sources used. We argue that our Activity Theory based approach is capable of integrating these cognitive aspects into the modelling process.

The integration of an *a posteriori* method of analysis with design methodologies is always challenging. One advantage AT has is that it is process oriented, which corresponds to a view on systems design where the deployed system itself is not static and where the system is able to incorporate new knowledge over time [Aamodt, 1995]. Activity Theory has its blind spots, such as modelling the user interaction of the interface level. However, in this particular work we are not focusing on user interfaces; thus, these deficiencies do not affect this work directly. Still, one of our future goal is to combine AT with other theories into a framework of different methods supporting the systems design process [Cassens, 2005].

Nevertheless, one of the next steps is to formalise the relationship between different elements of an AT based analysis and the knowledge contained in the different contextual aspects of our model. This more formalised relationship is being put to the test on a context modelling task, using an AT based analysis of a socio-technical system to support the design of our context-aware intelligent system (see for an example Kofod-Petersen and Mikalsen [2005a] for a description of the system).

We have recently initiated a project where everyday situations in a health care setting are being observed and documented. These observations are being used to test the situation assessment capabilities of our system. We have used a modelling approach based on Cultural Historical Activity Theory. This allows us to identify the different actions the medical staff is involved with and the artefacts and information sources used.

We have already instantiated a context model for this scenario using the topology described earlier in this paper. We are currently in the process of populating the model based on our observations. At the same time, we are refining our knowledge engineering methodologies for translating the findings into a knowledge model.

Our system also includes an agency part, which is described in Lech and Wienhofen [2005]. Based on the context-aware situation assessment being carried out, this agency supplies context-sensitive problem solving [Gundersen and Kofod-Petersen, 2005]. We are in the process of extending the analysis of the situations to model the way our decompose agent decomposes and solves problems.

Acknowledgements

Part of this work was carried out in the AmbieSense project, which was supported by the EU commission (IST-2001-34244).

Bibliography

Agnar Aamodt. Knowledge-intensive case-based reasoning in CREEK. In Peter Funk and Pedro A. González Calero, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, pages 1–15, Berlin Heidelberg, 2004. Springer.

Agnar Aamodt. Explanation-driven case-based reasoning. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-Based Reasoning: Proceedings of the 1st European Workshop EWCBR 1993 – Selected Papers*, volume 837 of *Lecture Notes in Computer Science*, pages 274–288, Berlin Heidelberg, 1994. Springer. ISBN 3-540-58330-0.

Agnar Aamodt. Knowledge acquisition and learning by experience – the role of case-specific knowledge. In Gheorghe Tecuci and Yves Kodratoff, editors, *Machine Learning and Knowledge Acquisition – Integrated Approaches*, chapter 8, pages 197–245. Academic Press, 1995.

Susanne Bødker. Applying activity theory to video analysis: How to make sense of video data in human-computer interaction. In Nardi [1996b], pages 147–174.

Susanne Bødker. Activity theory as a challenge to systems design. In Hans-Erik Nissen, Heinz K. Klein, and Rudy Hirschheim, editors, *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pages 551–564. North Holland, 1991.

Patrick Brézillon and Jean-Charles Pomerol. Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62(3):223–246, 1999.

Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991a.

Rodney A. Brooks. Planning is just a way of avoiding figuring out what to do next. Technical report, MIT, 1987.

Rodney A. Brooks. New approaches to robotics. *Science*, 253:1227–1232, 1991b.

Jörg Cassens. Knowing what to explain and when. In Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, Technical Report 142-04, pages 97–104, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Jörg Cassens. User aspects of explanation aware cbr systems. In Maria Francesca Costabile and Fabio Paternó, editors, *Human-Computer Interaction – INTERACT 2005*, volume 3585 of *Lecture Notes in Computer Science*, pages 1087–1090, Rome, 2005. Springer. URL http://dx.doi.org/10.1007/11555261_111.

Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Hamid R. Ekbia and Ana G. Maguitman. Context and relevance: A pragmatic approach. *Lecture Notes in Computer Science*, 2116:156–169, 2001.

Geraldine Fitzpatrick. *The Locales Framework: Understanding and Designing for Cooperative Work.* Ph.d. thesis, The University of Queensland, Australia, 1998.

Morten Fjeld, Kristina Lauche, Martin Bichsel, Fred Voorhoorst, Helmut Krueger, and Mathias Rauterberg. Physcial and virtual tools: Activity theory applied to the design of groupware. *CSCW*, 11(1-2):153–180, 2002.

James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

Ayse Göker and Hans Inge Myrhaug. User context and personalisation. In *Workshop proceedings for the sixth European Conference on Case Based Reasoning*, 2002.

Odd Erik Gundersen and Anders Kofod-Petersen. Multiagent based problem-solving in a mobile environment. In Eivind Coward, editor, *Norsk Informatikkonferance 2005, NIK 2005*, pages 7–18. Institutt for Informatikk, Universitetet i Bergen, November 2005. ISBN 82-519-2081-7.

Martha Dørum Jære, Agnar Aamodt, and Pål Skalle. Representing temporal knowledge for case-based prediction. In Susan Craw and Alun Preece, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2002*, number 2416 in Lecture Notes in Artificial Intelligence, pages 174–188, Aberdeen, 2002. Springer.

Manasawee Kaenampornpan and Eamonn O'Neill. Modelling context: an activity theory approach. In Panos Markopoulos, Berry Eggen, Emilie Aarts, and James L. Croeley, editors, *Ambient Intelligence: Second European Symposium on Ambient Intelligence, EUSAI 2004*, volume 3295 of *Lecture Notes in Computer Science*, pages 367–374. Springer, November 2004.

Manasawee Kaenampornpan and Eamonn O'Neill. Integrating history and activity theory in context aware system design. In *Proceedings of the 1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE)*, May 2005.

Anders Kofod-Petersen and Marius Mikalsen. An architecture supporting implementation of context-aware services. In Patrik Floréen, Greger Lindén, Tiina Niklander, and Kimmo Raatikainen, editors, *Workshop on Context Awareness for Proactive Systems (CAPS 2005), Helsinki, Finland*, pages 31–42. HIIT Publications, June 2005a. ISBN 952-10-2518-2.

Anders Kofod-Petersen and Marius Mikalsen. Context: Representation and reasoning – representing and reasoning about context in a mobile environment. *Revue d'Intelligence Artificielle*, 19(3):479–498, 2005b.

Mikko Korpela, Hettie A. Soriyan, and Karen C. Olufokunbi. Activity analysis as a method for information systems development. *Scandinavian Journal of Information Systems*, 12:191–210, 2001.

Kari Kutti. Activity theory as a potential framework for human-computer interaction research. In Nardi [1996b], pages 17–44.

Bruno Latour. *Science in Action: How to Follow Scientists and Engineers Through Society*. Harvard University Press, 1988. ISBN 0674792912.

Till C. Lech and Leendert W. M. Wienhofen. Ambieagents: A scalable infrastructure for mobile and context-aware information services. In

AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, pages 625–631, New York, NY, USA, 2005. ACM Press.

Doug Lenat. The dimensions of context-space. Technical report, Cycorp, Austin, TX, October 28 1998. URL http://www.cyc.com/doc/context-space.pdf.

Aleksei N. Leont'ev. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

Yang Li, Jason I. Hong, and James A. Landay. Topiary: a tool for prototyping location-enhanced applications. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 217–226, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: http://doi.acm.org/10.1145/1029632.1029671.

Chistopher Lueg. Representaion in pervasive computing. In *Proceedings of the Inaugural Asia Pacific Forum on Pervasive Computing*, 2002a.

Christopher Lueg. Looking under the rug: On context-aware artifacts and socially adept technologies. In *Proceedings of the Workshop The Philosophy and Design of Socially Adept Technologies at the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2002)*. National Research Council Canada NRC 44918, April 2002b.

David McSherry. Mixed-initative dialogue in case-based reasoning. In *Workshop Proceedings ECCBR-2002*, Aberdeen, 2002.

Daisy Mwanza. Mind the gap: Activity theory and design. Technical Report KMI-TR-95, Knowledge Media Institute, The Open University, Milton Keynes, 2000.

Bonnie A. Nardi. A brief introduction to activity theory. *KI – Künstliche Intelligenz*, 1:35–36, 2003.

Bonnie A. Nardi. Activity theory and human computer interaction. In *Context and Consciousness* Nardi [1996b], pages 7–16.

Bonnie A. Nardi, editor. *Context and Consciousness*. MIT Press, Cambridge, MA, 1996b.

Allen Newell. The knowledge level. *Artificial Intelligence*, (18):87–127, 1982.

Pinar Öztürk and Agnar Aamodt. A context model for knowledge-intensive case-based reasoning. *International Journal of Human Computer Studies*, 48:331–355, 1998.

Michael Polanyi. *Personal Knowledge: Towards a Post-critical Philosophy*. N.Y.: Harper & Row, 1964.

Amanda Quek and Hanifa Shah. A comparative survey of activity-based methods for information systems development. In Isabel Seruca, Joaquim Filipe, Slimane Hammoudi, and Jose Cordeiro, editors, *Proceedings of sixth International Conference on Enterprise Information Systems (ICEIS 2004)*, volume 5, pages 221–229, Porto, Portugal, 2004. ISBN 972-8865-00-7.

Mahadev Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17, August 2001. URL citeseer.ist.psu.edu/satyanarayanan01pervasive.html.

Thomas M. Strat. Employing contextual information in computer vision. In *DARPA93*, pages 217–229, 1993.

Lucy A. Suchman. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, New York, NY, USA, 1987. ISBN 0-521-33137-4.

Peter Totterdell and Paul Rautenbach. Adaptation as a design problem. In *Adaptive User Interfaces*, pages 59–84. Academic Pres, 1990.

Lew S. Vygotsky. *Mind in Society*. Harvard University Press, Cambridge, MA, 1978.

Lew S. Vygotsky. *Ausgewählte Schriften Bd. 1: Arbeiten zu theoretischen und methodologischen Problemen der Psychologie*. Pahl-Rugenstein, Köln, 1985.

Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.

Mikael Wiberg and Charlotte Olsson. Designing artifacts for context awareness. In *Proceedings of IRIS 22 Enterprise Architectures for Virtual Organisations*, pages 49–58, Jyväskylä, 1999. Dept. of Computer Science and Information Systems, University of Jyväskylä.

Elisabetta Zibetti, Vicenc Quera, Francesc Salvador Beltran, and Charles Tijus. Contextual categorization: A mechanism linking perception and knowledge in modeling and simulating perceived events as actions. *Modeling and using context*, pages 395–408, 2001.

Using Activity Theory to Model Context Awareness: a Qualitative Case Study

G

Authors:
Jörg Cassens and Anders Kofod-Petersen

Abstract:
In this paper, we describe an approach to modelling context-aware systems starting on the knowledge level. We make use of ideas from Activity Theory to structure the general context model and to assess empirical data. We further on describe how the data-driven and the model-driven aspects of our approach are combined into a single knowledge model. We outline the design of an empirical study conducted to gather information about a concrete workplace environment. This information is then used to populate our context model. We describe also how the collected data can be used to validate our approach.

Main Result:
The context model proposed in Paper F is implemented in a context aware system for a hospital ward domain, and a qualitative empirical study of its performance in a simulated environment is given.

Published in:
Geoff C. J. Sutcliffe and Randy G. Goebel, editors, *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, pages 619–624, Melbourne Beach, 2006. AAAI Press.

My main contributions to the paper:

- Activity theory background
- Overview on related work
- Definition of goals for gathering data

The following aspects were jointly developed by the authors:

- Design of the study

G.1 Introduction

The area of context-awareness in pervasive computing has gained considerable momentum over the last years. Not only in the number of researchers dealing with this issue, but the scenarios and visions have also grown more sophisticated. Originally, Weiser proposed the world of ubiquitous computing as a world populated with ordinary items augmented to assist people in their day to day activities [Weiser, 1991].

Since then the complexity of the tasks that pervasive computing has to solve has steadily grown. The systems envisioned today are often proactive and described as intelligent environments; or ambient intelligence. Examples of these systems are the Aura system in Satyanarayanan [2001, p. 3], which, for example, is able to infer that sensitive information should not be presented at a talk, as unfamiliar faces are present. A similar example is available in Ducatel et al. [2001, p. 4], where Maria's visa has been negotiated automatically, thus allowing her to walk right through immigrations when arriving at her foreign destination.

Just attempting to approach these visions is a daunting task. The degree of pro-activity, autonomous behaviour, and complicated reasoning abilities is staggering. However, one of the central issues when autonomous systems are to function in an environment is the ability to perceive and make sense of that environment. Systems as described in the above examples are situated [Brooks, 1987] in the environment, and to a large degree inseparable from it [Gibson, 1979].

The area of context awareness attempts to deal with the issues of modelling, representing, and to some degree reasoning about the environment. However, historically there has been a close connection between the concept of context and location, often they have been regarded as synonymous. This is not surprising, as we, the users, are mobile. However, one very important aspect of situations that has largely been ignored is activity. We believe that focusing on activities will allow us to gain a better understanding of context and context awareness.

Several interesting approaches to investigate activities have been proposed; such as Actor-Network Theory [Latour, 1988], Situated Action [Suchman, 1987] or the Locales Framework [Fitzpatrick, 1998]. Another fascinating starting point is Activity Theory, which is based on the works of Vygotsky [1978]; Leont'ev [1978]. In this paper, we propose the use of Activity Theory to model context and describe situations.

Most of the recent research in context aware systems has been largely technology driven. It is "...driven by what is technically feasible rather than by what might be helpful in a situation." [Lueg, 2002b, p. 1] One main obstacle is the lack of a common understanding of what constitutes context. This lack of common understanding is by no means surprising, since no common theory on context understanding in humans seems to exist. Thus, it would be unreasonable to expect a common theory for artificial entities.

However, it is reasonable to assume that knowledge and reasoning play an important role when humans assess situations. Thus, it seems feasible to regard context in artificial systems from a knowledge level perspective [Newell, 1982]. This will give systems the advantage of reasoning about context, rather than relying on pattern matching only.

Furthermore, as IT systems are used by humans in social settings, it is viable to perform an analysis of context on the level of socio-technical systems [Lueg, 2002a]. In fact, the integration of intelligent systems into workplace environments marks a shift from mere tool usage to partnership between humans and intelligent artefacts.

This work is organised as follows: First, a short introduction to Activity Theory is given. Second, we describe the context model utilised in this work. This is followed by a demonstration on the use of Activity Theory to identify contextual information. Afterwards, we discuss which information is needed in order to design a system for a hospital ward scenario and how the data is gathered. The next section details how the context model is populated with domain knowledge and information about specific situations. Finally, a conclusion and outlook on future work is given.

G.2 Activity Theory

Activity Theory (AT) is a descriptive psychological framework helping to understand the unity of consciousness and activity. It is best described with a set of basic principles. These guiding principles include [Bannon and Bødker, 1991]:

- **Hierarchical structure of activity:** Activities (the topmost category) are composed of goal-directed actions. These actions are performed consciously. Actions, in turn, consist of non-conscious operations.
- **Object-orientedness:** Objective and socially or culturally defined properties. Our way of doing work is grounded in a praxis which is shared by our co-workers and determined by tradition. The way an artefact is used and the division of labour influence the design. Hence, artefacts pass on the specific praxis they are designed for.
- **Mediation:** Human activity is mediated by tools, language, etc. The artefacts as such are not the object of our activities, but appear already as socio-cultural entities.

- **Continuous Development:** Both the tools used and the activity itself are constantly reshaped. Tools reflect accumulated social knowledge, hence they transport social history back into the activity and to the user.
- **Distinction between internal and external activities:** In contrast to traditional cognitive psychology, Activity Theory emphasises that internal mental processes cannot be properly understood when separated from external activities, that is the interaction with the outside world.

A basic notion of Activity Theory is that the subject participating in an activity does so because he wants to achieve a certain goal. His interest is directed towards the object of an activity which he tries to use and modify to achieve an anticipated outcome. His interaction with this object is mediated by tools, creating the basic triangle of Subject, Object, and Mediating Artefact.

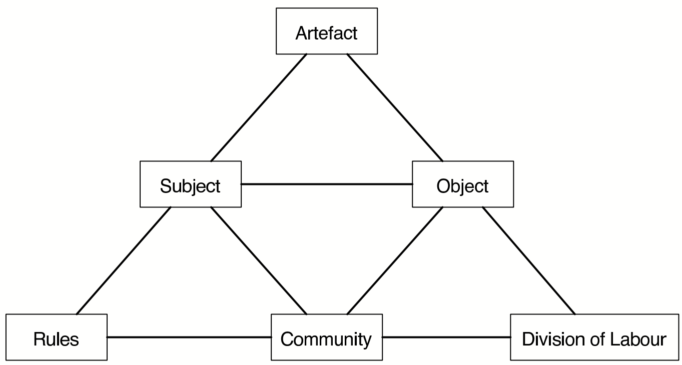

Figure G.1.: Cultural Historical Activity Theory (CHAT)

Since we consider social activities, the acting subject is part of a community. The relations between the acting subject and the community as well as between the community and the object are mediated by a set of rules and the division of labour (since the desired outcome is anticipated to be shared by the community, a solitary view on the relation between one subject and the object would miss important aspects).

The expanded model, including a community component and other mediators, is commonly referred to as Cultural Historical Activity Theory (CHAT). It is often depicted as the triangle shown in Figure G.1.

G.3 Context Model

The context model utilised in this work assumes a subjective view on situations. This is in contrast to the prevailing view where context normally describes an objectively defined situation. We argue that any experience is personal, thus the choice of contextual parameters and their weight will also be personal.

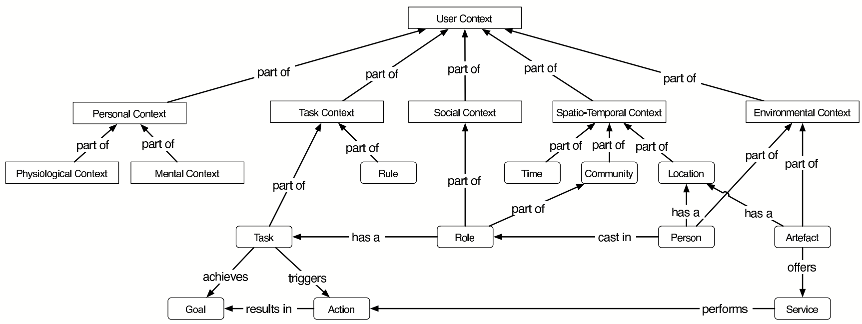

Figure G.2.: Context model

If we were to take this argument to the extreme, we could argue that not only the experiences are personal, but also the model and the representation. However, since our goal is to build artefacts that are useful and feasible to develop, we have chosen a pragmatic view on how to model context. The model is based on the definition given by Dey [2001]:

> Context is the set of suitable environmental states and settings concerning a user, which are relevant for a situation sensitive application in the process of adapting the services and information offered to the user.

This definition does not explicitly state that context is regarded as knowledge. However, following the argument in the introduction, we argue that context must be viewed from a knowledge perspective. Concurrently, we support the view maintained by Brézillon and Pomerol that context is not a special kind of knowledge. They argue that the knowledge regarded as context is dependent on the circumstances: "...knowledge that can be qualified as 'contextual' depends on the context!" [Brézillon and Pomerol, 1999, p.7]

Keeping to the pragmatic view on building artefacts, we impose a taxonomy on the context model during the design phase (Figure G.2). This taxonomy incorporates the tradition in context aware systems, and the general concepts found in Activity Theory. The taxonomy divides context into five sub-categories [Kofod-Petersen and Mikalsen, 2005]:

1. **Environmental context:** This part captures the user's surrounding, such as things, services, people, and information accessed by the user.
2. **Personal context:** This part describes the mental and physical information about the user, such as mood, expertise and disabilities.
3. **Social context:** This describes the social aspects of the user, such as information about the different roles a user can assume.
4. **Task context:** the task context describes what the user is doing, it can describe the user's goals, tasks and activities.
5. **Spatio-temporal context:** This type of context is concerned with attributes like: time, location and the community present.

The context model is represented as a multi-relational semantic network. It is used within the Creek framework for knowledge intensive Case-Based Reasoning (CBR) [Aamodt and Plaza, 1994; Aamodt, 2004]. The model allows for the inference of relationships between concepts by construction of contextual dependent paths between them. One important feature is the ability to match two features that are syntactical different, by explaining why they are similar [Aamodt, 1994; Jære et al., 2002].

G.4 Activity Theory for Identifying Context

We have further on brought this established knowledge model from the domain of context aware computing together with our activity theoretic approach to context awareness in order to design a context model which is sound from a psychological perspective.

Our interest in Activity Theory for context awareness is two-fold. On one hand, we use an activity theoretic model to build and justify a general knowledge model for capturing context related knowledge. This is a top-down, or model driven, approach to capture the essential aspects on the knowledge level. On the other hand, we use the same activity theoretic model to design empirical studies in the same setting where we later want to deploy a context aware system. In this second phase, done in a bottom-up way, or data driven, the data gathered in this process is used to populate the knowledge model with domain- and situation-specific knowledge.

Since our system builds on the knowledge intensive Case-Based Reasoning (CBR) methodology [Aamodt, 2004], the domain-specific knowledge gets incorporated into the general knowledge model of the system and the situation-specific knowledge takes the form of cases.

The top-down approach of building the knowledge model is described more thoroughly in Kofod-Petersen and Cassens [2006], so we will only describe it shortly here and go into more detail on the data-driven part later in this paper.

The contextual knowledge we want to capture includes knowledge about the acting subjects, the objects towards which activities are directed, the information sources accessed, and the community as well as knowledge about the mediating components, like rules or tools. To this end, we have proposed a mapping from the basic structure of an activity into the taxonomy of contextual knowledge as depicted in Table G.1. We can for example see that the personal context contains information we would associate with the acting subject itself.

Table G.1.: Basic aspects of an activity and their relation to a taxonomy of contextual knowledge

CHAT aspect	**Context Category**
Subject	Personal
Object	Task
Community	Spatio-Temporal
Mediating Artefact	Environmental
Mediating Rules	Task
Mediating Division of Labor	Social

We would like to point out that we do not think that a strict one to one mapping exists or is desirable at all. Our view on contextual knowledge is contextualised itself in the sense that different interpretations exist, and what is to be considered contextual information in one setting is part of the general knowledge model in another one. Likewise, the same piece of knowledge can be part of different categories based on the task at hand.

Other work on the use of AT in modelling context has been conducted e.g. by Kaenampornpan and O'Neill [2004]. The authors focus on modelling features of the world according to an activity theoretic model, but they do not carry out a knowledge level analysis of the activities. This is in contrast to our own approach, and we argue that our knowledge intensive approach has the advantage of giving the system the ability to reason about context so that it does not have to rely on pattern matching only. This is advantageous especially in situations where not all the important features are accessible by the system, for example because of limits of sensory input in mobile applications.

An interesting feature in Kaenampornpan and O'Neill's further work is the notion of history of context. The history is used to elicit a user's current goal [Kaenampornpan and O'Neill, 2005]. We do not explicitly address the problem of representing the user history in context models. The application area we are considering in this article features a set of relatively well defined situations, and information about the user's goal is included in the cases of the underlying CBR system.

Li et al. [2004] propose an activity based design tool for context aware applications. In contrast to our proposal, the authors focus on supporting the designer of context aware applications with a rapid prototyping tool, not on the use of Activity Theory in the Context model itself.

Wiberg and Olsson [1999] make also use of Activity Theory. The main issue here is the design of context aware tangible artefacts. The usage situation is well defined upfront and no reasoning about the context has to take place. This differs considerably from our own approach.

G.5 Gathering Data

We now have a well defined semantic network serving as a knowledge model which is sound both from an Activity Theory viewpoint and from the tradition of context-aware computing. The next step is to populate the model with data from real world situations.

The setting for our empirical study is supporting medical personnel at a hospital ward. The persons involved deal with different activities, like ward rounds, pre-ward round meetings, and different forms of examination. The staff has to access a large variety of different information systems. The main goal is to have a system which makes the information sources needed in different situations available pro-actively. To this end, the system must first identify the activity the system's user is involved in, identify his role, and then query the information sources which are likely to be accessed.

To gather data about this work processes, we have designed forms for a study which allow us to focus on different parts of an activity theoretic analysis of the work process. The forms had to meet certain requirements:

- It should be possible to clearly identify the different activities the users were involved in. Further on, the goal for each situation should be identified, even if the users did not explicitly state these goals. This would enable us to identify the different outcomes anticipated by the users, and eventually could help us building a model capturing the hierarchical structure of activities.

- The artefacts used should be identified, and different forms of use of these artefacts should be recorded. This would give us hints about the mediating role of artefacts. Special interest should be given to the use of infomation sources.

- The different entities involved in the activity as depicted in the basic triangle (see Figure G.1) should, if observable, be described in order to be able to directly connect the data collected to the knowledge level model.

- By observing the praxis of using artefacts, deeper insight on externalisation of cognitive processes can be gained. Although this is not in the scope of our current work, a study design which takes this aspect into consideration could help us evaluating the capabilities an intelligent system would have to provide to its users in order to be seen as an intelligent partner.

- Although a truly intelligent system would be able to adapt itself to completely new situations, we consider the usage situation, e.g. with regard to the governing rules and the capabilities of the tools used, as being relatively constant. Therefore, our study design did not particularly deal with issues of continuous development.

At the same time, the resulting form could not be too extensive since it was to be filled out by a single person observing the activities. The end result was a form which captured essentially the following aspects:

- **Location:** The room where the situation occurred
- **User:** The user of the system
- **Role:** The role of the user
- **Present:** Other persons present
- **Role:** The role of each of the persons present
- **Patient:** The ID of the patient in question
- **Time:** The time of day

- **Source:** Information sources and targets
- **I/O:** The direction of the information flow
- **Information:** Type of information

The data was collected through a period of one month at the St. Olavs Hospital in Trondheim, Norway. A medical student followed several employees and recorded the situations that occurred throughout the days on the forms we had designed for this task.

G.6 Populating the Context Model

The context aware system we are describing in this article is realised within the Creek framework for knowledge intensive CBR [Aamodt, 2004]. The knowledge components of Creek are modelled as a semantic network. The semantic network for our context aware application integrates the following components:

1. The basic knowledge, which holds the generic concepts necessary for modelling the general domain and case knowledge. See Aamodt [2004] for a more thorough description.
2. The general taxonomy as described in Section "Context Model".
3. General aspects of the activities, such as roles, artefacts, communities, and the relation between them as described in Section "Activity Theory for Identifying Context".
4. The adaptation of the generic model to the work environment at hand, in this case the hospital. The adaptation to each specific scenario consists itself of two different parts. The task is:
    a) To enrich the context model with domain specific information, like which artefacts were used and which services they offered and consumed, and
    b) to populate the context model by adding concrete situations (cases) that were observed.

G.6.1 Domain Specific Information

In order to adapt the generic context aware system we have described to a particular working environment, the tasks performed in this environment, the communities of labour existing, and the specific artefacts used, we have to enrich the knowledge model with specific information about the environment modelled. This enrichment includes: the different locations at the wards; the roles that employees, patients, and visitors assume; the classes of persons encountered in the wards; artefacts and services they offer and consume.

The empirical study performed was used to extract the necessary knowledge and model it in the Creek framework. A typical example for this kind of knowledge would be the observation that we have two distinct types of health workers (nurses and physicians), and that there are different types of physicians (e.g. consulting physicians, temporary physicians, and assistant physicians). Further more, this physicians could assume different roles, like group leader or examiner. Another example would be that five distinct laboratories are used, and that an ECG was placed at specific locations.

This knowledge was extracted from the collected data and modelled manually into the system.

G.6.2 Situation Specific Information

The data set contains 360 situations, 197 for cardiology and 163 for gastroenterology. The empirical evaluation of the system's performance is done in two steps. First, a qualitative evaluation of the data from the cardiology ward was carried out in order to review the context model and the integration on the knowledge level. This work has been completed. Second, data from both wards will be used in an quantitative evaluation of the final system architecture. This second step has not been completed yet.

The data describing the context of situations includes some information which could easily be sensed through available hardware, like location and the users involved and the time of the situation occurring. Some of the other data might not be easily available, like the presence of a patient's relatives.

However, since we are mainly concerned with methodological issues in this paper, we have not addressed the more technical aspects of collecting and fusing sensory information yet.

For the quantitative evaluation, approximately half of the situations were fed into the system manually, thereby giving the system a set of initial cases to reason about. The second half was used to test whether the system could successfully classify situations and identify the correct information sources needed.

The 197 situations at the cardiology ward which have been incorporated into the systen are distributed as described in Table G.2.

Table G.2.: Distribution of observed data for cardiology

Situation	AL7	AL9	AL14	OL9	∑
Pre pre ward	5				5
Pre ward round	7	22	11	26	66
Ward round	7	21	11	26	65
Examination		8	2	9	19
Post work		8	9	13	30
Pre discharge			2	4	6
Heart meeting		1		1	2
Discharge meeting				4	4

Eight different types of situations have been identified in the data set. Four different physicians were observed, where three were assistant physicians (AL7, AL9, and AL14) and one was a consultant physician (OL9). Beside these, several nurses, patients, and relatives were present in different situations.

The first qualitative analysis has shown that the CBR system was able to successfully identify new situations based on the initial set of cases. Further on, based on the knowledge about the sequence of actions contained in the previously seen cases, the system was able to identify the correct sequence of actions needed in the ongoing activity.

G.7 Conclusion and Future Work

We have shown that context aware intelligent systems can benefit from the socio-technical analysis made possible by applying Activity Theory. Moreover, taking socio-technical aspects into account is a necessity when intelligent systems are not used as mere tools but are designed to be more of a partner in a work process. It is beneficial to be able to make use of a sound psychological framework when defining a knowledge model as well as when constructing guidelines for observations. Our approach described in this paper can be used to design studies in real world settings which can be used as starting points for the deployment of context aware systems.

We have outlined how the observational data can be integrated into a knowledge level model to form a coherent multi-relational semantic network, which allows for the perception of the environment, reasoning about context to identify situations, and problem-solving based on this understanding.

Based on the data for the cardiology ward, we have populated our existing general model with domain- and situation-specific knowledge. At the same time, we have focused on identifying generic solution strategies corresponding to the situations we have discovered.

After the context aware system has successfully identified the current context and the potential goals of the human actors, the knowledge contained in the specific cases together with the domain-specific knowledge about possible courses of action make it feasible to support the human activities by offering guidance and retrieve necessary information.

As for empirical validation, we have performed an initial qualitative assessment of the system's integrity, and tested the ability of our reasoning component to correctly identify new situations on a subset of the collected data. Our initial results indicate that a knowledge intensive approach to combine situational data with general and domain specific knowledge can be regarded as being very promising when tackling the intricate problem of identifying situations. The next step is to execute a full simulation of the system on all available data.

On the methodological side, we will use the results from our empirical work to further formalise the relationship between different aspects of an AT based analysis and the different knowledge containers we can utilise in our system. Equally important is the development of a methodological approach to study design and data assessment.

G.8 Acknowledgements

Part of this work was carried out in the AmbieSense project, which was supported by the EU commission (IST-2001-34244). We would like to thank several of our colleagues for the cooperation on the empirical work; in particular Inger Dybdahl Sørby for facilitating the data collection at St. Olavs Hospital.

Bibliography

Agnar Aamodt. Knowledge-intensive case-based reasoning in CREEK. In Peter Funk and Pedro A. González Calero, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, pages 1–15, Berlin Heidelberg, 2004. Springer.

Agnar Aamodt. Explanation-driven case-based reasoning. In Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors, *Topics in Case-Based Reasoning: Proceedings of the 1^{st} European Workshop EWCBR 1993 –*

Selected Papers, volume 837 of *Lecture Notes in Computer Science*, pages 274–288, Berlin Heidelberg, 1994. Springer. ISBN 3-540-58330-0.

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994.

Liam Bannon and Susanne Bødker. Beyond the interface: Encountering artifacts in use. In John M. Carroll, editor, *Designing Interaction: Psychology at the Human-Computer Interface*, pages 227–253. Cambridge University Press, Cambridge, 1991.

Patrick Brézillon and Jean-Charles Pomerol. Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62(3):223–246, 1999.

Rodney A. Brooks. Planning is just a way of avoiding figuring out what to do next. Technical report, MIT, 1987.

Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Geraldine Fitzpatrick. *The Locales Framework: Understanding and Designing for Cooperative Work.* Ph.d. thesis, The University of Queensland, Australia, 1998.

James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

Martha Dørum Jære, Agnar Aamodt, and Pål Skalle. Representing temporal knowledge for case-based prediction. In Susan Craw and Alun Preece, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2002*, number 2416 in Lecture Notes in Artificial Intelligence, pages 174–188, Aberdeen, 2002. Springer.

Manasawee Kaenampornpan and Eamonn O'Neill. Modelling context: an activity theory approach. In Panos Markopoulos, Berry Eggen, Emilie Aarts, and James L. Croeley, editors, *Ambient Intelligence: Second European Symposium on Ambient Intelligence, EUSAI 2004*, volume 3295 of *Lecture Notes in Computer Science*, pages 367–374. Springer, November 2004.

Manasawee Kaenampornpan and Eamonn O'Neill. Integrating history and activity theory in context aware system design. In *Proceedings of the 1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE)*, May 2005.

Anders Kofod-Petersen and Jörg Cassens. Using activity theory to model context awareness. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers*, volume 3946 of *Lecture Notes in Computer Science*, pages 1–17, Edinburgh, 2006. Springer. ISBN 3-540-33587-0. URL http://dx.doi.org/10.1007/11740674_1.

Anders Kofod-Petersen and Marius Mikalsen. Context: Representation and reasoning – representing and reasoning about context in a mobile environment. *Revue d'Intelligence Artificielle*, 19(3):479–498, 2005.

Bruno Latour. *Science in Action: How to Follow Scientists and Engineers Through Society*. Harvard University Press, 1988. ISBN 0674792912.

Aleksei N. Leont'ev. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

Yang Li, Jason I. Hong, and James A. Landay. Topiary: a tool for prototyping location-enhanced applications. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 217–226, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: http://doi.acm.org/10.1145/1029632.1029671.

Christopher Lueg. Looking under the rug: On context-aware artifacts and socially adept technologies. In *Proceedings of the Workshop The Philosophy and Design of Socially Adept Technologies at the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2002)*. National Research Council Canada NRC 44918, April 2002a.

Christopher Lueg. Operationalizing context in context-aware artifacts: Benefits and pitfalls. *Information Science – The International Journal of an Emerging Discipline*, 5(2):42–47, 2002b.

Allen Newell. The knowledge level. *Artificial Intelligence*, (18):87–127, 1982.

Mahadev Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17, August 2001. URL citeseer.ist.psu.edu/satyanarayanan01pervasive.html.

Lucy A. Suchman. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, New York, NY, USA, 1987. ISBN 0-521-33137-4.

Lew S. Vygotsky. *Mind in Society*. Harvard University Press, Cambridge, MA, 1978.

Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.

Mikael Wiberg and Charlotte Olsson. Designing artifacts for context awareness. In *Proceedings of IRIS 22 Enterprise Architectures for Virtual Organisations*, pages 49–58, Jyväskylä, 1999. Dept. of Computer Science and Information Systems, University of Jyväskylä.

Designing Explanation Aware Systems: The Quest for Explanation Patterns

H

Authors:
Jörg Cassens and Anders Kofod-Petersen

Abstract:
Systems in general, and intelligent systems in particular, need to be able to explain their behaviour to their users or partners. Previously, a number of different user goals that explanations can support have been identified. Likewise, different kinds of explanations have been proposed. The problem remains how these abstract concepts can be made fruitful for the design of intelligent systems – they must be connected to software engineering methodologies. The work presented here builds on the concept of patterns and suggests using problem frames as a tool for requirements engineering. We further on propose to connect these problem frames with other design patterns as a tool supporting the implementation process.

Main Result:
This paper deals with software engineering problems of designing explanation aware systems. The five explanation goals introduced in Paper E are revisited and a problem frame diagram for each of the goals is presented. This goals are intended to enable software engineers to explicitly model explanatory capabilities.

Published in:
Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Explanation-Aware Computing – Papers from the 2007 AAAI Workshop*, number WS-07-06 in Technical Report, pages 20–27, Vancouver, BC, 2007. AAAI Press.

My main contributions to the paper:

- Background on explanations
- Survey of related work
- Final version of explanation problem frames
- Construction of example

The following aspects were jointly developed by the authors:

- Initial version of explanation problem frames

H.1 Introduction

Although a wide variety of knowledge engineering methodologies exists (see the following section), there seems to be a lack of methods focusing on the peculiarities of explanatory knowledge. These special features of explanations include in particular their role in enhancing the user experience by adding a level of self reflection about the actions of the system and the importance of explanations to gain the user's trust into the system's capabilities.

In human to human interaction, the ability to explain its own behaviour and course of action is a prerequisite for a meaningful interchange, therefore a truly intelligent system has to provide comparable capacities. In order to make sure that the system has sufficient knowledge about itself and that potential interests a user can have towards explanations can be satisfied, the design of explanatory capabilities should be an integral component of the system's design process.

Additionally, if we have an interest in the widespread adoption of explanation aware systems, we have to integrate the methods focussing on knowledge aspects with other software development methodologies to make the design of intelligent system accessible for a large group of software engineers. To this end, it is our aim to develop a formal notation to support a design methodology which is based on experiences both in the intelligent systems and software development communities. We therefore propose the use of patterns, especially the use of problem frames, in order to start the discussion of such methodologies.

The structure of this article is as follows: first, we give a short overview about the notion of explanations as used in this paper. Second, we refer to some related work both from the knowledge engineering and the software engineering disciplines. In the following step, we propose a description of user goals following Jackson's notion of problem frames. Thereafter, we present a simplified example of explanation aware re-engineering of an existing ambient intelligent solution before concluding with some remarks about further work.

H.2 Explanations

The ability to generate explanations is an important aspect of any intelligent system [Roth-Berghofer and Cassens, 2005; Sørmo et al., 2005]. We deem it necessary to investigate the explanatory capabilities from an early point onward in the design process to assure that the finished system can sufficiently explain itself. Therefore, an analysis of explanatory needs of both user and system should be part of the requirements engineering process. It is equally important to provide the system designers with methods to assure that explanatory knowledge and methods can be integrated into the application.

We have previously presented a framework for explanations in intelligent systems with a special focus on case-based reasoning [Sørmo et al., 2005]. Specifically, we have identified five goals that explanations can satisfy. The goal of *transparency* is concerned with the system's ability to explain how an answer was reached. *Justification* deals with the ability to explain why the answer is good. When dealing with the importance of a question asked, *relevance* is the goal that must be satisfied. *Conceptualisation* is the goal that handles the meaning of concepts. Finally, *learning* is in itself a goal, as it teaches us about the domain in question. These goals are defined from the perspective of a human user. His expectations on what constitutes a good explanation is situation dependend and has a historic dimension [Leake, 1995].

Roth-Berghofer has explored some fundamental issues with different useful kinds of explanation and their connection to the different knowledge containers of a case-based reasoning system [Roth-Berghofer, 2004]. Based on earlier findings from natural language explanations in expert systems, five different kinds of explanation are identified: *conceptual explanations*, which map unknown new concepts to known ones, *why-explanations* describing causes or justifications, *how-explanations* depicting causal chains for an event, *purpose-explanations* describing the purpose or use of something, and *cognitive explanations* predicting the behaviour of intelligent systems. Roth-Berghofer further on ties these different kinds of explanation to the different knowledge containers of case-based reasoning systems [Richter, 1995], namely *case base*, *similarity measure*, *adaptation knowledge*, and *vocabulary*.

Building on the last two works, we have earlier started to investigate a combined framework of user goals and explanation kinds [Roth-Berghofer and Cassens, 2005]. The goal of this work was to outline a design methodology that starts from an analysis of usage scenarios in order to be able to identify possible expectations a user might have towards the explanatory capabilities of an intelligent system. The requirements recognised can further on be used to identify which kind of knowledge has to be represented in the system, and which knowledge containers are best suited for this task. In that work, we have also identified the need for a socio-psychological analysis of workplaces in order to be able to design systems which can meaningful engage in socio-technical interactions.

We have also previously proposed the use of activity theory, a theory of human interaction with other humans and technical artefacts from industrial and organisational psychology, to investigate when explanations are important to the user [Cassens, 2004]. The same theory has shown its usefulness in designing a case-based reasoning system geared towards ambient intelligence [Kofod-Petersen and Cassens, 2006]. This work has recently been extended to explicitly take explanatory capabilities into account.

We are now in the process of investigating how these different aspects – a socio-technical analysis, user goals with explanations, and the different kinds of explanations – can fit into a design methodology which can be handled by knowledge and system engineers.

H.3 Related Work

The use of patterns [Alexander et al., 1977] is common for different software engineering approaches. Patterns can be used in different software development phases and they can have different foci. We can also identify knowledge engineering approaches making use of patterns.

When we look towards the software engineering world, we can see that patterns are used in different phases of the design process.

Early on in the requirements engineering process, *problem frames* [Jackson, 2001] are a method to classify software development problems. Problem frames look out into the world and attempt to describe the problem and its solution in the real world. Problem frames introduce concepts like 'Information Display' and 'Commanded Behaviour'.

Jackson's set of basic problem frames can be extended to be better able to model domain specific aspects. For example, Hatebur and Heisel [2005] introduce new problem frames for security problems. Their proposal includes problem frames for issues like 'Accept Authentication' and 'Secure Data Transmission'. They also provide architectural patterns connected to these problem frames.

On the software architecture level, we find *architecture patterns* [Avgeriou and Zdun, 2005]. At this level, we encounter concepts like 'Blackboards', 'Model-View-Controller', or 'Pipes and Filters'.

For finer grained software development close to the actual implementation, one can make use of design patterns which look inside towards the computer and its software [Gamma et al., 1995]. Design patterns deal with concepts like 'Factories', 'Facade', and 'Decorater'.

Patterns can also be useful for modelling non-functional requirements. HCI design patterns are such a special class of patterns. Rossi et al. [2000] introduce HCI patterns for hypermedia applications (like 'Information on Demand' and 'Process Feedback'). Another collection of HCI patterns can be found in van Welie and Trætteberg [2000], covering aspects like

'Wizards' or 'Preferences'.

Some research has been done on the issue of how patterns on different levels are related with each other. For example, Wentzlaff and Specker [2006] apply case-based reasoning to construct design patterns from developed problem frames. The problem part of the cases consist of problem frames, and the solution part is a corresponding HCI pattern.

Choppy et al. [2006] relate architectural patterns to problem frames. The design problem at hand can be divided into multiple frames, and the authors offer a modular approach to refining the problem frames into architectural patterns.

Methods and languages which use patterns and focus explicitly on the knowledge aspects of system design exist as well. There are for example efforts to provide reusable architectures by describing the abilities of (a library of) generic problem solving methods. An example for a component model is the Unified Problem-Solving Method Development Language UPML, cf. Fensel et al. [1999].

Plaza and Arco [1999] describe an application of the UPML model to case-based reasoning (CBR). They propose the ABC software architecture, based on the three components *task description*, *domain model*, and *adaptors*. The authors focus on the reuse part of the CBR cycle [Aamodt and Plaza, 1994] and interpret problem-solving as constructing a (case-specific) model of the input problem.

The INRECA [Bergmann et al., 2003] methodology is aimed at developing (industrial) case-based reasoning applications. Software process models from existing CBR applications are stored in an experience base which is structured at three levels. The *common generic level* is a collection of very generic processes, products, and methods for CBR applications. At the *cookbook level*, we find software models for particular classes of applications (so called recipes). At the *specific project level*, experiences from particular projects are stored. We can identify the recipes at the cookbook level as patterns.

Another well-known approach can be found with the CommonKADS methodology [Schreiber et al., 2000]. It encompasses both a *result perspective* with a set of models of different aspects of the knowledge based system and its environment, and a *project management perspective* starting from a spiral life-cycle model that can be adapted to the particular project.

The CommonKADS template knowledge model provides a way of (partially) reusing knowledge models in new applications and can be understood as patterns in the software engineering sense of the word.

Building on the KADS model and extending it explicitly towards software engineering, Gardner et al. [1998] introduce the notion of KADS Objects. The KADS Object framework allows direct support for object-oriented decomposition and utilises research on human cognition. The authors also supply a library of generic problem-solving templates, which themselves can be seen as software engineering patterns.

Unfortunately, despite the fact that a lot of work has been done on knowledge engineering methodologies and in particular the reuse of experience gained, it seems that little attention has been paid to the specifics of explanatory knowledge outlined above.

H.4 Explanation Problem Frames

The main purpose of any problem frame [Jackson, 2001] is to propose a machine which improves the combined performance of itself and its environment by describing the machine's behaviour in a specification. The most important approach is to address the frame concern. To explain ones behaviour a problem frame must be constructed that relates the behaviour the system shows to different parts of knowledge used by the system to support the chosen course of action in a specification.

Jackson [2001] originally described five different basic frames, each of which comes in different *flavours* and *variants*: 'required behaviour', 'commanded behaviour', 'information display', 'simple workpieces' and 'transformation'. Each basic frame has its own requirements, domain characteristics, domain involvement, and frame concern.

In general, a problem frame assumes a user driven perspective. Except for the 'required behaviour' basic frame, each frame assumes that the user is in control and dictates the behaviour of the machine. Since intelligent systems (ideally) take a much more pro active approach and mixed initiative issues become relevant, new problem frames addressing these topics have to be developed. For the course of this paper, we will focus exclusively on frames targeting explanatory aspects and will not discuss other types of problem frames.

Problem frames can be described by problem frame diagrams. These diagrams consist basically of dashed ovals, representing the requirements, plain rectangles, denoting application domains, and a rectangle with a double vertical stripe, standing for the machine (or software machine) domain to be developed. These entities become the nodes of the frame diagram. They are connected by edges, representing shared phenomena and denoting an interface. Dashed edges refer to requirement references. Dashed arrows designate constraining requirement references.

The domains can be of different types, indicated by a letter in the lower right corner. Here, a 'C' stands for a *causal* domain whose properties include predictable causal relationships among its phenomena. A

'B' denotes a *biddable* domain which lacks positive predictable internal behaviour. Biddable domains are usually associated with user actions. Finally, an 'X' marks a *lexical* domain. Such a domain is a physical representation of data and combines causal and symbolic phenomena.

In the software development process, problem frames are used in the following way. First, we start with a *context diagram*, which consists of domain nodes and their relations, but without the requirements. Afterwards, the context diagram is divided into sub problems. The resulting sub problems should, whenever possible, relate to existing generic *problem frames*. These generic problem frames are hereby instantiated do describe the particular sub problem at hand.

In the remainder of this section, we propose a set of new generic problem frames to capture aspects of explanations connected to the aforementioned different user goals identified in [Sørmo et al., 2005].

H.4.1 Transparency Explanation

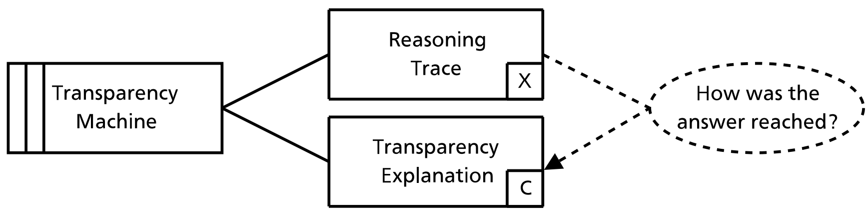

Figure H.1.: Transparency Explanation. An explanation supporting this goal gives the user some insight into the inner working of the system. To this end, the system inspects its own reasoning trace when formulating the explanation.

This goal is concerned with an explanation of how the system reached the answer.
"I had the same problem with my car yesterday, and charging the battery fixed it."
The goal of an explanation of this kind is to impart an understanding of how the system found an answer. This allows the users to check the system by examining the way it reasons and allows them to look for explanations for why the system has reached a surprising or anomalous result.

The frame diagram depicted in Figure H.1 highlights that in order to support the transparency goal, the software system has to inspect its reasoning trace and represent the relevant facts of its reasoning process to the user. We expect a transparency explanation usually to be given after the reasoning process has terminated.

H.4.2 Justification Explanation

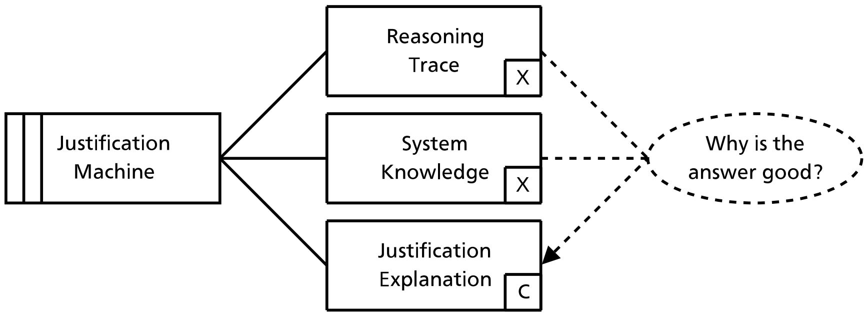

Figure H.2.: Justification Explanation. In contrast to the transparency explanation, the user is here not primarily interested in why the system exposes a particular behaviour, but wants to have evidence supporting that this behaviour is correct. Therefore, other knowledge has to be taken into account besides the reasoning trace.

When supporting the justification goal, we want to explain why the answer given is a good answer.
"You should eat more fish - your heart needs it!"
"My predictions have been 80% correct up until now."
This is the goal of increasing the confidence in the advice or solution offered by the system by giving some kind of support for the conclusion suggested by the system. This goal allows for a simplification of the explanation compared to the actual process the system goes through to find a solution. Potentially, this kind of explanation can be completely decoupled from the reasoning process, but it may also be achieved by using additional background knowledge or reformulation and simplification of knowledge that is used in the reasoning process.

A explanation supporting the justification goal, as shown in Figure H.2, has not only to take the reasoning of the machine into account, but it will also make use of other parts of the system's knowledge in order to generate after the fact explanations supporting its actions or decisions. Since justification explanations complement transparency explanations, we expect it to be given usually after the reasoning process has terminated.

H.4.3 Relevance Explanation

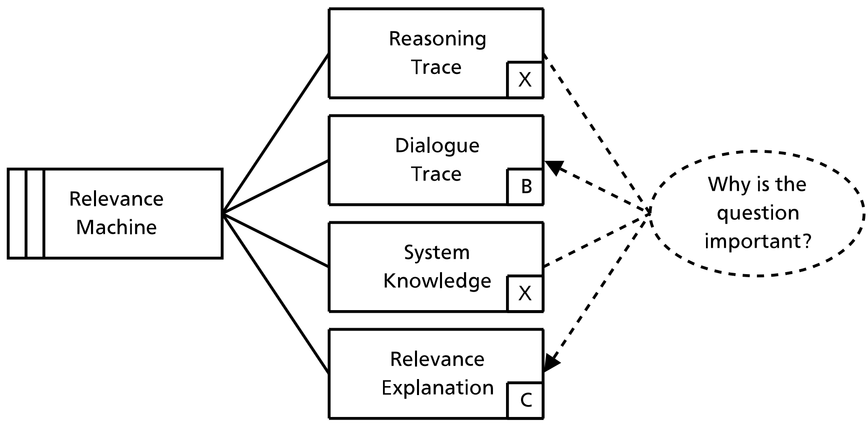

Figure H.3.: Relevance Explanation. An explanation supporting this goal should instil confidence by indicating that the system's behaviour is connected to the task at hand. Consequently, the reasoning and dialogue traces should be taken into account as well as other (domain) knowledge.

An explanation targeting this goal gives hints about why a question asked is relevant.
"I ask about the more common failures first, and many users do forget to connect the power cable."
An explanation of this type would have to justify the strategy pursued by the system. This is in contrast to the previous two goals that focus on the solution. The reasoning trace type of explanations may display the strategy of the system implicitly, but it does not argue why it is a good strategy. In conversational systems, the user may wish to know why a question asked by the system is relevant to the task at hand. It can also be relevant in other kinds of systems where a user would like to verify that the approach used by the system is valid.

Since this goal, depicted by the frame diagram in Figure H.3, is of particular interest for mixed initiative systems, the explaining machine has to relate its explanation both to its own dialogue with the user (and here in particular the questions asked by the system or the actions performed), the reasoning trace (in oder to relate to the situation the system assumes it is in) and the system knowledge relevant. In contrast to the first two goals, an explanation supporting this goal is important to be given during the reasoning process of the system.

H.4.4 Conceptualisation Explanation

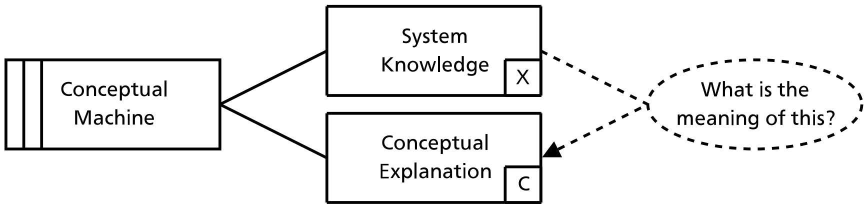

Figure H.4.: Conceptualisation Explanation. By giving a conceptualisation explanation, the system explicates its own conceptualisation of the domain or the task at hand to the user. Hence, it will connect the concept to be explained with its own knowledge components.

The conceptualisation goal deals with the need to clarify the meaning of concepts.
"By 'conceptualisation' we mean the process of forming concepts and relations between concepts."
One of the lessons learned after the first wave of expert systems had been analysed was that the users did not always understand the terms used by a system. This may be because the user is a novice in the domain, but also because different people can use terms differently or organise the knowledge in different ways. It may not be clear, even to an expert, what the system means when using a specific term, and he may want to get an explanation of what the system means when using it.

This explanation machine, represented with the frame diagram depicted in Figure H.4, builds on its own system knowledge. This highlights the fact that explanations supporting this goal should set unknown concepts in the context of the other knowledge the system has, and which is expected to be shared with the user already. Conceptualisation explanations are important both during the reasoning process (e.g. in addition to a relevance explanation) and after the reasoning process has terminated (e.g. in addition to a justification explanation).

H.4.5 Learning Explanation

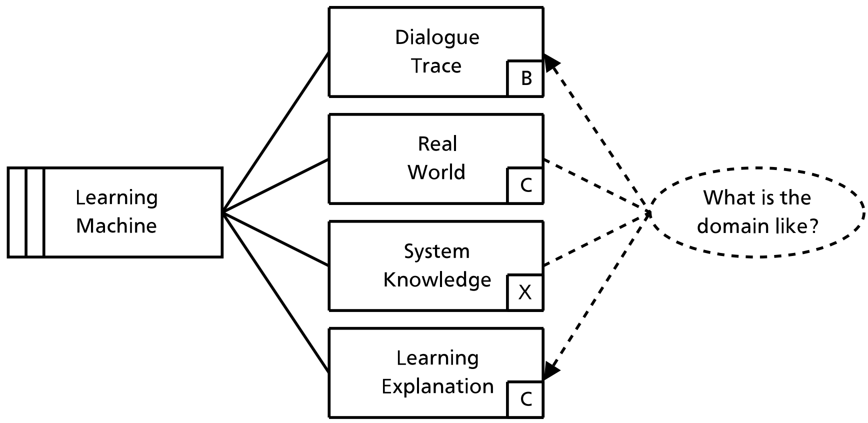

Figure H.5.: Learning Explanation. This goal is special, since it focuses on the user's interest in the application domain (hence the real world), and not on some particular behaviour of the system.

The learning goal focuses on the interest of the user to learn something about the application domain.
"When the headlights won't work, the battery may be flat as it is supposed to deliver power to the lights."
This goal is of specific interest for educational applications, which have learning as the primary goal of the whole system. In these systems, we cannot assume that the user will understand even definitions of terms, and may need to provide explanations at different levels of expertise. The goal of the system is typically not only to find a good solution to a problem, but to explain the solution process to the user in a way that will increase his understanding of the domain. The goal can be to teach more general domain theory or to train the user in solving problems similar to those solved by the system. In other words, the explanation is often more important than the answer itself.

The Figure H.5 highlights this fact by pointing out that the explanatory machine has to connect its own system knowledge with the real world (representing the application domain) in order to generate explanations supporting the user in gaining a better understanding of the application domain. The explanation given should also relate to the system's assumptions about its user, which is influenced by the dialogue trace. Because of the nature of this goal, it will usually be important during the system's reasoning process.

H.5 Example

Like mentioned in the introduction, we have designed and implemented an ambient intelligent application [Cassens and Kofod-Petersen, 2006] where the main purpose is to identify ongoing situations and proactively acquire digital information required by the persons present.

The system consists of three main components: one component acquiring data from the environment relevant for classifying the situation [Kofod-Petersen and Mikalsen, 2005]; one component assessing the ongoing situation through the use of case-based reasoning, which we understand as being context aware [Kofod-Petersen and Aamodt, 2006]; and finally one component conducting a task decomposition to solve the problem in the ongoing situation, which we understand as being context sensitive [Gundersen and Kofod-Petersen, 2005].

To exemplify how problem frames can assist us in building explanation aware applications, let us look at a typical engineering task where we start with the existing system and want to re-engineer it to include explanations.

As an example, we have the instance where the system correctly classifies an ongoing situation as a pre-ward round. A pre-ward round is a particular type of meeting that occurs every morning. Here the physician in charge and the nurse in charge go over the status of the patients on the ward, and decide on the treatment plan. The current condition of the patients in question is reviewed in light of any changes, test results, and the like.

We know from the knowledge acquisition and modelling process [Cassens and Kofod-Petersen, 2006] that the goal of this type of situation can be decomposed into the following sequence of tasks:

1. Acquire name of patient;

2. Acquire changes in patient's condition since yesterday;

3. Acquire any new results from tests;

4. Examine, and possible change, medication scheme;

5. Note changes in treatment.

If we focus on the context sensitive part of the system, its main purpose is to examine the artefacts, represented by agents, in the environment and find those that can supply relevant information. If we examine a particular pre-ward round situation, here one occurring at the cardiology ward, the problem can be decomposed as depicted in Figure H.10.

Figure H.10 demonstrates how the initial problem of finding the name of the patient can be facilitated by the *Patient List Agent*. Further on, the 'Acquire Information' task is decomposed into one task that acquires changes which are supplied by the *Electronic Patient Record*, the *WiseW* application and the *Patient Chart*, and another task that acquires results which can be delivered by the *Patient Chart* and the *WiseW* application. So far this application only supplies information without any explanation of its behaviour.

In order to demonstrate how the explanation goal problem frames can be used to model explanatory needs in the problem domain, we will start with a simplified problem diagram for our application (Figure H.6). We have modified Jackson's *information display* problem frame (Figure H.7) and used it as a starting point for the diagram. You can see three domains representing (groups of) the agents mentioned in Figure H.10 and explained above.

Additionally, you see the 'Display' domain which stands for the information display of the system and 'System Knowledge' for deciding which data sources to use. For the sake of clarity and simplicity, we have abstracted away the sensor parts of as well as the context aware parts of our example application and focus solely on the information gathering and display parts.

Let us now assume that the results displayed by the system are of such a nature that the physician using the system requires an explanation. Let us further focus on the transparency and justification explanations.

The transparency and justification explanations are related in the sense that they to some degree serve the same purpose. Namely, to persuade the user of the validity of the proposed solution and/or the validity of the problem solving approach chosen by the system. In the work presented here, we decide upon which of the two explanations to present as a function of the user's level of competence. That is, expert users are subject to transparency explanations and novice users to justification explanations [Mao and Benbasat, 2000].

To model the explanatory capabilities of the system, we want to integrate the explanation sub problems described by the two problem frame diagrams for the *Transparency* and the *Justification* goal with the original

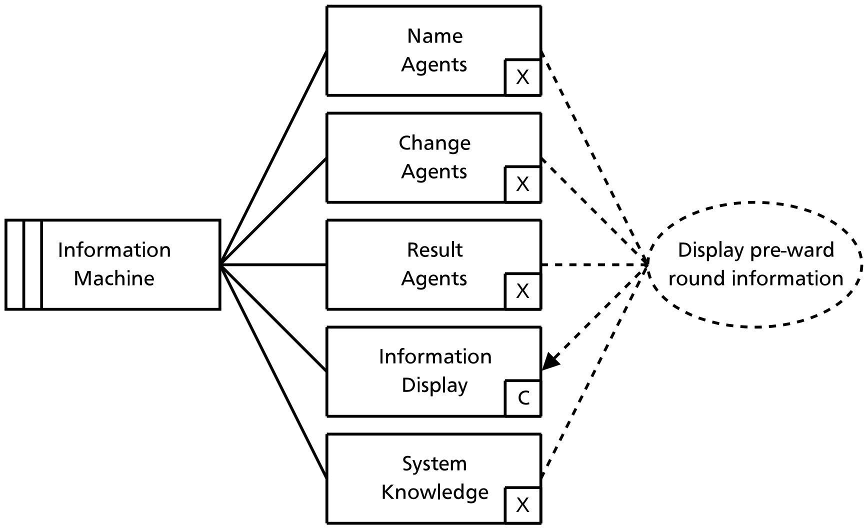

Figure H.6.: Simplified problem diagram for an ambient intelligent system for displaying medical information in pre-ward rounds.

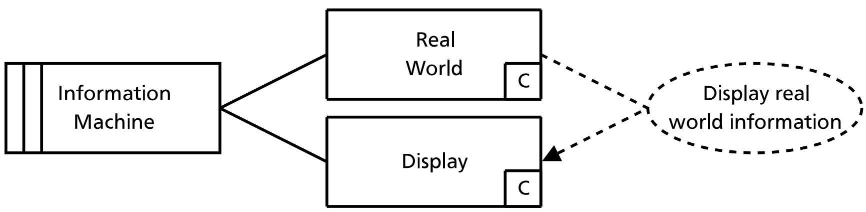

Figure H.7.: Jackson's information display problem frame diagram.

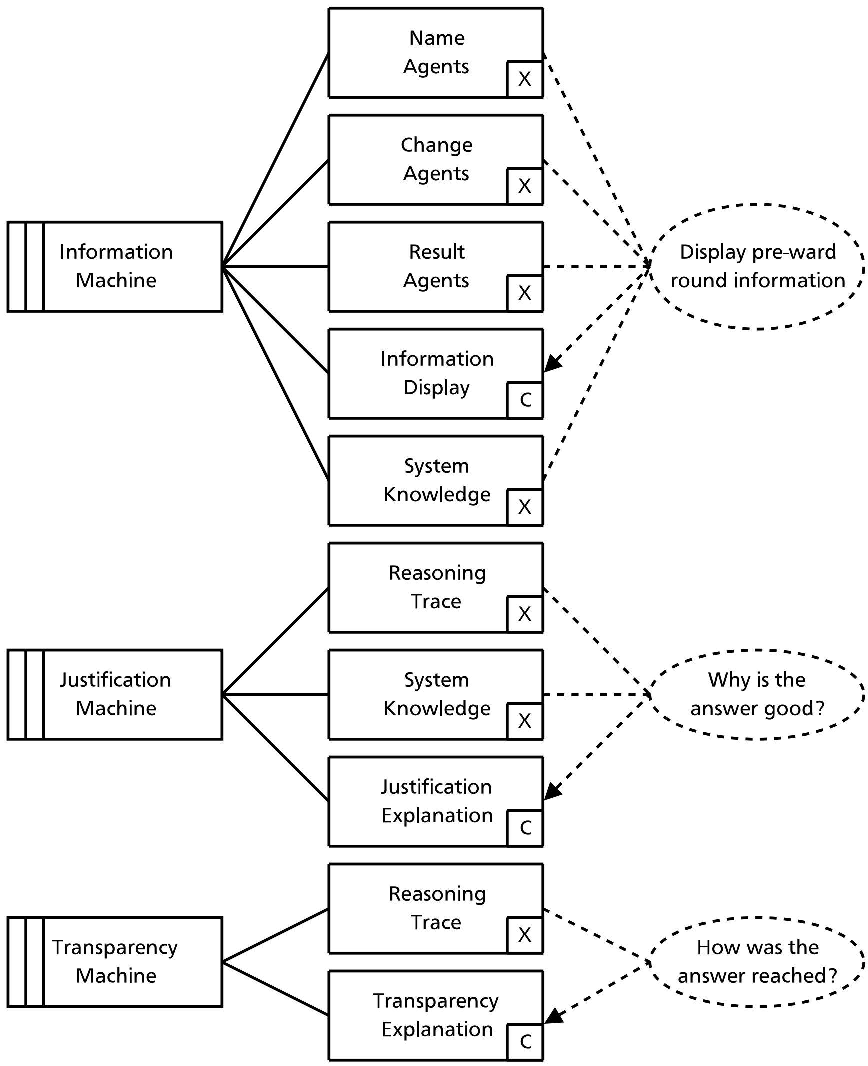

Figure H.8.: Simplified problem diagram with explanation problem frames added.

application problem diagram. The goal is to compose a single problem diagram for the three sub problems depicted in Figure H.8.

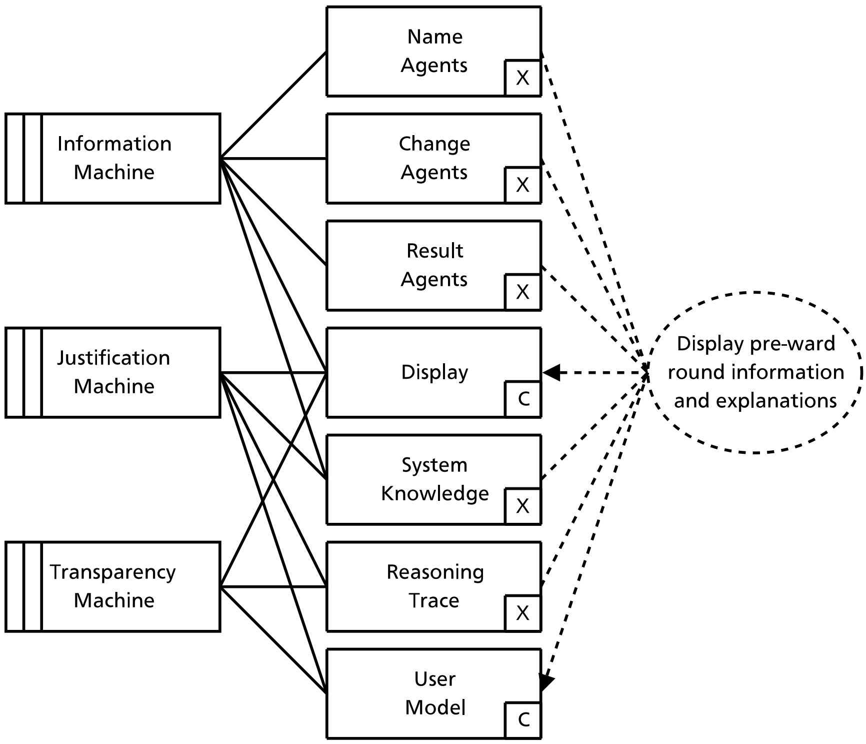

Figure H.9.: Simplified problem diagram with explanation problem frames integrated and user model added.

Figure H.9 shows one possible problem diagram for an explanation aware patient information display system. In order to be able to chose the right type of explanation – transparency or justification – we have included a user model component in this diagram. The need for this domain became clear when we tried to integrate the two different explanation machines. Please note also that we now share one common display for both the information delivery and explanation delivery.

The problem diagram depicted in Figure H.9 is a simplified version of a real world diagram. The solution shown is probably not the best one possible, but it can be seen that different sub problems can be composed into a larger problem diagram. Some of the domains of the problems can be shared, whereas others might only be used by one or some sub problems (please note the user model in our example).

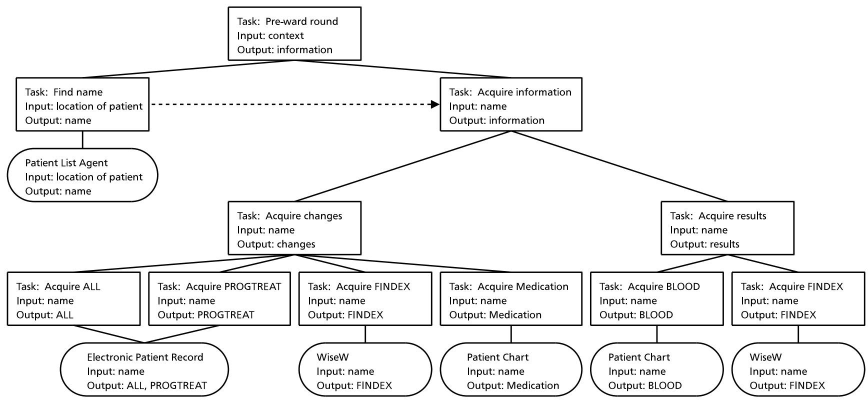

Figure H.10.: Decomposition tree for a pre-ward round situation, as constructed by our context sensitive component.

After we have included the explanatory machine in our problem diagram, we can re-visit the problem described above. The expert user physician might wish to know how the particular combination of information displayed was reached. According to the transparency explanation problem frame, this explanation can be achieved by displaying the reasoning trace. This can for example be done by showing that the top task 'Preward round' was selected as a function of the classification, by displaying how the decomposition tree looks like, and by supplying information about the agents which were selected.

For the justification explanation, the novice user physician would like to know why this combination of information is any good. This can be achieved by relating the reasoning trace to the domain model of the system. For example, according to the domain model, the 'Acquire Medication' task could be satisfied not only by the *Patient Chart* but also by the *Electronic Patient Record*. However, as the *Electronic Patient Record* agent was busy serving other requests only the *Patient Chart* could respond to this request.

This example shows how the use of explanation goal problem frames can explicate which kind of explanatory knowledge is necessary to support the explanatory needs of the users of the system. It can help identifying the structure of the problem at hand and the knowledge components required. It can further on be used as a means of communication between prospective users, software engineers, and knowledge engineers.

H.6 Conclusion and Future Work

This is ongoing work, but we have sketched how a set of problem frames targeted specifically towards explanatory capabilities of knowledge based systems can support the engineering process of explanation aware systems. With the explicit use of patterns, we have started to formalise the previously introduced notions of explanation goals and explanation kinds.

Until now, we have only looked out into the environment in which the intelligent system has to function, and proposed a formal notation for the description of user goals. These outward looking descriptions should further be connected with another view looking inwards towards the implementation of the system.

There are two directions of research we want to explore. One direction is to amend the explanation problem frames and to further analyse the relation between explanation goals and explanation kinds. To this end, we have to formalise the previously proposed stepwise refinement process from goals to kinds [Roth-Berghofer and Cassens, 2005] so that we can construct combined patterns for goals and kinds. By this, we are going to populate the proposed model with examples for how the outward directed view on the non functional user requirements for explanation aware systems can be combined with the inward directed view of necessary explanatory knowledge.

The other direction is aimed at relating the proposed explanation goal problem frames with architectural patterns. Ideally, this would enable us to discuss explanation issues, knowledge aspects, HCI aspects, and functional requirements at a very early stage of the development process without any a priori assumptions about the problem solving methods used.

Further on, it is necessary to extend the formalism at "both ends", meaning that we on one hand have to revisit our analysis of how the necessary knowledge to support the different explanation kinds can be represented in the actual system, and that we on the other hand have to refine our socio-technical analysis to end up with a psychologically plausible model for elucidating the explanatory needs of the prospected users.

H.7 Acknowledgements

Part of this work has been supported by Accenture Innovation Lab Norway.

Bibliography

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994.

Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. Oxford University Press, New York, 1977.

Paris Avgeriou and Uwe Zdun. Architectural patterns revisited – a pattern language. In *Proceedings of the tenth European Conference on Pattern Languages of Programs (EuroPlop 2005)*, pages 1–39, Irsee, 2005.

Ralph Bergmann, Klaus-Dieter Althoff, Sean Breen, Mehmet Göker, Michel Manago, Ralph Traphöner, and Stefan Wess. *Developing Industrial Case-Based Resoning Applications: The INRECA Methodology*, volume 1612 of *Lecture Notes in Computer Science*. Springer, Berlin, second edition, 2003.

Jörg Cassens and Anders Kofod-Petersen. Using activity theory to model context awareness: a qualitative case study. In Geoff C. J. Sutcliffe and Randy G. Goebel, editors, *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference*, pages 619–624, Melbourne Beach, 2006. AAAI Press. ISBN 978-1-57735-261-7. URL `http://www.aaai.org/Library/FLAIRS/flairs06contents.php`.

Jörg Cassens. Knowing what to explain and when. In Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, Technical Report 142-04, pages 97–104, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Christine Choppy, Denis Hatebur, and Maritta Heisel. Component composition through architectural patterns for problem frames. In *APSEC '06: Proceedings of the XIII Asia Pacific Software Engineering Conference*, pages 27–34. IEEE Computer Society, 2006. doi: http://dx.doi.org/10.1109/APSEC.2006.27.

Dieter Fensel, Richard Benjamins, Stefan Decker, Mauro Gaspari, Rix Groenboom, William Grosso, Mark Musen, Enrico Motta, Enric Plaza, Guus Schreiber, Rudi Studer, and Bob Wielinga. The component model of upml in a nutshell. In *WWW Proceedings WICSA1, 1st Working IFIP Conference on Software Architectures*, San Antonio, Texas, 1999.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston, MA, 1995.

Karen M. Gardner, Alexander Rush, Michael K. Crist, Robert Konitzer, and Bobbin Teegarden. *Cognitive Patterns*. Cambridge University Press, Cambridge, UK, 1998.

Odd Erik Gundersen and Anders Kofod-Petersen. Multiagent based problem-solving in a mobile environment. In Eivind Coward, editor, *Norsk Informatikkonferance 2005, NIK 2005*, pages 7–18. Institutt for Informatikk, Universitetet i Bergen, November 2005. ISBN 82-519-2081-7.

Denis Hatebur and Maritta Heisel. Problem frames and architectures for security problems. In Rune Winther, Bjoern Axel Gran, and Gustav Dahll, editors, *Proceedings of the 24th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, number 3688 in Lecture Notes in Computer Science, pages 390–404, Berlin, 2005. Springer.

Michael Jackson. *Problem Frames – Analysing and Structuring Software Development Problems*. Addison-Wesley, Boston, MA, 2001.

Anders Kofod-Petersen and Agnar Aamodt. Contextualised ambient intelligence through case-based reasoning. In Thomas R. Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *Proceedings of the Eighth European Conference on Case-Based Reasoning (ECCBR 2006)*, volume 4106 of *Lecture Notes in Computer Science*, pages 211–225, Berlin, September 2006. Springer.

Anders Kofod-Petersen and Jörg Cassens. Using activity theory to model context awareness. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers*, volume 3946 of *Lecture Notes in Computer Science*, pages 1–17, Edinburgh, 2006. Springer. ISBN 3-540-33587-0. URL http://dx.doi.org/10.1007/11740674_1.

Anders Kofod-Petersen and Marius Mikalsen. Context: Representation and reasoning – representing and reasoning about context in a mobile environment. *Revue d'Intelligence Artificielle*, 19(3):479–498, 2005.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

Ji-Ye Mao and Izak Benbasat. The use of explanations in knowledge-based systems: Cognitive perspectives and a process-tracing analysis. *Journal of Managment Information Systems*, 17(2):153–179, 2000.

Enric Plaza and Josep L. Arco. The abc of adaptation: Towards a software architecture for adaptation-centered cbr systems. Research Report 99-22, Artificial Intelligence Research Institute (IIIA), Bellaterra, Spain, 1999. http://www.iiia.csic.es/.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Gustavo Rossi, Daniel Schwabe, and Fernando Lyardet. User interface patterns for hypermedia applications. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 136–142, New York, NY, 2000. ACM Press. ISBN 1-58113-252-2. doi: http://doi.acm.org/10.1145/345513.345288.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Peter Funk and Pedro A. González Calero, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, pages 389–403, Berlin Heidelberg, 2004. Springer.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, pages 451–464, Chicago, 2005. Springer. URL http://dx.doi.org/10.1007/11536406_35.

Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde, and Bob Wielinga. *Knowledge Engineering and Management – The CommonKADS Methodology*. MIT Press, Cambridge, MA, 2000.

Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in case-based reasoning – perspectives and goals. *Artificial Intelligence Review*, 24(2):109–143, October 2005. ISSN 0269-2821. URL http://dx.doi.org/10.1007/s10462-005-4607-7.

Martijn van Welie and Hallvard Trætteberg. Interaction patterns in user interfaces. In *Proceedings of the seventh Pattern Languages of Programs Conference (PLoP)*, Monticello, 2000. URL `http://citeseer.ist.psu.edu/vanwelie00interaction.html`.

Ina Wentzlaff and Markus Specker. Pattern-based development of user-friendly web applications. In *ICWE '06: Workshop Proceedings of the sixth International Conference on Web Engineering*, New York, NY, 2006. ACM Press.

Explanations and Context in Ambient Intelligent Systems

I

Authors:

Anders Kofod-Petersen and Jörg Cassens

Abstract:

Ambient intelligent systems are context aware by perceiving and reasoning about their environment, they perceive the needs of their users and proactively respond to these needs by being context sensitive. Users do not interact with these systems by traditional means only, but also through behavioural interfaces. This combination of mixed initiative systems and unconventional interfaces puts strong requirements on the explanatory capabilities of any system. The work presented here focuses on explaining the behaviour of an ambient intelligent systems to its users. It demonstrates how explanations can be combined with context to deal with the different types of explanations that are required for a meaningful interaction of a system and its users.

Main Result:

The relationships between *context awareness* and *context sensitivity* on the one hand and explanations as a *means of reasoning* and a *means of communication* with the user on the other hand are explored. It is proposed how concepts from activity theory can be used to address the different goals a user can have towards explanations, and it is dicussed how these goals can be satisfied in the different phases of the use of the system.

Published in:

Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors, *Modeling and Using Context – CONTEXT 2007*, volume 4635 of *LNCS*, pages 303–316, Roskilde, Denmark, 2007. Springer.

Copyright:

My main contributions to the paper:

- Background on explanations and activity theory
- Original version of the description of relationships between context and explanations
- Drafting of the example

The following aspects were jointly developed by the authors:

- The mapping between AT concepts and explanation goals is the result of long sessions in front of a whiteboard

I.1 Introduction

Recent insights into both ubiquitous and pervasive computing have lead to the realisation that to achieve the scenarios and visions proposed, systems must be viewed as more complex than initially argued by Weiser [1991]. This has lead to the developments jointly labelled as *ambient intelligence* [Ducatel et al., 2001]. The explicit focus on intelligence stands in stark contrast to the original argument by Weiser, where: "no revolution in artificial intelligence is needed – just the proper embedding of computers into the everyday world" [Weiser, 1991, p. 3].

The core of an ambient intelligent systems is the ability to appreciate the system's environment, be aware of persons in this environment, and respond intelligently to their needs. To realise the abilities of an ambient intelligent system, three main areas of responsibility can be identified [Kofod-Petersen and Aamodt, 2006]: first, the initial responsibility of *perceiving* the world that the system inhabits; second, the responsibility of being aware of the environment and reason about ongoing situations, which traditionally has been labelled as *context awareness*; and third, exhibit appropriate behaviour in ongoing situations by being *context sensitive* [Kofod-Petersen and Aamodt, 2006; Yau et al., 2002].

Arguably one of the most important aspects of an ongoing situation is the activity that is occurring. For an ambient intelligent system to function it must be able to reason about its own, as well as other ongoing activities. Systems that aim at exhibiting the properties connected with ambient intelligence must be more than mere reactive systems, where deliberation and reasoning plays an important part.

Marx [1867] demonstrates this difference by arguing that even though a spider conducts operations that resemble those of a weaver, and a bee humbles many an architect, there is a significant difference between them. Even the worst architects raise the structures in their imagination before they are erected in reality. At the end of each labour-process, we get a result that already existed in the imagination of the labourer. The labourer not only affects the materials used, but also realises a purpose that gives the law to his modus operandi, and to which the labourer's will must be subordinated. Besides the exertion of the bodily organs, the process demands that, during the whole operation, the workman will be steadily in consonance with his purpose. This means close attention. The less he is attracted by the nature of the work, and the mode in which it is carried on, and the less, therefore, he enjoys it as something which gives play to his bodily and mental powers, the more close his attention is forced to be.

The elementary factors of the labour-process are: *i*) the personal activity of the labourer, *ii*) the subject of the work, and *iii*) its instruments.

This is also the starting point for activity theory. To capture the activity related aspects of any situation, activity theory [Vygotsky, 1978; Leont'ev, 1978] can be used to acquire and model the relevant knowledge. Briefly, activity theory considers activities as a set of actions and operations on an object. These actions and operations are conducted by a labourer, or subject, to achieve an already imagined outcome. The subject's actions and operations are mediated by the use of certain instruments, or artefacts. We will elaborate on this later.

According to Turing [1950], one indication that a system *is* intelligent is its ability to *appear* intelligent; i.e. by passing the Turing test. Therefore, we need to understand what makes humans appear intelligent.

Following Kant, human understanding has as a necessary constituent the ability to conceptualise perceived phenomena (structured through 'categories of understanding') through an active, discursive process of making sense of the intuitive perception [Kant, 1787, p. 58].

In later works, Kant gives us a more detailed description of his understanding of human reason. He makes clear that the human ability of reasoning has perceptivity (attentio), abstraction (abstractio), and reflection (reflexio) as its necessary preconditions [Kant, 1798, p. 138].

Further on, it is important to note that the ability of human beings to act freely, the ability to initiate a causal chain from freedom, is coupled with his ability to act morally (Kant describes freedom as the ratio essendi of the moral law, while the moral law is the ratio cognoscendi of freedom [Kant, 1788, p. 4]). Kant couples the ability to act morally (and thus freely) with the ability to give a rational explanation of the behaviour in his categorical imperative – "Act so that the maxim of thy will can always at the same time hold good as a principle of universal legislation" [Kant, 1788, p. 30]. Therefore, we can ascribe the ability of explaining ones behaviour and motives to every rational being, that means to every intelligent entity. We therefore count explanatory capabilities, in particular the ability to explain ones own understanding of the world and ones own behaviour, as a necessary precondition for appearing intelligent.

I.2 Background

One approach to realising intelligent behaviour is by employing *case-based reasoning* [Aamodt and Plaza, 1994]. This method springs from understanding reasoning as an explanation process [Schank, 1986]. Our understanding of common occurrences assists us in comprehending stories, in such a way that details omitted or assumed implicitly do not make a story incomprehensible for us. Our general knowledge about situations, the expectations, and the behaviour which should be exhibited are stored in what has been referred to in psychology as *scripts* [Schank and Abelson, 1977], which are closely related to the concept of *schema* [Bartlett, 1932; Rumelhart, 1980].

Sørmo et al. [2005] present a framework for explanations in intelligent systems with a special focus on case-based reasoning. Specifically, they identify five goals that explanations can satisfy. The goal of *transparency* is concerned with the system's ability to explain how an answer was reached. *Justification* deals with the ability to explain why the answer is good. When dealing with the importance of a question asked, *relevance* is the goal that must be satisfied. *Conceptualization* is the goal that handles the meaning of concepts. Finally, *learning* is in itself a goal, as it teaches us about the domain in question. These goals are defined from the perspective of a human user. His expectation on what constitutes a good explanation is situation dependend and has a historic dimension [compare e.g. Leake, 1995].

Roth-Berghofer has explored some fundamental issues with different useful kinds of explanations and their connection to the different knowledge containers of a case-based reasoning system [Roth-Berghofer, 2004]. Based on earlier findings from natural language explanations in expert systems, five different kinds of explanation are identified: *conceptual explanations*, which map unknown new concepts to known ones, *why-explanations* describing causes or justifications, *how-explanations* depicting causal chains for an event, *purpose-explanations* describing the purpose or use of something, and *cognitive explanations* predicting the behaviour of intelligent systems. Roth-Berghofer, further on, ties these different kinds of explanation to the different knowledge containers of case-based reasoning systems [Richter, 1995], namely case base, similarity measure, adaptation knowledge, and vocabulary.

Building on these two works, we have earlier started to investigate a combined framework of user goals and explanation kinds [Roth-Berghofer and Cassens, 2005]. The goal of this work was to outline a design methodology that starts from an analysis of usage scenarios in order to be able to identify possible expectations a user might have towards the explanatory capabilities of an intelligent system. The requirements recognised can further on be used to identify which kind of knowledge has to be represented in the system, and which knowledge containers are best suited for this task. In this work, we have identified the need for a socio-psychological analysis of workplaces in order to be able to design systems that can meaningful engage in socio-technical interactions.

In order to further explore the assumed advantages of designing systems from a socio-technical perspective, we have later on investigated the use of theories from industrial and organisational psychology in designing a case-based reasoning system geared towards ambient intelligence. The work presented here shows how the user-centric explanation goals can be satisfied by relating kinds of explanations in context awareness and context sensitivity with a socio-technical approach to modelling context.

I.3 Use of Activity Theory as a Means to Model Context

We have published some work on using activity theory to model context awareness [Kofod-Petersen and Cassens, 2006; Cassens and Kofod-Petersen, 2006]. Although we have discussed the use of this theoretical framework to help understand when to deliver an explanation [Cassens, 2004], we have not previously explored how to make use of the same theoretical framework for designing explanatory capabilities for context aware systems. We will now outline how these deficiencies can be overcome.

First in this section, we will give a short summary of aspects of activity theory that are important for this work. See Nardi [2003] for a short introduction to activity theory and Bødker [1991]; Nardi [1996a] for deeper coverage. The theoretical foundations of activity theory in general can be found in the works of Vygotsky and Leont'ev Vygotsky [1978]; Leont'ev [1978]; Vygotsky [1985].

Activity theory is a descriptive tool to help understand the unity of consciousness and activity. Its focus lies on individual and collective work practice. Some of its basic properties are:

- **Hierarchical structure of activity:** Activities (the topmost category) are composed of goal-directed actions. These actions are performed consciously. Actions, in turn, consist of non-conscious operations.

  If an action fails, the operations comprising the action can get conceptualised and might become conscious actions in the next attempt to reach the overall goal. This is referred to as a *breakdown situation*.

- **Object-orientedness:** Objective and socially or culturally defined properties. Our way of doing work is grounded in a praxis which is shared by our co-workers and determined by tradition. The way an artefact is used and the division of labour influences the design. Hence, artefacts pass on the specific praxis they are designed for.

- **Mediation:** Human activity is mediated by tools, language, etc. The artefacts as such are not the object of our activities, but appear already as socio-cultural entities.

- **Continuous Development:** Both the tools used and the activity itself are constantly reshaped. Tools reflects accumulated social knowledge, hence they transport social history back into the activity and to the user.

- **Distinction between internal and external activities:** Traditional cognitive psychology focuses on what is denoted internal activities in activity theory, but it is emphasised that these mental processes cannot be properly understood when separated from external activities, that is the interaction with the outside world.

We have used an expanded model of activity theory, the Cultural Historical Activity Theory [CHAT, compare e.g. Kutti, 1996; Mwanza, 2000], in order to analyse the use of technical artefacts as instruments for achieving a predefined goal in the work process as well as the role of social components, like the division of labour and community rules.

We have linked these different aspects of an activity to related categories of context in order to build a psychologically plausible context model. At the same time, we have used the model to guide our analysis of the work processes to be modeled into the system. However, we have not exploited all of the above mentioned aspects of activity theory in order to gain insight into the expectations and needs of the prospective users with regard to explanations.

I.4 Explanations and Context

The term explanation can have different foci. Either as goals that explanations can satisfy or as kinds of explanations that can be given. In addition, Leake identifies three different facets of explanations within the context of case-based reasoning [Leake, 2004]:

- Using explanations to support the case-based reasoning process

- Generating explanations by case-based reasoning

- Using cases for explaining system results to an external user

With our notion of user goals, we can subsume the last two facets as both being targeted towards the user of the system. In our understanding, showing the case to the user is a special case of 'generating explanations by case-based reasoning', making use of the case-based reasoning assumption that similar problems have similar solutions. Provided that the user has some knowledge about the similarity function and that the case structure is understandable by the user, the displayed case acts as an explanation to the user [see e.g. Sørmo et al., 2005; Cunningham et al., 2003]. We are left with two functions of an explanation, as described in Aamodt [1991]: first, enhancing and promoting the reasoning process. Second, delivering some knowledge about the reasoning process, its results, or implication to the user. We call the first aspect the *system centric view* on explanation and the second one the *user centric view* on explanation:[1]

- Explanation as **part of the reasoning process** itself.
  Example: a knowledge intensive case-based reasoning system can use its domain knowledge to explain the absence/variation of feature values.

- Giving explanations of the found solution, its application, or the reasoning process **to the user**.
  Example: in an engine failure diagnosis system, the user gets an explanation on why a particular case was matched.

We have earlier argued that an ambient intelligent system consists of three layers, each with their own responsibility [Kofod-Petersen and Aamodt, 2006]. The top layer is responsible for *perceiving* the world and order the perceived data into a coherent context structure on which reasoning is possible. The awareness layer is responsible for assessing the context and classify an ongoing situation. This layer demonstrates the ability of *context awareness*. Finally, the third layer is responsible for selecting and executing suitable behaviour based on the classification done in the awareness layer. This ability is referred to as *context sensitivity*.

In this architecture, context serves two purposes. Initially it is used as a focussing lens on the part of the world that can be perceived. Here the context limits the parts of the knowledge that the system uses to classify the situation. The second use of context is in the context sensitivity layer, where context is viewed as a lens that focuses the part of the system's knowledge that is to be used to satisfy the goal of the situation.

Figure I.1 depicts the dual use of context. Initially the Situation Context is what the context aware part uses to execute the case-based reasoning process that classifies the situation. Once the situation has been classified a suitable goal for this situation is found. This goal further limits the part of the context that is necessary for the context sensitivity part to exhibit appropriate behaviour. The goal as well as the context are made available to the context sensitivity part, as is indicated by the Goal arrow and the Goal Context in the figure.

[1] This distinction is valid not only for case-based reasoning systems.

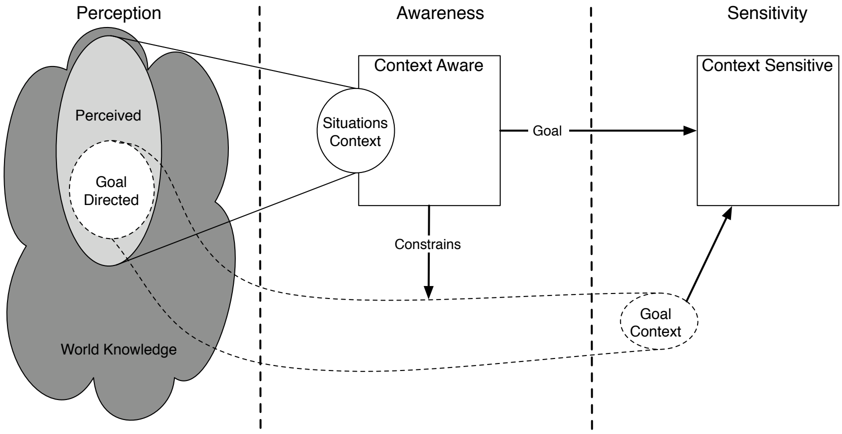

Figure I.1.: Dual use of Context

For the purpose of this paper we will disregard the perception layer of the architecture as the perception layer demonstrates no reasoning capabilities, and only structures perceived data syntactically. Following our earlier arguments introduced in Kofod-Petersen and Aamodt [2006], we identify these two aspects as two distinct steps in the reasoning process:

- **Context Awareness:** Trying to detect which situation the system is in.
  Example: An ambient intelligent system for supporting health personnel figures out that the user is on a ward-round because of the time of the day, the location, and the other persons present.

- **Context Sensitivity:** Acting according to the situation the system thinks it is in.
  Example: the same system fetches the newest versions of electronic patient records of all patients in the room from the hospital systems. When the user stands close to the bed of a patient, the system automatically displays them.

Combining these views on explanation and on context, we end up with two dimensions of inquiry as depicted in Table I.1. This table shows the four different areas where explanations can be required, divided into a system centric and user centric view.

Table I.1.: Context and Explanations

	Context Awareness	Context Sensitivity
System Centric	Generate an explanation to **recognise** the situation	**Identify** the behaviour the system should expose
User Centric	**Elucidate** why the system identifies a particular situation	**Explicate** why a certain behaviour was chosen

In the system centric view where explanations are a part of the reasoning process it is possible to initially generate an explanation used to *recognise* the situation. In this step we are using explanations to find out what situation we are in, by explaining similarities between a new situation and previously experienced situations. Following the recognition of a situation we can now use explanations to identify appropriate behaviour.

When dealing with the user centric view we can initially use explanations to elucidate why the system assumes that we are in a certain situation. The system can use all available sources of knowledge in order to gain the user's confidence in its capabilities. In the situation where the the system is required to explicate the behaviour that it exhibits, the explanation is used to explain why it takes a specific action.

As described above, activity theory has been used to recognise contextual facets of a work situation. By integrating the knowledge necessary for supporting the different explanatory goals of the user with this contextual information, the explanatory capabilities of the system are coupled with the different contexts. Hence, the hypothesis is that this explanatory knowledge will indeed primarily be used in the appropriate context.

We will now explore the relations between the basic properties of activity theory and explanation goals.

Hierarchical structure of activity: The fact that activities are hierarchically structured, and that changes in these structures occur, facilitates certain explanation goals. Actions that are performed often will be transformed into operations. Vice versa, if an anticipated outcome of an operation does not occur, non-conscious operation will become conscious actions. This is called a breakdown situation. The explanatory capabilities of a system should support this. In fact two goals are relevant in these situations:

- **Transparency:** If parts of the non-conscious operations are carried out by artefacts, the system might need sufficient knowledge to explain the artefacts inner working in case of a breakdown.

- **Relevance:** If an artefact involved in an action can behave differently than expected, it should be made clear why the unexpected behaviour occurred.

Object-orientedness: In the activity theoretical sense of the term object-oriented, the meaning of this term is twofold. On one hand, it highlights that all human activities have an objective, a goal, and therefore points towards the mental part of an activity. On the other hand, it refers to the fact that this mental objectives are directed towards the physical world. This holds for automated processes insofar as the automation already assumes a goal, and is supposed to support this goal:

- **Transparency:** It should be possible for a system to explain its relation to the physical processes.
- **Justification:** An intelligent system should be able to explain its goals to the user.

Mediation: Every activity will incorporate some tools, be it physical (like machinery) or psychological artefacts (like language). If parts of the activity are carried out by an intelligent artefact, this artefact both acts as a mediator in the physical world and as a mediator of the psychological processes of the user:

- **Justification:** The system should be able to explain the connection between its actions and the reasoning process.

Continuous development: The aspect of continuous development deals with the continuous change in the way we interact with the world. Both the user's activities and the artefacts used are changing. It should be noted that this includes the necessity for an intelligent system to adopt to changes over time:

- **Learning:** The system should be able to support the user's learning processes. If the system is extended, or new capabilities are included, the system should be able to act as teacher. It should therefore incorporate knowledge about how the new component facilitates the problem solving process.

Distinction between internal and external activities: Activity theory tries to overcome the dichotomy of mental processes and the outside world by focussing on the relation between internal and external activities. It is therefore crucial that the system supports the user in building an understanding of the artefacts used.

- **Conceptualisation:** The system should support the user's understanding of it by providing means of explaining its own world model to him.

Not all explanation goals can be satisfied by an activity theoretical perspective alone. Some goals can only be satisfied by inspecting other parts of the knowledge model, either in all cases or for certain situations. As an example, when recognising a situation the transparency goal can be satisfied by supplying a trace of the reasoning process used for classification. The different sources of knowledge required to satisfy the different goals will be further discussed in the following section.

I.5 Identifying Explanations Kinds from Goals

With the combination of explanations and context described above it is now possible to identify different kinds of explanations by identifying the explanation goals of the user. The four different areas where explanations can be required are shown in Table I.1. For the purpose of this paper we will focus on the user centric perspective where explanations are used to elucidate why the system identifies a particular situation and to explicate why a certain behaviour was chosen.

As we have stated before, users do not interact with an ambient intelligent system by traditional means only but also through behavioural interfaces. This means also that if the system gets everything right, it should be unobtrusive and supportive. The main situations where explanations are necessary are when something goes wrong, i.e. the system does not recognise the correct context or follows a path of actions which the user perceives as wrong, unusual, or unexpected. So while we do not dismiss the option that a user wants some explanation from the system even if it does what the user expects, we do not focus on this aspect in this paper. But it has to be kept in mind that the system's explanatory capabilities should also cover its ability to explain itself when nothing goes wrong. This is of special importance during the beginning of the use of the system in order to gain the user's trust into the system.

We would also like to point out that we have chosen not to consider the learning goal in this paper. The learning goal is specifically targeted towards educational systems. The goal of such a system is typically not only to find a good solution to a problem, but to explain the solution process to the user in a way that will increase his understanding of the domain. We do not consider this type of systems at the time being.

I.5.1 Context Awareness

In case of the context aware user centric perspective the system might *misclassify* a situation. In this case the system must satisfy the goals of *transparency, justification* and *conceptualisation*. In case of *transparency* and *justification* they explain the process through which the classification was reached. The choice between a *transparency* or *justification* goal is governed by the user's proficiency level. Where an expert user will require *transparency*, a novice user requires *justification*[2]. These two goals map to the 'how' and 'why' explanation kinds, where 'how' explains the causal chain of events leading to the classification, and 'why' justifies why the system thinks that the answer is good. The knowledge required to supply these kinds of explanations is found within the reasoning method, e.g. similarity measures in case-based reasoning.

I.5.2 Context Sensitivity

When dealing with the context sensitive user centric perspective, the system has two main situations in which explicating is required (not counting the situation where the system exhibits flawless behaviour). These two main situations are when the system exhibits *wrong behaviour* for the situation, and when it exhibits *unexpected behaviour*. Both of these situations can result in a breakdown situation as defined by activity theory. In case of a *wrong behaviour* the system's goal is not in line with the user's goal, any operations performed by the user will fail and become actions, thus a breakdown situation is occurring. In this case, the system displaying wrong behaviour must satisfy the same goals as when misclassifying the situation. This means that the *transparency/justification* goals must be satisfied. As with the case of misclassification, these goals map to the 'why' and 'how' kinds, which require knowledge about the reasoning process employed. In addition, the hierarchical structure principle in activity theory can guide the process through which these goals are satisfied.

From a user perspective, the system can *behave unexpectedly* in several different manners: it can request an *unexpected action* from the users, non-user actions can be performed by a *new or alternative person* or by a *new or unexpected artefact*.

When the system requests a *new action* from the user, a breakdown situation occurs, and the user must respond consciously. Again, the goals of *transparency/justification* must be satisfied. In addition, the system must satisfy the *relevance* goal by explaining the relevance of the requested action, and in case of previously unperformed actions *conceptualization* is

[2]This separation will be used consistently throughout the rest of the paper.

required. The hierarchical structure principle in activity theory can guide the process through which the *transparency/justification* and *relevance* goals are satisfied, whereas the *conceptualization* goal can be satisfied by inspecting the specific domain model. The *relevance* goal maps to the 'purpose' kind where an explanation of the purpose of the requested action gives the relevance. Finally the *conceptualization* goal maps to the 'conceptualization' kind, mapping unknown concepts to known ones.

If a non-user action is performed by a *new or alternative artefact*, three goals must be satisfied: *transparency/justification, relevance* and *conceptualization*. The mediation principle in activity theory can guide the process where the *transparency/justification* goal is satisfied, whereas the *relevance* and *conceptualization* goals are satisfied by inspecting the specific domain model. As aforementioned the *transparency/justification* goals map to the 'why' and 'how' kinds and the *conceptualization* goal maps to the 'conceptualization' kind. In the case of the *relevance* goal, this maps to the 'purpose' kind when dealing with alternative artefacts by describing the purpose of this artefact. When dealing with a new artefact, the *relevance* goal also maps to the 'conceptualization' kind.

For non-user actions performed by a *new or alternative person*, the description is similar to the one for artefacts. However, one important distinction exists. In the activity theoretical part of the knowledge model persons are part of the community that cooperates with the user through a division of labour. However, our current modelling of persons and roles does not distinguish between the two. Thus, even though an action is performed by a new or alternative person, the fact that the role is unchanged means that the activity as viewed from the system is unchanged. This is contrary to the way artefacts are perceived, where using a new or alternative artefact to perform an action will result in a change in the activity. This means that no activity theoretical principle can guide the satisfaction of the goals, thus other parts of the knowledge model must be inspected.

I.6 Example

We will briefly investigate the relations between context and explanation by the means of an example. Let us consider the following scenario: We have a case-based diagnostic system for aircraft failures. An engineer is equipped with an intelligent mobile assistant and one of his tasks is to diagnose the probable causes of engine problems. Let us assume that the engineer is working both at his home base and at line stations where faults have occurred.

Scenario 1 – Misclassified Context: Let us assume that our engineer is going to work with the head of engineering on a new schedule for sending engineers to line stations. He is doing administrative work and not working on technical problems. The time of this meeting, however, is at a time where there is usually a briefing with all engineers, and the system also recognises that some of the other people usually participating at this meeting are present. However, instead of being in a meeting room, we are at the office of the head of the engineering group, a fact which contradicts the assumption of being in the briefing. The system might now explain away this unusual facet by generalising that both the meeting room and the office are rooms and that an office to a certain degree serves the same purpose as a meeting room. Therefore, the system assumes that we are in a briefing and delivers fault information about the airplanes which are scheduled to be worked on.

When this error becomes obvious to the engineer, he might want to check why the system displayed this kind of information. So we are in the *explicate* phase of Table I.1. If he is an expert user of the system, he might have an interest what lead to the problem, so his goal is *transparency*. The kind of explanation helpful is a 'why' explanation, in particular one where the system displays the best matched cases and that it has classified the office as a general kind of room.

Scenario 2 – New Artefact Used: Let us now assume that the engineer is working on a diagnostic task and, in the course of this task, needs access to some performance data. This is recognised by the system. The knowledge source usually used for this kind of data is temporarily not available, so the system queries a different system which was added recently. This comes as a suprise to the engineer who was not aware of neither the unavailability of the first system nor the existence of the second.

The engineer now wants to know why the data from the new system is helpfull, he has a *relevance* goal. This can be supported by a 'purpose' kind of explanation, and by inspecting its own domain knowledge, the system can describe the purpose of the new data source, for example by explaining that the new data source is a backup system for performance data.

I.7 Summary and Future Work

This paper builds on a view of ambient intelligence encompassing first an understanding of the situation (context awareness) and then decisions on behaviour (context sensitivity). It has been argued that in both phases, explanations can be viewed from a system centric as well as a user centric perspective. It has further been described how explanations play a key role in ambient intelligent systems as a necessary prerequisite for a system being perceived as intelligent by human users.

The conceptual framework presented here describes how explanations can be used in the different parts of an ambient intelligent system. Further on, it describes how knowledge about requirements for explanations which can fulfil different user goals can be gained. We have introduced a means of taking user goals into account which is both psychologically plausible and in line with the tradition in context aware computing.

We have further on outlined how an understanding for user goals can be obtained both from an activity theoretic analysis of the activity environment and from the general and domain specific knowledge encompassed in the system at hand.

We have described how different user goals for explanations are related to different kinds of explanation and by this have outlined what knowledge a system has to contain in order to fulfil the user's goals. However, we have not yet tied this into a detailed methodology for intelligent systems design.

The three layered conceptual architecture (perception – awareness – sensitivity) combined with our conceptual model of explanations in ambient intelligence gives a foundation for the development of explanation aware applications. The different goals a user might have towards explanations together with their mapping to kinds and the inclusion of socio-technical analytic methods help us integrating the explanatory capabilities of the application at an early stage of the design process.

Our current implementation of an ambient intelligent case-based reasoning system can cater to the system centric perspective of explanations to some degree, but this has to be developed further. Regarding the user centric perspective, the current application does support the transparency, conceptualization and justification goals, where the latter is only supported partially due to the underlying issues with plausible inheritance in the current Java implementation. For the other goals, further implementation work is necessary.

Another aspect that deserves further attention is our model of the division of labour. In order to reconcile our view on artifacts and humans, we have to find ways to integrate the modelling of different persons as subjects into our generic context model.

Additionally, we want to augment existing design guidelines with methods for the analysis of social aspects which can lead to a better understanding of the environment in which the ambient intelligent system has to function than ad-hoc methods can give. It is also important to note that we have not yet fully utilised some aspects of our theoretical foundations in organisational psychology, like the notions of breakdown situations or functional organs in activity theory, or the use of semiotics for the organisation of the user interface itself.

Bibliography

Agnar Aamodt. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. PhD thesis, Norwegian Institute of Technology, Department of Computer Science, Trondheim, May 1991. URL http://www.idi.ntnu.no/grupper/su/publ/phd/aamodt-thesis.pdf.

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994.

Frederic C. Bartlett. *Remembering: A study in Experimental and Social Psychology*. Cambridge University Press, 1932.

Susanne Bødker. Activity theory as a challenge to systems design. In Hans-Erik Nissen, Heinz K. Klein, and Rudy Hirschheim, editors, *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pages 551–564. North Holland, 1991.

Jörg Cassens and Anders Kofod-Petersen. Using activity theory to model context awareness: a qualitative case study. In Geoff C. J. Sutcliffe and Randy G. Goebel, editors, *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference*, pages 619–624, Melbourne Beach, 2006. AAAI Press. ISBN 978-1-57735-261-7. URL http://www.aaai.org/Library/FLAIRS/flairs06contents.php.

Jörg Cassens. Knowing what to explain and when. In Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, Technical Report 142-04, pages 97–104, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An evaluation of the usefulness of case-based reasoning explanation. In Kevin D. Ashley and Derek Bridge, editors, *Case-Based Reasoning Research and Development: Proceedings ICCBR 2003*, number 2689 in Lecture Notes in Artificial Intelligence, pages 122–130, Berlin Heidelberg, 2003. Springer.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Immanuel Kant. *Anthropologie in pragmatischer Hinsicht*. Akademie, 1798.

Immanuel Kant. *Kritik der praktischen Vernunft*. Akademie, 1788.

Immanuel Kant. *Kritik der reinen Vernunft (2. Auflage)*. Akademie, 1787.

Anders Kofod-Petersen and Agnar Aamodt. Contextualised ambient intelligence through case-based reasoning. In Thomas R. Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *Proceedings of the Eighth European Conference on Case-Based Reasoning (ECCBR 2006)*, volume 4106 of *Lecture Notes in Computer Science*, pages 211–225, Berlin, September 2006. Springer.

Anders Kofod-Petersen and Jörg Cassens. Using activity theory to model context awareness. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers*, volume 3946 of *Lecture Notes in Computer Science*, pages 1–17, Edinburgh, 2006. Springer. ISBN 3-540-33587-0. URL http://dx.doi.org/10.1007/11740674_1.

Kari Kutti. Activity theory as a potential framework for human-computer interaction research. In Nardi [1996b], pages 17–44.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

David B. Leake. Workshop introduction: ECCBR-04 workshop on explanation in CBR. Slides, last access 22.02.2007, 2004. URL https://www.cs.tcd.ie/research_groups/mlg/ecbrws2004/.

Aleksei N. Leont'ev. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

Karl Marx. *Das Kapital. Kritik der politischen Ökonomie. Band 1: Der Produktionsprozeß des Kapitals*, volume 23 of *Marx Engels Werke (MEW)*. Dietz-Verlag, Berlin, 1867.

Daisy Mwanza. Mind the gap: Activity theory and design. Technical Report KMI-TR-95, Knowledge Media Institute, The Open University, Milton Keynes, 2000.

Bonnie A. Nardi. A brief introduction to activity theory. *KI – Künstliche Intelligenz*, 1:35–36, 2003.

Bonnie A. Nardi. Activity theory and human computer interaction. In *Context and Consciousness* Nardi [1996b], pages 7–16.

Bonnie A. Nardi, editor. *Context and Consciousness*. MIT Press, Cambridge, MA, 1996b.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Peter Funk and Pedro A. González Calero, editors, *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, pages 389–403, Berlin Heidelberg, 2004. Springer.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, pages 451–464, Chicago, 2005. Springer. URL `http://dx.doi.org/10.1007/11536406_35`.

David E. Rumelhart. Schemata: The building blocks of cognition. In Rand J. Spiro, Bertram C. Bruce, and William F. Brewer, editors, *Theoretical Issues in Reading and Comprehension*. Lawrence Erlbaum Assoc Inc, 1980.

Roger C. Schank. *Explanation Patterns – Understanding Mechanically and Creatively*. Lawrence Erlbaum, New York, 1986.

Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals and Understanding*. Erlbaum, 1977.

Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in case-based reasoning – perspectives and goals. *Artificial Intelligence Review*, 24(2):109–143, October 2005. ISSN 0269-2821. URL `http://dx.doi.org/10.1007/s10462-005-4607-7`.

Alan Turing. Computing machinery and intelligence. *Mind*, LIX(236): 433–460, October 1950.

Lew S. Vygotsky. *Mind in Society*. Harvard University Press, Cambridge, MA, 1978.

Lew S. Vygotsky. *Ausgewählte Schriften Bd. 1: Arbeiten zu theoretischen und methodologischen Problemen der Psychologie*. Pahl-Rugenstein, Köln, 1985.

Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.

Stephen S. Yau, Yu Wang, and Fariaz Karim. Development of situation-aware application software for ubiquitous computing environments. In *29th Annual International Computer Software and Applications Conference (COMPSAC'02)*, volume 1, pages 107–112, Edinburgh, Scotland, July 2002. IEEE Computer Society.

J

Explanations and Case-Based Reasoning in Ambient Intelligent Systems

Authors:
Jörg Cassens and Anders Kofod-Petersen

Abstract:
Interacting with intelligent systems in general and ambient intelligent systems in particular, requires that these systems have the ability to build a trust relationship with the users. The ability to explain its own behaviour is one of the most important abilities that such a system can exhibit to gain trust. We argue that explanations are not just an addition to an ambient intelligent system rather it is an approach to the design and implementation of such a system. Explanations are useful both for the reasoning process itself and as a means of communicating with the users. In this paper, we present a knowledge intensive approach for identifying different contexts and generating a course of action depending on the context found. We explore the use of explanations both as a means of reasoning and as a means of communication with the user.

Main Result:
A context aware system for a hospital ward environment is introduced, and it is shown how explanation needs in different phases of the use of the system are addressed.

Published in:
David C. Wilson and Deepak Khemani, editors, *ICCBR-07 Workshop Proceedings*, pages 167–176, Belfast, Northern Ireland, 2007.

My main contributions to the paper:

- Background on explanation and CBR
- Original version of facets of explanation and context
- Description of the different relations

The following aspects were jointly developed by the authors:

- Final version of facets of explanation and context

J.1 Introduction

Ambient intelligent systems are described by their ability to be aware of its users, perceive their needs and respond intelligently to them [Ducatel et al., 2001]. To achieve this, such a system must exhibit pro-activity and reasoning. Thus, contrary to traditional systems where the user is in charge and the system plays a passive role, ambient intelligent system can assume responsibility and behave proactively. The shift from passive systems, to what could be regarded as a partnership between humans and intelligent artefacts, fosters the need for social adept system [Marsh, 1995], in such a way that intelligent systems have to show certain abilities traditionally ascribed to humans [Pieters, 2001]. Among these abilities, we would count a system's ability to explain its behaviour.

Further, the nature of an ambient intelligent system dictates that the way we traditionally interact with computer systems are substituted with multi modal interfaces and in particular behavioural interfaces. It is not only the input devices where behavioural interfaces are used; the main output device for an ambient intelligent system is its behaviour. Thus, a user now runs the risk of having to master the methods of behavioural psychology to use an ambient intelligent system. Relying solely on observing the behaviour of an ambient intelligent system will, at best, give us a limited understanding of its behaviour. It will not allow us to consider any internal organisation. Yet, to persuade the user of a system's usability and credibility, explanations are in order

The work presented here argues that explanations are not just an addition to an ambient intelligent system; rather it is an approach to the design and implementation of such a system. Explanations are a useful approach to both the reasoning process from a system centric perspective, as well as a means of communication viewed from a user centric perspective.

This paper is structured as follows: First, an overview on how explanations and case-based reasoning are related is given. Secondly, an introduction to the use of case-based reasoning in ambient intelligence, as well as a short overview of the system employed here, is presented. Thirdly, a description of the theoretical framework underpinning our approach to the use of explanations and case-based reasoning in ambient intelligence is given. A summary and pointers to future work ends the paper.

J.2 Explanations and Case-based Reasoning

Originally, case-based reasoning did emerge from an understanding of reasoning as an explanation process [Schank, 1983, 1986]. Schank describes explanations as the most common method used by humans to support their decision making. An understanding of common occurrences assists in comprehending stories; in such a way that details omitted or assumed implicitly do not make a story incomprehensible.

Sørmo et al. [2005] present a framework for explanations in intelligent systems with a special focus on case-based reasoning. Specifically, they identify five goals that explanations can satisfy. The goal of *transparency* is concerned with the system's ability to explain how an answer was reached. *Justification* deals with the ability to explain why the answer is good. When dealing with the importance of a question asked, *relevance* is the goal that must be satisfied. *Conceptualization* is the goal that handles the meaning of concepts. Finally, *learning* is in itself a goal, as it teaches us about the domain in question.

Roth-Berghofer has explored some fundamental issues with different useful kinds of explanation and their connection to the different knowledge containers of a case-based reasoning system [Roth-Berghofer, 2004]. Five different kinds of explanation are identified: *conceptual explanations*, which map unknown new concepts to known ones, *why-explanations* describing causes or justifications, *how-explanations* depicting causal chains for an event, *purpose-explanations* describing the purpose or use of something, and *cognitive explanations* predicting the behaviour of intelligent systems. Roth-Berghofer further on ties these different kinds of explanation to the different knowledge containers of case-based reasoning systems [Richter, 1995], namely case base, similarity measure, adaptation knowledge, and vocabulary.

Building on the last two works, we have earlier started to investigate a combined framework of user goals and explanation kinds [Roth-Berghofer and Cassens, 2005]. The goal of this work was to outline a design methodology that starts from an analysis of usage scenarios in order to be able to identify possible expectations a user might have towards the explanatory capabilities of an intelligent system. The requirements recognised can be used to identify which kind of knowledge has to be represented in the system, and which knowledge containers are best suited for this task. In this work, we have identified the need for socio-psychological analyses of workplaces in order to be able to design systems which can meaningful engage in socio-technical interactions. The advantages of designing systems from a socio-technical perspective has been investigated through the use of activity theory as a method for designing an ambient intelligent case-based reasoning system [Kofod-Petersen and Cassens, 2006].

J.3 Case-based Reasoning and Ambient Intelligence

Weiser, who coined the term *ubiquitous computing*, did explicitly state that artificial intelligence was unimportant when realising the visions described [Weiser, 1991, p.3]. This view that proper embedding of computers is sufficient does also to a large degree saturate the field of *pervasive computing*. However, the insight that achievement of the visions described in pervasive computing does indeed require some degree of reasoning has lead to the developments jointly labelled as *ambient intelligence*.

The IST Advisory Group to the European Commission (ISTAG) describes ambient intelligence as human beings surrounded by intelligent interfaces, supported by computing and network technology embedded in everyday objects. More importantly, the environment should be aware of the presence of a person, perceive the needs of this person and respond intelligently to them [Ducatel et al., 2001].

The ability to be aware of the environment, reason about ongoing situations and decide about appropriate behaviour is closely linked with being knowledgeable about the world. Case-based reasoning in general, and knowledge intensive case-based reasoning in particular [Díaz-Agudo and González-Calero, 2000; Aamodt, 2004], appears to be a promising candidate for reasoning about situations and behaviour in an ambient intelligent setting.

Zimmermann reports on case-based reasoning used to generate recommendations based on the user's context in a mobile environment [Zimmermann, 2003]. The user context is encapsulated inside cases to facilitate comparison of contexts, generating recommendations based on case similarities, and learning of user behaviour. The cases are structured around a context describing the user's environment as the findings and a recommendation for a particular audio file as the solution.

Along the same lines of adapting solutions to users is the work by Ma et al., where case-based reasoning is used to adapt the behaviour of smart homes to user preferences [Ma et al., 2005]. The cases are represented as frames, where the findings are: the user, the environment, the time, and the values of the active devices. When new cases are instantiated; the similarities between the new and existing cases are calculated and a setting for the appliances in the house is selected.

Recently Bénard et al. have investigated the use of case-based reasoning as a mechanism for selecting suitable behaviour in different situations [Bénard et al., 2006]. They propose an agent-based architecture that uses perceived information, or context, as the findings of a case and the proposed action as the solution. The existing cases in the case base are preclassified situations modelled by a domain expert. The authors briefly describe how the case-based reasoning process approaches the problem of behaviour selection. Initially a new context is compared to the findings in the existing cases and the best matching case is retrieved. Secondly, the retrieved case is adapted to fit the new context.

The work presented here builds on the experience gained by applying case-based reasoning to situation assessment in an ambient intelligent system [Kofod-Petersen and Aamodt, 2003]. The architecture developed and system implemented approaches ambient intelligence by separating the main responsibilities into three layers.

The architecture has been implemented as an ambient intelligent system in a hospital ward [Kofod-Petersen and Aamodt, 2006]. The personnel at the hospital ward are involved in many different activities, such as doing ward rounds, meetings and different forms of examinations. The system's main purpose is to recognise ongoing situations and proactively acquired digital information relevant for the user.

Figure J.1 depicts the functional system architecture. The Context Middleware [Kofod-Petersen and Mikalsen, 2005b] provides an infrastructure that perceives the environment by collecting and maintaining a coherent model of available context. The CREEK agent implements and extends the knowledge intensive case-based reasoning method CREEK [Aamodt, 2004], which is responsible for assessing occurring situations through context awareness. This is done by constructing an unsolved case where the findings are the context received from the Context Middleware. This case is then matched to existing cases and the best matching case contains the goal for this particular type of situation. This goal is then transmitted to the Decomposer Agent, which decomposes according to the existing artefacts and persons described in the context. Each of these entities are described by an Application Agent that supplies information. These two types of agents are together responsible for executing suitable system behaviour by being context sensitive.

The knowledge model utilised in this system is structured as a multi-relational semantic network, where each of the five different parts are integrated. Three parts of the knowledge model are universal for any CREEK application: the top-level ontology, called ISOPOD; the domain-specific model of general and factual knowledge; and the case base. In addition to these, two parts have been developed for the ambient intelligent system, namely the Basic Context Model and the Activity Theory Model.

The Basic Context Model is structured around a meronomy based on tradition in pervasive computing [Kofod-Petersen and Mikalsen, 2005b; Göker and Myrhaug, 2002], which imposes a structure that facilitates easy development of context sensitive applications [Kofod-Petersen and

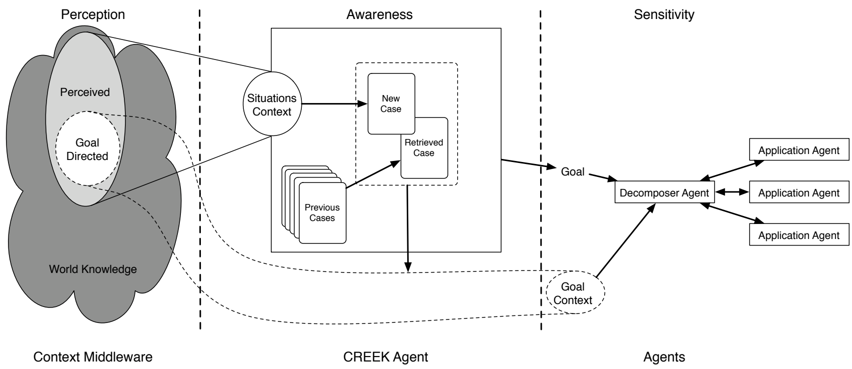

Figure J.1.: Functional System Architecture

Mikalsen, 2005a]. The Activity Theory Model captures knowledge regarding activities, which is one of the most important aspect of situations [Cassens and Kofod-Petersen, 2006].

J.4 Facets of Explanation and Context

As outlined in Section J.2, the term explanation can have different focii in artificial intelligence. In his introduction to the ECCBR-04 Workshop on Explanation in case-based reasoning, David Leake [2004] has identified three different facets of explanation and case-based reasoning:

- Using explanations to support the case-based reasoning process
- Generating explanations by case-based reasoning
- Using cases for explaining system results to an external user

With our notion of user goals, we can subsume the last two facets as both being targeted towards the user of the system. In our understanding, showing the case to the user is a special case of 'generating explanations by case-based reasoning', making use of the case-based reasoning assumption that similar problems have similar solutions. Provided that the user has some knowledge about the similarity function and that the case structure is understandable by the user, the displayed case acts as an explanation to the user [see e.g. Sørmo et al., 2005; Cunningham et al., 2003]. We are left with two functions of an explanation, as described in Aamodt [1991]: first, enhancing and promoting the reasoning process.

Table J.1.: Explanations

	Context Awareness	**Context Sensitivity**
System Centric	Generate an explanation to **recognise** the situation	**Identify** the behaviour the system should expose
User Centric	**Elucidate** why the system identifies a particular situation	**Explicate** why a certain behaviour was chosen

Second, delivering some knowledge about the reasoning process, its results, or implication to the user. We call the first aspect the *system centric view* on explanation and the second one the *user centric view* on explanation:

- Explanation as **part of the reasoning process** itself.
  Example: a knowledge intensive case-based reasoning system can use its domain knowledge to explain the absence/variation of feature values.

- Giving explanations of the found solution, its application, or the reasoning process **to the user**.
  Example: in an engine failure diagnosis system, the user gets an explanation on why a particular case was matched.

When we now look at the use of context, we can again identify different views on context during the reasoning process and the use of the system. First, the system has to identify an unknown situation out of its internal state and the percieved state of the world. Here the system has to find out which situation it is in. Second, the system has to act according to the percieved state of the world and an assumed context. Following our earlier argumentation introduced in Kofod-Petersen and Aamodt [2006], we identify these two aspects as two distinct steps in the reasoning process:

- **Context Awareness:** Trying to figure out which situation the system is in.
  Example: An ambient intelligent system for supporting health personnel recognises that the user is on a ward-round because of the time of the day, the location, and the other persons present.

- **Context Sensitivity:** Acting according to the situation the system thinks it is in.

> *Example:* the same system fetches the newest versions of electronic patient records of all patients in the room from the hospital systems. When the user stands close to the bed of a patient, the system automatically displays them.

Combing these views on explanation and on context, we end up with two dimensions of inquiry as depicted in Table J.1.

J.5 Using Explanations in an Ambient Intelligent System

We will further investigate the relationship between context and explanations by examining an example from our current implementation. At this time, our system is capable of assessing ongoing situations in a simulated hospital (cardiology) ward domain. The system's main purpose is to identify ongoing situations and proactively acquire digital information required by the persons present.

Recognise:

In this step, we are *using explanations to recognise the current ongoing situation*. The system uses all available resources in its reasoning process. Let us assume that the ongoing situation is a *ward round*. Normally ward rounds take place in a patient room, however the current situation is occurring in the hallway. This discrepancy can be explained away by the system generalising that both locations can indeed contain patient beds. When CREEK retrieves a matching case, the system has no explicit knowledge stating that a hallway can contain hospital beds. The initial match is of a syntactical nature only. However, it can use its general knowledge and the reasoning mechanism of plausible inheritance to generate an explanation supporting the hypothesis that beds can be located in the hallway, for example because they are both some kinds of rooms, and beds are some kinds of objects which have a room as a location. Therefore, as all other parameters are consistent with a ward round, the system assumes that it is indeed a ward round situation.

The explanation used by the system in this example states that a hallway is a room and can therefore contain a hospital bed.

Elucidate:

We now want to *generate an explanation for the user that tells the user why the system assumes a certain situation*. The system will make use of all available sources of knowledge in order to gain the user's confidence in its capabilities. It will also have to consider the user's goals when chosing a specific explanation. It has been shown that simply presenting the reasoning trace is not always sufficient (it can even be counter productive) [Majchrzak and Gasser, 1991; Gregor and Benbasat, 1999]. The system might therefore generate an after-the-fact explanation, which for example justifies its assumption instead. Since the ward round situation is occurring at an unusual place, CREEK will point out the time of the day, the availability of the other expected participants, and the fact that hallways might contain beds, as the reason for its assumption instead of only displaying its generalisation of the location.

The explanation shown to the user is a justification of the system's believe of being on a ward round.

Identify:

After the system has successfully identified the context, it is *using explanations to generate a plan for a reasonable course of action*. Now, it is using only the knowledge sources important for the situation at hand (the context is acting as a focus lense [Kofod-Petersen and Aamodt, 2006]). When we now presume that the system has recognised that we are on a ward round, discussing medical conditions and treatments with several patients, it will try to prepare all the relevant information to be presented to the user. This includes all test results. The system can now ask other available artefacts for test results on the user, and the medical images database can offer a MR image whereas the patient record offers a textual description of the MRI. Because of limitations of handheld devices, the system will for example not be able to display high resolution MR images. When choosing which of the artefacts to query, the system will reject the medical image database and only query the electronic patient record database.

The explanation used by the system is based on the knowledge that a high resolution image displaying device is not available on a ward round.

Explicate:

Looking at the user centric part again, we are now in need of *generating an explanation for why the systems takes a specific action*. The system will take into account which situation it assumes it is in and the possible goals the user might have for an explanation. In executing its plan, the system proposes its user to visit the isolation room with patients which should be kept seperate. The user is surprised since he is not aware that any of the patient he should see at the ward round is in the isolation room, and no information on this was exchanged in the morning briefing. The system can then generate an explanation that shows the relevance of the proposal by pointing out that one particular patient had to be moved to the isolation room for medical reasons since the time of the morning meeting, and this information was available via the patient information system. This explanation would not be useful if it had not been established already that we are on a ward round and the aim was to visit the patients. Vice versa, if the system would generate a justification for its assumption of being on a ward round, this would still not satisfy the need of the user to know why he should go to the isolation room.

The explanation shown to the user is pointing out the relevance of performing a particular action, namely visiting the isolation room.

J.6 Summary and Future Work

This paper has presented an approach to combine explanations and case-based reasoning in context awareness. It has been argued that explanations can be viewed from both a system centric and user centric perspective. It has further been described how explanations play a key role in ambient intelligent systems, where traditional interaction has been changed, and division of responsibility has been shifted from user initiated to mixed initiative.

The current implementation does support some of the goals seen from an user centric perspective; that is *elucidating* why it assumes that a given situation is occurring. The *transparency* goal is satisfied by CREEK's ability to graphically represent the case-match and the underlying semantic network representation, as well as a textual representation of the overall match strength and individual feature matches. A textual representation of explanations used in the reasoning process is also shown to the user.

The *conceptualization* goal is supported by providing a means to the user to explore the relation of the case to the underlying knowledge base. By examining the relations of the case features to this underlying knowledge structure, the user can gain some insight into the conceptual model the system has of the domain at hand.

The *justification* goal is to some degree supported in our current implementation. CREEK does explain why two different features are similar by means of plausible inheritance [Sørmo, 2000].

The most important future development is to apply the facets of explanation to the context sensitive part of our system. Here the Decomposer and Application agents must be able to use a system centric perspective on identifying a suitable behaviour in a given situation; as well as explicating why a given behaviour was chosen to the user.

Another direction for future research we want to explore is to tie our findings back in to our earlier work on a design methodology for explanation-aware intelligent systems with a socio-technical perspective. Our research has shown that ambient intelligent systems can benefit from a psychologically plausible knowledge model, but we have not yet explored the relation to the different knowledge containers in detail.

Additionally, we want to augment existing design guidelines with methods for the analysis of social aspects that can lead to a better understanding than ad-hoc methods. It is also important to note that we have not yet fully utilised some aspects of our theoretical foundations in organisational psychology, like the notions of breakdown situations or functional organs in Activity Theory, or the use of Semiotics for the organisation of the user interface itself.

Bibliography

Agnar Aamodt. Knowledge-intensive case-based reasoning in CREEK. In Funk and Calero [2004], pages 1–15.

Agnar Aamodt. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. PhD thesis, Norwegian Institute of Technology, Department of Computer Science, Trondheim, May 1991. URL http://www.idi.ntnu.no/grupper/su/publ/phd/aamodt-thesis.pdf.

Kevin D. Ashley and Derek Bridge, editors. *Case-Based Reasoning Research and Development: Proceedings ICCBR 2003*, number 2689 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2003. Springer.

Romain Bénard, Cyril Bossard, and Pierre De Loor. Context's modelling for participative simulation. In Sutcliffe and Goebel [2006], pages 613–618. ISBN 978-1-57735-261-7. URL http://www.aaai.org/Library/FLAIRS/flairs06contents.php.

Jörg Cassens and Anders Kofod-Petersen. Using activity theory to model context awareness: a qualitative case study. In Sutcliffe and Goebel [2006], pages 619–624. ISBN 978-1-57735-261-7. URL http://www.aaai.org/Library/FLAIRS/flairs06contents.php.

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An evaluation of the usefulness of case-based reasoning explanation. In Ashley and Bridge [2003], pages 122–130.

Belén Díaz-Agudo and Pedro González-Calero. An architecture for knowledge-intensive CBR systems. In *Advances in Case-Based Reasoning:*

Proceedings EWCBR 2000, number 1898 in Lecture Notes in Artificial Intelligence, Berlin, 2000. Springer.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Peter Funk and Pedro A. González Calero, editors. *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2004. Springer.

Ayse Göker and Hans Inge Myrhaug. User context and personalisation. In *Workshop proceedings for the sixth European Conference on Case Based Reasoning*, 2002.

Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, 23 (4):497–530, 1999.

Anders Kofod-Petersen and Agnar Aamodt. Case-based situation assessment in a mobile context-aware system. In Antonio Krüger and Rainer Malaka, editors, *Artificial Intelligence in Mobile Systems (AIMS 2003)*, pages 41–49. Universität des Saarlandes, October 2003.

Anders Kofod-Petersen and Agnar Aamodt. Contextualised ambient intelligence through case-based reasoning. In Thomas R. Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *Proceedings of the Eighth European Conference on Case-Based Reasoning (ECCBR 2006)*, volume 4106 of *Lecture Notes in Computer Science*, pages 211–225, Berlin, September 2006. Springer.

Anders Kofod-Petersen and Jörg Cassens. Using activity theory to model context awareness. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers*, volume 3946 of *Lecture Notes in Computer Science*, pages 1–17, Edinburgh, 2006. Springer. ISBN 3-540-33587-0. URL http://dx.doi.org/10.1007/11740674_1.

Anders Kofod-Petersen and Marius Mikalsen. An architecture supporting implementation of context-aware services. In Patrik Floréen, Greger Lindén, Tiina Niklander, and Kimmo Raatikainen, editors, *Workshop on Context Awareness for Proactive Systems (CAPS 2005), Helsinki, Finland*, pages 31–42. HIIT Publications, June 2005a. ISBN 952-10-2518-2.

Anders Kofod-Petersen and Marius Mikalsen. Context: Representation and reasoning – representing and reasoning about context in a mobile environment. *Revue d'Intelligence Artificielle*, 19(3):479–498, 2005b.

David B. Leake. Workshop introduction: ECCBR-04 workshop on explanation in CBR. Slides, last access 22.02.2007, 2004. URL `https://www.cs.tcd.ie/research_groups/mlg/ecbrws2004/`.

Tinghuai Ma, Yong-Deak Kim, Qiang Ma, Meili Tang, and Weican Zhou. Context-aware implementation based on cbr for smart home. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob 2005)*, pages 112–115. IEEE, IEEE Computer Society, August 2005.

Ann Majchrzak and Les Gasser. On using artificial intelligence to integrate the design of organizational and process change in us manufacturing. *AI and Society*, 5:321–338, 1991.

Stephen Marsh. Exploring the socially adept agent. In *Proceedings of the First International Workshop on Decentralized Intelligent Multi Agent Systems (DIMAS 1995)*, pages 301–308, 1995.

Wolter Pieters. Free will and intelligent machines. Project Report, NTNU Trondheim, 2001.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Funk and Calero [2004], pages 389–403.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, pages 451–464, Chicago, 2005. Springer. URL `http://dx.doi.org/10.1007/11536406_35`.

Roger C. Schank. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, Cambridge, 1983.

Roger C. Schank. *Explanation Patterns – Understanding Mechanically and Creatively*. Lawrence Erlbaum, New York, 1986.

Frode Sørmo. Plausible inheritance semantic network inference for case-based reasoning. M.sc. thesis, Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science, Trondheim, 2000.

Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in case-based reasoning – perspectives and goals. *Artificial Intelligence Review*, 24(2):109–143, October 2005. ISSN 0269-2821. URL `http://dx.doi.org/10.1007/s10462-005-4607-7`.

Geoff C. J. Sutcliffe and Randy G. Goebel, editors. *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference*, Melbourne Beach, 2006. AAAI Press. ISBN 978-1-57735-261-7. URL `http://www.aaai.org/Library/FLAIRS/flairs06contents.php`.

Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.

Andreas Zimmermann. Context-awareness in user modelling: Requirements analysis for a case-based reasoning application. In Ashley and Bridge [2003], pages 718–732.

Making Use of Abstract Concepts – Systemic-Functional Linguistics and Ambient Intelligence

K

Authors:
Jörg Cassens and Rebekah Wegener

Abstract:
An operational model of context is particularly important for the successful integration of new technical artefacts into complex processes. One of the challenges for ambient intelligence is to embed technical artefacts into human work processes in such a way that they support the sense making processes of human actors instead of placing new burdens upon them. This paper examines some of the strengths and current limitations in a systemic functional model of context. We propose that the dimensions that are relevant to modelling are those that have the most consequence for meaning, particularly how the participants involved in a social process construe that social process.

Main Result:
Semiotics is one of the three views on intelligent systems in workplace environments introduced earlier (see Paper D). In this paper, we make use of a specific semiotic theory, systemic functional linguistics, and show its use in ambient intelligence. We focus hereby on abstract concepts.

Accepted for:
IFIP AI 2008 – The Second IFIP International Conference on Artificial Intelligence in Theory and Practice. 7-10 September, 2008. Milan, Italy.

My main contributions to the paper:

- Background on ambient intelligence
- General remarks on semiotics in computing

The following aspects were jointly developed by the authors:

- Discussion of the relations between semiotics and ambient intelligence

K.1 Introduction

If we want an artefact to be considered intelligent, it must exhibit intelligent behaviour. What we generally refer to when we say this is behaviour that is contextually appropriate. An ability to accurately read context is important for any animal if it is to survive, but it is especially important to social animals and of these perhaps humans have made the most out of being able to read context, where such an ability is tightly linked to reasoning and cognition [Leake, 1995].

The necessity of exhibiting some kind of intelligent behaviour has lead to the developments jointly labelled as *ambient intelligence* [Ducatel et al., 2001]. But to successfully create intelligent artefacts, the socio-technical processes and their changes through the use of mediating artefacts have to be examined more closely. This paper focuses on how a social-semiotic theory of language, in which context is seen as integral to understanding communication, can be usefully employed to create ambient intelligence both in architectural aspects and in intelligent response aspects. Ambient intelligence and its requirements from semiotics is further discussed in section K.2 below.

Not only do the artefacts present themselves through semiosis, but the process of creating and developing these artefacts is a semiotic process. Semiotics, as we interpret it in this paper through a Hallidayian model (see section K.3), can be helpful not just in understanding the use of the artefacts and their role in an overall sense making process, but also in understanding and modelling the creation process. The relationship between semiotics and ambient intelligence is outlined in section K.4 below. A semiotic analysis before beginning to develop artefacts can provide a rich description of the environment in which devices will be embedded, and this will ultimately lead to better devices, not only in the immediate instance, but for future devices as well. In this paper we discuss one particular ways in which semiotics can be useful, namely in defining abstract concepts, see in section K.5.

We conclude this paper by pointing to future work in this area. While we have focused on devices designed to interact closely with a single user, humans typically interact in groups, so it will be necessary to consider the impact of this for environments where not all users share the same meaning system.

K.2 Ambient Intelligence

In understanding human cognition and reasoning, disciplines such as neuroscience, psychology, sociology, linguistics, and philosophy have had to take a stance on context as a concept. Setting aside the more mechanistic views taken on reasoning, which typically need not consider context at all, positions on context tend to fall into two broad domains: those who see context as vast and unable to be coded and those who see it as vast but able to be coded. This divides roughly along the same lines as the relativism debate: those who believe in an ultimate reality and those who believe reality is relative. For most this debate can remain largely theoretical impinging little on day to day research, for the field of artificial intelligence however, this debate has very real consequences. Because of the need to study reasoning in the real world, ambient intelligence has, like fields such as anthropology, been forced to work with context however underelaborated the models.

At the core of any ambient intelligent system lies its ability to take account of its environment, be aware of persons in this environment, and respond intelligently to the persons needs and actions. Several authors have identified three main aspects to realise the abilities of an ambient intelligent system [for example Kofod-Petersen and Aamodt, 2006; Yau et al., 2002]: first, the initial act of *perceiving* the world that the system inhabits; second, being aware of the environment and reasoning about ongoing situations, (traditionally labelled *context awareness*); and third, exhibit appropriate behaviour in ongoing situations (often called being *context sensitive*).

The idea that ambient intelligent systems have to exhibit awareness makes modelling context a genuine artificial intelligence problem. The artefact to be designed has to display some kind of behaviour which makes it possible for human users to subscribe to its reasoning behaviour as being the behaviour of an intelligent actant.

We consider that, besides the state changing activity, communication and sense making processes occurring belong to the most consequential aspects of an ongoing situation, and therefore the context. In this paper, we are considering how intelligent devices can be integrated in the overall sense making process during these activities. That is, we consider the communication processes between the different actors involved, be they human or artificial.

Intelligent computing devices, as additional actants, are construed against the backdrop of an existing social context. At the same time, like human actors, they bring their own history and abilities into this social context, thus reconstruing the whole socio-technical process. If intelligent devices are to be useful in a given social context, we have to understand the interdependencies of these relations, and the ways in which intelligent devices can change and be integrated into the existing communication processes.

Since these devices will serve specific purposes, they will not simply observe, but will have to actively interact with other actants. The behaviour of the artefacts will change the situation, and these changes have to be meaningful and useful for the human actants if the integration of the artefacts is to be successful. Therefore, we argue that for an ambient intelligent system to function, it must be able to reason about its own, as well as other's ongoing activities and communications. Ambient intelligence requires more than mere reactive systems. Deliberation and reasoning must play an important part, and this means understanding meaning making systems and how they are utilised in context.

K.3 Semiotics

Semiotics, or the study of sign systems, "has a past which acts on its present and its future" [Hodge and Kress, 1988]. It is not our intention in this paper to review the body of work surrounding semiotics though we are mindful of the impact of this work on the field today, in particular the work of Saussure [1966], Peirce [1904] and Voloshinov [1973]. For a comprehensive account of semiotics as it is applied to computing we recommend works such as Gudwin and Queiroz [2006] (in particular Andersen and Brynskov [2006] and Clarke et al. [2006]) as well as de Souza [2005]. The intelligent artefacts that we consider in this paper are an integral part of social interaction. They change the sense making process on the side of the human users as well as their own functioning as signs (contextualised by the users). Ideally, the artefact should be able to adapt to its use and user, and the means for this adaptation will have to be laid out by the designers.

In this research, we have used the social semiotics outlined by Halliday (see for example Halliday [1978] and Halliday and Matthiessen [2004]). Halliday combines the strengths of the approaches of Saussure, Pierce, and Voloshinov. He brings together the tradition of relational thinking from Saussure, the understanding that different modalities have consequences for the structure of meanings from Pierce, and from Voloshinov, the insistence that the sign is social.

Halliday's Systemic Functional Theory of language (SFL) is a social semiotic theory that sets out from the assumption that humans are social beings that are inclined to interact [Halliday, 1978]. In this paper we examine the value of the SFL notion of context, which views context as all the features of a social process relevant to meaning making. These features are organised into 3 core parameters of context: Field, Tenor and Mode, where **field** is *"the nature of the social activity..."*, **tenor** is *"the na-*

*ture of social relations...", and **mode** is "the nature of contact..."* [Hasan, 1999]. Context, in SFL is one of four linguistic levels (see below), which are related realizationally rather than causally, meaning that patterns on one level both construe and construct patterns on another level. Halliday manages the complexity of language by modelling it as a multidimensional system. The most crucial dimensions of this multidimensional system for our purposes are: stratification and instantiation. We examine how these key notions of SFL make this model of context valuable for AI.

Stratification: Halliday uses a stratified model of language that incorporates the levels of the expression plane (including sound systems - phonetics and phonology, gesture, pixels etc), lexicogrammar (lexis/grammar - or wording and structure), semantics (the meaning system) and context (culture and situation - elements of the social structure as they pertain to meaning). Description on each stratum is functionally organised into systems. All levels can be represented as networks of options with the networks rendering any degree of complexity by combining 5 primitives:

- **or:** option between X or Y
- **and:** option between X and Y
- **only if:** only if x and y
- **both:** both X and Y
- **iteration:** re-enter the system and choose over.

By building in values for probabilities we arrive at a weighted description that is customised to the 'typical-actual' of a given situation type (or register, see below). Individual situations, roles, or participants can be profiled by their pathways through the networks and/or by the ensemble of options across the levels which are most typically invoked [Halliday and Matthiessen, 2004].

Instantiation: Halliday uses a tripartite representation of language, which has language as system, language as behaviour and language as knowledge. Language as system encapsulates the abstract structure of language. This accounts for the regularised (though changeable) patternings that we see in language. It is this regularity that makes prediction and a certain degree of formalism (at least of a functional nature) possible. Language as behaviour looks at the activity of language, while language as knowledge looks at the way in which we know language. But we do not do these things independently. We do not know language as a set of abstract rules. Rather we know language in the sense of knowing how to

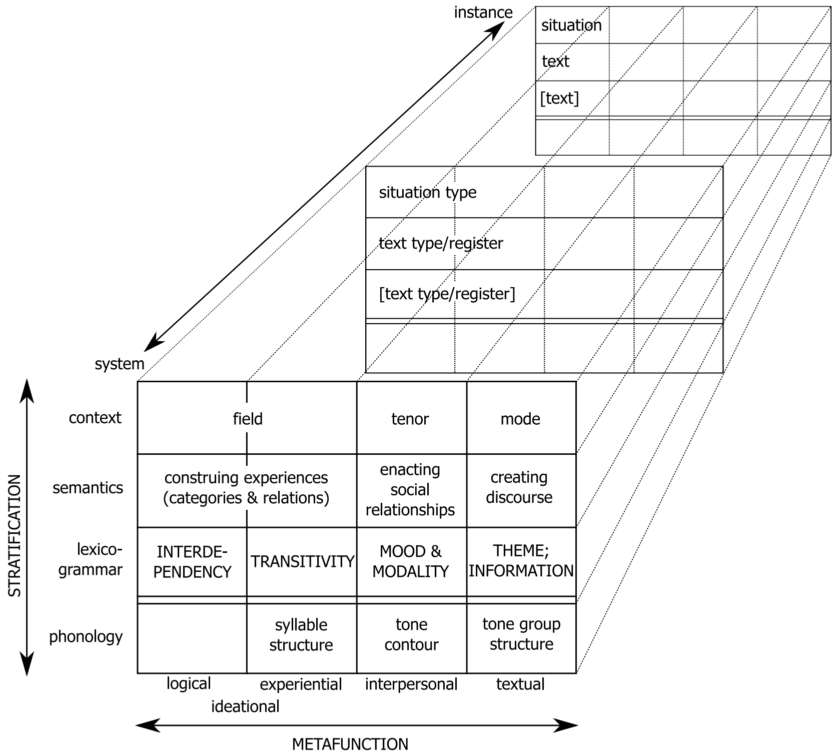

Figure K.1.: The dimensions of language - Halliday and Matthiessen

use it, in the sense of knowing how to communicate with others [Halliday, 1978]. In practice these things occur together. When we try to build a device, it is language behaviour and knowledge that we face, yet it is the seemingly inaccessible system that we need to encode in order to produce intelligent seeming behaviours and knowledge in the device.

The concept that encapsulates this problem is what Halliday calls the cline of instantiation. This is a way of looking at the relationship between System (which at the level of context means the culture) and Instance (which at the level of context means the situation that we are in). This is represented in figure K.1. Here we see in the foreground the system view of language, and its grounding in the instance.

Instances that share a similar function, e.g. instances of ward rounds in hospitals, typically share a similar structure. Halliday refers to these situation types as registers and they represent a functional variety of language [Halliday and Matthiessen, 2004]. The value of register is that we do not have to describe everything. Register can be thought of as an aperture on the culture. So, we are not faced with the full complexity of the culture. This does not mean that we do not keep the culture in mind. Any picture of a part of the system necessarily has the full system behind it. With register we set out from the instance, but keep in mind that each instance is a take on the system. Our notion of what constitutes an instance is shaped by our understanding of the culture/system. So, although Halliday represents the relationship between system and instance as a cline of instantiation, it is probably best understood as a dialectic since the two are never actually possible without each other. Register does not so much sit between system and instance, as it is a take on system and instance at the one time. It is the culture brought to bear on the instance of the social process.

For ambient intelligence, this means that we are not faced with the unhelpful uniqueness of each instance, because we are viewing it through the system and therefore foregrounding the shared aspects. Neither are we confronted with the seemingly impossible task of transcribing the infinity of culture, because we are viewing the culture through the aperture of the instance.

K.4 Semiotics in Ambient Intelligence

Interaction is a process of exchanging and interpreting symbols referring to objects. The user of a computer systems sees his interaction with the system against this background. When typing a letter, a user does not send mere symbols, but signs to the computer, and the feedback from the machine, the pixels on the screen, are interpreted as signs: to the user, the computer is a semiotic machine. The question that arises is whether a computer is actually itself taking part in the sense making process.

If we follow the lead of philosophers such as Kant, human understanding has as a necessary constituent the ability to conceptualise perceived phenomena through an active, discursive process of making sense of the intuitive perception [Kant, 1787, p. 58]. This poses problems for computer systems which are only processing signals, lacking the necessary interpreting capabilities humans have. They only manipulate symbols without conceptualising them. However, intelligence is in the eye of the beholder, and it can be argued that even mere signal processing units can appear as sign processors to the human if they mimic sufficiently human behaviour.

The pragmaticist approach, by contrast, following for example Peirce and Dewey, avoids this question altogether by focusing not on whether the machine is itself a sense maker, but on how its use changes the ongoing socio-technical process, and whether it can mediate the sense making process. From this point of view, the computer can be a sense making agent if its actions are appropriate in terms of the user's expectations.

Both approaches lead to a change in the issues we deal with when constructing an ambient intelligent system. The problem is transformed from one where the issue is to build a machine which itself realizes a sense making process to one in which the issue is to build a computer that displays actions appropriate to the context it is in and that exhibits sufficient sign processing behaviour.

We argue that, in order to make an ambient intelligent system that behaves intelligently in a context, it must be able to execute actions that make a difference to the overall sense making process in a given context.[1]

One important challenge here is the features that allow the system to display its abilities. This can be described as a communication problem: the system has to interpret the actions of the user and perceived contextual information in a meaningful way and itself present results that make sense for the user. This process of sense-making is highly interactive: an intelligent partner in a communication process asks (meaningful) questions if an unclear situation occurs and is able to explain its own actions. Therefore, it is desirable that the artefacts mimic some abilities usually ascribed to humans (e.g. explanatory capabilities). Contextually appropriate explanatory abilities are essentially a meaning making phenomenon. This means that semiotics should be well positioned to assist in understanding how such abilities can be introduced to artefacts.

K.5 Abstract Concepts

Abstraction, or the ability to create a more general category from a set of specifics by whatever principle is arguably one of the most useful mental tools that humans possess [Butt, 2006]. Indeed Whorf [1956] suggests that the abstract categories that form part of our everyday life and language, are typically below conscious attention and only become apparent through linguistic analysis.

In the hospital environment, 'emergency' has a specific meaning that is distinct from the meaning in other contexts. Not only is there a hospital specific meaning (culture specific), but the meaning varies according

[1] Which differs from the interaction with traditional systems in which case the sense-making falls wholly on the side of the human user: You do not expect a text processor to understand your letter, but you expect an ambient intelligent system to display behaviour as if it understands relevant parts of the context you are in.

to the situation as well (situation specific). To function intelligently in context, artefacts must be able to recognise 'emergency' and respond appropriately. They may need, for example, to "be quiet" while the doctor deals with an 'emergency' or they may need to "provide new information" needed by the doctor in an 'emergency'.

To account for these complexities, a rich, but targeted, description of the culture is needed. To do this we will use the notions of register and generic structure potential [Hasan, 1994] and a contextual model of language. In order to establish what emergency means in this context we need to see its place in the system. That means we need to understand how it fits in the hospital culture. Understanding the richness of the culture is part of adequately embedding a device into that culture. Not doing so runs the risk of producing an artefact unsuited to its purpose and thus unintelligent. Part of what makes something (appear) intelligent is the ability to read and respond to the context. Context here is not just the immediate setting of the artefacts, or the context of situation, but the culture of which that setting is a part.

Consider the meaning of 'emergency' for a ward round. The notion of a ward round is itself a functional abstraction of all the behaviours, relations, and communications that go into completing a ward round. We are able to recognise from experience that certain behaviours by different participants, combined with certain roles and relations (e.g. ward doctor, ward nurse, patient, specialist) combined with the exchange of certain types of information (receiving information, requesting information, giving information) together constitute a ward round. None of these behaviours, relations or communications on their own constitutes a ward round, indeed, they are each necessary parts of other hospital functions as well. By studying many instances we arrive at a 'typical', or a generic structure potential for a ward round. This does not mean that there will not be variation. This perspective on the context is a necessary part of understanding what a ward round is, but it is not the only perspective that is necessary. Ward rounds must also be seen from the perspective of how they fit into the hospital culture. Ward rounds are a part of the function of the hospital, which, can be said to be restoration of health. In order to make artefacts capable of dealing with change, it is necessary to consider the culture in which ward rounds are embedded. the function ward rounds perform in the hospital culture is to monitor health. Because it has a 'monitoring' function within the hospital culture, the ward round will be able to be interrupted by 'emergencies' from the wider hospital.

By building up a picture not only of what a ward round is, but also of how it fits into the broader hospital culture, we are better able to see its function, and thus what the meaning of 'emergency' is likely to be in this situation. There are two broad categories of emergency: those constituting an interruption to the ward round (when the hospital culture impinges on the ward round) and those constituting a change to the ward round (when there is internal variation in the ward round context). Because the first involves changes to the field (a new topic, ward, and focus), tenor (very different participants and role relations), it is likely to require a 'new information response'. The second, will not involve changes to the field or tenor, or at least not major changes and so is likely to require a 'be quiet and await query' response. By utilising the notion of register to limit what we have to consider in the culture, and the concept of generic structure potential to model a typical view of the situation based on our study of the instances, we are able to better understand the context of the ward round and how to model abstract concepts for this context.

K.6 Conclusion and further work

In this paper we have considerered one of several ways that semiotics can be made fruitful in ambient intelligence. We have discussed how the notion of abstract concepts and the use of the analytical tool of register can help us to avoid an Althusserian trap of the last instance [Althusser, 1962] in our modelling efforts.

Systemic Functional Linguistics offers a unified approach to many of the issues in ambient intelligence. If we utilise this approach to semiotics, then we are in a position to draw on the work that is being done for other projects where it is relevant to the context that we study. For example, there is a lot of work being done around the world on hospital environments for different purposes. When this work is carried out using SFL, we can generalise the findings to our own domain. This significantly reduces the work load and labour cost, and still provides a rich description of the context. This is only possible because we consider what is shared between contexts while keeping in mind that each instance is a unique take on the system.

This research has suggested many areas of future investigation. In this project we have focused on the individual, but the sign making process is a negotiated process. It is not simply one meaner that has to be considered. In any exchange there are always at least two meaners, and more typically more than two. Multiparticipant communication represents a challenge to modelling. We have to keep in mind that others may share our conceptualisations and meanings only to a certain extent. When ambient intelligent systems link different people this is an important thing to remember. The closer a person is in our social network the more likely they are to share our meanings, while the further out in our social network the less likely they are to share meanings. In the hospital environment, ambient intelligent devices can belong to different groups of users. Should we model them in a way that the assistant of a nurse is more likely to share concepts with the assistant of another nurse than that of a physician?

Ambient intelligent systems will have deal with these kinds of challenges. Another point to consider is where in the network the system itself sits. What is the relation of the system to its user? To other pervasive devices? To their users? We are effectively dealing with a case of dialectal variation. Certain users may find some signs transparent and others not, while other users may find the exact opposite. If ambient intelligent systems are used to link people how do they best utilise signs to do this? This issue becomes very important when health care professionals from different cultural and language backgrounds have to interact.

Another issue we would like to explore further is the extent to which it is possible to relate a semiotic approach to ambient intelligent systems design to other socio-technical theories already in use in the field of ambient intelligence. A promising candidate is for example activity theory. Bødker and Andersen [2005] have outlined some properties of a socio-technical approach taking advantage of ideas from both theoretical frameworks, and we would like to extend this to cover specific aspects of SFL and Cultural-Historical Activity Theory (CHAT). This will potentially extend the number of projects from which we can borrow findings, meaning a richer description of the hospital environment.

Another point we have not fully explored yet is the relation of concepts from SFL with specific methods from the field of artificial intelligence. For example, the notion of genres in SFL seems to be a likely candidate for knowledge poor lazy learning mechanisms, while the descriptive power of the register might be exploitable in knowledge intensive or ontology based approaches. A promising candidate to combine these aspects is knowledge-intensive case-based reasoning.

Bibliography

Louis Althusser. Contradiction and overdetermination. In *For Marx*. Penguin Press, 1962. URL http://www.marxists.org/reference/archive/althusser/1962/overdetermination.htm.

Peter Bøgh Andersen and Martin Brynskov. The semiotics of smart appliances and pervasive computing. In *Semiotics and Intelligent Sys-*

tems Development Gudwin and Queiroz [2006], pages 211–255. ISBN 1599040638.

Susanne Bødker and Peter Bøgh Andersen. Complex mediation. *Journal of Human Computer Interaction*, 20(4):353–402, 2005.

David Butt. Multimodal representations of remote past and projected futures: from human origins to genetic counselling. In Masa-Chiyo Amano, editor, *Multimodality: towards the most efficient communications by humans. 21st century COE Program International Conference Series NO. 6*. Graduate school of letters, Nagoya University, 2006.

Rodney Clarke, Aditya Ghose, and Aneesh Krishna. Systemic semiotics as a basis for an agent oriented conceptual modeling methodology. In *Semiotics and Intelligent Systems Development* Gudwin and Queiroz [2006], pages 256–285. ISBN 1599040638.

Clarisse Sieckenius de Souza. *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press, 2005. ISBN 0262042207. QA76.9.H85 .D465 2005.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Ricardo Gudwin and Joao Queiroz. *Semiotics and Intelligent Systems Development*. IGI Publishing, Hershey, PA, USA, 2006. ISBN 1599040638.

Michael A.K. Halliday. *Language as a Social Semiotic: the social interpretation of language and meaning*. University Park Press, 1978.

Michael A.K. Halliday and Christian M.I.M. Matthiessen. *An Introduction to Functional Grammar, Third edition*. Arnold, London, UK, 2004.

Ruqaiya Hasan. Situation and the definition of genre. In Allen Grimshaw, editor, *What's going on here? Complementary Analysis of Professional Talk: volume 2 of the multiple analysis project*. Ablex, Norwood N.J., 1994.

Ruqaiya Hasan. Speaking with reference to context. In Mohsen Ghadessy, editor, *Text and Context in Functional Linguistics*. John Benjamins, Amsterdam, 1999.

Robert Hodge and Gunther Kress. *Social Semiotics*. Cornell, UK, 1988.

Immanuel Kant. *Kritik der reinen Vernunft (2. Auflage)*. Akademie, 1787.

Anders Kofod-Petersen and Agnar Aamodt. Contextualised ambient intelligence through case-based reasoning. In Thomas R. Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *Proceedings of the Eighth European Conference on Case-Based Reasoning (ECCBR 2006)*, volume 4106 of *Lecture Notes in Computer Science*, pages 211–225, Berlin, September 2006. Springer.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

Charles Sanders Peirce. New elements (kaina stoicheia). In Carolyn Eisele, editor, *The New Elements of Mathematics by Charles S. Peirce*, volume 4, Mathematical Philosophy, pages 235–263. 1904. URL http://www.cspeirce.com/menu/library/bycsp/stoicheia/stoicheia.htm.

Ferdinand de Saussure. *Course in General Linguistics*. McGraw-Hill, 1966.

Valentin N. Voloshinov. *Marxism and the Philosophy of language*. Seminar Press, New York, 1973.

Benjamin L. Whorf. *Language, Thought and Reality (ed. J. Carroll)*. MIT Press, Cambridge, MA, 1956.

Stephen S. Yau, Yu Wang, and Fariaz Karim. Development of situation-aware application software for ubiquitous computing environments. In *29^{th} Annual International Computer Software and Applications Conference (COMPSAC'02)*, volume 1, pages 107–112, Edinburgh, Scotland, July 2002. IEEE Computer Society.

Modelling with Problem Frames: The Knowledge Needed for Explanation-Aware Ambient Intelligent Systems

L

Authors:
Jörg Cassens and Anders Kofod-Petersen

Abstract:
Explanations are an important vehicle to convey information in everyday human-human interaction. They enhance the knowledge of the communication partners in a way that furthers acceptance and understanding for statements made or actions taken. Systems in general, and intelligent systems in particular, should also have the capacity of explaining their behaviour to users. To this end, it is important to be able to model the explanatory needs of the human users and connect those to explanatory capabilities of the system. This work looks at problem frames for explanations and investigates how problem frames can be utilised to elicitate, analyse, and specify explanation-specific requirements. These requirements are then coupled to the different ways in which explanations can be used in ambient intelligent systems. The results can help designers in modelling the knowledge needed to support the explanatory capabilities of the system.

Main Result:
Building on the concept of explanation problem frames we have introduced earlier (see Paper H), we further develop problems frames for ambient intelligent and explanation aware systems. We give an example of how to use the newly developed frames in requirements engineering. To this end, we take a step towards the re-design of an existing application from the hospital ward domain and show how explanation patterns can help to model requirements towards the explanatory capabilities of the system.

Submitted to:
ECAI 2008 – The 18th European Conference on Artificial Intelligence. 21-25 July 2008. Patras, Greece.

Copyright:

My main contributions to the paper:

- Discussion of related work
- Example for re-design
- Connection to user-centric/system-centric explanations

The following aspects were jointly developed by the authors:

- Refinements of explanation problem frames

L.1 Introduction

Ambient intelligence describes environments where human beings are surrounded by intelligent artefacts supported by computing and network technology. Such environments augment everyday objects such as furniture and clothes. In addition, an ambient intelligent environment should be aware of the presence of a person, perceive the needs of this person, and respond to them in an unobtrusive and intelligent manner [Ducatel et al., 2001]. Ambient intelligence can be seen as the intersection of pervasive computing, ubiquitous computing, and AI.

The ability to explain itself, it's reasoning and actions, has been identified as one core capability of any intelligent entity [Sørmo et al., 2005]. The question of what is considered to be a good explanation is context dependent [Leake, 1995], leading to the necessity to design the explanatory capabilities of the system together with the modelling of the different situations the ambient intelligent system is likely to encounter.

The work presented in this paper targets the requirements elicitation, analysation, and specification processes. We make use of the notion of problem frames [Jackson, 2001], which appears to be a promising method both in helping to elicit requirements and in later transformation of design documents into actual systems. Cassens and Kofod-Petersen have previously suggested additional problem frames that target explanatory capabilities explicitly [Cassens and Kofod-Petersen, 2007], and building on their work we will demonstrate how problem frames can be put to use in revealing limitations of an existing ambient intelligent systems design and can help to take needs into account arising from explanatory capabilities when (re-) designing such a system.

L.2 Related Work

The use of patterns [Alexander et al., 1977] is common for different software engineering approaches. Patterns can be used in different software development phases and they can have different foci. We can also identify knowledge engineering approaches that make use of patterns for the development of intelligent systems.

There are several methods and languages that use patterns and focus explicitly on the knowledge aspects of system design. For example, the goal of the INRECA [Bergmann et al., 2003] methodology is to support the development of (industrial) case-based reasoning (CBR) applications. Software process models from existing CBR applications are stored in an experience base that is structured at three levels. The *common generic level* is a collection of very generic processes, products, and methods for CBR applications. At the *cookbook level*, we find software models for particular classes of applications (so called recipes). At the *specific project level*, experiences from particular projects are stored. We can identify the recipes at the cookbook level as patterns.

Another well-known approach is the CommonKADS methodology [Schreiber et al., 2000]. It is based on two different views on the development process of knowledge based systems: the *result perspective* encompasses a set of models of different aspects of the knowledge based system and its environment, and the *project management perspective* starts from a spiral life-cycle model that can be adapted to the particular project. The CommonKADS template knowledge model provides a way of (partially) reusing knowledge models in new applications and can be understood as patterns in a software engineering sense.

When we look towards the software engineering world, we can see that patterns are used in different phases of the design process.

On the software architecture level, we find *architecture patterns* [Avgeriou and Zdun, 2005]. At this level, we encounter concepts like 'Blackboards', 'Model-View-Controller', or 'Pipes and Filters'. For finer grained software development close to the actual implementation, one can make use of design patterns that look inside towards the computer and its software [Gamma et al., 1995]. Design patterns deal with concepts like 'Factories', 'Facade', and 'Decorater'.

Early on in the requirements engineering process, *problem frames* [Jackson, 2001] is a method to classify software development problems. Problem frames look out into the world and attempt to describe the problem and its solution in the real world. Problem frames introduce concepts like 'Information Display' and 'Commanded Behaviour'.

Jackson's set of basic problem frames can be extended to be better able to model domain specific aspects. For example, Hatebur and Heisel [2005] introduce new problem frames for security problems. Their proposal includes problem frames for issues like 'Accept Authentication' and 'Secure Data Transmission'. They also provide architectural patterns connected to these problem frames.

Hall and Rapanotti [2005] have introduced extensions to the basic problem frames that will better facilitate socio-technical systems. They introduce a 'user interaction frame', and employ the model-view-controller perspective to ease decomposition. We will build on this results in our own work on ambient intelligent systems as a special class of socio-technical systems where user interaction is not only achieved via explicit communication, but also through the behaviour of both system and user.

L.3 Problem Frames

The main purpose of any problem frame [Jackson, 2001] is to propose a machine that improves the combined performance of itself and its environment by describing the machine's behaviour in a specification. Jackson originally described five different basic frames. In general, a problem frame assumes a user driven perspective. Most basic frames assume that the user is in control and dictates the behaviour of the machine. Since intelligent systems (ideally) take a much more pro-active approach and mixed initiative issues become relevant, new problem frames addressing these topics have to be developed. For the course of this paper, we will focus on frames targeting explanatory aspects and will not discuss other types.

Problem frames can be described by problem frame diagrams. These diagrams consist basically of dashed ovals, representing the requirements, plain rectangles, denoting application domains, and a rectangle with a double vertical stripe, standing for the machine (or software machine) domain to be developed. These entities become the nodes of the frame diagram. They are connected by edges, representing shared phenomena and denoting an interface. Dashed edges refer to requirement references. Dashed arrows designate constraining requirement references.

The domains can be of different types, indicated by a letter in the lower right corner. Here, a 'C' stands for a *causal* domain whose properties include predictable causal relationships among its phenomena. A 'B' denotes a *biddable* domain that lacks positive predictable internal behaviour. Biddable domains are usually associated with user actions. Finally, an 'X' marks a *lexical* domain. Such a domain is a physical representation of data and combines causal and symbolic phenomena.

L.4 Problem Frames and Ambient Intelligence

Following the definition of ambient intelligence [Ducatel et al., 2001], in general an Ambient Intelligent system can be fitted into a *Required Behaviour* problem frame. Fig. L.1 illustrates this. `AmI!C1` is the phenomena shared between the machine and the environment, and controlled by the systems; that is the actuators. The `E!C2` is the phenomena shared between the machine and the environment, which is not controlled by the machine; that is the sensors. Finally, `C3` refers to the behaviour that the machine is to exhibit.

Even though required behaviour frames are generally and by definition suitable for ambient intelligent systems, some special cases exist where explicit user interaction is required other than through *behavioural interfaces*.

Figure L.1.: Ambient Intelligent Systems as a Controlled Behaviour Frame

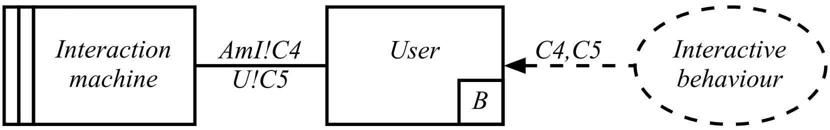

Figure L.2.: User Interaction Frame [adopted from Hall and Rapanotti, 2005]

It has been argued that any adaptive system in general, and an ambient intelligent system in particular, must exhibit the ability to explain its behaviour [Kofod-Petersen and Cassens, 2007]. This ability requires that the system is able to communicate with the user through suitable interfaces, such as displays. In addition, the user should have the option to explicitly request an explanation of the system's behaviour. Thus, a suitable problem frame is required to capture this.

Following the argumentation of Hall and Rapanotti [2005], we employ the *User Interaction Frame*, depicted in Fig. L.2. `AmI!C4` is the symbolic phenomena shared between the machine and the user, where the machine can display information to the user. `U!C5` is the causal shared phenomena between the machine and the user, where the user initiates commands. Finally, `C4,C5` are the rules of conduct between the machine and the user.

Again following Hall and Rapanotti, we can combine these two frames into an *Interactive Ambient Intelligence Frame*, as depicted in Fig. L.3. Here, interactive, explanatory capabilities are combined with the environment controlling aspects of ambient intelligent systems. This aggregation differs significantly from the original *required behaviour frame* [Jackson, 2001]. The behaviour of the ambient intelligent system is not mainly guided by explicit input from the user, but is a result of the pro-activeness of the system and implicit interaction (for example the location of the user). But it opens up for direct interaction, for example by the user requesting an explanation. This will, however, not directly command the whole behaviour

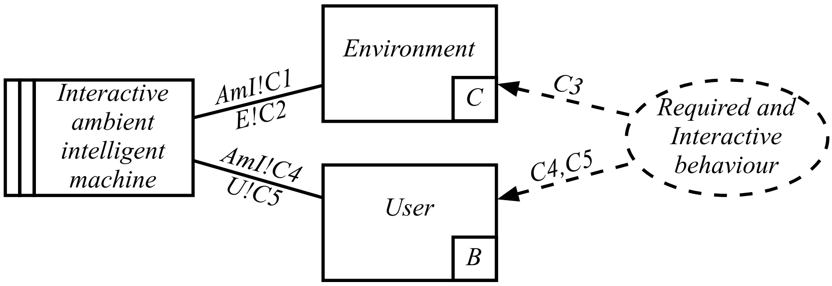

Figure L.3.: Interactive Ambient Intelligence Frame

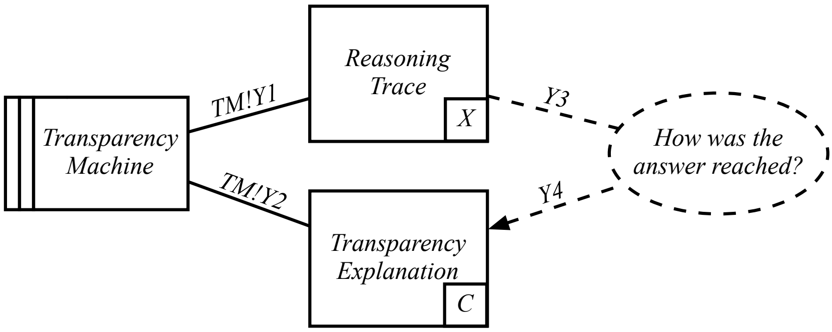

Figure L.4.: Transparency Explanation

of the system, but only a small part of it. In that sense it further on differs from the *user commanded frame* in Hall and Rapanotti [2005] as the system can take actions that are not triggered through commands explicitly issued by the user.

L.4.1 Explanation Problem Frames

Taking a closer look at Fig. L.3, we will see that the upper part captures the behaviour of the ambient intelligent system, whereas the lower part represents the interactive properties of the system. We will use this part of the general frame to model the explanation abilities. To this end, however, we have to decompose the lower part in order to model different types of explanation.

The list of explanation requirements can be described as a list of the

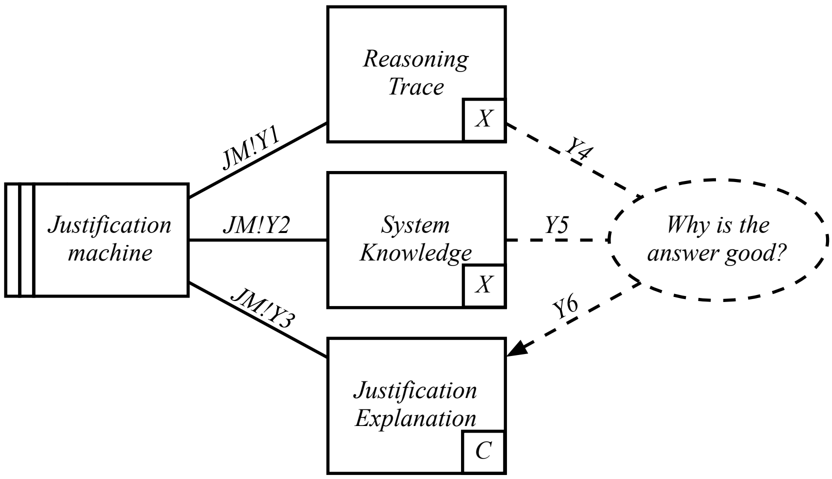

Figure L.5.: Justification Explanation

explanation goals that a system must be able to satisfy. Sørmo et al. identify five different explanations goals that a system might have to handle [Sørmo et al., 2005]. This work has been further expanded in Kofod-Petersen and Cassens [2007], where the explanation goals have been combined with the ambient intelligence paradigm. Our own work focuses on the four goals that are not related to applications as an educational tool: The goal of *transparency* is concerned with the system's ability to explain how an answer was reached. *Justification* deals with the ability to explain why the answer is good. When dealing with the importance of a question asked, *relevance* is the goal that must be satisfied. Finally, *conceptualisation* is the goal that handles the meaning of concepts.

We will briefly describe problem frames for the goals we consider in this paper. Frames for other goals exist as well. The *transparency goal* is concerned with how the system finds a solution to a given problem. This allows the user to inspect the reasoning process to identify the cause of any abnormal behaviour. The transparency explanation frame is depicted in Fig. L.4. Here the `Reasoning Trace` is a lexical domain, which allows the `Transparency Machine` to read the reasoning trace trough the shared phenomena `TM!Y1`. The `Transparency Explanation` is the causal domain, which the machine can control through the shared phenomena `TM!Y2`. In short, the `Transparency Machine` has to inspect the reasoning trace and present the relevant information to its user.

Table L.1.: Context and Explanations

	Context Awareness	**Context Sensitivity**
System Centric	Explanations to **recognise** the situation	**Identify** the desired system behaviour
User Centric	**Elucidate** why a situation was identified	**Explicate** why a certain behaviour was chosen

The *justification goal* is closely related to the *transparency goal*. Justification can be seen as a simplification of the reasoning process that the system actually goes through. The main purpose of this explanation goal is to convince the user that the reasoning is sound. Fig. L.5 displays the problem frame of a justification explanation goal. This frame resembles the one for transparency explanations, with the addition of the lexical domain `System Knowledge`. This domain facilitates the expansion of a transparency explanation by allowing the `Justification Machine` to inspect the system's knowledge through the shared phenomena `JM!Y2`.

L.5 Hospital Ward System

The application in question is an ambient intelligent information system for hospital wards. The system was developed through cooperation with the cardiology ward at an university hospital. Following the definition of ambient intelligence in Ducatel et al. [2001], the system *perceives* its environment, becomes *aware* of ongoing situations, and is *sensitive* to the idiosyncrasies of the particular situations. The main purpose of this system is to perceive the information needs of its user in ongoing situations (such as specific journals, test results, and treatment plans), and pro-actively fetch the required information.

The existing system is build around a multi-agent platform. *Perception* is handled by a Context Middleware, the situation *awareness* is build around a case-based reasoning [Kolodner, 1993] system, and the acquisition of relevant information (*sensitivity*) is facilitated by dynamic task decomposition. Situations were identified and the knowledge model populated through an ethnographical study conducted at the cardiology ward. The whole system was implemented using the Jade [Bellifemine et al., 2003] agent framework.

In our system, we use explanations in two distinct ways: first, enhancing and promoting the reasoning process; called the *system centric* view. Second, delivering some knowledge about the reasoning process, its results, or implication to the user; called the *user centric* view. Table L.1

shows how these two different usages of explanations relate to the *awareness* and *sensitivity* aspects of our system. For the purpose of this paper we will disregard the perception layer of the architecture as the perception layer demonstrates no reasoning capabilities, and only structures perceived data syntactically.

In the setting of our general frame for interactive ambient intelligent systems depicted in Figure L.3, the *system centric* explanations relate the upper part, whereas the *user centric* explanations relate to the lower part. The explanation problem frames for user goals can be put to use in the lower part or *user centric* view, but modified versions reflecting the system's intentions are important for the upper part or *system centric* view as well. For the remainder of this paper, we describe how to make use of explanation problem frames for the explication aspect, describing the user centric and context sensitive use of explanations.

L.5.1 Example

To clarify the functionality of this system, we will present a small example. It sketches an execution taken from a simulated system run, using the real data set gathered at the cardiology ward. In this case we are dealing with a *pre-ward round* situation. A pre-ward round is a particular type of meeting that occurs every morning. Usually, the physician in charge and the nurse in charge are present. They discuss each of their patients, including their current condition, any changes, and the treatment plan.

The Context Middleware monitors the different sensors in the environment, and discovers a change, which provokes a change in the current context. This change in the context is transferred to the CBR sub-system, which retrieves the best matching case based on the sensor values.

In this example, the CBR component retrieves a case describing another pre-ward round. Having identified the ongoing situation as a pre-ward round, the CBR engine now extracts the goal of this type of situation. In this case, the goal is to gather the relevant information. This goal is sent to the sensitivity part to be solved.

The Sensitivity part of this system receives the goal and matches it to a general decomposition tree that contains the tasks required to satisfy the goal. In this example the task tree that matches a pre-ward round goal is as follows:

1. Acquire name of patient.
2. Acquire changes in patient's conditions since yesterday.
3. Acquire any new results from tests.

4. Examine, and possible change, medication scheme.
5. Note changes in treatment.

Each solvable task is matched to an action performed by an available, willing and able agent. The system currently offers 19 different agents, each representing one information system at the hospital ward. Together these 19 agents offers 21 different information services. The initial problem of finding the name of the patient can be facilitated by the *Patient List Agent*. Further on, the 'Acquire Information' task is decomposed into one task that acquires changes which are supplied by the *Electronic Patient Record*, the *WiseW* application and the *Patient Chart*, and another task that acquires results which can be delivered by the *Patient Chart* and the *WiseW* application. This plan is now executed and the information acquired through the different agents is returned to the user; thus ending an execution cycle of this system.

L.6 Redesigning the Existing Application

In the first incarnation, explanatory capabilities were not explicitly included in the design specifications. However, the socio-technical theory used in the study design and application allows us to elicit the possible explanation goals users of the system might have. Therefore, a re-design on the grounds of the data already gathered is feasible.

L.6.1 Example

Revisiting the example of the previous section, we have the instance where the system correctly classifies an ongoing situation as a pre-ward round. If we focus on the context sensitive part of the system, its main purpose is to examine the artefacts, represented by agents, in the environment and find those that can supply relevant information. So far this application only supplies information without any explanation of its behaviour.

In order to demonstrate how the explanation goal problem frames can be used to model explanatory needs in the problem domain, we will start with a simplified problem diagram for our application (Fig. L.6). This is essentially modelling the behaviour of the system without direct user interaction and reflecting the capabilities the existing system was designed to have. This part is a decomposition of the upper part of the *Interactive Ambient Intelligence Frame* from Fig. L.3. We have modified Jackson's *information display* problem frame and used it as a starting point for the diagram. You can see three domains representing (groups of) the agents mentioned above.

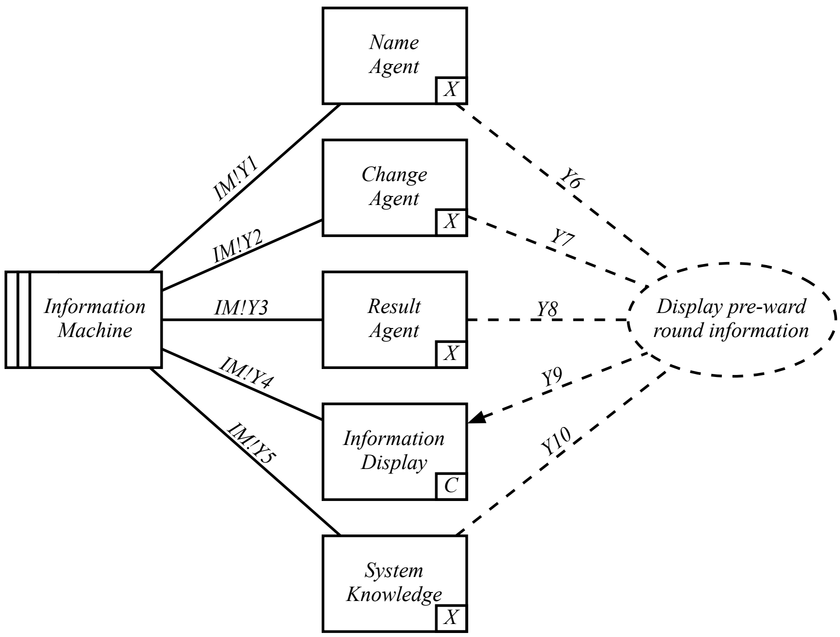

Figure L.6.: Ambient Information Display

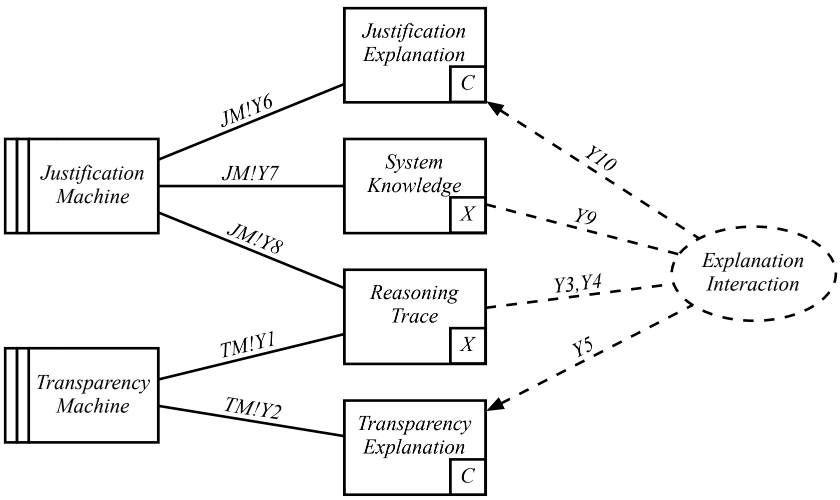

Figure L.7.: Transparency and Justification Explanation

Additionally, you see the `Display` domain which stands for the information display of the system and `System Knowledge` for deciding which data sources to use. For the sake of clarity and simplicity, we have abstracted away the sensor parts of as well as the context aware parts of our example application and focus solely on the information gathering and display parts. Let us now assume that the results displayed by the system are of such a nature that the physician using the system requires an explanation. Let us for the sake of simplicity of the example further focus on a subset of the identified explanation goals.

We want to integrate the explanation sub problems described by the two problem frame diagrams for the *Transparency* and the *Justification* goal to model the explanatory capabilities of the system. This combination is covered in the frame depicted in Fig. L.7. This model is a decomposition of the lower part of the *Interactive Ambient Intelligence Frame* from Fig. L.3.

By integrating Fig. L.6 modelling the ambient intelligence capabilities without explicit user interaction and Fig. L.7 capturing direct user involvement exemplified for explanations, we have a complete (albeit simplified) model of our explanation aware ambient intelligent system. We can now re-visit the example problem described above. The expert user physician might wish to know how the particular combination of information displayed was reached. According to the transparency explanation problem frame, this explanation can be achieved by displaying the reasoning trace.

This can for example be done by showing that the top task 'Pre-ward round' was selected as a function of the classification, by displaying how the decomposition tree looks like, and by supplying information about the agents selected.

For the justification explanation, the novice user physician would like to know why this combination of information is any good. This can be achieved by relating the reasoning trace to the domain model of the system. For example, according to the domain model, the 'Acquire Medication' task could be satisfied not only by the *Patient Chart* but also by the *Electronic Patient Record*. However, as the *Electronic Patient Record* agent was busy serving other requests only the *Patient Chart* could respond to this request.

L.6.2 Analysing the existing application

The results of our ethnographical study are pointing towards the necessity to support four of the five different user goals introduced by Sørmo et al. [2005], namely *transparency*, *justification*, *relevance*, and *conceptualisation*. This can be expressed in design specification documents which explicitly include the explanatory needs. When we look at the existing application, we can see that it does support the *transparency*, *conceptualisation*, and *justification* goals, where the latter is even only supported partially.

The fact that the system lacks certain explanatory capabilities is hardly surprising since they were not the main focus of the earlier implementation. However, the use of problem frames in general and explanation problem frames in particular helps us in identifying the deficiencies of the existing design, understanding and communicating explanatory needs, as well as exploring possible solutions to overcome these deficiencies.

Since we have started with data from an existing workplace analysis, we have not tested the hypothesis that (explanation) problem frames can be put to use in communicating with prospective users during the requirements elicitation phase. But we assume problems frames can enhance the cooperation between the requirements engineers and these users, as indicated by Phalp and Cox [2000].

In the requirements analysis, introducing explanation frames facilitates the explication and formalisation of the findings of our ethnographical study and thereby deepens our understanding of the problem domain.

The use of problem frames as a method during requirement specification aids us in checking the completeness of the specification and helps us to incorporate explanatory needs which could otherwise be overlooked. This should also lead to a design which encompasses more features. If we had done the original system specification with the help of (explanation) problem frames, the missing support for the *relevance* goal would have been uncovered.

An explanation aware requirements specification is also fruitful in the transition from design to implementation. Earlier work by Roth-Berghofer and others has coupled explanation goals with the knowledge containers of case-based reasoning systems [Roth-Berghofer and Cassens, 2005]. Having an explicit representation of explanation goals helps in identifying requirements for the knowledge containers, thereby easing the way from a specification document to the structure and actual content of the knowledge containers.

L.7 Conclusion and Future Work

We have shown how the use of explanation goal specific problem frames can help in recognising and explicating design requirements resulting from the perceived necessity of intelligent systems to be able to explain their own reasoning and behaviour. By testing our suggestions on an existing application, we have further on addressed how the use of (explanation) problem frames can lead to additional insights and can help testing the completeness of the requirements specification.

There are several issues which we have not addressed in this paper and which are left for further work. For example, we have to further explore the relation between the design documents and the actual implementation. Our results show that problem frames help us to identify which explanatory knowledge and mechanism should be provided, but the methods for the next step in identifying the missing knowledge containers and suggesting remedies have to be extended over the existing work on the relation between explanation goals, explanation kinds, and knowledge containers. We are considering to couple problem frames with design patterns to give system designers hints about the implementation issues.

Bibliography

Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. Oxford University Press, New York, 1977.

Paris Avgeriou and Uwe Zdun. Architectural patterns revisited – a pattern language. In *Proceedings of the tenth European Conference on Pattern Languages of Programs (EuroPlop 2005)*, pages 1–39, Irsee, 2005.

Fabio Bellifemine, Giovanni Caire, Agostino Poggi, and Giovanni Rimassa. Jade – a white paper. *TILAB EXP "in search of innovation"*, 3 (3):6–19, September 2003.

Ralph Bergmann, Klaus-Dieter Althoff, Sean Breen, Mehmet Göker, Michel Manago, Ralph Traphöner, and Stefan Wess. *Developing Industrial Case-Based Resoning Applications: The INRECA Methodology*, volume 1612 of *Lecture Notes in Computer Science*. Springer, Berlin, second edition, 2003.

Jörg Cassens and Anders Kofod-Petersen. Designing explanation aware systems: The quest for explanation patterns. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Explanation-Aware Computing – Papers from the 2007 AAAI Workshop*, number WS-07-06 in Technical Report, pages 20–27, Vancouver, BC, 2007. AAAI Press. ISBN 978-1-57735-333-1. URL `http://www.aaai.org/Library/Workshops/ws07-06.php`.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston, MA, 1995.

Jon Hall and Lucia Rapanotti. Problem frames for sociotechnical systems. In Jose Luis Mate and Andres Silva, editors, *Requirements Engineering for Sociotechnical Systems*, pages 318–339. Idea Group Publishing, 2005.

Denis Hatebur and Maritta Heisel. Problem frames and architectures for security problems. In Rune Winther, Bjoern Axel Gran, and Gustav Dahll, editors, *Proceedings of the 24th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, number 3688 in Lecture Notes in Computer Science, pages 390–404, Berlin, 2005. Springer.

Michael Jackson. *Problem Frames – Analysing and Structuring Software Development Problems*. Addison-Wesley, Boston, MA, 2001.

Anders Kofod-Petersen and Jörg Cassens. Explanations and context in ambient intelligent systems. In Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors, *Modeling and Using Context – CONTEXT 2007*, volume 4635 of *Lecture Notes in Computer*

Science, pages 303–316, Roskilde, Denmark, 2007. Springer. ISBN 978-3-540-74254-8. URL http://dx.doi.org/10.1007/978-3-540-74255-5_23.

Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, 1993.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995.

Keith Phalp and Karl Cox. Picking the right problem fram – an empirical study. *Empirical Software Engineering*, 5:215–228, 2000.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, pages 451–464, Chicago, 2005. Springer. URL http://dx.doi.org/10.1007/11536406_35.

Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde, and Bob Wielinga. *Knowledge Engineering and Management – The CommonKADS Methodology*. MIT Press, Cambridge, MA, 2000.

Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in case-based reasoning – perspectives and goals. *Artificial Intelligence Review*, 24(2):109–143, October 2005. ISSN 0269-2821. URL http://dx.doi.org/10.1007/s10462-005-4607-7.

M Further Publications

In this chapter, you will find an annotated list of publications published between 2000 and 2008 and relevant to the topic of the thesis which are not included in full text. These are largely earlier workshop versions – the revised and extended versions published later are included in the thesis.

M.1 User Aspects of Situated CBR Systems

Author:
Jörg Cassens

Abstract:
This paper deals with embedding CBR systems into human decision processes. It is heading towards Human Computer Interaction in a wide sense, in particular the question of how to design systems which are not only tools in a conventional sense, but have intelligent abilities.

When AI systems are considered not as a replacement of, but a supplement to human work, the question of an adequate form of interaction arises. An AI system is to a certain degree trespassing the boundary of the computer system as a tool, and extending this to it acting as a partner in a work flow.

Main Result:
Introduction of actor network theory and semiotics as tools for understanding issues of personalisation.

Published in:
Pedro Gonzálex-Calero, David Aha, and Mehmet Göker, editors, *Proceedings of the ECCBR 2002 Workshops*, Aberdeen, 2002.

Copyright:

M.2 Case-Based Explanations: Considering the Goals

Authors:
Frode Sørmo and Jörg Cassens

Abstract:
In this paper, we present a short overview of different theories of explanation. We argue that the goals of the user should be taken into account when deciding what is a good explanation for a given CBR system. Some general types relevant to many Case-Based Reasoning (CBR) systems are identified and we use these goals to identify some limitations in using the case as an explanation in CBR systems.

Main Result:
Introduction of the four goals *Justification*, *Transparency*, *Relevance*, and *Learning*. This is an earlier version of Paper E.

Published in:
Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, number 142-04 in Technical Report of the Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, pages 165–174, Madrid, 2004.

M.3 Goals and Kinds of Explanations in Case-Based Reasoning

Authors:
Thomas R. Roth-Berghofer, Jörg Cassens and Frode Sørmo

Abstract:
Research on explanation in Case-Based Reasoning (CBR) is a topic that gains momentum. In this context, fundamental issues on what are and to which end do we use explanations have to be reconsidered. This article presents a prelimenary outline of the combination of two recently proposed classifications of explanations based on the type of the explanation itself and user goals which should be fulfilled.

Main Result:
This is a first view on combining the notions of user goals and explanation kinds into one framework. Paper C is a later and more detailed inquiry into this topic.

Published in:
Klaus Dieter Althoff, Andreas Dengel, Ralph Bergmann, Markus Nick, and Thomas R. Roth-Berghofer, editors, *WM 2005: Professional Knowledge Management - Experiences and Visions*, pages 264–268, Kaiserslautern, 2005.

Copyright:

M.4 Activity Theory and Context-Awareness

Authors:
Anders Kofod-Petersen and Jörg Cassens

Abstract:
A lot of research has been done in the area of context-aware computing. Even though, the term context seems often not to be well defined. We attribute this problem partly to the fact that research often focuses on syntactical and technical issues of contextuality and does not take a knowledge level perspective on context. When including the knowledge level, some sort of analysis is required on what aspects need to be modelled. In this paper, we propose the use of an Activity Theory (AT) based approach on modelling components, and outline how it can be combined with the AmbieSense context modelling framework we have proposed earlier.

Main Result:
Introducing a knowledge level perspective on context modeling. Connecting concepts from activity theory with different, established categories of concept. Earlier version of Paper F.

Published in:
Stefan Schulz, David B. Leake, and Thomas R. Roth-Berghofer, editors, *IJCAI-05 Workshop on Modelling and Retrieval of Context – Working Notes*, pages 1–12, Edinburgh, 2005.

M.5 Explanatory Capabilities in the CREEK Knowledge-Intensive Case-Based Reasoner

Authors:
Anders Kofod-Petersen, Jörg Cassens, and Agnar Aamodt

Abstract:
The ability to give explanations for its reasoning and behaviour is a core capability of an intelligent system. There are a number of different goals a user can have towards such explanations. This paper presents how the knowledge intensive case-based reasoning framework CREEK can support some of these different goals in an ambient intelligence setting.

Main Result:
The paper looks at how CREEK can support the different user goals introduced earlier.

To be published in:
Proceedings SCAI 2008 - The 10th Scandinavian Conference on Artificial Intelligence. Stockholm, 2008.

Part III

Postscript

List of Figures

List of Tables

List of Publications

The following articles relevant to the contents of this thesis have been published by the author in the years 2000 to 2008.

Bibliography

Jörg Cassens. User Aspects of Situated CBR Systems. In *Proceedings of the ECCBR 2004 Workshops*, Aberdeen, 2002. Robert Gordon University.

Jörg Cassens. Knowing What to Explain and When. In Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, number 142-04 in Technical Report of the Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, pages 97–104, Madrid, 2004. URL http://www.idi.ntnu.no/~cassens/work/publications/download/2004-ECCBR-WS-AT.pdf.

Jörg Cassens. User Aspects of Explanation Aware CBR Systems. In Maria Francesca Costabile and Fabio释ternó, editors, *Human-Computer Interaction – INTERACT 2005*, volume 3585 of *LNCS*, pages 1087–1090, Rome, 2005. Springer. URL http://dx.doi.org/10.1007/11555261_111.

Jörg Cassens. A Work Context Perspective on Mixed-Initiative Intelligent Systems. In *Proceedings of the IJCAI 2003 Workshop on Mixed-Initiative Intelligent Systems*, pages 30–35, Acapulco, 2003. URL http://lalab.gmu.edu/miis/papers/Cassens-Jorg.pdf.

Jörg Cassens. Zum Verhältnis der Informatik zu anderen Fachdisziplinen. In Frieder Nake, Arno Rolf, and Dirk Siefkes, editors, *Informatik - Aufregung zu einer Disziplin*, number 235 in Berichte, pages 36 – 38, Heppenheim, 2001. Universität Hamburg, Fachbereich Informatik.

Jörg Cassens and Anders Kofod-Petersen. Explanations and case-based reasoning in ambient intelligent systems. In David C. Wilson and

Deepak Khemani, editors, *ICCBR-07 Workshop Proceedings*, pages 167–176, Belfast, Northern Ireland, 2007a. ISBN 978-1-85923-222-4. URL http://ceur-ws.org/Vol-271/paper01.pdf.

Jörg Cassens and Anders Kofod-Petersen. Designing explanation aware systems: The quest for explanation patterns. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Explanation-Aware Computing – Papers from the 2007 AAAI Workshop*, number WS-07-06 in Technical Report, pages 20–27, Vancouver, BC, 2007b. AAAI Press. ISBN 978-1-57735-333-1. URL http://www.aaai.org/Library/Workshops/ws07-06.php.

Jörg Cassens and Anders Kofod-Petersen. Using Activity Theory to Model Context Awareness: a Qualitative Case Study. In Geoff C. J. Sutcliffe and Randy G. Goebel, editors, *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, pages 619–624, Melbourne Beach, 2006. AAAI Press. ISBN 978-1-57735-261-7. URL http://www.aaai.org/Library/FLAIRS/flairs06contents.php.

Jörg Cassens and Rebekah Wegener. Making use of abstract concepts – systemic-functional linguistics and ambient intelligence. In *Proceedings IFIP AI 2008 – The Second IFIP International Conference on Artificial Intelligence in Theory and Practice*, to appear.

Zoran Constantinescu-Fulöp and Jörg Cassens. It's Magic: SourceMage GNU/Linux as HPC Cluster OS. In *LinuxTag 2003*, Karlsruhe, 2003. LinuxTag.

Anders Kofod-Petersen and Jörg Cassens. Explanations and context in ambient intelligent systems. In Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors, *Modeling and Using Context – CONTEXT 2007*, volume 4635 of *LNCS*, pages 303–316, Roskilde, Denmark, 2007. Springer. ISBN 978-3-540-74254-8. URL http://dx.doi.org/10.1007/978-3-540-74255-5_23.

Anders Kofod-Petersen and Jörg Cassens. Activity Theory and Context-Awareness. In Stefan Schulz, David B. Leake, and Thomas R. Roth-Berghofer, editors, *IJCAI-05 Workshop on Modelling and Retrieval of Context – Working Notes*, pages 1–12, Edinburgh, 2005. IJCAI. URL http://ceur-ws.org/Vol-146/paper1.pdf.

Anders Kofod-Petersen and Jörg Cassens. Using Activity Theory to Model Context Awareness. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005*, *Revised Selected Papers*, volume 3946 of *LNCS*, pages 1–17, Edinburgh, 2006. Springer. ISBN 3-540-33587-0. URL http://dx.doi.org/10.1007/11740674_1.

Anders Kofod-Petersen, Jörg Cassens, David B. Leake, and Stefan Schulz, editors. *Proceedings of the 4th International Workshop on Modeling and Reasoning in Context (MRC 2007)*, number 112 in Computer Science Research Report, Roskilde, Denmark, 2007. Roskilde University Center. URL http://www.ruc.dk/dat_en/research/reports/112/.

Anders Kofod-Petersen, Jörg Cassens, and Agnar Aamodt. Explanatory capabilities in the CREEK knowledge-intensive case-based reasoner. In *Proceedings SCAI 2008 – The 10th Scandinavian Conference on Artificial Intelligence*, to appear.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping Goals and Kinds of Explanations to the Knowledge Containers of Case-Based Reasoning Systems. In Héctor Muñoz-Avila and Francesco Ricci, editors, *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *LNAI*, pages 451–464, Chicago, 2005. Springer. URL http://dx.doi.org/10.1007/11536406_35.

Thomas R. Roth-Berghofer, Jörg Cassens, and Frode Sørmo. Goals and Kinds of Explanations in Case-Based Reasoning. In Klaus Dieter Althoff, Andreas Dengel, Ralph Bergmann, Markus Nick, and Thomas Roth-Berghofer, editors, *WM 2005: Professional Knowledge Management – Experiences and Visions*, pages 264–268, Kaiserslautern, 2005. DFKI.

Frode Sørmo and Jörg Cassens. Explanation Goals in Case-Based Reasoning. In Pablo Gervás and Kalyan May Gupta, editors, *Proceedings of the ECCBR 2004 Workshops*, number 142-04 in Technical Report of the Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, pages 165–174, Madrid, 2004. URL http://www.idi.ntnu.no/~cassens/work/publications/download/2004-ECCBR-WS-goals.pdf.

Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in Case-Based Reasoning – Perspectives and Goals. *Artificial Intelligence Review*, 24(2):109–143, October 2005. ISSN 0269-2821. URL http://dx.doi.org/10.1007/s10462-005-4607-7.

Rebekah Wegener, Jörg Cassens, and David Butt. Start making sense: Systemic functional linguistics and ambient intelligence. *Revue d'Intelligence Artificielle*, Special issue on Modelling and Reasoning on Context, to appear.

Citation Index

Bibliography

The following bibliography lists all references used throughout this thesis, both in Part 1 (Research Overview) and in Part 2 (Publications).

Joint Bibliography

Agnar Aamodt. Knowledge-intensive case-based reasoning in CREEK. In Funk and Calero [2004], pages 1–15.

Agnar Aamodt. *A Knowledge-Intensive, Integrated Approach to Problem Solving and Sustained Learning*. PhD thesis, Norwegian Institute of Technology, Department of Computer Science, Trondheim, May 1991. URL http://www.idi.ntnu.no/grupper/su/publ/phd/aamodt-thesis.pdf.

Agnar Aamodt. Explanation-driven retrieval, reuse, and learning of cases. In Michael M. Richter, Stefan Wess, Klaus-Dieter Althoff, and Frank Maurer, editors, *EWCBR-93: First European Workshop on Case-Based Reasoning*, number SR-93-12 (Technical Report), pages 279–284, Kaiserslautern, 1993. University of Kaiserslautern.

Agnar Aamodt. Explanation-driven case-based reasoning. In Wess et al. [1994], pages 274–288. ISBN 3-540-58330-0.

Agnar Aamodt. Knowledge acquisition and learning by experience – the role of case-specific knowledge. In Gheorghe Tecuci and Yves Kodratoff, editors, *Machine Learning and Knowledge Acquisition – Integrated Approaches*, chapter 8, pages 197–245. Academic Press, 1995.

Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1):39–59, March 1994.

Peter Achinstein. *The Nature of Explanation*. Oxford University Press, Oxford, 1983.

David W. Aha and Ian Watson, editors. *Case-Based Reasoning Research and Development: Proceedings ICCBR 2001*, number 2080 in Lecture Notes in Artificial Intelligence, Vancouver, 2001. Springer.

Vincent Aleven and Kevin D. Ashley. Teaching case-based argumentation through a model and examples: Empirical evaluation of an intelligent learning environment. In B. du Boulay and R. Mizoguchi, editors, *Artificial Intelligence in Education, Proceedings of AI-ED 1997 World Conference*, pages 87–94, Amsterdam, 1997. IOS Press.

Vincent Aleven and Kenneth R. Koedinger. An effective metacognitive strategy: Learning by doing and explaining with a computer based cognitive tutor. *Cognitive Science*, 26:147–179, 2002.

Christopher Alexander, Sara Ishikawa, Murray Silverstein, Max Jacobson, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language*. Oxford University Press, New York, 1977.

Klaus-Dieter Althoff and Wolfgang Wilke. Potential uses of case-based reasoning in experience based construction of software systems and business process support. In Ralph Bergmann and Wolfgang Wilke, editors, *Proc. of the Fifth German Workshop on Case-Based Reasoning, Centre for Learning Systems and Applications*, number LSA-97-01E (Technical Report), pages 31–38. University of Kaiserslautern, 1997.

Louis Althusser. Contradiction and overdetermination. In *For Marx*. Penguin Press, 1962. URL http://www.marxists.org/reference/archive/althusser/1962/overdetermination.htm.

Jim Alty, Ben Shneiderman, and Pattie Maes. Direct manipulation vs. interface agents. *Interactions*, pages 42–61, 11+12 1997.

Peter Bøgh Andersen. What semiotics can and cannot do for HCI. *Knowledge-Based Systems*, 14:419–424, 2001.

Peter Bøgh Andersen and Martin Brynskov. The semiotics of smart appliances and pervasive computing. In *Semiotics and Intelligent Systems Development* Gudwin and Queiroz [2006], pages 211–255. ISBN 1599040638.

Aristotle. *Metaphysik*. rowohlts enzyklopädie. Rowohlt, Hamburg, 1994.

Aristotle. *Analytica Posteriora*. Akademie, 1998. URL http://classics.mit.edu/Aristotle/posterior.html.

Kevin D. Ashley and Derek Bridge, editors. *Case-Based Reasoning Research and Development: Proceedings ICCBR 2003*, number 2689 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2003. Springer.

Paris Avgeriou and Uwe Zdun. Architectural patterns revisited – a pattern language. In *Proceedings of the tenth European Conference on Pattern Languages of Programs (EuroPlop 2005)*, pages 1–39, Irsee, 2005.

Liam Bannon and Susanne Bødker. Beyond the interface: Encountering artifacts in use. In John M. Carroll, editor, *Designing Interaction: Psychology at the Human-Computer Interface*, pages 227–253. Cambridge University Press, Cambridge, 1991.

Frederic C. Bartlett. *Remembering: A study in Experimental and Social Psychology*. Cambridge University Press, 1932.

Fabio Bellifemine, Giovanni Caire, Agostino Poggi, and Giovanni Rimassa. Jade – a white paper. *TILAB EXP "in search of innovation"*, 3 (3):6–19, September 2003.

Juan José Bello-Thomás, Pedro González-Calero, and Belén Díaz-Agudo. JColibri: an object-oriented framework for building CBR systems. In Funk and Calero [2004], pages 32–46.

Romain Bénard, Cyril Bossard, and Pierre De Loor. Context's modelling for participative simulation. In Sutcliffe and Goebel [2006], pages 613–618. ISBN 978-1-57735-261-7. URL http://www.aaai.org/Library/FLAIRS/flairs06contents.php.

Ralph Bergmann, Gerhard Pews, and Wolfgang Wilke. Explanation-based similarity: A unifying approach for integrating domain knowledge into case-based reasoning for diagnosis and planning tasks. In Wess et al. [1994], pages 182–196. ISBN 3-540-58330-0.

Ralph Bergmann, Klaus-Dieter Althoff, Sean Breen, Mehmet Göker, Michel Manago, Ralph Traphöner, and Stefan Wess. *Developing Industrial Case-Based Resoning Applications: The INRECA Methodology*, volume 1612 of *Lecture Notes in Computer Science*. Springer, Berlin, second edition, 2003.

Enrico Blanzieri. A cognitive framework for personalization of the cbr cycle. In *Workshop Proceedings ECCBR-2002*, Aberdeen, 2002.

Susanne Bødker. Applying activity theory to video analysis: How to make sense of video data in human-computer interaction. In Nardi [1996b], pages 147–174.

Susanne Bødker. Activity theory as a challenge to systems design. In Hans-Erik Nissen, Heinz K. Klein, and Rudy Hirschheim, editors, *Information Systems Research: Contemporary Approaches and Emergent Traditions*, pages 551–564. North Holland, 1991.

Susanne Bødker and Peter Bøgh Andersen. Complex mediation. *Journal of Human Computer Interaction*, 20(4):353–402, 2005.

William F. Brewer, Clark A. Chinn, and Ala Samarapungavan. Explanations in scientists and children. *Minds and Machines*, 8:119–136, 1998.

Patrick Brézillon. Context modeling: Task model and practice model. In Kokinov et al. [2007], pages 122–135. ISBN 978-3-540-74254-8. URL http://dx.doi.org/10.1007/978-3-540-74255-5_10.

Patrick Brézillon and Jean-Charles Pomerol. Contextual knowledge sharing and cooperation in intelligent assistant systems. *Le Travail Humain*, 62(3):223–246, 1999.

Sylvain Bromberger. An approach to explanation. In Ronald J. Butler, editor, *Analytical Philosophy*, volume 2, pages 72–105. Basil Blackwell, Oxford, 1965.

Sylvain Bromberger. Why questions. In Robert G. Colodny, editor, *Mind and Cosmos*, pages 86–111. Pittsburgh University Press, Pittsburgh, 1966.

Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991a.

Rodney A. Brooks. Planning is just a way of avoiding figuring out what to do next. Technical report, MIT, 1987.

Rodney A. Brooks. New approaches to robotics. *Science*, 253:1227–1232, 1991b.

Stefanie Brüninghaus and Kevin D. Ashley. Combining case-based and model-based reasoning for predicting the outcome of legal cases. In Ashley and Bridge [2003], pages 65–79.

Bruce G. Buchanan and Edward H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. Addison Wesley, Reading, 1984.

David Butt. Multimodal representations of remote past and projected futures: from human origins to genetic counselling. In Masa-Chiyo Amano, editor, *Multimodality: towards the most efficient communications* *by humans. 21st century COE Program International Conference Series NO. 6*. Graduate school of letters, Nagoya University, 2006.

Jörg Cassens and Anders Kofod-Petersen. Using activity theory to model context awareness: a qualitative case study. In Sutcliffe and Goebel [2006], pages 619–624. ISBN 978-1-57735-261-7. URL http://www.aaai.org/Library/FLAIRS/flairs06contents.php.

Jörg Cassens and Anders Kofod-Petersen. Designing explanation aware systems: The quest for explanation patterns. In Roth-Berghofer et al. [2007], pages 20–27. ISBN 978-1-57735-333-1. URL http://www.aaai.org/Library/Workshops/ws07-06.php.

Jörg Cassens. A work context perspective on mixed-initiative intelligent systems. In *Proceedings of the IJCAI 2003 Workshop on Mixed-Initiative Intelligent Systems*, pages 30–35, Acapulco, 2003.

Jörg Cassens. Knowing what to explain and when. In Gervás and Gupta [2004], pages 97–104.

Jörg Cassens. User aspects of explanation aware cbr systems. In Maria Francesca Costabile and Fabio Paternó, editors, *Human-Computer Interaction – INTERACT 2005*, volume 3585 of *Lecture Notes in Computer Science*, pages 1087–1090, Rome, 2005. Springer. URL http://dx.doi.org/10.1007/11555261_111.

Balakrishnan Chandrasekaran, Michael C. Tanner Tanner, and John R. Josephson. Explaining control strategies in problem solving. *IEEE Expert*, 4(1):9–15, 1989.

William Cheetham and Joseph Price. Measures of solution accuracy in case-based reasoning. In Funk and Calero [2004], pages 106–118.

Christine Choppy, Denis Hatebur, and Maritta Heisel. Component composition through architectural patterns for problem frames. In *APSEC '06: Proceedings of the XIII Asia Pacific Software Engineering Conference*, pages 27–34. IEEE Computer Society, 2006. doi: http://dx.doi.org/10.1109/APSEC.2006.27.

William J. Clancey. The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence*, 20(3):215–251, 1983.

Rodney Clarke, Aditya Ghose, and Aneesh Krishna. Systemic semiotics as a basis for an agent oriented conceptual modeling methodology. In *Semiotics and Intelligent Systems Development* Gudwin and Queiroz [2006], pages 256–285. ISBN 1599040638.

Paul Cohen. Evaluation and case-based reasoning. In Kristian J. Hammond, editor, *Proceedings of a Workshop on Case-Based Reasoning*, pages 168–172, San Mateo, CA, USA, 1989. DARPA, Morgan Kaufmann. ISBN 1-55860-068-X.

Paul Cohen and Adele Howe. How evaluation guides AI research: The message still counts more than the medium. *AI Magazine*, 9(4):35–43, 1988.

Daniel Cohnitz. Explanations are like salted peanuts. In Ansgar Beckermann and Christian Nimtz, editors, *Proceedings of the Fourth International Congress of the Society for Analytic Philosophy*, 2000. http://www.gap-im-netz.de/gap4Konf/Proceedings4/titel.htm [Last access: 2004-08-11].

Michael T. Cox, editor. *Mixed Initiative Intelligence (MII): Proceedings of the AAAI Workshop*, 1999.

Susan Craw and Alun Preece, editors. *Advances in Case-Based Reasoning: Proceedings ECCBR 2002*, number 2416 in Lecture Notes in Artificial Intelligence, Aberdeen, 2002. Springer.

Pádraig Cunningham, Dónal Doyle, and John Loughrey. An evaluation of the usefulness of case-based reasoning explanation. In Ashley and Bridge [2003], pages 122–130.

Clarisse Sieckenius de Souza. *The Semiotic Engineering of Human-Computer Interaction (Acting with Technology)*. The MIT Press, 2005. ISBN 0262042207. QA76.9.H85 .D465 2005.

René Descartes. *Meditationen über die Grundlagen der Philosophie*. Number 27 in Philosophische Bibliothek. Felix Meiner Verlag, Hamburg, 1641.

Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

Belén Díaz-Agudo and Pedro González-Calero. An architecture for knowledge-intensive CBR systems. In *Advances in Case-Based Reasoning: Proceedings EWCBR 2000*, number 1898 in Lecture Notes in Artificial Intelligence, Berlin, 2000. Springer.

Dónal Doyle, Pádraig Cunningham, Derek Bridge, and Yusof Rahman. Explanation oriented retrieval. In Funk and Calero [2004], pages 157–168.

Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. ISTAG scenarios for ambient intelligence in 2010. Technical report, IST Advisory Group, 2001.

Bruce Edmonds. Towards implementing free will. In *AISB 2000 symposium on How to Design a Functioning Mind*, Birmingham, April 2000.

Robert G. Eggleston. Mixed-initiative transactions: A cognitive engineering approach to interface agent modeling. In Cox [1999].

Hamid R. Ekbia and Ana G. Maguitman. Context and relevance: A pragmatic approach. *Lecture Notes in Computer Science*, 2116:156–169, 2001.

Boi Faltings. Probabilistic indexing for case-based prediction. In David B. Leake and Enric Plaza, editors, *Case-Based Reasoning Research and Development: Proceedings ICCBR 1997*, volume 1266 of *Lecture Notes in Artificial Intelligence*, pages 611–622, Berlin Heidelberg, 1997. Springer.

Dieter Fensel, Richard Benjamins, Stefan Decker, Mauro Gaspari, Rix Groenboom, William Grosso, Mark Musen, Enrico Motta, Enric Plaza, Guus Schreiber, Rudi Studer, and Bob Wielinga. The component model of upml in a nutshell. In *WWW Proceedings WICSA1, 1st Working IFIP Conference on Software Architectures*, San Antonio, Texas, 1999.

Geraldine Fitzpatrick. *The Locales Framework: Understanding and Designing for Cooperative Work.* Ph.d. thesis, The University of Queensland, Australia, 1998.

Morten Fjeld, Kristina Lauche, Martin Bichsel, Fred Voorhoorst, Helmut Krueger, and Mathias Rauterberg. Physcial and virtual tools: Activity theory applied to the design of groupware. *CSCW*, 11(1-2):153–180, 2002.

Kenneth M. Ford, Alberto J. Cañas, and John Coffey. Participatory explanation. In *Proceedings FLAIRS 1993*, pages 111–115, 1993. URL `http://www.ihmc.us/users/acanas/Publications/ParticipatoryExplanation/ParticipatoryExplanation.htm`.

Michael Friedman. Explanation and scientific understanding. *Journal of Philosophy*, 71:5–19, 1974.

Peter Funk and Pedro A. González Calero, editors. *Advances in Case-Based Reasoning: Proceedings ECCBR 2004*, number 3155 in Lecture Notes in Artificial Intelligence, Berlin Heidelberg, 2004. Springer.

Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston, MA, 1995.

Karen M. Gardner, Alexander Rush, Michael K. Crist, Robert Konitzer, and Bobbin Teegarden. *Cognitive Patterns*. Cambridge University Press, Cambridge, UK, 1998.

Pablo Gervás and Kalyan May Gupta, editors. *Proceedings of the ECCBR 2004 Workshops*, Technical Report 142-04, Madrid, 2004. Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid.

James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

Ayse Göker and Hans Inge Myrhaug. User context and personalisation. In *Workshop proceedings for the sixth European Conference on Case Based Reasoning*, 2002.

Linda S. Gottfredson. Mainstream science on intelligence: An editorial with 52 signatories, history, and bibliography. *Intelligence*, 24(1):13–23, 1997.

Georg Gottlob and Toby Walsh, editors. *Proceedings of the 18th International Conference on Artificial Intelligence (IJCAI 2003)*, Acapulco, 2003. Morgan Kaufmann.

Shirley Gregor and Izak Benbasat. Explanations from intelligent systems: Theoretical foundations and implications for practice. *MIS Quarterly*, 23 (4):497–530, 1999.

Ricardo Gudwin and Joao Queiroz. *Semiotics and Intelligent Systems Development*. IGI Publishing, Hershey, PA, USA, 2006. ISBN 1599040638.

Odd Erik Gundersen and Anders Kofod-Petersen. Multiagent based problem-solving in a mobile environment. In Eivind Coward, editor, *Norsk Informatikkonferance 2005, NIK 2005*, pages 7–18. Institutt for Informatikk, Universitetet i Bergen, November 2005. ISBN 82-519-2081-7.

Kalyan Moy Gupta, David W. Aha, and Nabil Sandhu. Exploiting taxonomic and causal relations in conversational case retrieval. In Craw and Preece [2002].

Winfried Hacker. *Allgemeine Arbeitspsychologie*. Huber, 1998.

Jon Hall and Lucia Rapanotti. Problem frames for sociotechnical systems. In Mate and Silva [2005], pages 318–339.

Michael A.K. Halliday. *Language as a Social Semiotic: the social interpretation of language and meaning*. University Park Press, 1978.

Michael A.K. Halliday and Christian M.I.M. Matthiessen. *An Introduction to Functional Grammar, Third edition*. Arnold, London, UK, 2004.

Joseph Hanna. Probabilistic explanation and probabilistic causality. In *Phil. Soc. Am.*, volume 2. 1982.

Gilbert Harman. The inference to the best explanation. *The Philosophical Review*, 74(1):88–95, January 1965.

Thomas C. Hartrum and Scott A. DeLoach. Design issues for mixed-initiative agent systems. In Cox [1999].

Ruqaiya Hasan. Situation and the definition of genre. In Allen Grimshaw, editor, *What's going on here? Complementary Analysis of Professional Talk: volume 2 of the multiple analysis project*. Ablex, Norwood N.J., 1994.

Ruqaiya Hasan. Speaking with reference to context. In Mohsen Ghadessy, editor, *Text and Context in Functional Linguistics*. John Benjamins, Amsterdam, 1999.

Denis Hatebur and Maritta Heisel. Problem frames and architectures for security problems. In Rune Winther, Bjoern Axel Gran, and Gustav Dahll, editors, *Proceedings of the 24th International Conference on Computer Safety, Reliability and Security (SAFECOMP)*, number 3688 in Lecture Notes in Computer Science, pages 390–404, Berlin, 2005. Springer.

Carl G. Hempel. *Aspects of Scientific Explanation*. Free Press, New York, 1965.

Carl G. Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, (15):135–175, 1948.

Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the ACM 2000 Conference on Computer Supported Cooperative Work*, pages 241–250, 2000. URL `citeseer.ist.psu.edu/herlocker00explaining.html`.

Robert Hodge and Gunther Kress. *Social Semiotics*. Cornell, UK, 1988.

Michael Jackson. *Problem Frames – Analysing and Structuring Software Development Problems*. Addison-Wesley, Boston, MA, 2001.

Martha Dørum Jære, Agnar Aamodt, and Pål Skalle. Representing temporal knowledge for case-based prediction. In Craw and Preece [2002], pages 174–188.

R. Burke Johnson and Anthony J. Onwuegbuzie. Mixed methods research: A research paradigm whose time has come. *EDUCATIONAL RESEARCHER*, 33(7):14–26, October 2004. doi: 10.3102/0013189X033007014. URL http://dx.doi.org/10.3102/0013189X033007014.

John R. Josephson and Susan G. Josephson, editors. *Abductive Inference Computation: Philosophy, Technology*. Cambridge University Press, New York, 1994.

Manasawee Kaenampornpan and Eamonn O'Neill. Modelling context: an activity theory approach. In Panos Markopoulos, Berry Eggen, Emilie Aarts, and James L. Croeley, editors, *Ambient Intelligence: Second European Symposium on Ambient Intelligence, EUSAI 2004*, volume 3295 of *Lecture Notes in Computer Science*, pages 367–374. Springer, November 2004.

Manasawee Kaenampornpan and Eamonn O'Neill. Integrating history and activity theory in context aware system design. In *Proceedings of the 1st International Workshop on Exploiting Context Histories in Smart Environments (ECHISE)*, May 2005.

Immanuel Kant. *Anthropologie in pragmatischer Hinsicht*. Akademie, 1798.

Immanuel Kant. *Kritik der praktischen Vernunft*. Akademie, 1788.

Immanuel Kant. *Kritik der reinen Vernunft (2. Auflage)*. Akademie, 1787.

Viktor Kaptelinin. Computer-mediated activity: Functional organs in social and developmental contexts. In Nardi [1996b], pages 45–68.

Frank C. Keil and Robert A. Wilson, editors. *Explaining Explanation*. Bradford Books, Boston, MA, 2000.

Philip Kitcher. Explanation, conjunction, and unification. *Journal of Philosophy*, 73:207–12, 1976.

Anders Kofod-Petersen and Agnar Aamodt. Case-based situation assessment in a mobile context-aware system. In Antonio Krüger and Rainer Malaka, editors, *Artificial Intelligence in Mobile Systems (AIMS 2003)*, pages 41–49. Universität des Saarlandes, October 2003.

Anders Kofod-Petersen and Agnar Aamodt. Contextualised ambient intelligence through case-based reasoning. In Thomas R. Roth-Berghofer, Mehmet H. Göker, and H. Altay Güvenir, editors, *Proceedings of the Eighth European Conference on Case-Based Reasoning (ECCBR 2006)*, volume 4106 of *Lecture Notes in Computer Science*, pages 211–225, Berlin, September 2006. Springer.

Anders Kofod-Petersen and Jörg Cassens. Activity theory and context-awareness. In Stefan Schulz, David B. Leake, and Thomas R. Roth-Berghofer, editors, *IJCAI-05 Workshop on Modelling and Retrieval of Context – Working Notes*, pages 1–12, Edinburgh, 2005. IJCAI. URL http://ceur-ws.org/Vol-146/paper1.pdf.

Anders Kofod-Petersen and Jörg Cassens. Using activity theory to model context awareness. In Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors, *Modeling and Retrieval of Context: MRC 2005, Revised Selected Papers*, volume 3946 of *Lecture Notes in Computer Science*, pages 1–17, Edinburgh, 2006. Springer. ISBN 3-540-33587-0. URL http://dx.doi.org/10.1007/11740674_1.

Anders Kofod-Petersen and Jörg Cassens. Explanations and context in ambient intelligent systems. In Kokinov et al. [2007], pages 303–316. ISBN 978-3-540-74254-8. URL http://dx.doi.org/10.1007/978-3-540-74255-5_23.

Anders Kofod-Petersen and Marius Mikalsen. An architecture supporting implementation of context-aware services. In Patrik Floréen, Greger Lindén, Tiina Niklander, and Kimmo Raatikainen, editors, *Workshop on Context Awareness for Proactive Systems (CAPS 2005), Helsinki, Finland*, pages 31–42. HIIT Publications, June 2005a. ISBN 952-10-2518-2.

Anders Kofod-Petersen and Marius Mikalsen. Context: Representation and reasoning – representing and reasoning about context in a mobile environment. *Revue d'Intelligence Artificielle*, 19(3):479–498, 2005b.

Boicho Kokinov, Daniel C. Richardson, Thomas R. Roth-Berghofer, and Laure Vieu, editors. *Modeling and Using Context – CONTEXT 2007*, volume 4635 of *Lecture Notes in Computer Science*, Roskilde, Denmark, 2007. Springer. ISBN 978-3-540-74254-8. URL http://dx.doi.org/10.1007/978-3-540-74255-5.

Janet L. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann Publishers, San Mateo, 1993.

Janet L. Kolodner. Educational implications of analogy: A view from case-based reasoning. *American Psychologist*, 52(1):57–66, 1997.

Janet L. Kolodner and David B. Leake. A tutorial introduction to case-based reasoning. In David B. Leake, editor, *Case-Based Reasoning: Experiences, Lessons, & Future Directions*. MIT Press & AAAI Press, Cambridge, MA, 1996.

Mikko Korpela, Hettie A. Soriyan, and Karen C. Olufokunbi. Activity analysis as a method for information systems development. *Scandinavian Journal of Information Systems*, 12:191–210, 2001.

Mikko Korpela, Anja Mursu, and Hettie A. Soriyan. Information systems development as an activity. *CSCW*, 11(1-2):111–128, 2002.

Phyllis Koton. Reasoning about evidence in causal explanations. In *Proceedings of the AAAI-88*, volume 1, pages 256–261, Cambridge, MA, 1988. AAAI Press/MIT Press.

Kari Kutti. Activity theory as a potential framework for human-computer interaction research. In Nardi [1996b], pages 17–44.

Mansur Lalljee, Margaret Watson, and Peter White. Attribution theory: Social and functional extensions. In *The Organization of Explanations*. Blackwell, Oxford, 1983.

Pat Langley. User modeling in adaptive interfaces. In Judy Kay, editor, *User Modeling: Proceedings of the seventh International Conference, UM-99*, pages 357–370, Banff, Alberta, 1999. Springer.

Bruno Latour. *Science in Action: How to Follow Scientists and Engineers Through Society*. Harvard University Press, 1988. ISBN 0674792912.

Bruno Latour. Technology is society made durable. In John Law, editor, *A Sociology of Monsters*, pages 103–131. Routledge, 1991.

David B. Leake. *Evaluating Explanations: A Content Theory*. Lawrence Erlbaum Associates, New York, 1992.

David B. Leake. Goal-based explanation evaluation. In *Goal-Driven Learning*, pages 251–285. MIT Press, Cambridge, 1995a.

David B. Leake. Abduction, experience, and goals: A model of everyday abductive explanation. *Journal of Experimental and Theoretical Artificial Intelligence*, (7):407–428, 1995b.

David B. Leake. CBR in context: The present and future. In *Case-Based Reasoning: Experiences, Lessons, & Future Directions* Leake [1996b], pages 3–30.

David B. Leake, editor. *Case-Based Reasoning: Experiences, Lessons, & Future Directions*. AAAI Press/MIT Press, Menlo Park, 1996b.

David B. Leake. Workshop introduction: ECCBR-04 workshop on explanation in CBR. Slides, last access 22.02.2007, 2004. URL https://www.cs.tcd.ie/research_groups/mlg/ecbrws2004/.

David B. Leake and David McSherry. Special issue on explanation in case-based reasoning. *Artificial Intelligence Review*, 24(2), October 2005. ISSN 0269-2821.

David B. Leake, Andrew Kinley, and David C. Wilson. Learning to integrate multiple knowledge sources for case-based reasoning. In Pollack [1997], pages 246–251. URL citeseer.nj.nec.com/leake97learning.html.

David B. Leake, Larry Birnbaum, Kristian Hammond, Cameron Marlow, and Hao Yang. An integrated interface for proactive, experience-based design support. In *Proceedings of the sixth International Conference on Intelligent User Interfaces*, pages 101–108, Santa Fe, 2001a.

David B. Leake, Ana Maguitman, and Alberto Cañas. Assessing conceptual similarity to support concept mapping. In *Proceedings of the* 15^{th} *International Florida Artificial Intelligence Research Society Conference*, pages 172–186, Menlo Park, 2001b. AAAI Press.

Till C. Lech and Leendert W. M. Wienhofen. Ambieagents: A scalable infrastructure for mobile and context-aware information services. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 625–631, New York, NY, USA, 2005. ACM Press.

John D. Lee and Katrina A. See. Trust in automation: Designing for appropriate reliance. *Human Factors*, 46(1):50–80, 2004.

Doug Lenat. The dimensions of context-space. Technical report, Cycorp, Austin, TX, October 28 1998. URL http://www.cyc.com/doc/context-space.pdf.

Doug Lenat and Edward Feigenbaum. On the thresholds of knowledge. In *Proceedings IJCAI 1987*, pages 1173–1182, 1987.

Mario Lenz, Brigitte Bartsch-Spörl, Hans-Dieter Burkhard, and Stefan Wess, editors. *Case-Based Reasoning Technology: From Foundations to Applications*, volume 1400 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin, 1998.

Aleksei N. Leont'ev. *Activity, Consciousness, and Personality*. Prentice-Hall, 1978.

Yang Li, Jason I. Hong, and James A. Landay. Topiary: a tool for prototyping location-enhanced applications. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 217–226, New York, NY, USA, 2004. ACM Press. ISBN 1-58113-957-8. doi: http://doi.acm.org/10.1145/1029632.1029671.

Chistopher Lueg. Representaion in pervasive computing. In *Proceedings of the Inaugural Asia Pacific Forum on Pervasive Computing*, 2002a.

Christopher Lueg. Looking under the rug: On context-aware artifacts and socially adept technologies. In *Proceedings of the Workshop The Philosophy and Design of Socially Adept Technologies at the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2002)*. National Research Council Canada NRC 44918, April 2002b.

Christopher Lueg. Operationalizing context in context-aware artifacts: Benefits and pitfalls. *Information Science – The International Journal of an Emerging Discipline*, 5(2):42–47, 2002c.

Tinghuai Ma, Yong-Deak Kim, Qiang Ma, Meili Tang, and Weican Zhou. Context-aware implementation based on cbr for smart home. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob 2005)*, pages 112–115. IEEE, IEEE Computer Society, August 2005.

Ann Majchrzak and Les Gasser. On using artificial intelligence to integrate the design of organizational and process change in us manufacturing. *AI and Society*, 5:321–338, 1991.

Ji-Ye Mao and Izak Benbasat. The use of explanations in knowledge-based systems: Cognitive perspectives and a process-tracing analysis. *Journal of Managment Information Systems*, 17(2):153–179, 2000.

Stephen Marsh. Exploring the socially adept agent. In *Proceedings of the First International Workshop on Decentralized Intelligent Multi Agent Systems (DIMAS 1995)*, pages 301–308, 1995.

Karl Marx. *Das Kapital. Kritik der politischen Ökonomie. Band 1: Der Produktionsprozeß des Kapitals*, volume 23 of *Marx Engels Werke (MEW)*. Dietz-Verlag, Berlin, 1867.

Stewart Massie, Susan Craw, and Nirmalie Wiratunga. Visualisation of case-based reasoning for explanation. In Gervás and Gupta [2004], pages 135–144.

Jose Luis Mate and Andres Silva, editors. *Requirements Engineering for Sociotechnical Systems*. Idea Group Publishing, 2005.

Gavin P. McArdle and David C. Wilson. Visualising case-base usage. In Lorraine McGinty, editor, *Workshop Proceedings ICCBR 2003*, number 4/2004 in Technical Report, pages 105–114, Trondheim, 2003. Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science (IDI).

John McCarthy. What is ai? Internet, 2007. URL `http://www-formal.stanford.edu/jmc/whatisai/`. Last visited 2007-12-09.

Drew McDermott. A critique of pure reason. *Journal of Computational Intelligence*, 3(3):151–160, August 1987.

Lorraine McGinty and Barry Smyth. On the role of diversity in conversational recommender systems. In Ashley and Bridge [2003], pages 276–290.

David McSherry. Interactive case-based reasoning in sequential diagnosis. *Applied Intelligence*, 14:65–76, 2001.

David McSherry. Mixed-initative dialogue in case-based reasoning. In *Workshop Proceedings ECCBR-2002*, Aberdeen, 2002.

David McSherry. Explanation in case-based reasoning: an evidential approach. In Brian Lees, editor, *Proceedings of the eighth UK Workshop on Case-Based Reasoning*, pages 47–55, Cambridge, 2003.

David McSherry. Explaining the pros and cons of conclusions in CBR. In Funk and Calero [2004], pages 317–330.

David McSherry. Explanation in recommender systems. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 179–197. ISSN 0269-2821.

David McSherry. Strategic induction of decision trees. In Rob Milne and Max A. Bramer, editors, *Proceedings of ES98*, pages 15–26, 1998.

Eric Monteiro. Actor-network theory. In Claudio U. Ciborra, editor, *From Control to Drift*, pages 71–83. Oxford University Press, 2000.

Héctor Muñoz-Avila and Francesco Ricci, editors. *Case Based Reasoning Research and Development – ICCBR 2005*, volume 3630 of *Lecture Notes in Artificial Intelligence*, Chicago, 2005. Springer.

J. William Murdock, David W. Aha, and Len A. Breslow. Assessing elaborated hypotheses: An interpretive case-based reasoning approach. In Ashley and Bridge [2003], pages 332–346.

Daisy Mwanza. Mind the gap: Activity theory and design. Technical Report KMI-TR-95, Knowledge Media Institute, The Open University, Milton Keynes, 2000.

Frieder Nake. Human-computer interaction – signs and signals interfacing. *Languages of Design*, 2:193–205, 1994.

Bonnie A. Nardi. A brief introduction to activity theory. *KI – Künstliche Intelligenz*, 1:35–36, 2003.

Bonnie A. Nardi. Activity theory and human computer interaction. In *Context and Consciousness* Nardi [1996b], pages 7–16.

Bonnie A. Nardi, editor. *Context and Consciousness*. MIT Press, Cambridge, MA, 1996b.

Ulric Neisser, Gwyneth Boodoo, Thomas J. Bouchard Jr., A. Wade Boykin, Nathan Brody, Stephen J. Ceci, Diane F. Halpern, John C. Loehlin, Robert Perloff, Robert J. Sternberg, and Susana Urbina. Intelligence: Knowns and unknowns. *American Psychologist*, 51(2):77–101, 1996.

Allen Newell. The knowledge level. *Artificial Intelligence*, (18):87–127, 1982.

Conor Nugent and Pádraig Cunningham. A case-based explanation system for black-box systems. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 163–178. ISSN 0269-2821.

Lean Suan Ong, Barry Shepherd, Loong Cheong Tong, Francis Seow-Choen, Yik Hong Ho, Choong Leong Tang, Yin Seong Ho, and Kelvin Tan. The colorectal cancer recurrence support (CARES) system. *Artificial Intelligence in Medicine*, 11(3):175–188, 1997.

Pinar Öztürk and Agnar Aamodt. A context model for knowledge-intensive case-based reasoning. *International Journal of Human Computer Studies*, 48:331–355, 1998.

Charles Sanders Peirce. New elements (kaina stoicheia). In Carolyn Eisele, editor, *The New Elements of Mathematics by Charles S. Peirce*, volume 4, Mathematical Philosophy, pages 235–263. 1904. URL `http://www.cspeirce.com/menu/library/bycsp/stoicheia/stoicheia.htm`.

Keith Phalp and Karl Cox. Picking the right problem fram – an empirical study. *Empirical Software Engineering*, 5:215–228, 2000.

Jean Piaget. *The Construction of Reality in the Child*. Routledge and Kegan Paul, London, 1955.

Wolter Pieters. Free will and intelligent machines. Project Report, NTNU Trondheim, 2001.

Enric Plaza and Josep L. Arco. The abc of adaptation: Towards a software architecture for adaptation-centered cbr systems. Research Report 99-22, Artificial Intelligence Research Institute (IIIA), Bellaterra, Spain, 1999. `http://www.iiia.csic.es/`.

Enric Plaza, Eva Armengol, and Santiago Ontañón. Explanatory power of symbolic similarity in case-based reasoning. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 145–161. ISSN 0269-2821.

Michael Polanyi. *Personal Knowledge: Towards a Post-critical Philosophy*. N.Y.: Harper & Row, 1964.

Martha Pollack, editor. *Proceedings of the 15th International Conference on Artificial Intelligence (IJCAI 1997)*, Nagoya, 1997. Morgan Kaufmann.

Andrew Potter. A discourse approach to explanation aware knowledge representation. In Roth-Berghofer et al. [2007], pages 56–63. ISBN 978-1-57735-333-1. URL `http://www.aaai.org/Library/Workshops/ws07-06.php`.

Pearl Pu and Li Chen. Trust building with explanation interfaces. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 93–100, New York, NY, USA, 2006. ACM. ISBN 1-59593-287-9. doi: http://doi.acm.org/10.1145/1111449.1111475.

Amanda Quek and Hanifa Shah. A comparative survey of activity-based methods for information systems development. In Isabel Seruca, Joaquim Filipe, Slimane Hammoudi, and Jose Cordeiro, editors, *Proceedings of sixth International Conference on Enterprise Information Systems (ICEIS 2004)*, volume 5, pages 221–229, Porto, Portugal, 2004. ISBN 972-8865-00-7.

James Reilly, Kevin McCarthy, Lorraine McGinty, and Barry Smyth. Explaining compound critiques. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 199–220. ISSN 0269-2821.

Elaine Rich. User modeling via stereotypes. *Cognitive Science*, (3):329–354, 1979.

Michael M. Richter. The knowledge contained in similarity measures. Invited Talk at the First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, 1995.

Michael M. Richter. Remarks on current explanation research in artificial intelligence, 2005. Personal notes by Thomas R. Roth-Berghofer.

Michael M. Richter. *Prinzipien der Künstlichen Intelligenz*. B. G. Teubner, Stuttgart, 2. edition, 1992.

Michael M. Richter. Generalized planning and information retrieval. Technical report, University of Kaiserslautern, Artificial Intelligence – Knowledge-based Systems Group, 1997.

Gustavo Rossi, Daniel Schwabe, and Fernando Lyardet. User interface patterns for hypermedia applications. In *AVI '00: Proceedings of the working conference on Advanced visual interfaces*, pages 136–142, New York, NY, 2000. ACM Press. ISBN 1-58113-252-2. doi: http://doi.acm.org/10.1145/345513.345288.

Thomas R. Roth-Berghofer. Explanations and case-based reasoning: Foundational issues. In Funk and Calero [2004], pages 389–403.

Thomas R. Roth-Berghofer. *Knowledge Maintenance of Case-Based Reasoning Systems – The SIAM Methodology*, volume 262 of *Dissertationen zur Künstlichen Intelligenz*. Akademische Verlagsgesellschaft Aka GmbH/IOS Press, Berlin, 2003.

Thomas R. Roth-Berghofer and Jörg Cassens. Mapping goals and kinds of explanations to the knowledge containers of case-based reasoning systems. In Muñoz-Avila and Ricci [2005], pages 451–464. URL `http://dx.doi.org/10.1007/11536406_35`.

Thomas R. Roth-Berghofer, Jörg Cassens, and Frode Sørmo. Goals and kinds of explanations in case-based reasoning. In Klaus Dieter Althoff, Andreas Dengel, Ralph Bergmann, Markus Nick, and Thomas Roth-Berghofer, editors, *WM 2005: Professional Knowledge Management – Experiences and Visions*, pages 264–268, Kaiserslautern, 2005. DFKI.

Thomas R. Roth-Berghofer, Stefan Schulz, and David B. Leake, editors. *Explanation-Aware Computing – Papers from the 2007 AAAI Workshop*, number WS-07-06 in Technical Report, Vancouver, BC, 2007. AAAI Press. ISBN 978-1-57735-333-1. URL http://www.aaai.org/Library/Workshops/ws07-06.php.

David E. Rumelhart. Schemata: The building blocks of cognition. In Rand J. Spiro, Bertram C. Bruce, and William F. Brewer, editors, *Theoretical Issues in Reading and Comprehension*. Lawrence Erlbaum Assoc Inc, 1980.

Ulrich Ruschig. Korruption der wissenschaft. In Jörg Cassens, editor, *Tagungsbändchen JOINT-95*, number 3 in Schriftenreihe Kritische Wissenschaft, pages 3–17, Oldenburg, 1995. akki.

Wesley Salmon. Statistical explanation. In Robert G. Colodny, editor, *The Nature and Function of Scientific Theories*, pages 173–231. Pittburgh University Press, Pittburgh, 1971.

Wesley Salmon. *Scientific Explanation and the Causal Structure of the World*. Princeton University Press, Princeton, 1984.

Mahadev Satyanarayanan. Pervasive computing: Vision and challenges. *IEEE Personal Communications*, 8(4):10–17, August 2001. URL citeseer.ist.psu.edu/satyanarayanan01pervasive.html.

Ferdinand de Saussure. *Course in General Linguistics*. McGraw-Hill, 1966.

Guus Schreiber, Hans Akkermans, Anjo Anjewierden, Robert de Hoog, Nigel Shadbolt, Walter Van de Velde, and Bob Wielinga. *Knowledge Engineering and Management – The CommonKADS Methodology*. MIT Press, Cambridge, MA, 2000.

Roger C. Schank. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press, Cambridge, 1983.

Roger C. Schank. *Explanation Patterns – Understanding Mechanically and Creatively*. Lawrence Erlbaum, New York, 1986.

Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals and Understanding*. Erlbaum, 1977.

Roger C. Schank and David B. Leake. Creativity and learning in a case-based explainer. *Artificial Intelligence*, 40(1–3):353–385, 1989.

Roger C. Schank, Alex Kass, and Christopher K. Riesbeck, editors. *Inside Case-Based Explanation*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1994.

Barry Smyth and Mark T. Keane. Adaptation-guided retrieval: Questioning the similarity assumption in reasoning. *Artificial Intelligence*, 102(2): 249–293, 1998.

Barry Smyth and Paul McClave. Similarity vs. diversity. In Aha and Watson [2001], pages 347–361.

Barry Smyth and Lorraine McGinty. The power of suggestion. In Gottlob and Walsh [2003], pages 127–132.

Alan Sokal and Jean Bricmont. *Eleganter Unsinn*. dtv, München, 2001.

Frode Sørmo. Plausible inheritance semantic network inference for case-based reasoning. M.sc. thesis, Norwegian University of Science and Technology (NTNU), Department of Computer and Information Science, Trondheim, 2000.

Frode Sørmo. Case-based student modeling using concept maps. In Muñoz-Avila and Ricci [2005], pages 492–506.

Frode Sørmo and Agnar Aamodt. Knowledge communication and CBR. In Pedro González-Calero, editor, *Proceedings of the ECCBR-02 Workshop on Case-Based Reasoning for Education and Training*, pages 47–59, Aberdeen, 2002.

Frode Sørmo and Jörg Cassens. Explanation goals in case-based reasoning. In Gervás and Gupta [2004], pages 165–174. URL `http://www.idi.ntnu.no/~cassens/work/publications/download/2004-ECCBR-WS-goals.pdf`.

Frode Sørmo, Jörg Cassens, and Agnar Aamodt. Explanation in case-based reasoning – perspectives and goals. In *Artificial Intelligence Review* [Leake and McSherry, 2005], pages 109–143. ISSN 0269-2821. URL `http://dx.doi.org/10.1007/s10462-005-4607-7`.

Peter Spieker. *Natürlichsprachliche Erklärungen in technischen Expertensystemen*. Dissertation, University of Kaiserslautern, 1991.

Thomas M. Strat. Employing contextual information in computer vision. In *DARPA93*, pages 217–229, 1993.

Lucy A. Suchman. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, New York, NY, USA, 1987. ISBN 0-521-33137-4.

Geoff C. J. Sutcliffe and Randy G. Goebel, editors. *Proceedings of the 19^{th} International Florida Artificial Intelligence Research Society Conference*, Melbourne Beach, 2006. AAAI Press. ISBN 978-1-57735-261-7. URL http://www.aaai.org/Library/FLAIRS/flairs06contents.php.

William R. Swartout. What kind of expert should a system be? xplain: A system for creating and explaining expert consulting programs. *Artificial Intelligence*, 21:285–325, 1983.

William R. Swartout and Johanna D. Moore. Explanation in second generation expert systems. In Jean-Marc David, Jean-Paul Krivine, and Reid Simmons, editors, *Second Generation Expert Systems*, pages 543–585. Springer, Berlin, 1993.

William R. Swartout and Stephen W. Smoliar. On making expert systems more like experts. *Expert Systems*, 4(3):196–207, 1987.

Carsten Tautz, Klaus-Dieter Althoff, and Markus Nick. Learning from project experience – a case study on experience factory. In *Proc. GI-Fachgruppentreffen Maschinelles Lernen (FGML'00)*, Sankt Augustin, 2000. GMD.

Gheorge Tecuci, Mihai Boicu, Kathryn Wright, and Seok Won Lee. Mixed-initiative development of knowledge bases. In Cox [1999].

Paul Thagard. *Computational Philosophy of Science*. MIT Press/Bradford Books, Boston, 1988.

Paul Thagard. Explanatory coherence. *Behavioral and Brain Sciences*, 12(3): 435–467, 1989.

Paul Thagard and Keith Holyok. Why indexing is the wrong way to think about analog retrieval. In *Proceedings of the DARPA Workshop on Case-Based Reasoning*, pages 36–40, San Mateo, 1989. Morgan Kaufman.

Guy Tiberghien. Context and cognition: Introduction. *European Bulletin of Cognitive Psychology*, 6(2):105–119, 1986.

Peter Totterdell and Paul Rautenbach. Adaptation as a design problem. In *Adaptive User Interfaces*, pages 59–84. Academic Pres, 1990.

Alan Turing. Computing machinery and intelligence. *Mind*, LIX(236): 433–460, October 1950.

Bas van Fraassen, editor. *The Scientific Image*. Clarendon Press, Oxford, 1980.

Martijn van Welie and Hallvard Trætteberg. Interaction patterns in user interfaces. In *Proceedings of the seventh Pattern Languages of Programs Conference (PLoP)*, Monticello, 2000. URL `http://citeseer.ist.psu.edu/vanwelie00interaction.html`.

Valentin N. Voloshinov. *Marxism and the Philosophy of language*. Seminar Press, New York, 1973.

Lew S. Vygotsky. *Mind in Society*. Harvard University Press, Cambridge, MA, 1978.

Lew S. Vygotsky. *Ausgewählte Schriften Bd. 1: Arbeiten zu theoretischen und methodologischen Problemen der Psychologie*. Pahl-Rugenstein, Köln, 1985.

Douglas Walton. Dialogical models of explanation. In Roth-Berghofer et al. [2007], pages 1–9. ISBN 978-1-57735-333-1. URL `http://www.aaai.org/Library/Workshops/ws07-06.php`.

Ian Watson. Case-based reasoning is a methodology, not a technology. *Knowledge-Based Systems*, 12(5-6):303–308, 1999a.

Ian Watson. Survey of CBR application areas, 1999b. Invited Talk at the 3rd International Conference on Case-Based Reasoning ICCBR.

Mark Weiser. The computer for the 21st century. *Scientific American*, pages 94–104, September 1991.

Ina Wentzlaff and Markus Specker. Pattern-based development of user-friendly web applications. In *ICWE '06: Workshop Proceedings of the sixth International Conference on Web Engineering*, New York, NY, 2006. ACM Press.

Stefan Wess, Klaus-Dieter Althoff, and Michael M. Richter, editors. *Topics in Case-Based Reasoning: Proceedings of the 1st European Workshop EWCBR 1993 – Selected Papers*, volume 837 of *Lecture Notes in Computer Science*, Berlin Heidelberg, 1994. Springer. ISBN 3-540-58330-0.

Benjamin L. Whorf. *Language, Thought and Reality (ed. J. Carroll)*. MIT Press, Cambridge, MA, 1956.

Mikael Wiberg and Charlotte Olsson. Designing artifacts for context awareness. In *Proceedings of IRIS 22 Enterprise Architectures for Virtual Organisations*, pages 49–58, Jyväskylä, 1999. Dept. of Computer Science and Information Systems, University of Jyväskylä.

Michael R. Wick and William B. Thompson. Reconstructive expert system explanation. *Artificial Intelligence*, 54(1–2):33–70, 1992.

Stephen S. Yau, Yu Wang, and Fariaz Karim. Development of situation-aware application software for ubiquitous computing environments. In *29th Annual International Computer Software and Applications Conference (COMPSAC'02)*, volume 1, pages 107–112, Edinburgh, Scotland, July 2002. IEEE Computer Society.

L. Richard Ye and Paul E. Johnson. The impact of explanation facilities on user acceptance of expert systems advice. *MIS Q.*, 19(2):157–172, 1995. ISSN 0276-7783.

Gabriele Zenobi and Pádraig Cunningham. An approach to aggregating ensembles of lazy learners that supports explanation. In Craw and Preece [2002], pages 436–447.

Elisabetta Zibetti, Vicenc Quera, Francesc Salvador Beltran, and Charles Tijus. Contextual categorization: A mechanism linking perception and knowledge in modeling and simulating perceived events as actions. *Modeling and using context*, pages 395–408, 2001.

Andreas Zimmermann. Context-awareness in user modelling: Requirements analysis for a case-based reasoning application. In Ashley and Bridge [2003], pages 718–732.