Using Activity Theory to Model Context Awareness

Anders Kofod-Petersen and Jörg Cassens

Department of Computer and Information Science (IDI),
Norwegian University of Science and Technology (NTNU),
7491 Trondheim, Norway,
{anderpe|cassens}@idi.ntnu.no,
http://www.idi.ntnu.no/

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.

1 Introduction

The original vision of ubiquitous computing proposed by Weiser [1] 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 [2, p. 3], and the *automagic* way that Maria gets help on her business trip in [3, p. 4]. More examples and comments can be found in [4]. 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 [5] perspective or an analysis of context on the level of socio-technical systems [6].

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 [7], Situated Action [8] or the Locales Framework [9]. One of the most intriguing theories, however, is Activity Theory based on the works of Vygotsky and Leont'ev [10,11,12]. 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 [13,14]. 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 5, the knowledge model, including context employed in this work, is described. Finally, some pointers for future work are presented.

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) [15].

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 [16] 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 [17], 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 [16, 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 [18,19,20], and in ecological psychology by J. J. Gibson [21].

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 [22], where context is applied to select the most suitable algorithm for recognition in computer vision, and by Öztürk and Aamodt [23] who utilised context to improve the quality and efficiency of Case-Based Reasoning.

Strat [22] 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 [23]. 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. [24] 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.

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 [25] for a short introduction to AT and [26,27] for deeper coverage. The theoretical foundations of AT in general can be found in the works of Vygotsky and Leont'ev [10,11,12].

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. 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.

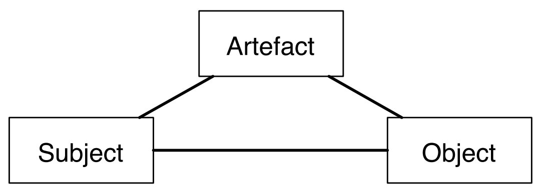

Fig. 1. Activity Theory: The basic triangle of Mediation

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 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.

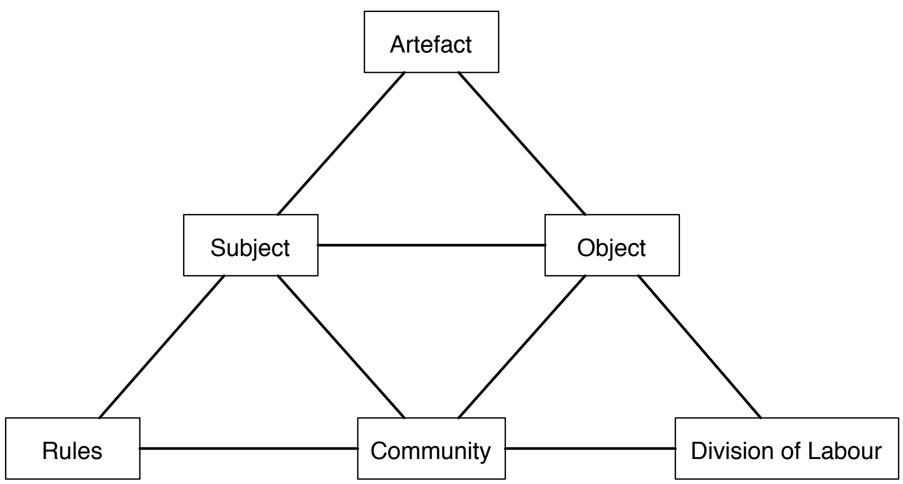

Fig. 2. Cultural Historical Activity Theory: Expanded triangle, incorporating the community and other mediators.

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. [28,29]). The expanded model (depicted in Fig. 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.

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.

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 1 (the taxonomy is described in more detail in Section 5). We can see that the personal context contains information we would associate with the acting subject itself.

Table 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.

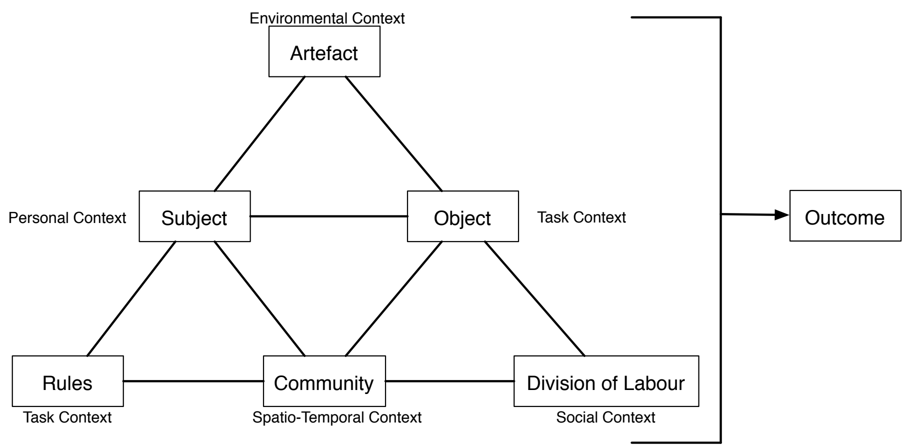

Fig. 3. Mapping from Activity Theory to context model

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.

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 [30]. 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 [31]. 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 and Landay [32] 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 [33] 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 [34]. It has also been used in the design of augmented reality systems, as reported in [35] and for a posteriori analysis of computer systems in use [36]. A comparative survey of five different AT based methods for information systems development with pointers to additional examples was conducted by Queak and Shah [37].

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 [38].

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 [24], 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 [15].

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 [39], 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 [40] 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!" [40, 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. 4). This taxonomy is inherited from the context-aware tradition and adopted to make use of the general concepts we find in Activity Theory.

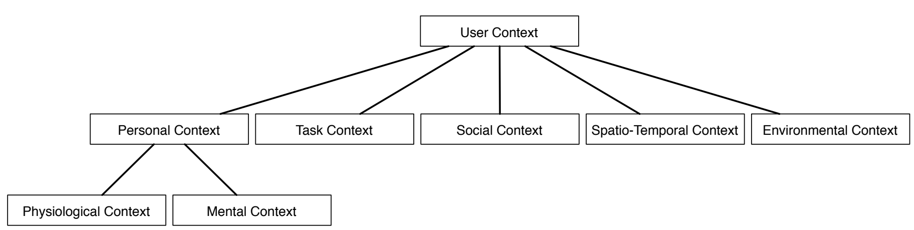

Fig. 4. Context taxonomy

The context is divided into five sub-categories (a more thorough discussion can be found in [41] or [42]):

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. 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 [43]. 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 [44,45].

The generic concepts are partly gathered through the use of activity theoretic analysis. These concepts include the six aspects shown in Fig. 3. The top-level taxonomy including the concepts acquired from AT is depicted in Fig. 5. The context model is now primed to model situations and the activities occurring within them.

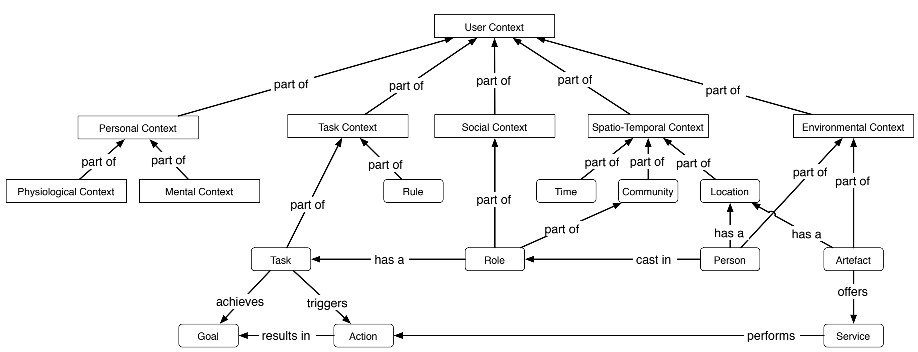

Fig. 5. Populated context structure

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*.

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 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 [46]. 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 [47].

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 [48] 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 [49]. Based on the context-aware situation assessment being carried out, this agency supplies context-sensitive problem solving [50]. 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).

References

1. Weiser, M.: The computer for the 21st century. Scientific American (1991) 94–104
2. Satyanarayanan, M.: Pervasive computing: Vision and challenges. IEEE Personal Communications **8** (2001) 10–17
3. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: ISTAG Scenarios for Ambient Intelligence in 2010. Technical report, IST Advisory Group (2001)
4. Lueg, C.: Representaion in Pervasive Computing. In: Proceedings of the Inaugural Asia Pacific Forum on Pervasive Computing. (2002)
5. Newell, A.: The Knowledge Level. Artificial Intelligence **18** (1982) 87–127
6. Lueg, C.: 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 (2002)
7. Latour, B.: Science in Action: How to Follow Scientists and Engineers Through Society. Harvard University Press (1988)
8. Suchman, L.A.: Plans and Situated Actions: The Problem of Human-Machine Communication. Cambridge University Press, New York, NY, USA (1987)
9. Fitzpatrick, G.: The Locales Framework: Understanding and Designing for Cooperative Work. Ph.d. thesis, The University of Queensland, Australia (1998)
10. Vygotski, L.S.: Mind in Society. Harvard University Press, Cambridge, MA (1978)
11. Vygotski, L.S.: Ausgewählte Schriften Bd. 1: Arbeiten zu theoretischen und methodologischen Problemen der Psychologie. Pahl-Rugenstein, Köln (1985)
12. Leont'ev, A.N.: Activity, Consciousness, and Personality. Prentice-Hall (1978)
13. McSherry, D.: Mixed-Initative Dialogue in Case-Based Reasoning. In: Workshop Proceedings ECCBR-2002, Aberdeen (2002)
14. Totterdell, P., Rautenbach, P.: Adaptation as a Design Problem. In: Adaptive User Interfaces. Academic Pres (1990) 59–84

15. Polanyi, M.: Personal Knowledge: Towards a Post-critical Philosophy. N.Y.: Harper & Row (1964)
16. Ekbia, H.R., Maguitman, A.G.: Context and relevance: A pragmatic approach. Lecture Notes in Computer Science **2116** (2001) 156–169
17. Lenat, D.: The Dimensions of Context-Space. Technical report, Cycorp, Austin, TX (1998)
18. Brooks, R.A.: Planning is Just a Way of Avoiding Figuring Out What to Do Next. Technical report, MIT (1987)
19. Brooks, R.A.: Intelligence without Representation. Artificial Intelligence **47** (1991) 139–159
20. Brooks, R.A.: New Approaches to Robotics. Science **253** (1991) 1227–1232
21. Gibson, J.J.: The Ecological Approach to Visual Perception. Houghton Mifflin (1979)
22. Strat, T.M.: Employing contextual information in computer vision. In: DARPA93. (1993) 217–229
23. Öztürk, P., Aamodt, A.: A context model for knowledge-intensive case-based reasoning. International Journal of Human Computer Studies **48** (1998) 331–355
24. Zibetti, E., Quera, V., Beltran, F.S., Tijus, C.: Contextual categorization: A mechanism linking perception and knowledge in modeling and simulating perceived events as actions. Modeling and using context (2001) 395–408
25. Nardi, B.A.: A Brief Introduction to Activity Theory. KI – Künstliche Intelligenz (2003) 35–36
26. Bødker, S.: Activity theory as a challenge to systems design. In Nissen, H.E., Klein, H., Hirschheim, R., eds.: Information Systems Research: Contemporary Approaches and Emergent Traditions. North Holland (1991) 551–564
27. Nardi, B.A., ed.: Context and Consciousness. MIT Press, Cambridge, MA (1996)
28. Kutti, K.: Activity Theory as a Potential Framework for Human-Computer Interaction Research. [27] 17–44
29. Mwanza, D.: Mind the Gap: Activity Theory and Design. Technical Report KMI-TR-95, Knowledge Media Institute, The Open University, Milton Keynes (2000)
30. Kaenampornpan, M., O'Neill, E.: Modelling context: an activity theory approach. In Markopoulos, P., Eggen, B., Aarts, E., Croeley, J.L., eds.: Ambient Intelligence: Second European Symposium on Ambient Intelligence, EUSAI 2004. Volume 3295 of Lecture Notes in Computer Science., Springer Verlag (2004) 367–374
31. Kaenampornpan, M., O'Neill, E.: 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). (2005)
32. Li, Y., Hong, J.I., Landay, J.A.: Topiary: a tool for prototyping location-enhanced applications. In: UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology, New York, NY, USA, ACM Press (2004) 217–226
33. Wiberg, M., Olsson, C.: Designing artifacts for context awareness. In: Proceedings of IRIS 22 Enterprise Architectures for Virtual Organisations, Jyväskylä, Dept. of Computer Science and Information Systems, University of Jyväskylä (1999) 49–58
34. Korpela, M., Soriyan, H.A., Olufokunbi, K.C.: Activity analysis as a method for information systems development. Scandinavian Journal of Information Systems **12** (2001) 191–210
35. Fjeld, M., Lauche, K., Bichsel, M., Voorhoorst, F., Krueger, H., Rauterberg, M.: Physcial and Virtual Tools: Activity Theory Applied to the Design of Groupware. CSCW **11** (2002) 153–180

36. Bødker, S.: Applying Activity Theory to Video Analysis: How to Make Sense of Video Data in Human-Computer Interaction. [27]
37. Quek, A., Shah, H.: A Comparative Survey of Activity-based Methods for Information Systems Development. In Seruca, I., Filipe, J., Hammoudi, S., Cordeiro, J., eds.: Proceedings of 6th International Conference on Enterprise Information Systems (ICEIS 2004). Volume 5., Porto, Portugal (2004) 221–229
38. Cassens, J.: Knowing What to Explain and When. In Gervás, P., Gupta, K.M., eds.: Proceedings of the ECCBR 2004 Workshops. Number 142-04 in Technical Report of the Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, Madrid (2004) 97–104
39. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing **5** (2001) 4–7
40. Brézillon, P., Pomerol, J.C.: Contextual knowledge sharing and cooperation in intelligent assistant systems. Le Travail Humain **62** (1999) 223–246
41. Göker, A., Myrhaug, H.I.: User context and personalisation. In: Workshop proceedings for the 6^{th} European Conference on Case Based Reasoning. (2002)
42. Kofod-Petersen, A., Mikalsen, M.: Context: Representation and Reasoning – Representing and Reasoning about Context in a Mobile Environment. Revue d'Intelligence Artificielle **19** (2005) 479–498
43. Aamodt, A.: Knowledge-intensive case-based reasoning in creek. In Funk, P., Calero, P.A.G., eds.: Advances in case-based reasoning, 7th European Conference, ECCBR 2004, Proceedings. (2004) 1–15
44. Aamodt, A.: Explanation-driven case-based reasoning. In Wess, S., Althoff, K., Richter, M., eds.: Topics in Case-based reasoning. Springer Verlag (1994) 274–288
45. Jære, M.D., Aamodt, A., Skalle, P.: Representing temporal knowledge for case-based prediction. In: Advances in case-based reasoning. 6th European Conference, ECCBR 2002, Lecture Notes in Artificial Intelligence, 2416, Springer Verlag (2002) 174–188
46. Aamodt, A.: Knowledge Acquisition and Learning by Experience – The Role of Case-Specific Knowledge. In Tecuci, G., Kodratoff, Y., eds.: Machine Learning and Knowledge Acquisition – Integrated Approaches. Academic Press (1995) 197–245
47. Cassens, J.: User Aspects of Explanation Aware CBR Systems. In Costabile, M.F., Paternó, F., eds.: Human-Computer Interaction – INTERACT 2005. Volume 3585 of LNCS., Rome, Springer (2005) 1087–1090
48. Kofod-Petersen, A., Mikalsen, M.: An Architecture Supporting implementation of Context-Aware Services. In Floréen, P., Lindén, G., Niklander, T., Raatikainen, K., eds.: Workshop on Context Awareness for Proactive Systems (CAPS 2005), Helsinki, Finland, HIIT Publications (2005) 31–42
49. Lech, T.C., Wienhofen, L.W.M.: 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, New York, NY, USA, ACM Press (2005) 625–631
50. Gundersen, O.E., Kofod-Petersen, A.: Multiagent Based Problem-solving in a Mobile Environment. In Coward, E., ed.: Norsk Informatikkonferance 2005, NIK 2005, Institutt for Informatikk, Universitetet i Bergen (2005) 7–18