# A Novel Approach for Creating Activity-Aware Applications in a Hospital Environment

Jakob E. Bardram

IT University of Copenhagen
Rued Langgaards Vej 7, DK-2300 Copenhagen, Denmark
bardram@itu.dk

**Abstract**. Context-aware and activity-aware computing has been proposed as a way to adapt the computer to the user's ongoing activity. However, deductively moving from physical context – like location – to establishing human activity has proved difficult. This paper proposes a novel approach to activity-aware computing. Instead of inferring activities, this approach enables the user to explicitly model their activity, and then use sensor-based events to create, manage, and use these computational activities adjusted to a specific context. This approach was crafted through a user-centered design process in collaboration with a hospital department. We propose three strategies for activity-awareness: context-based activity matching, context-based activity creation, and context-based activity adaptation. We present the implementation of these strategies and present an experimental evaluation of them. The experiments demonstrate that rather than considering context as information, context can be a relational property that links 'real-world activities' with their 'computational activities'.

## 1 Introduction

Context-aware computing has been proposed as a way to design mobile computer applications that adapt according to the physical context of the user, like his or her location. For example, if a mobile phone is inside a cinema, then it is automatically silenced [17]. In the nomadic work environment of a hospital, context-aware technologies seem a promising approach. For example, a physician using a PDA may be presented with relevant medical data on the patient, he or she is approaching [1, 19, 9].

However, deductively moving from physical context – like location – to establishing human activity has proved difficult [7]. The mobile phone in the cinema may belong to a service technician performing a repair activity, in which case silencing the phone is annoying. Similarly, a physician may be in front of a patient in order to prepare some medical equipment, like a catheter insertion, and would be more interested in a clinical guideline for this procedure rather than the patient's medical record [9].

In order to mitigate these challenges, several researchers have proposed to build activity-aware systems, i.e. systems which are able to recognize not only the context of the user, but the overall activity that the user is engaged in. For example, Favela et. al [9] argues that the challenge in the medical scenario is to make better sensor systems and machine learning algorithms for deducing the 'right' activity. Activity-aware computing typically entails sensing physical and digital context like location, time, movement, and usage of e.g. files, and then trying to deduce the overall activity of a person from a synthesis of current and historical sensor input. This synthesis is then used to adapt the computer to the activity of a user. In this paper, we propose another approach to activity-aware computing; instead of trying to deduce activities from lower-level sensor input, users help to explicitly define their activities as part of a computer system. This approach entails a range of benefits, both for system designers as well as for users, because it removes a huge part of the uncertainty associated with activity-aware computing based on activity inferring. It is relevant to note that this approach is not in conflict with context-aware computing and activity inference based on sensor input – as we shall see, our approach also make use of such sensor techniques. However, we view activity-aware computing as a way to apply these techniques within a framework for creating and managing real-world activities, which are defined by the users and not the computer.

## 2 Related Work

There is a wide range of systems and applications that support context-aware adaptation of an application to the specific context of the user. In the medical domain, researchers have been investigating how to build context-aware systems for mobile technologies in hospitals [13, 19, 1]. However, because these approaches to context awareness treat 'context' as a matter of representing context and then deduce human activity from it, they often run into the problem that the scope of contextual features is defined dynamically [7], in which case it is difficult to establish what a context event 'means', i.e. what kind of human activity is going on.

In order to support human activity more directly, a range of approaches seek to support activity management and representation. These are systems like Unified Activity Manager [16], Activity Explorer [18], and Activity Based Computing [3]. Because these systems require users to manually create, maintain, and clean up the computational representations of activities, they provide little or no support for connecting the computer representation of activities to the real-world human activities.

To address these issues, a set of systems has been proposed to automatically generate its task or activity representations from logs of low-level, interaction events. These systems include TaskTracer [8], UMEA [11], Kimura [15], and CAAD [20]. However, these kinds of low-level interaction event with a computer are of little use in e.g. a hospital environment where the important context events are related to the broader clinical context. Another set of systems tries to use extensive sensoring of an environment and run machine learning algorithms to deduce human activities of e.g. daily living (e.g.[14]). In a hospital domain, Favela et al. [9] has presented a neural network system that infers activities of clinicians. Our observations are similar to Favela et al., but we come to different conclusions. Whereas Favela et al. emphasize the need for better sensing and machine learning technology, we rather move in the direction of having the user to help explicitly represent relevant activities and intentions in the computer system.

In summary, there are three types of systems. First, context-aware systems that tries to sense, model, and reason about context, and use this for adapting an application. Second, systems which provide support for activity management. And third, systems for activity deduction and inference. However, there are no systems that uses context events to help users bridge between 'real-world activities' and 'computational activities', i.e. representations of activities which are already modeled and created by the user in the computer system. This latter approach of combining a real-world activity with its computational counterpart is the focus of our approach to activity-aware systems.

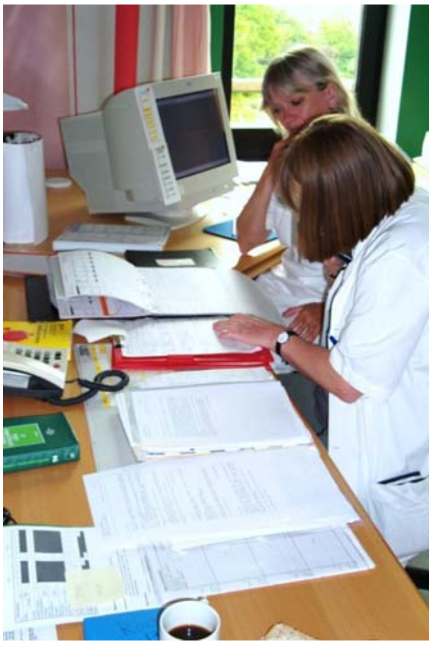

**Fig. 1**. Inside the office at the ward. Coordination and information management done by the physician and the nurse during the ward round.

## 3 Research Background

The present approach to activity-awareness is rooted in a long-term engagement in the design of pervasive computing technologies for hospital work. Clinicians in hospitals attend many concurrent activities in a very mobile, collaborative, and hectic environment, while accessing a wide range of information sources. For example, Figure 1 shows a nurse and a physician preparing for the ward round. In this case, they need to juggle several patient cases while using a large set of data sources and tools: the medical record, the medicine chart, the computerized lab system, the medicine handbook, referral letters, etc. A core challenge to the current work environment is that there is a substantial overhead associated with finding and aligning all the electronic and physical material that is needed in order to discuss a patient case and get ready for e.g. the ward round. Furthermore, once the physician and nurse leave the ward office, they need to get access to this information for each of the patients they may visit at their bedside.

The research site was an internal medicine department, specialized in hematology at a large metropolitan teaching hospital. The work at a hematology department is highly collaborative and involves a wide range of specialized doctors, like pathologists, hematologists, radiologists, and hematopathologists, as well as lab technicians, nurses, and care assistants. In order to investigate how activity-aware technologies could be designed for this time-critical, mobile, and collaborative work setting, we engaged in a user-centered participatory design process, which consisted of three main strands of research: (i) ethnographic field studies of work in hospitals, including participant observation and video analysis [21]; (ii) a series of design workshops applying the future workshop methodology [12]; and (iii) scenario-based evaluation [5] of prototypes in a simulated hospital environment. The project took place over a two-year period with 3two-month periods of ethnographic fieldwork, 12 open-ended interviews, and 11 design and evaluation workshops involving ca. 20 different clinicians from the hospital.

## 4 Designing for Activity Awareness

The field studies at the hematology department showed that clinicians need support for;(i) timely retrieval of large amount of medical information, which is relevant for specific patient activities; (ii) frequent and fast switching between activities; (iii) mobile support for activities; and (iv) continuous multi-tasking. This raised a number of questions on how to create computer support for activities, and especially how to design systems that are somehow aware of these activities and the context in which they take place. This was subsequently the focus of the design process.

The design for activity-aware computing builds upon the Activity-Based Computing (ABC) framework [4, 2, 3]. The ABC framework has been designed to support users– and especially clinicians working in hospitals – to manage a set of parallel and collaborative activities. In the framework, a real-world activity is supported by grouping resources like files, documents, records, and services into a bundle which is called a computational activity – or just ‘activity’. For example, at the hematology department, at least one activity would be associated with each patient, and such an activity would bundle relevant medical data like the medical record, the medicine chart, relevant clinical guidelines, x-ray images, etc. Clinicians access these activities through activity resume and suspend, which enables the user to bring up a computational activity and its associated resources on publicly available displays within the hospital, and dismiss of it once finished using it. This has the effect that patient information relevant for the ongoing treatment is fetched and displayed. An activity can be shared amongst a set of participants which all can resume and use the activity during patient treatment. For example, the physician and the nurse doing the ward round together would share activities on relevant patients, and hence both have access to the medical resources for the patients.

The goal of activity-based computing is to help users manage complex real-world activities that involve heavy information management on computers. In contrast to systems that try to deduce real-world human activity though sensing technology, the approach in the ABC framework is to have users explicitly create, define, and maintain the activities, which they find relevant. The framework has mechanisms for users to define their activities, arrange the computational resources associated with these activities, and manage the set of people who participate in an activity. As such, the

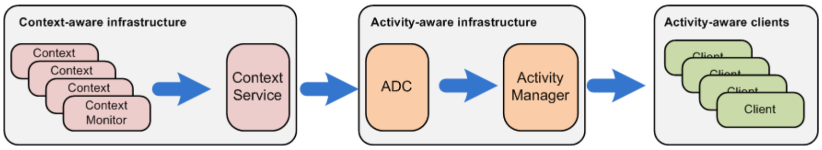

**Fig. 2.** A simplified picture of the ABC activity-aware architecture.

ABC framework has mechanisms for supporting the handling of large bundles of medical information relevant for different patient cases, and provides support for mobility and collaboration.

During the design and evaluation phases, it was, however, realized that a core challenge to the ABC framework was to help users manage the dynamic complexity of these clinical activities; i.e. to help users create, maintain, navigate, and handle such activities, their resources, and participants in order to constantly ensure that the computational activities are aligned with the real-world activities taking place in the work environment. To help users with this task of maintaining an association between real-world activities and their computational counterparts, we made the ABC framework activity-aware by having the framework maintain an awareness of the real-world activities going on, and thereby help the users to define, maintain, and resume/suspend appropriate computational activities. Based on the series of design workshops with the clinicians, we identified three relevant strategies for activity awareness: (i) context-based activity matching;(ii) context-based activity creation; and (iii) context-based activity adaptation.

### 4.1 Context-based Activity

The goal of context-based activity matching is to help users retrieve relevant information in a specific context. This is one of the earliest goals of context-aware computing, and has also been proposed in a medical setting – e.g. displaying the patient's medical record when the physician is near the patient's bed [1, 19]. In general, the goal is to somehow infer the activity of the user and use this for 'activity-aware information retrieval' [9].

From the field studies and the design workshops, it was evident that this kind of activity-aware information retrieval would be highly relevant in order to assist clinicians in handling the vast amount of medical information associated with patient treatment and care. However, taking a closer look, it is actually difficult to establish exactly what we mean by "display[ing] the medical record when a physician is in front of a patient's bed" [9;p.166]. The patient's medical record is in most cases extremely voluminous containing a wide variety of information ranging from hundreds of test results, tens of x-ray images, a very long medical continuation, medicine charts for dozens of medical prescriptions, extensive nursing notes, etc. Clearly we do not want to automatically bring all that up in front of the physician just because he happens to be approaching a patient's bed.

The approach that we propose by the concept of context-based activity matching is not to try to figure out what information is relevant for a specific activity, because this has already been defined by the user in the creation and population of his or her activities. The approach is therefore simply to make a link between specific context events – like the physician's location in front of the patient's bed – to a relevant activity. For example, the presence of the patient may trigger a suggestion to resume the computational activity associated with this patient's treatment, which will bring up the information that the physician has deemed to be relevant for the treatment of this particular patient. Another example is to associate the activity of handling medicine for a patient with the patient's pill tray, i.e. the tray nurses are using to hand out medicine according to the physician's prescriptions. When this pill tray is used in a specific context– e.g. in the medicine room or at the patient's bed side – the ABC framework will suggest the user to resume this 'handling medicine' activity.

The implementation of context-based activity matching makes use of the architectural components illustrated in Figure 2. Information about context is maintained by the context-aware infrastructure, which uses context monitors to sense context entities. An RFID scanner scanning an RFID-enabled pill tray is an example of a context monitor and the pill tray itself is an example of a context entity. Context information is stored and managed by a context service. The activity detection component (ADC) listens to context events in the context service, and based on a mapping between context entities (like the pill tray), the ADC can match context events with one or more activities. If a matching activity is discovered, the activity manager is notified, which again notifies relevant activity-aware clients.

### 4.2 Context-based Activity Creation

The goal of context-based activity creation is to assist users in creating relevant activities in different context. This is relevant because activity-based computing relies on users to create their own computational activities and populate them with relevant resources. Context-based activity creation simply comes up with suggestions for relevant activities including resources and participants. For example, creating 'Hematology Treatment' activities to be used for patients admitted to the hematology department.

The approach proposed by the concept of context-based activity creation is to use generic activity templates. These templates can be used by the user when manually creating a new activity. For example, a 'Hematology Treatment' template would be used by the physician to create an activity for the treatment of a specific patient. These activity templates can, however, also be used for context-based activity creation. The activity templates can take as input current context information about an entity and based on this context, the template populates and returns an instance of an activity. For example, once a patient is being admitted to the hematology department, this event can trigger the creation of a 'Hematology Treatment' for this patient, and the template would be able to fill in all the details about this patient, seek out relevant information in the medical record, set up a medicine chart, and prepare the ordering of relevant lab tests. Once this patient's physician and nurse is known, they will become participants of the activity, and hence be able to access the activity and its associated medical resources. Another example is to use an 'Operation Template' to create activities based on scheduled operations at a surgical ward. This template would populate an operation activity with relevant patient information and information about the operation, and add the scheduled clinicians as participant. This activity can then be used for easy information access during the actual surgical procedure, and for later follow up in the outpatient clinic.

In the ABC framework, context-based activity creation is implemented as part of the ADC. The ADC listens to relevant context events generated by different monitors, and when a certain context event occurs, the ADC uses an appropriate activity template to create a new activity based on the context information. This new activity is then forwarded to the activity manager, which again notifies the activity-aware clients. For example, an 'Operation Scheduler Monitor' may listen to the operation scheduling system. For every new patient being scheduled, the Context Service notifies the ADC which again uses an 'Operation Activity' template to create a new activity, which links to medical data relevant for the surgery for each patient.

### 4.3 Context-based Activity Adaptation

The goal of context-based activity adaptation is to help users to maintain their activities. Activity-based computing fundamentally relies on users to maintain and nurture their computational activities according to their need for supporting different real-world activities. Context-based activity adaptation is aimed at helping users in this maintenance task by using contextual information to continuously adapt the activities used in different context. As also pointed out by Favela et. al [9], the physician may be in front of a patient in order to prepare a catheter insertion, in which case the clinical guidelines and pharmacological databases, rather than patient's health record, may be appropriate in this context.

The approach taken in context-based activity adaptation is to investigate the context in which an activity is being resumed or suspended. If there are resources, services, or information that seems to be relevant in the current situation, these are added to the activity. For example, if we have some knowledge about the presence of a catheter insertion needle, then it would be relevant to add the guideline as a resource to the activity. Another example is when resuming an activity inside the operating room. In the operating room there are different services, which are relevant for the operation activity, but only available in here. For example, different audio and video recording services. These services are added to the activity if the activity is resumed inside the OR, but are not part of the activity otherwise. Similarly, inside the OR different resources like a clinical guideline for this particular type of operation and a patient safety checklist are also relevant in this context, and may thus be added to the activity.

Context-based activity adaptation is implemented using the activity template introduced above. Each activity maintains a link to its originating template, and this template contains the logic, which can adapt the activity according to the context in which it is being resumed or suspended. For example, when an activity based on the surgical template is being resumed inside an operating room, links to resources like video cameras and operating manuals for the equipment inside this specific operating room are added to the activity. These resources are removed again once the activity is suspended and no longer used inside the operating room.

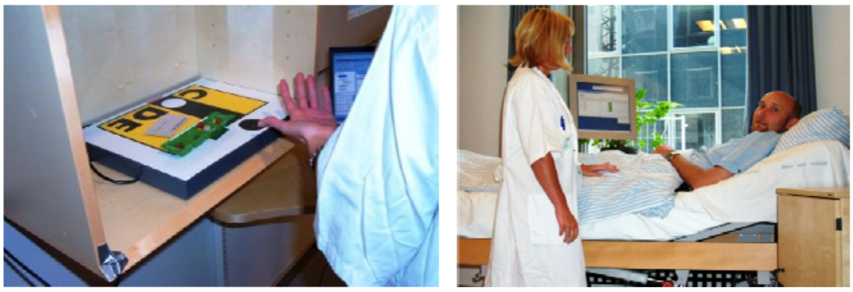

**Fig. 3.** Experimental setup. Left: the ‘Medicine Room’ showing the nurse, a table with an RFID reader, a pill tray on the table, and a display. Right: the ‘Patient Bed’ showing the nurse, the patient, and a build-in display.

## 5 Experiments and Evaluations

The concepts of activity-based computing, activity-awareness, and the three strategies presented above are of an abstract, conceptual nature. In order to establish how these principles could be implemented in use, a range of experiments and evaluations have been conducted. These experiments have been done throughout the two-year period. In total 11 whole-day workshops were conducted in an experimental facility, which was equipped like a medical ward, including bed rooms, a hallway, the ward office, a medical conference room, and a medicine room. In all these places, different public displays were available. For example, a large wall-display in the wards office, a build-in display in the hospital bed (Figure 3, right), and a touch display in the medicine room (Figure 3, left).

The workshops applied scenario-based design and evaluation methods [5], allowing the clinicians to try out different design alternatives. The scenarios were created based on the field studies and refined together with the clinicians before they were enacted. We expected the clinicians to relate primarily to the usability aspects – i.e. issues related to the user interface of our prototypes. We were, however, quite surprised how well we were able to present and discuss the conceptual aspects of our design with them, and we were often engaged in detailed and very sophisticated discussions of appropriate solutions to activity-based computing and activity-awareness in a clinical environment. From a methodological point of view, we hence did not make any formal ‘evaluation’ as such. From a exploratory research perspective, however, we had substantial feedback on our proposed designs, which – we would argue – to a much larger degree helped to establish an appropriate design for activity-aware computing (c.f. [10]).

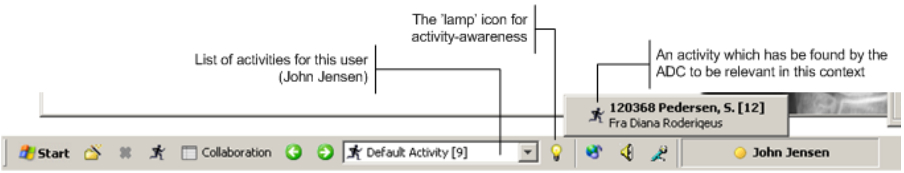
The 'lamp' icon for activity-awareness
List of activities for this user (John Jensen)
An activity which has be found by the ADC to be relevant in this context
120368 Pedersen, S. [12]
Fra Diana Roderiqeus
Start
Collaboration
Default Activity [9]
John Jensen

**Fig. 4.** The Activity Bar showing the ‘lamp’ icon, which is used to list the activities discovered by the ADC as relevant in this context.

### 5.1 Context-based Activity Matching

Context-based activity matching tries to match context events to relevant activities, which in the ABC framework is done through linkage between a context event and a computational activity. In the experimental setup, we created support for this kind of linkage based on the location of different physical objects. This was implemented using passive RFID tagging of various clinical objects, such as the patient, the patient’s paper-based medical record, x-ray photographs, and various medical utensils like pill trays and surgical instruments. In the event that such an object was seen, the ABC framework would notify the user that it had discovered an activity relevant in this context.

Figure 3 (left) shows a picture from one of our workshops where the nurse is preparing medicine inside the medicine room. She is using a pill tray for a specific patient and by placing this pill tray on the table (which has a build-in RFID scanner), the public display in the medicine room provides access to this patient’s medication activity.

One of the open questions was how to present to the user that an activity had been discovered. We experimented with three prototypes, each taking into account the lessons learned from the previous one. In the first prototype, just placing the pill tray on the table would trigger the ongoing activity to be suspended and the activity related to the pill tray (typically an activity on a patient) would be resumed. This had the effect that all previous medical data on the display were removed and medical data related to the patient would be displayed.

This approach of automatically displaying information based on the presence of a tagged physical object has been suggested by a number of researchers [22] and also in the medical domain [19]. Our experiments, however, showed that this was not an appropriate design solution; physical objects may be in proximity of a public display for many reasons and may hence not justify that the user’s current activity is being resumed and disappears from the screen. For example, the nurse working in a medicine room will be handling dozens of pill tray and some of these may be placed on the table just for convenience sake.

Based on this input, the second prototype presented a list of discovered activities as buttons in an ‘Activity Bar’ and the user could select the activity by pressing the corresponding button. Now the user would decide on the activity switching while easy access to the activity was maintained. The clinicians agreed that this solution was to prefer over the first one due to its non-intrusive nature. This activity bar had, however, some limitations regarding the number of activities it could hold. The third design of the prototype therefore listed the set of discovered activities in a pop-up menu accessible from a button on the activity bar. This third prototype implementation is shown in Figure 4 where the ‘lamp’ button blinks if there are any discovered activities. When clicking on the icon, the user has access to a list of activities, which the activity-awareness system deems to be relevant in this context.

A couple of lessons came out of these experiments. First, the users found the notion of linking activities and their associated resources to real-world objects and events very useful. It provided the important connection between the material world of patient treatment and the increasing amount of digital information and services being used as part of patient treatment. Second, the experiments showed that automatic activity switching should be done with outmost caution, and that activity matching with real-world context events should be done in a non-intrusive way.

### 5.2 Context-based Activity Creation

Context-based activity creation aims at creating activities based on relevant context events, which in the ABC framework is done using activity templates. In our experimental setup we created a set of templates, which could generate relevant activities indifferent context. For example, a template for handling medicine for a patient. This ‘medicine handling’ template was used to generate an activity in the medicine room; when the nurse placed a patient’s medicine tray on the table, this template was used to generate a new activity which was populated with resources relevant for medication of this particular patient. Resources were things like the medicine chart, relevant medical guidelines, and the medicine handbook (which holds a list of all available medicine). This activity was then added to the user’s list of discovered activities, and depending on the version of the prototype, the user had access to this activity in different ways as described above.

Another experiment was to use a logic inference engine as part of the activity discovery component (ADC) [6]. This was done in order to extend the one-to-one relationship between a context event and an activity by allowing a set of context events to be parsed into creating an activity. For example, in order to bring up relevant information when the nurse was taking the prepared medicine to the patient’s bedside, the ADC would react on the combination of the right pill tray for the patient, a certified nurse, and the presence of the patient. In this case, the ADC would create an activity, which could display the medicine chart for this patient, and in this chart highlight the medicine in the pill tray. This activity was then shown on the bed’s build-in display as shown in Figure 3 (right). This approach has the benefit of presenting and highlight-ing relevant information in this specific context and furthermore has the benefit of being able to detect, if the nurse is trying to give the patient the wrong pill tray.

The clinicians clearly acknowledged the benefit of increased patient safety during medication administration. However, they did not find the solution to scale to the work in a hospital, because several contingencies could arise even with this rather simple scenario; the patient could be away from the bed, the person giving the medicine may be a nursing assistant, and the pill tray may be missing. Furthermore, the clinicians argued that there was actually no need for generating new activities every time a context event occurred – the activity for handling medicine in the medicine room is the same activity as giving the medicine to the patient at the bedside. This approach would therefore automatically generate a large number of minor computational activities belonging to the same overall human activity. Consequently, the clinicians preferred the simple object-to-activity linking where both the pill tray as well as the patient would link to the overall activity of treating the patient.

These lessons lead to the current design of activity templates that help users create computational activities, which are aligned with real-world activities. One example is a template suited for the medical treatment done at the hematology department, which can be used when a patient is admitted to the department. Another template could retrieve and present a set of x-ray images in a way, which is suited for the radiology conference. The clinicians deemed these kinds of activity templates to be more suitable than the detailed ones regarding e.g. medicine handling.

### 5.3 Context-based Activity Adaptation

An important lesson coming out of our experiments with context-based activity matching and creation is, that context events not necessarily justify activity switching; in many cases there is no need for shifting from one overall activity to another, but rather to adjust the current activity. For example, when the nurse is placing a pill tray on the table she needs to view the medicine chart. However, in the case where the medicine chart for some reason is not part of the patient’s activity, the appropriate course of action would not be to switch activity, but merely to add the medicine chart to the patient’s activity. A similar argument can be made in the case where a physician is using the catheter insertion equipment in front of the patient. In this case, relevant medical guidelines can be added to the patient’s current activity. This strategy ensures that the overall context of the treatment is maintained, but supplemented with resources relevant in this specific situation. Observations like these lead to the design of context-based activity adaptation.

During the workshops we discussed this adaptation approach, and this kind of adaptation of the activity to its use in a specific context was in general appreciated by the clinicians as a solution to some of the issues raised above. Basically, the idea of having fever overall activities that evolve over time was preferred to the idea of creating several smaller activities suited to a specific context. The adaptation involving adding and removing resources and services like the medicine chart was also deemed as less intrusive compared to automatic activity switching; the overall activity was maintained active and only supplemented with additional resources. If these resources

– like the medicine chart – were considered inappropriate in this situation, it was quite easy to just close the medicine chart window. In other words; the consequence of showing additional resources to an activity is more balanced with the accuracy of scanning passive RFID tags.

## 6 Discussion and Conclusion

Traditionally in context-awareness research, context is viewed as a representational problem [7]. This means that context-awareness is inherently restricted to the context information that can be sensed and modeled, and progress in context-aware computing therefore relies on the basic expectation that we can make better sensors, models, and reasoning engines. In this line of reasoning, several researchers have proposed to create context-aware systems for use in hospitals, which reveal patient-related medical data when a clinician is close to the patient [13, 19, 1]. However, despite any technological development in sensing, modeling, and reasoning technologies, context-aware computing will always be limited to what has been sensed – any reasoning is no better that its raw data. Thus – if the patient is not nearby the physician, it is hard to use sensing of the patient as a means of locating the patient's medical record. Or – if the proximity of the patient in the bed wards brings up the medical record, is it then feasible to bring up the medicine chart as well?

Our approach to activity-aware computing is simpler; it basically asks the user to help define the computational activities that match whatever real-world activity he or she is doing, or wants to do. These computational activities embrace and provide easy access to resources, material, and services, which are relevant for performing the corresponding human activity. By having a model of the activity in the computer, the whole context can be restored – also resources which in no way can be deduced from the current context. Hence, physical context events, like the location of the patient, become links into a much richer computational context.

The drawback of this approach is, however, that users need to model, define, create, and nurture these computational activities. To mitigate this drawback, our approach to activity-aware computing helps users to continuously maintain and adapt their activities according to relevant information and events arising from the real-world context in which the corresponding human activities unfold. We have proposed three strategies for supporting activity-aware computing: context-based activity matching, context-based activity creation, and context-based activity adaptation. Through a user-centered design process, these strategies have been implemented in a set of prototypes, which were evaluated experimentally with a group of hospital clinicians.

Several lessons came out of these experiments. First, the support for context-aware activity matching based on linking computational activities with their associated resources and data, to context events from real-world objects were considered a useful and conceptually understandable strategy. It provided the linkage between the highly material world of patient treatment and the increasing amount of digital information and services being used as part of patient treatment. Second, the experiments showed that automatic activity switching should be done with outmost caution, and that activity matching with real-world context events should be done in a non-intrusive way. Third, the experiments showed that more advanced rules regarding the linkage between a set of context-events and a relevant activity may be too complicated to maintain and to use. The clinicians all agreed that having a one-to-one linkage between a physical object and an activity was more appropriate both conceptually as well as in use. Fourth, creating computational activities based on events in the real world was considered very useful for activity management in a hectic and ever-changing environment like a hospital. Based on our experiments, however, we also found that support for e.g. context-aware information display should be tied into the overall activity in which it makes sense; for example, presenting the medicine chart in the medicine room or the medical record at the patient's bedside does not in itself constitute an 'activity', but the medical chart and record are important resources that need to be made available in these settings. This leads to the fifth lesson learned, which revealed that the computational infrastructure should help users to constantly maintain and adapt their computational activities according to the flow of events happening as work unfold in the real world. Activity adaptation involving adding and removing resources and services, like the medicine chart in the medicine room, was deemed as less intrusive compared to automatic activity switching, because the overall activity was maintained active and supplemented with additional material.

The proposed approach to activity-aware computing as presented in this paper has only been implementing simple scenarios with easy detectable objects (like the pill tray) and has been subject to tests in an artificial lab test setup. A concern that can be raised about the application of this approach to the everyday working practices of complex environments, like hospital units, is whether the user intervention required to model the activity in question may become too demanding in terms of costs/benefits ratio, once the activity-based computing system becomes more sophisticated and it comprises several computational activities to be maintained. To address this scalability and complexity question, we are currently undertaking a new implementation of the proposed techniques that aims at handling far more complex context-based activity matching, context-based activity creation, and context-based activity adaptation. This would involve a large amount of real-world activities and be tested inside a hospital.

**Acknowledgments:** Claus Bossen and Henrik B. Christensen were participating in the study and design for hospital work. We are indebted for the enthusiastic participation from the clinicians at the Hematology department at the University Hospital in Aarhus – especially Jørgen S. Christensen. This work was funded by the Danish Research Council.

## References

1. J. E. Bardram. Applications of Context-Aware Computing in Hospital Work - Examples and Design Principles. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, pages 1574–1579. ACM Press, 2004.
2. J. E. Bardram. Activity-Based Computing: Support for Mobility and Collaboration in Ubiquitous Computing. *Personal and Ubiquitous Computing*, 9(5):312–322, July 2005.

3. J. E. Bardram, J. Bunde-Pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *Proceedings of CHI '06*, pages 211–220, New York, NY, USA, 2006. ACMPress.

4. J. E. Bardram and H. B. Christensen. Pervasive computing support for hospitals: An overview of the activity-based computing project. *IEEE Pervasive Computing*, 6(1):44–51,2007.

5. S. Bødker and E. Christiansen. Scenarios as springboards in design. In G. Bowker, L. Gasser, L. Star, and W. Turner, editors, *Social science research, technical systems and cooperative work*., pages 217–234. NJ: Erlbaum, 1997.

6. H. B. Christensen. Using Logic Programming to Detect Activities in Pervasive Healthcare. In *International Conference on Logic Programming, ICLP 2002*. Springer Verlag.

7. P. Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8:9–30, 2005.

8. A. N. Dragunov, T. G. Dietterich, K. Johnsrude, M. McLaughlin, L. Li, and J. L. Herlocker. *TaskTracer: a desktop environment to support multi-tasking knowledge workers*. In *Proceedings of the 10th international conference on Intelligent user interfaces*, p. 75–82. ACM Press, 2005.

9. J. Favela, M. Tentori, L. Castro, V. Gonzalez, E. Moran, and A. Martinez-Garcia. Activity recognition for context-aware hospital applications: Issues and opportunities for the deployment of pervasive networks. *Mobile Networks and Applications*, 12(2):155–171, 2007.

10. S. Greenberg and B. Buxton. Usability evaluation considered harmful (some of the time*)*. In *Proceeding of CHI '08*. p. 111–120, New York, NY, USA, 2008. ACM.

11. V. Kaptelinin. UMEA: Translating interaction histories into project contexts. In *Proceedings of the CHI'03*, p. 353–360. ACM Press, 2003.

12. F. Kensing and K. Halskov Madsen. Generating Visions: Future Workshops and Metaphorical Design. In J. Greenbaum and M. Kyng, editors, *Design at Work: Cooperative Design of Computer Systems*. p. 155–168. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

13. J. Kjeldskov and M. Skov. Supporting work activities in healthcare by mobile electronic patient records. In *Proceedings of the 6th Asia-Pacific Conference on Human-Computer Interaction*, 2004.

14. B. Logan, J. Healy, M. Philipose, E. M. Tapia, and S. Intille. A long-term evaluation of sensing modilitis for activity recognition. In *Proceedings UbiComp 2007*, p. 483–500.

15. B. MacIntyre, E. D. Mynatt, S. Vodia, K. M. Hansen, J. Tullio, and G. M. Support for Multitasking and Background Awareness Using Interactive Peripheral Displays. In *Proceeding of ACM User Interface Software and Technology 2001 (UIST01)*, pages 11–14, Orlando, Florida, USA, Nov. 2001.

16. P. Moody, D. Gruen, M. J. Muller, J. Tang, and T. P. Moran. Business activity patterns: A new model for collaborative business applications. *IBM Systems Journal*, 45(4):683–694,2006.

17. T. Moran and P. Dourish. Introduction to this special issue on context-aware computing. *Human-Computer Interaction*, 16:87–95, 2001.

18. M. J. Muller,W. Geyer, B. Brownholtz, E.Wilcox, and D. R. Millen. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proceedings of CHI '04*. p. 375–382. ACM Press, 2004.

19. M. Munoz, M. Rodriguez, J. Favela, A. Martinez-Garcia, and V. Gonzalez. Context-aware mobile communication in hospitals. *IEEE Computer*, 36(9):38–46, 2003.

20. T. Rattenbury and J. Canny. CAAD: An Automatic Task Support System. In *Proceedings of CHI'07*, p. 687–696. ACM Press, 2007.

21. L. A. Suchman and R. H. Trigg. Understanding Practise: Video as a medium for Reflection and Design. In J. Greenbaum and M. Kyng, editors, *Design at Work: Cooperative Design of Computer Systems*, p. 65–90. Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

22. R. Want, K. P. Fishkin, A. Gujar, and B. L. Harrison. Bridging physical and virtual worlds with electronic tags. In *Proceedings of CHI '99*, p. 370–377, 1999. ACM Press.