A Situated Model and Architecture for Distributed Activity-Based Computing

Steven Houben, Morten Esbensen and Jakob E. Bardram
The Pervasive Interaction Technology Lab
IT University of Copenhagen
{shou,mortenq,bardram}@itu.dk

ABSTRACT
In this position paper, we introduce the *situated activity model*, an activity theory informed activity-based computing (ABC) model that unites the description of *activity* and *situated contexts* into one computational model. By introducing a unified activity model, we seek to connect cross domain activity systems ranging from desktop systems to pervasive computing and beyond. In this paper, we describe (i) the situated activity model (SAM), (ii) a conceptual description of a generic cloud-based architecture for the prototyping and development of situated activity systems (SAS) and (iii) the value of the situated activity approach in different application domains.

Author Keywords
Activity, Activity-Based Computing, ABC, Situated Activity Model, Activity Theory, Context

ACM Classification Keywords
H.5.m Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION
Activity-based computing (ABC) is a term, originally proposed by Apple Research [15], to define a new interaction paradigm that supports activities that users perform rather then the tools they use. The purpose of task- or activity-based computing is to aggregate relevant resources and tools into a higher level structure, activity (or task), that represents an intention of work. Since its first introduction, many different approaches to ABC have been proposed to structure parallel work, context or even augmented interaction in the world. These approaches focus on different areas in Human-Computer Interaction (HCI), ranging from the classic desktop interface [1, 6, 14, 16] and more recently ubiquitous computing [2, 5, 12].

Activity theory [7] has been introduced to the HCI community to theorize the interaction between humans and computers using activity as a unit of analysis. Over the years, activity theory has been refined into models that provide a design space to translate concepts of activity theory to interaction design [9, 4]. In parallel, a bottom-up approach has been used to embed the theoretical notion of activity in work practices through empirical studies, resulting in guidelines and principles for frameworks or systems [1, 16] and meta models [13]. However, in her reflection on second wave HCI theories [3], Suzanne Bødker argues that the application of second wave theories, including activity theory, did not complete its achievements as it failed to describe the shared mediation and context between different *situations*.

Despite the myriad of both theoretical and system approaches to ABC, there is thus no generic activity structure, which implies that there is no unified approach to building or *connecting* ubicomp ABC systems. Because of this, comparing, evaluating, understanding and prototyping systems is difficult. Moreover, there is a gap between the model of activity (in the form of resources, services and contacts) and the use context of activity (in the form of tools or settings). There is thus a fragmentation in both activity *descriptions* and activity *systems*. As an approach to these problems, this paper first proposes the situated activity model (SAM), an activity theory informed ABC model that unites the description of *activity* and *situated contexts* into one computational construct. Second, it describes a cloud - based architecture that can be used for the development of interconnected situated activity systems (SAS). Finally, the paper discusses the potential value of the model and architecture for different application domains.

ACTIVITY THEORY
Activity theory (AT) is a psychological framework that describes human activity as a relation between the subject (S) (human or group that acts in the world), object (O) (which is acted upon and motivates the activity) and the community (C) (or social strata in which the activity is engaged) [7]. The motivation of activity is projected and reflected into an outcome, which is the contribution of activity.

The S-O-C relation is mediated by instruments, rules and division of labour. First, *instruments* provide the subject with a way to act in the world. They externalize the act in the world through enactment and are shaped by affordance and resistance. Second, *rules* define how the act of the subject is embedded in the social context. It socializes the act in the environment, culture and world. Third, *division of labour* structures the relation between the social strata and the object of the activity. It links the distribution of work among community to the hierarchical motive towards the object.

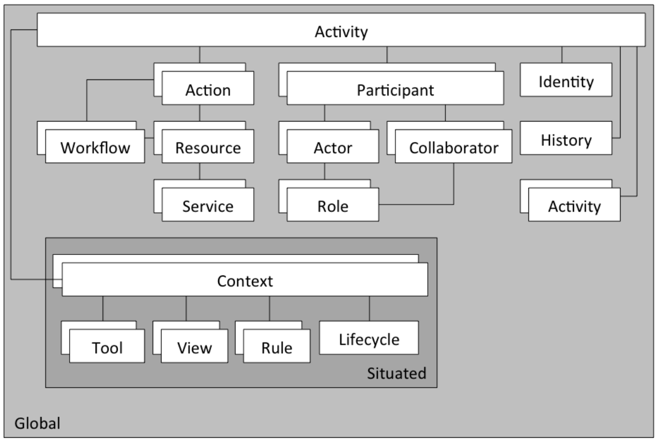

Figure 1. The situated activity model (SAM) is computational representation of activity that describes both global and situated context information.

Activity is not a fixed structure but a dynamic hierarchical interaction between the activity itself driven by motivation, conscious goal-directed actions and unconscious operations that are performed when certain conditions are met. Additionally, Engeström [8] categorized four fundamental processes that are an interwoven into this hierarchy. These processes are: (i) production, (ii) consumption, (iii) exchange and (iv) distribution. In summary, activities are why we do something; actions are what we do it; and operations are how we do [11].

As activity theory moves into its third generation, it has become clear that the unit of analysis is expanding from an individual analysis to a global analysis that comprises not only the individual, community, and artefacts but also the interconnectivity between activity systems. The focus is on networks of interacting activity systems, the dialogues between these systems and the multiple perspectives of these networks of activity [8]. Activity is not an isolated unit of analysis but an integral part of the psychological synthesis of life.

SITUATED ACTIVITY MODEL

The main purpose of Activity-Based Computing (ABC) systems is to lower the amount of configuration work needed to complete a task. We define *configuration work* as the amount of work needed to locate, open and arrange all necessary resources required to complete the objective of an activity. In traditional computing, the user is responsible for the (re-) configuration and maintenance of the workspace to fit the needs of the ongoing activities. ABC systems however can support this *configuration* of activities on three different levels:

1. Interlinking: The first level of configuration is the linking and logical association of activities with actions, resources, actors and community. At this level the resources and requirements of the activity are defined.

2. Situating: The second level situates the deployed activity in the local context through the setting. Several aspects of an activity, defined by the setting of the deployment, are highly coupled with the local context, and this configuration level accounts for this. The situating level describes how the activity can be used in a given context or situation.

3. Visualizing: The third level applies a visualization to the activity. As activities can be consumed on very different devices, different visualization techniques are available. This final level defines how the context of activity is presented.

The situated activity model (SAM) (Figure 1) draws from the basic concepts of AT to provide a computational activity representation that extends the existing activity-based computing [1] model with situated context. It thus merges both the descriptions of the *interlinking* and *situating* configuration level into one unified model that can be used to describe and visualize system-mediated activities on different platforms or systems. The model makes a distinction between *global* and *situated* information.

Global Activity Information

The global activity information is in many ways similar to the original activity-based computing model [1]. It is composed of several interlinked subcomponents that define the content of the activity. Each activity is subdivided into a set of actions, which are subtasks that are part of the activity. Actions structure how actors interact with the different *resources*, such as files and folder, and *services*, such as web services. Additionally, actions can be modelled as *workflows*, which are structured or unstructured sequences of actions that are imposed or defined by the participants.

Participants are human agents that are digitally represented in the system as part of the activity. Participants can be both *actors* and *collaborators*. Actors interact with the system, motivated to complete the *objective* for which the system was designed. They act on the system by using tools that are relevant and related to the activity. Although actors are part of the activity, they own, shape, define, consume, and share activities by interacting with the system. Collaborators are secondary actors that are not directly involved in the *situation* of the interaction of the activity, but contribute to its relevance. They represent external stakeholders that influence and define the object of the activity. Both the capabilities of actors and collaborators are defined through roles. Roles define what actions are accessible and executable by a participant.

Since every activity is an evolving structure, it is embedded into its own history. Each event that occurs within the activity is logged and stored into the activity itself for persistence and reflection. History can be used to track changes in parts the activity, create an awareness on different aspects of the activity or simply to visualize the evolution of the activity. Each activity is uniquely defined by an identity which consists of meta data such as a name, image or description and a unique reference number (e.g. GUID[1]). An activity can be connected to other activities thereby creating a hierarchical relationship or references between activities.

[1]Globally Unique Identifier

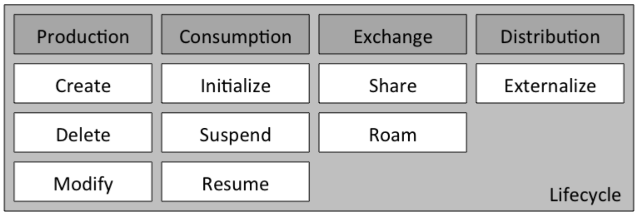

Figure 2. The subprocesses in a SAM lifecycle.

Situated Activity Information

Because a deployment of an activity description is always *situated* and thereby defined by context (environment and social setting in which the activity is used), it is also described by additional context dependent subcomponents. Note that an activity can have multiple contexts depending on different deployments. Tools represent both physical and digital artifacts that allow actors to interact with digital resources accessible through the system. A tool can thus be an application that is shown on a screen or a sensor network that is gathering information about the actor. Besides tools, actors also use physical artifacts that are transformed into tools by augmentation of the system. Tools are the local interface to the actions and digital resources of the activity and thus determine how the activity is consumed.

The setting in which the activity is engaged, is the environment, system or situation in which the activity occurs. It is represented to the user through a view of the setting, providing actors with a level of intelligibility. The view is a mental affordance and interactive tool that describes and demonstrates the capabilities of the system in the setting to the actor. A wall-mounted display or glyph [10] for example, can be a view to represent the capabilities of the setting for the current activity in a pervasive environment while an activity dock or taskbar can be used as a view for desktop systems. The view thus describes how the activity is situated and how it can be consumed. Rules define the policy and access to workflows or actions and are used by the local system to determine how the activity is handled.

Each deployed activity has a lifecycle that determines how the context of the activity is handled by the situated system. The lifecycle consists of the four processes identified by Engeström [7]. The processes are: (i) production (create, delete and modify), (ii) consumption (initialize, suspend and resume), (iii) exchange (share and roam) and (iv) distribution (externalize) (Figure 2).

BUILDING SITUATED ACTIVITY SYSTEMS

By allowing for the deployment of different activity systems that can be interconnected, activities are not confined within one system but can be consumed in all interconnected systems through adaptation of the context. To make the network of interconnected activity systems concrete on a system level, we propose a lightweight but standardized toolkit that can be used to design, prototype and develop interconnected situated activity systems (SAS). The purpose of the toolkit is to provide a lightweight but scalable framework for the development of activity-centric systems. Conceptually, the architecture of the toolkit is composed of two major components: (i) the activity cloud and the (ii) the situated activity system (Figure 3).

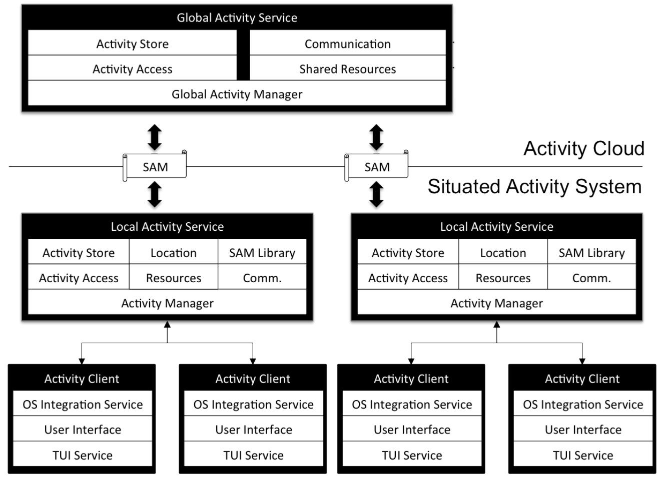

Figure 3. The architecture of the Activity Cloud Toolkit.

Activity Cloud

The main purpose of the activity cloud is to provide a central activity management system that connects different situated activity systems (SAS) to a cloud-based global activity service. The global activity service is composed of different cloud services that are accessible through the global activity manager. Local activity systems can access the cloud manager through a REST-based publish subscribe mechanism. The activity store is used to store SAM descriptions of different SAS. These descriptions are accessible by all connected situated activity systems based on the rules defined in the activity access service. These rules grant local activity systems with authorization to access, modify or manage stored activities. The activity access service also defines how activities and their resources are synchronized with the local activity service. Based on the specification of the local SAS, the cloud synchronization methods can be adjusted. The activity cloud also provides a mechanism to store resources that are shared or exchanged between different SAS's as it allows for the storage of data through a Binary Large Object (Blob) storage. The access to these shared files is managed by the activity manager and should be defined in the local SAS's

Situated Activity System

A situated activity system (SAS) is an SAM-based interaction design system that is composed of two main parts: a local distributed activity service and activity clients. Although both components are architecturally separated, they can be physically used on one device. The local activity service provides support for local activity roaming, sharing and access over all devices that are part of the situated activity system.

The distributed activity manager has a local activity store and access mechanism that has two purposes. First, it is used to manage the activities that are used and shared by different local activity clients. All activity clients are thus connected to a central repository that handles synchronization, distribution and location tracking of the clients. Second, the local activity service is connected to the global activity service for persistence and real time updates from outside the activity system. Additionally, the local service also provides support for real-time communication between different clients. All devices that are part of the activity system are connected to the local activity manager through an activity client. This client itself is composed of: (i) activity integration services for specific platforms or devices that merge the activity representation into the existing experience, (ii) an activity or task-based user interface (ABC/UI) design and (iii) a tangible or wearable computing layer that connects physical objects with the activity client based on the SAM.

The three components, Global Activity Service, Local Activity Service, and Activity Client correspond in their roles to the three configurations levels as described earlier. The Global Activity Service provides global accessible storage and access to SAM descriptions of different SAS. The Local Activity Service is able to situate activity models in the context of the deployment - e.g. through location tracking information - as defined in the second configuration level. Finally, the Activity Client provides a UI to visualize the activity as described in the third configuration level.

ONGOING AND FUTURE WORK

We are currently in the process of developing the basic toolkit and underlying activity cloud infrastructure. The goal is to test the toolkit (both the model and architecture) by building ABC support for two very different domains: nomadic work in a hospital and global software development collaboration. We choose these settings as previous and current research in the work practices of these domains, allows us to gain great insight in to how work is structured and conducted. Furthermore, we will be able to test our systems in real use settings for end-user validation.

Global Software Development

Global Software Development (GSD) is a development method where the production of software is carried out in multiple locations. The geographical displacement of software teams in GSD however introduces physical, temporal, and cultural challenges. One of the main methods to overcome these challenges is to make extensive use of groupware technologies that allows for collaboration across distances. We have identified a number of different aspects of GSD for which we see an applicability for SAM:

- With the physical distance between teams, all resources should be shared on the network. We can allow this using the cloud activity manager.
- There are clear situated activity systems in GSD, namely those identified with the different physical locations. These different systems however are highly interlinked and share an outcome. We can model the systems through different local activity systems, all connected to the activity cloud.
- An activity-centric approach to GSD allows for the linking of different design artifacts, e.g. source code, design documents or project plans, in to activities
- All physical items, such as PC's tablets, digital Scrum boards etc. can be connected to the local activity manager allowing all time access to relevant information.

Workflow in Hospital Units

Based on observations in the hospital, we describe the workflow of clinicians in a patient ward as nomadic. Nomadic workflow – as compared to mobile work – essentially means that clinicians, instead of being able to carry a laptop or other devices and sit at desks in different location, roam through different departments of the hospital while doing their work. Hence, their work is heavily influenced by collaborations with other clinicians, they regularly use public and shared devices and their general work pattern is characterized by planning and re-planning. Additionally, next to digital artefacts, clinicians also use a large set of medical tools and objects that play an important role in the care of patient.

- Because the entire workflow in a hospital is focused towards patients and their care, it maps on the activity-centric approach of the SAM. All patient information as well as the tools to provide the care can be united into one meaningful structure.
- The nomadic workflow implies that patient information should be ubiquitous, interconnected with stakeholders and available on different locations at different times. The activity cloud can be used to provide this global availability of information. Additionally, it can also be used to provide new activity-centric functionality such as remote patient access or monitoring.
- On a local level, the general patient information can be deployed as a situated activity system. These systems can be an entire department but also a single room that is hosting multiple patients. Because of the link to the global activity cloud, these systems can interact with each other, thereby simplifying collaboration and providing a level of awareness on the patient beyond the local department.
- Physical objects that are related to the patient, such as blood samples, paper documents or medication, can be connected to the activity of the patient.

CONCLUSION

In this paper, we propose a new model and architecture to prototype and develop activity-centric systems. The situated activity model (SAM) unites the description of *activity* and *situated contexts* into one computational representation and can be used to model system-mediated activity. We discussed the theoretical ground of the model, the conceptual architecture of the toolkit and their potential application in different domains. Future work includes supporting multiple platforms, refining the toolkit and deploying situated activity systems in different domains to validate the approach.

ACKNOWLEDGEMENTS

This work was supported by the EU Marie Curie Network iCareNet under grant number 264738.

REFERENCES

1. J. Bardram, J. Bunde-Pedersen, and M. Soegaard. Support for activity-based computing in a personal computing operating system. In *Proc. of CHI '06*, pages 211–220. ACM.
2. J. E. Bardram, J. Bunde-Pedersen, A. Doryab, and S. Sørensen. Clinical surfaces - activity-based computing for distributed multi-display environments in hospitals. In *Proc. of INTERACT '09*, pages 704–717. Springer-Verlag.
3. S. Bødker. When second wave hci meets third wave challenges. In *Proc. of NordiCHI '06*, pages 1–8. ACM.
4. S. Bødker. *Applying Activity Theory to Video Analysis: How to Make Sense of Video Data in Human-Computer Interaction*, chapter 7, pages 148–174. MIT Press, 1996.
5. S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, and J. e. a. Froehlich. Activity sensing in the wild: a field trial of ubifit garden. In *Proc. of CHI '08*, pages 1797–1806. ACM.
6. 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 *Proc. of IUI '05*, pages 75–82. ACM.
7. Y. Engestrøm. *Learning by expanding: an activity-theoretical approach to developmental research*. Orienta-Konsultit Oy., 1987.
8. Y. Engeström. Expansive Learning at Work: toward an activity theoretical reconceptualization. *Journal of Education and Work*, 14:133–156, 2001.
9. V. Kaptelinin, B. A. Nardi, and C. Macaulay. Methods & tools: The activity checklist: a tool for representing the "space" of context. *interactions*, 6(4):27–39, 1999.
10. F. Kawsar, J. Vermeulen, K. Smith, K. Luyten, and G. Kortuem. Exploring the design space for situated glyphs to support dynamic work environments. In *Proc of Pervasive'11*, pages 70–78. Springer-Verlag.
11. J. Lave. *Cognition in Practice*. Cambridge University Press, 1988.
12. Y. Li and J. A. Landay. Activity-based prototyping of ubicomp applications for long-lived, everyday human activities. In *Proc of CHI '08*, pages 1303–1312. ACM.
13. T. P. Moran. Unified activity management: Explicitly representing activity in work-support systems. workshop paper at ecscw. In *Proc. of ECSCW 2005*.
14. 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 *Proc. of CHI '04*, pages 375–382. ACM.
15. D. A. Norman. *The invisible computer : why good products can fail, the personal computer is so complex, and information appliances are the solution*. 1. mit press paperback ed edition, 1999.
16. S. Voida and E. D. Mynatt. "it feels better than filing": Everyday work experiences in an activity-based computing system. In *Proc. of CHI '09*, pages 259–268. ACM Press.