Activity-Based Computing: Computational Management of Activities Reflecting Human Intention

Jakob E. Bardram, Steven Jeuris, Steven Houben

■ *An important research topic in artificial intelligence is automatic sensing and inferencing of contextual information, which is used to build computer models of the user's activity. One approach to building such activity-aware systems is the notion of activity-based computing (ABC). ABC is a computing paradigm that has been applied in personal information-management applications as well as in ubiquitous, multidevice, and interactive surface computing. ABC has emerged as a response to the traditional application- and file-centered computing paradigm, which is oblivious to a notion of a user's activity spanning heterogeneous devices, multiple applications, services, and information sources. In this article, we present ABC as an approach to contextualize information, and present our research into designing activity-based computing technologies.*

In today's world of ubiquitous computing, an abundance of information is constantly at hand wherever you go, whenever you desire. This can make it challenging for users to find the actual information they need in between these vast amounts of resources. Additionally, unrelated information can easily distract users from the original work they set out to do, leading separate activities to intertwine. Purposefully multitasking or not, there is an overhead associated with managing different parallel activities. The presence of too much information negatively affecting the user's work is commonly referred to as information overload.

Providing contextualized and relevant information to the user can alleviate this problem, hiding irrelevant data while pushing important information to the foreground. However, the question remains what this "context" is composed of. Historically, *context* is a broadly interpretable term usually encompassing the where, when, who, how, or why of any given situation.

Different areas of research are trying to provide contextualized information to the user, each approaching the problem from a different angle. Although there are clear overlaps among fields, a discrepancy on focus and terminology is apparent. Personal information-management (PIM) research tries to empower users by providing extensive tool support by

 ISSN 0738-4602

which to manage and access their information. Human-computer interaction (HCI) follows a more user-oriented approach, where the user is usually placed central during system design. Artificial intelligence (AI) research focuses on making systems smarter, making them context-aware and even responsive to ongoing activities using activity recognition. In other words, PIM, HCI, and AI emphasize information, users, and automation respectively. These different perspectives all lead to useful insights and solutions to what is in essence a common problem.

In this article, we present our approach to contextualizing information called activity-based computing (ABC), applying methods and insights gained from all three fields. We reflect back on 10 years of research of this particular approach, providing an overview of the state of the art of ABC and its relationship to research in AI. ABC defines activity as the context associated to human intention, thus also including the cognitive context. By definition, what defines a concrete activity is user specific, since it relies on which intention is expressed by or is meaningful to the user. Although a standardized and common ontology has still not been established, the basic model of ABC has not changed over the years and rather has been extended to explore different aspects of what constitutes an activity in more detail. We have applied the notion of ABC in system support for many different domains, including personal information management in an office setting, mobile and collaborative work in hospitals, wet-lab research in biology labs, and in software engineering. This abundance of empirical evidence has provided us with a deep insight into the benefits and challenges of applying the concepts and technologies of ABC. We use this as the basis for a discussion of recurring issues in ABC, and how this relates to research in AI.

Context and Human Intention

A common approach to providing contextualized information to the user is context-aware computing. Its traditional definition is intentionally broad:

> A system is context-aware if it uses context to provide relevant information or services to the user, where relevancy depends on the user's task. (Dey, 2001, p. 5)

An early criticism against context-aware computing is "there are human aspects of context that cannot be sensed or even inferred by technological means" (Bellotti and Edwards 2001). A key challenge is detecting human intention — what drives users to perform certain actions or tasks. Context-aware systems cannot be designed always to act correctly on our behalf. To prevent possible conflicts due to incorrect presumptions of human intention, intelligibility has been brought forward as a design principle.

> *Intelligibility* — Context aware systems that seek to act upon what they infer about the context must be able to represent to their users what they know, how they know it, and what they are doing about it. (Bellotti and Edwards, 2001, p. 201)

We argue that taking into account human intention is not only important for context-aware systems, but for any system that provides contextualized information, automated or not. Having digital support of users' activity contexts, aligned with the user's mental model of them, allows for new interaction techniques otherwise not possible. Although activity recognition traditionally focused on detecting low-level human actions, more recent work also focuses on capturing goal-oriented stateful activities (Brdiczka and Bellotti 2011). The resulting activity awareness could allow systems to respond to users' activities in meaningful ways. For example, by monitoring and detecting so-called activities of daily living (ADL) of patients in a nursing home, early warning signals can be triggered automatically when irregular behavior is detected (Tentori and Favela 2008). Within ABC, human intention and its surrounding context are part of the main perspective considered during system design.

Activity-Based Computing

A large number of observational studies show that users often structure their work within the context of higher-level activities in desktop environments (González and Mark 2004; Boardman and Sasse 2004; Bergman, Beyth-Marom, and Nachmias 2006). Users thus not only reason within the context of activities but sometimes also feel a need to externalize them when using current computing systems. Although this can be seen as a necessity due to limitations of current computing systems, it also has some advantages. Manually subdividing work within the context of activities makes them more explicit, which supports episodic memory during later revisitation (Whittaker 2011).

In the early 1980s Bannon et al. (1983) described a vision in which the computer provided "an alternative organization of user commands which preserves their task specificity." One of the earliest implementations of this vision was the Rooms system (Henderson and Card 1986). Activity-based computing (ABC) as an interaction paradigm was originally coined by Apple Research (Norman 1999), which makes activities first-class computational objects. Apple never published this or implemented it in any of its solutions, but subsequently a number of activity-centered systems have been researched, each focusing on different application domains and technological aspects. Figure 1 provides an overview of many of these systems, which we will introduce and discuss next.

Our research group has been researching ABC for more than 10 years with a special focus on providing ABC support for ubiquitous computing (Christensen and Bardram 2002). The central goal is to provide a computing platform that allows the user to focus on

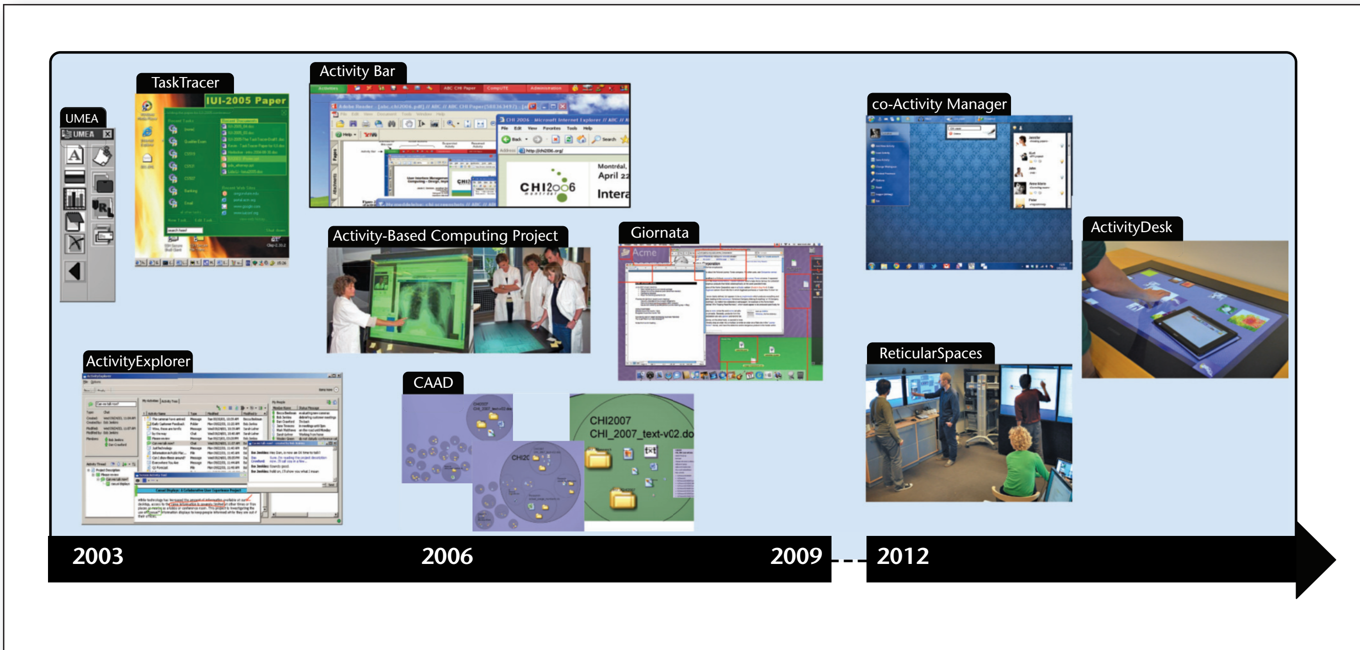

Figure 1. A Historical Overview of Activity-Based Computing Systems Since 2003.

higher-level collaborative activities rather than low-level application and data management. We situate this goal in Mark Weiser's original ideas of transparency and computation as a ubiquitous background resource. In a ubiquitous computing world, where users are using a multitude of heterogeneous computing devices, the need for supporting the users at the activity level becomes essential.

ABC Principles

Our research on ABC has been crystallized into six ABC principles (Bardram 2009). These principles are grounded both in theoretical models of human cognition and activity, as well as in empirical research involving the design and evaluation of ABC technologies and applications. Although the principles themselves evolved over time from their original definition, the core concepts remain unchanged.

Activity-Centered

Work is organized into activities, which are higher-level computational constructs that encapsulate all resources, tools, and communication mechanisms into one goal-oriented interaction model. By moving away from classic application-oriented interfaces to multidevice activity-oriented work spaces, users are presented with logical units of work combined with the tools required to perform that work.

Activity Multiplexing

By supporting activity suspension and resumption, users can easily switch between different activity contexts. Suspending an activity means its state is stored and removed from the active work space, while resuming an activity restores it. This feature supports parallel activities (multitasking) and interruptions in work.

Activity Roaming

Activities are stored in an infrastructure and hence can be accessed from multiple devices. This allows a user to suspend an activity on one device and resume it on another, thereby allowing the user to roam between devices. The context of an ongoing activity can also be spread across devices, allowing for multidevice interaction on one common activity context. Users are presented with awareness cues and overviews on the distributed state and accessibility of the activities.

Activity Adaptation

Activities adapt to the capabilities of the device(s) on which they are resumed. Hence, an activity might look quite different whether it is resumed on a wall-sized display or on a smartphone. A subset of an activity's context can be displayed when it is spread across several devices.

Activity Sharing

Because activities are distributed, they can also be shared among users. Shared activities can be accessed and modified by all related participants. Accessing activities simultaneously allows for synchronous collaborative setups. Alternatively, asynchronous exchange of information is possible when separate users suspend and resume an activity. By attaching messages or other objects to the activity, all related participants are notified of changes, thus providing users with awareness about what changed and on who is working on what activity.

Context-Awareness
Since activities are computational constructs that transcend a single user or device, they need to be aware of their usage context such as location, type of device, amount of users and other factors. The process of detecting, selecting and managing the activity and its resources is a semi-automatic process involving both the users as well as automated sensing and inferencing of context information.

Activity-Based Technologies and Systems

Over the years, a wide range of activity-centered computing technologies and applications have been build. In this section we provide an overview of systems that were designed with the ABC principles in mind, or adhere to them to a large extent. We will discuss the systems within the particular context they were designed for, demonstrating the generalizability of the model.

Desktop Systems

Since the seminal work on Rooms (Henderson and Card 1986) many desktop systems supporting activities have been described. UMEA (Kaptelinin 2003) is a first example of a system that monitors users' behavior within self-defined projects. Activity context is built up automatically within the currently selected activity, providing the user with an overview of interaction history.

TaskTracer (Dragunov et al. 2005) and CAAD (Rattenbury and Canny 2007) use advanced data collection frameworks to monitor the user's interaction with the system, automatically creating activity representations. Although the work of defining activities is automated, TaskTracer still allows for manual construction as well.

Activity Explorer (Muller et al. 2004), the Activity Bar (Bardram, Bunde-Pedersen, and Soegaard 2006), and Giornata (Voida, Mynatt, and Edwards 2008) are examples of systems that fully rely on users to define meaningful activities. Giornata and the Activity Bar reframe the desktop interface for personal computers to be activity-centered, for OS X and Windows XP respectively. They provide activity-centered management and sharing of context like windows, files and contacts by integrating with the traditional desktop operating system. Activity Explorer (Muller et al. 2004) is an example of a multiuser communication and collaboration tool resembling an email client. Although being an external application which is less integrated with the operating system, it was the first system to show that you can meaningfully structure communication and collaboration processes within shared activity abstractions.

Extending our research on the Activity Bar, we have recently introduced the co-ActivityManager (Houben et al. 2013). The co-ActivityManager is shown in figure 2 and is an activity-centered desktop system supporting three main features; (1) activity-centered computing in dedicated activity work spaces; (2) activity sharing and collaboration through shared status updates, sharing of resources, and online communication; and (3) activity management through a dedicated activity task bar. The goal of the system is explicitly to integrate communication and collaboration channels into activity-centered computing support.

The co-ActivityManager was deployed for a period of 14 days in a multidisciplinary software development team. The study showed that the activity-centered work space supports different individual and collaborative work configuration practices and that activity-centered collaboration is a two-phase process consisting of an activity sharing and per activity coordination phase. An analysis of the activities created by users showed different granularities of goals and time spans used. Participants organized activities in three categories: (1) ad hoc activities (such as a to-do), (2) short-term activities (for example day to day work) and (3) long-term activities (for example, ongoing collaborative projects). Despite this new insight into how users appropriate different granularities of activities within activity-centered systems, it is still unclear how these different activities relate to each other and how they fit into the entire shared life cycle of daily work.

Ubiquitous Computing Systems

ABC has also been applied to ubiquitous computing and distributed user interface systems, demonstrating its merit as a context model in a multidevice environment. When moving away from a single user or device scenario to a more pervasive environment many of the multitasking and interruption problems are greatly amplified.

Our initial research into activity-based computing took its outset in the design of a ubiquitous computing infrastructure to support the nomadic, collaborative, and time-critical work in hospitals (Christensen and Bardram 2002; Bardram 2009). This research introduced an activity-based infrastructure consisting of distributed middleware and specialized user interfaces optimized for medical work. All services, applications, and resources related to patient care are bundled in distributed activities. For example, medical records, information on the medicine administered, and medical images for a patient are linked together in an activity, which is shared across clinicians involved in the patient's treatment and care. This infrastructure was extended to incorporate large interactive displays as depicted in figure 3 (Bardram et al. 2009) and automatic activity detection (Doryab, Togelius, and Bardram 2012), resulting in increased activity awareness.

More recently, we have extended this research to create support for distributed activities across multi-

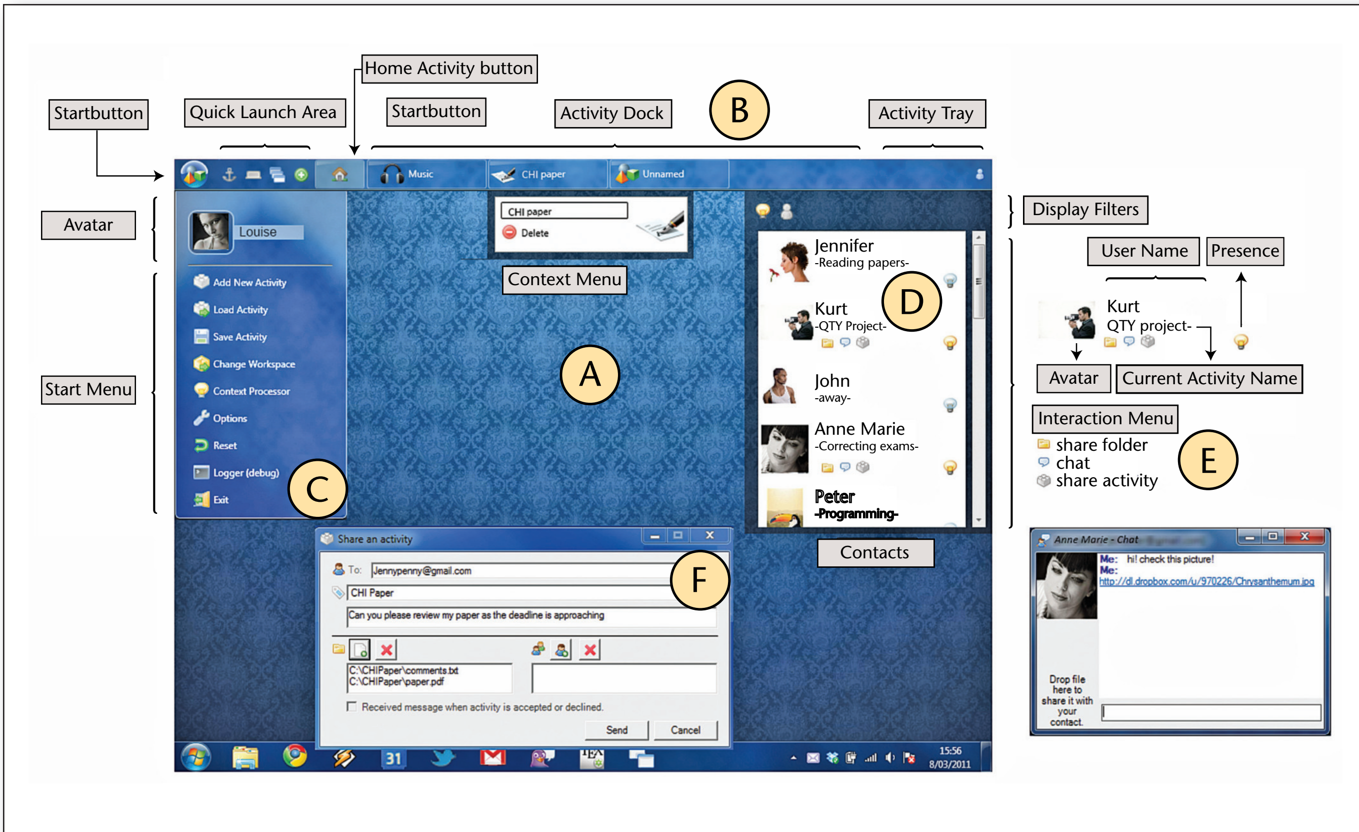

Figure 2. The Interface of Co-ActivityManager.

This interface (Houben et al. 2013) consists of (A) a per activity work space, (B) an activity task bar to visualize activities to the user, (C) an activity start menu to manage activities and applications, and (D) a collaboration manager to interact and share with contacts. Each contact is visualized with an avatar, name, and status field. The interaction menu (E) can be used to share a folder, chat, or share an activity (F).

ple devices in the ReticularSpaces infrastructure (Bardram et al. 2012). ReticularSpaces is an activity-based smart-space system designed to support unified interaction with applications and documents through ReticUI, a novel distributed user interfaces design; management of the complexity of tasks between users and displays; mobile users in local, remote, or nomadic settings; and collaboration among local and remote users. ReticularSpaces was deployed in a smart-space environment and was exposed to end users through scenario-based evaluation.

The study showed that users found the use of activities intuitive in a multidevice context as it allowed them to move tasks between devices and collaborating users. However, it also highlighted a number of open issues with device-specific visualization of activities and cross-device interaction. While ReticUI duplicated the same user interface on all devices with only small adaptations, users argued that this adaptation should go much further and that interfaces as well as information density should be much more tailored to the type of device.

Interactive Surface and Multidevice Interaction

Mobile devices, such as smartphones and tablets, have become an intrinsic part of people's everyday life. Together with laptops and desktop computers, these devices have become part of a device ecology in which each device acts as a specialized portal into users' personal or shared information space. The user-device mapping is quickly changing from being a one-to-one to a one-to-many or even to a many-to-many relation. However, using multiple devices introduces a configuration overhead as users have to manually reconfigure all devices according to ongoing activities. Especially in an environment such as an office (figure 4), where the use of multiple devices is more common, the process of configuring them in context of ongoing activities is cumbersome.

The ActivityDesk system shown in figure 5 (Houben and Bardram 2013) is our latest research in support for multidevice management. ActivityDesk is an activity-centered interactive desk that supports

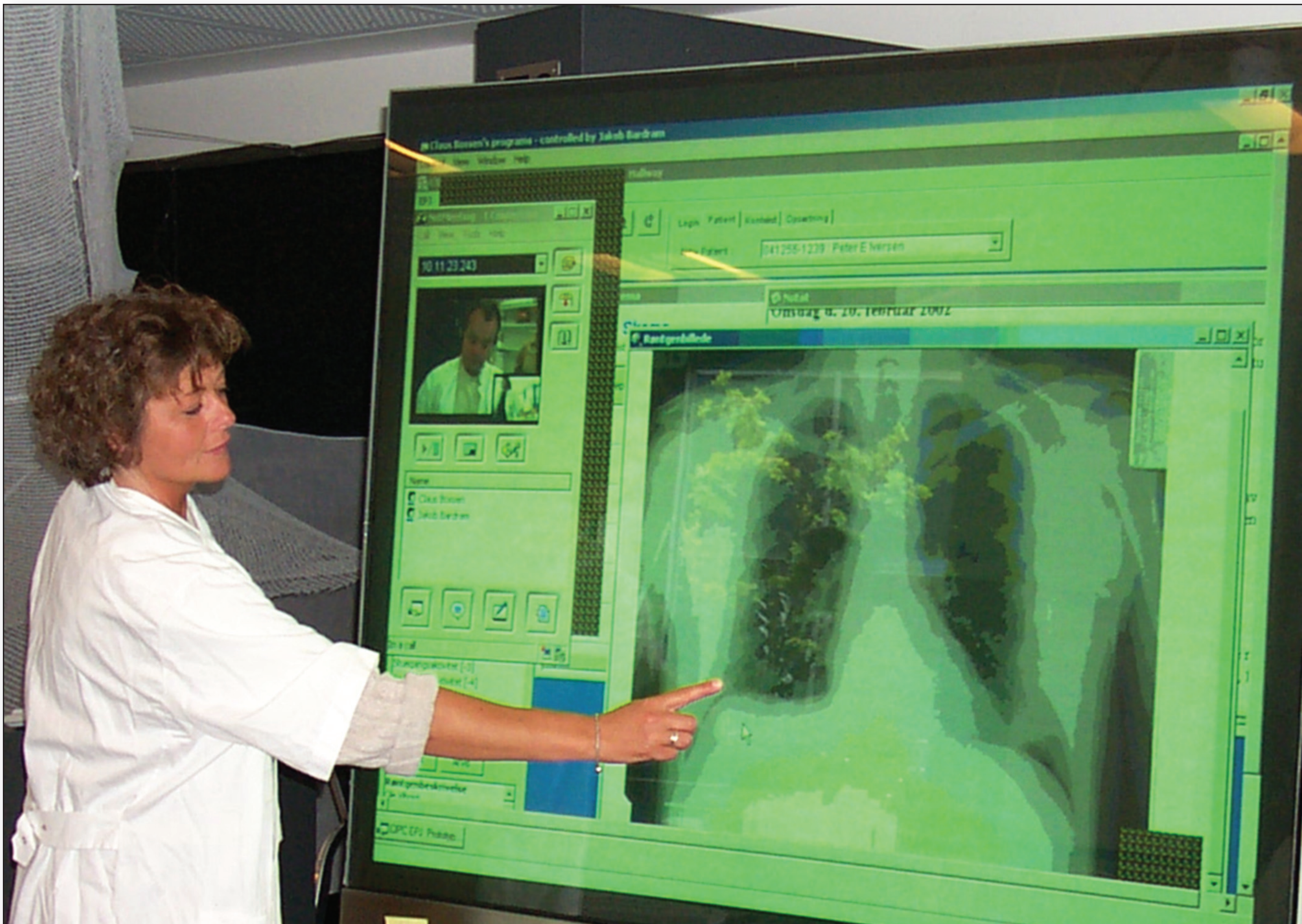

Figure 3. The Original ABC Project.

This original project (Bardram 2009) focused on supporting mobile and collaborative workflows in hospitals using activity as an organizational unit to structure patient data. This image shows a nurse engaged in an online activity sharing session (including video) with a radiologist while sharing resources like X-ray images on a large interactive display.

multidevice configuration work and work-space aggregation into a personal ad hoc smart space for knowledge workers. The main goal of ActivityDesk is to reduce the configuration work required to use multiple devices simultaneously by using an interactive desk as an activity-centered configuration space. Through ActivityDesk a shared work space across devices can be set up in a tangible and visible way, after which information is automatically distributed across all participating devices.

Lessons Learned

Based on our research on ABC and the design of different activity-centered systems we are able to derive some common findings. While ABC has been shown to be particularly advantageous in certain settings, there are still several recurring questions that remain unanswered. In this section we provide an overview of our main findings so far and the open issues that will need to be addressed in future work.

Benefits

Over the many research projects, the core idea of activity-centered computing has proven to be extremely robust; significant improvements in human-computer interaction are achieved by allowing users to organize computational resources into logical bundles of activities, which can be subject to distribution, sharing, adaptation, and suspension and resumption.

Activity Appropriation

Users found computational activities a useful construct to organize information and communication processes. They reported being more focused and appreciated the ability to switch quickly between different information contexts. In desktop environments, we observed that users generally closed fewer

Figure 4. RecticularSpaces.

This figure (Bardram et al. 2012) depicts an activity-based smart-space setup, comprising large wall-based displays, horizontal tabletop displays, laptops, and tablet computers that all run ReticUI, the unified user interface.

resources, since they could simply be hidden by switching to another activity. Activities within ABC allow for additional ways by which to organize work, on top of traditional organizational strategies. Systems such as co-ActivityManager do not impose a formal activity model on users but rather provide them with an additional activity abstraction that can be used to organize and structure work on their desktop. Activities were created both up front and retrospectively as activities evolve over time. Post hoc activity creation usually occurs when an ongoing activity context becomes too large and is split into multiple activities. Users appropriate activities differently, including differences in intention and duration. Activities were created for short undefined ad hoc work, to long-running collaborative projects.

Long-Term Use

There is a learning curve associated with using activities. Initially most users are reluctant to create too many activities as they see the effort of doing so greater than the estimated benefits. However, after having used an ABC system over longer periods of time and having experienced the advantages first hand, users start to create activities even for smaller one-hour tasks. Structuring work within the context of activities becomes especially interesting once more elaborate work needs to be done. All users who were confronted with multitasking on a daily basis preferred ABC over the traditional approach.

Collaboration

Users reported that the process of constructing and sharing activities with each other helped them reflect on their work and that of their collaborators. Activity management and sharing thus increases the awareness of users about ongoing processes within the working context.

Within a collaborative setting, users found that sharing information and setting up collaborative processes in the context of an activity significantly helped them manage ongoing work. Users would, for example, asynchronously share an activity (which in a desktop configuration contained files and contacts) with other users to create a shared starting point for a collaborative project. After this initial sharing process, the scope and definition of the activity can be changed and appropriated by all users depending on their role inside the project. Communication and collaboration channels such as logs, chat windows,

Figure 5. The ActivityDesk System.

ActivityDesk (Houben and Bardram 2013) is an activity-based information space for knowledge workers that aggregates different devices and their containing resources into one interactive desktop space.

or shared folders can be included in the activity representation.

Multidevice

Within a distributed setting where information is spread across several devices, activity-centered computing proved to provide additional advantages to users. A computational unit representing an activity across devices and locations offers a more consistent mental model to the user. Information is automatically distributed across devices within the work context it relates to, removing the overhead of having to transfer resources manually. Devices can become activity visualizers that are part of a larger activity ecology rather than independent computing entities. Activities can be replicated across all devices to support a synchronized view, but additionally activity resources can also be divided among several devices, allowing for cross-device activity representations. Different roles can be allocated to devices. For example, in ActivityDesk the desk itself becomes a master activity manager that manages the rendering of activities and resources on "slave" devices.

Open Issues

Our work with ABC has, however, also left us with a range of open questions, which we would like to address — potentially in collaboration with researchers working in the field of AI. These issues are in particular concerned with the modeling of human activity and how activities are managed and handled in daily use.

Human Intent

Ideally, computational activities reflect user intent. For example, in a hospital setting, an activity-centered electronic medical record would provide support for patient-related activities, like a "prescribe medicine for patient Hansen" activity. Such an activity would help a physician to locate and bring up relevant information on a public display or on a mobile tablet computer. This, however, requires the system to know what is relevant in this activity, and — as argued in the introduction — in order to prevent possible annoyances due to incorrect presumptions of human intention, intelligibility needs to be supported.

During our close collaboration with users (including hospital clinicians), it is evident that activity-based computing support should avoid the need for users to provide lengthy descriptions whenever they define a new activity. Even the relatively simple action of naming an activity when it is created is often too distracting. Coupling activity with intent needs to be light weight or should be automated. For example, light-weight support could involve suggesting activity descriptions based on names of resources used during post hoc activity construction. Depending on the scenario (such as a critical hospital environment) even fully automated activity construction might be preferred.

Early work on task-based computing (Sousa and Garlan 2002) actually argued that given more sophisticated context monitoring, the less the task-based computing system has to rely on explicit indications from a user concerning their intentions. Thus intelligent context-aware and activity-recognition systems can help identify and capture user activity and intent on the fly. Early in our research, we experimented with contextual triggering of activities (Christensen 2002), but this can be extended to have automatic sensing of contextual information, which then is used for semiautomatic generation and maintenance of the activity model in the computer system.

Activity Life Cycle

One particularly challenging part of the ABC principles is the light weightness of activities. Because an activity carries no semantic, it is often hard to distinguish one activity from another. One of the recurrent observations was that one activity merged into another — there were no clear demarcations between the ending of one activity and the beginning of another. For example, while prescribing medicine to one patient, a physician would look up medical data for another patient. This was a source for much confusion, as, for example, an activity labeled "Prescription for Mr. Hansen" would end up also displaying medical data for Mrs. Pedersen. Furthermore, it was not always easy to judge when to create a new activity. Should a new activity be created per patient, or per type of activity (such as prescribing medicine)?

Adding semantic information to activities is an obvious challenge for data mining and data processing approaches. For example, the linkage between an activity, its main object (such as a specific patient), and associated data and tools could be maintained through continuous data mining techniques. Moreover, semantic technologies would help maintain consistent semantics across the ABC system.

Organizing and Managing Activities

Another conceptual as well as practical challenge with the ABC principles concerns scalability; it is hard to tell how well the ABC principles would scale based on our current research systems and implementations. A real-world deployment of an activity-centered electronic medical record system in a modern hospital would be required to handle a significant number of patients, users, physical artifacts, and real-world activities. It is by no means straightforward to scale the current research systems to these numbers. If not carefully designed, a user would in no time accumulate a significant number of activities that are being outdated at a fast pace. Hence, automatic tools and methods for handling, linking together, and navigating in a complex web of activities are needed. Furthermore, ways of cleaning up more or less automatically are essential. All of this calls for automated data mining, learning, and pattern-matching techniques.

Conclusion

In this article we presented activity-based computing as an approach to contextualize human-computer interaction. We have surveyed findings from more than 10 years of research in ABC systems: how users perceive computational activities, how they define them, which ones they find useful, and how they affect their work. Our current research includes further investigation of the activity life cycle, and providing support to organize and manage large collections of shared activities as part of a collaborative workflow. In particular, we are exploring how globally distributed software development can benefit from ABC's principles.

Creating contextualized information systems is a multidisciplinary challenge. To this end we are looking more into what we can learn from related fields. We hope by providing an overview of ABC and its open issues, new cross-disciplinary insights can be gained. Empirical results within personal information management can tell us which PIM tools are used and how they relate to the activity life cycle. In particular we see opportunities for the AI community to automate parts of activity management, possibly addressing some of the open issues. However, likewise we argue for areas where explicit user processes should be maintained. We suggest a hybrid approach of explicitly defined activities, enhanced by AI to predict and suggest activity operations. This can reduce activity construction costs while simultaneously supporting episodic memory by making operations more explicit.

Acknowledgments

The research on activity-based computing has been supported by several partners, including the Danish Council for Strategic Research and the EU Marie Curie Program under grant number 264738 (the iCareNet network).

References

Bannon, L.; Cypher, A.; Greenspan, S.; and Monty, M. L. 1983. Evaluation and Analysis of Users' Activity Organization. In *Proceedings of the 1983 ACM SIGCHI Conference on Human Factors in Computing Systems,* 54–57. New York: Association for Computing Machinery.

Bardram, J. E. 2009. Activity-Based Computing for Medical

Work in Hospitals. *ACM Transactions on Computer-Human Interactaction* 16(2) Article 10.

Bardram, J. E.; Bunde-Pedersen, J.; Doryab, A.; and Srensen, S. 2009. Clinical Surfaces: Activity-Based Computing for Distributed Multi-Display Environments in Hospitals. In *Proceedings of the Conference on Human Computer Interaction,* volume 5727 of Lecture Notes in Computer Science, 704–717. Berlin: Springer.

Bardram, J. E.; Bunde-Pedersen, J.; and Soegaard, M. 2006. Support for Activity-Based Computing in a Personal Computing Operating System. In *Proceedings of the 2006 ACM SIGCHI Conference on Human Factors in Computing Systems,* 211–220. New York: Association for Computing Machinery.

Bardram, J. E.; Gueddana, S.; Houben, S.; and Nielsen, S. 2012. ReticularSpaces: Activity-Based Computing Support for Physically Distributed and Collaborative Smart Spaces. In *Proceedings of the 2012 ACM SIGCHI Conference on Human Factors in Computing Systems,* 2845–2854. New York: Association for Computing Machinery. dx.doi.org/10.1145/2207676.2208689

Bellotti, V., and Edwards, K. 2001. Intelligibility and Accountability: Human Considerations in Context-Aware Systems. *Human-Computer Interaction* 16(2–4): 193–212. dx.doi.org/10.1207/S15327051HCI16234_05

Bergman, O.; Beyth-Marom, R.; and Nachmias, R. 2006. The Project Fragmentation Problem in Personal Information Management. In *Proceedings of the 2006 ACM SIGCHI Conference on Human Factors in Computing Systems,* 271–274. New York: Association for Computing Machinery.

Boardman, R., and Sasse, M. A. 2004. Stuff Goes into the Computer and Doesn't Come Out: A Cross-Tool Study of Personal Information Management. In *Proceedings of the 2004 ACM SIGCHI Conference on Human Factors in Computing Systems,* 583–590. New York: Association for Computing Machinery. dx.doi.org/10.1145/985692.985766

Brdiczka, O., and Bellotti, V. 2011. Identifying Routine and Telltale Activity Patterns in Knowledge Work. In *Proceedings of the IEEE 2011 Conference on Semantic Computing,* 95–101. Piscataway, NJ: Institute for Electrical and Electronics Engineers.

Christensen, H. B. 2002. Using Logic Programming to Detect Activities in Pervasive Healthcare. In *Proceedings of the 2002 International Conference on Logic Programming,* volume 2401 of Lecture Notes in Computer Science, 421–436. Berlin: Springer. dx.doi.org/10.1007/3-540-45809-3_8

Christensen, H. B., and Bardram, J. E. 2002. Supporting Human Activities — Exploring Activity-Centered Computing. In *Proceedings of the 2002 International Conference on Ubiquitous Computing,* volume 2498 of Lecture Notes in Computer Science, 107–116. Berlin: Springer.

Dey, A. K. 2001. Understanding and Using Context. *Personal Ubiquitous Computing* 5(1):4–7. dx.doi.org/10.1007/s007790170019

Doryab, A.; Togelius, J.; and Bardram, J. 2012. Activity-Aware Recommendation for Collaborative Work in Operating Rooms. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces,* 301–304. New York: Association for Computing Machinery. dx.doi.org/10.1145/2166966.2167023

Dragunov, A. N.; Dietterich, T. G.; Johnsrude, K.; McLaughlin, M.; Li, L.; and Herlocker, J. L. 2005. TaskTracer: A Desktop Environment to Support Multi-Tasking Knowledge Workers. In *Proceedings of the 10th International Conference on Intelligent User Interfaces,* 75–82. New York: Association for Computing Machinery.

González, V. M., and Mark, G. 2004. Constant, Constant, Multi-Tasking Craziness: Managing Multiple Working Spheres. In *Proceedings of the 2004 ACM SIGCHI Conference on Human Factors in Computing Systems*, 113–120. New York: Association for Computing Machinery. dx.doi.org/10.1145/985692.985707

Henderson, Jr., D. A., and Card, S. 1986. Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface. *ACM Transactions on Graphics* 5(3): 211–243. dx.doi.org/10.1145/24054.24056

Houben, S., and Bardram, J. E. 2013. ActivityDesk: Multi-Device Configuration Work Using an Interactive Desk. In *CHI 2013 Extended Abstracts on Human Factors in Computing Systems,* 721–726. New York: Association for Computing Machinery.

Houben, S.; Bardram, J. E.; Vermeulen, J.; Luyten, K.; and Coninx, K. 2013. Activity-Centric Support for Ad-Hoc Knowledge Work: A Case Study of Co-Activity Manager. In *Proceedings of the 2013 ACM SIGCHI Conference on Human Factors in Computing Systems,* 2263–2272. New York: Association for Computing Machinery.

Kaptelinin, V. 2003. UMEA: Translating Interaction Histories into Project Contexts. In *Proceedings of the 2003 ACM SIGCHI Conference on Human Factors in Computing Systems,* 353–360. New York: Association for Computing Machinery.

Muller, M. J.; Geyer, W.; Brownholtz, B.; Wilcox, E.; and Millen, D. R. 2004. One-hundred Days in an Activity-Centric Collaboration Environment Based on Shared Objects. In *Proceedings of the 2004 ACM SIGCHI Conference on Human Factors in Computing Systems,* 375–382. New York: Association for Computing Machinery. dx.doi.org/10.1145/985692.985740

Norman, D. A. 1999. *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. Cambridge, MA: The MIT Press.

Rattenbury, T., and Canny, J. F. 2007. CAAD: An Automatic Task Support System. In *Proceedings of the 2007 ACM SIGCHI Conference on Human Factors in Computing Systems,* 687–696. New York: Association for Computing Machinery. dx.doi.org/10.1145/1240624.1240731

Sousa, J. P., and Garlan, D. 2002. Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments. In *Proceedings of the Third Working IEEE/IFIP International Conference on Software Architecture,* volume 224 of IFIP Conference Proceedings, 29–43. Dordrecht, Netherlands: Kluwer.

Tentori, M., and Favela, J. 2008. Activity-Aware Computing for Healthcare. *IEEE Pervasive Computing* 7(2): 51–57. dx.doi.org/10.1109/MPRV.2008.24

Voida, S.; Mynatt, E. D.; and Edwards, W. K. 2008. Re-framing the Desktop Interface Around the Activities of Knowledge Work. In *Proceedings of the ACM SIGCHI Conference on User Interface Software Technology,* 211–220. New York: Association for Computing Machinery.

Whittaker, S. 2011. Personal Information Management: From Information Consumption to Curation. *ARIST* 45(1): 1–62.

Jakob E. Bardram (itu.dk/people/bardram) is a professor at the IT University of Copenhagen (ITU) and runs the Pervasive Interaction Technology (PIT) Lab. His research interests are ubiquitous computer systems, object-oriented software architecture, computer-supported cooperative work (CSCW); and human-computer interaction (HCI). The main application area of this research is within health care, especially pervasive health care.

Steven Jeuris is a Ph.D. student in the PIT Lab at the IT University of Copenhagen, working on next-generation technologies for global software development (NexGSD) where he applies ABC principles throughout the life cycle of software development to improve tool integration and knowledge transfer between coworkers.

Steven Houben is a research associate at the Intel Collaborative Research Institute on Sustainable and Connected Cities (ICRI-Cities) and UCL Interaction Centre (UCLIC) working on multidevice interaction techniques, physical computing and sensor-based systems..