Activity-Centric Computing Systems

Jakob E. Bardram*
Technical University of Denmark
Richard Petersens Plads
Kgs. Lyngby DK-2800
jakba@dtu.dk

Steven Jeuris
Technical University of Denmark
Richard Petersens Plads
Kgs. Lyngby DK-2800
sjeu@dtu.dk

Paolo Tell
IT University of Copenhagen
Rued Langgaards Vej
Copenhagen DK-2300
pate@itu.dk

Steven Houben
Lancaster University
South Drive, Campus of Lancs.
Lancaster LA1 4WA
s.houben@lancaster.ac.uk

Stephen Voida
University of Colorado Boulder
Boulder, Colorado CO 80309-0315
svoida@colorado.edu

ABSTRACT

Key Insights

- Activity-Centric Computing (ACC) addresses deep-rooted information management problems in traditional application-centric computing by providing a unifying computational model for human goal-oriented 'activity,' cutting across system boundaries.
- We provide a historical review of the motivation for and development of ACC systems, and highlight the need for broadening up this research topic to also include low-level system research and development.
- ACC concepts and technology relate to many facets of computing; they are relevant for researchers working on new computing models and operating systems, as well as for application designers seeking to incorporate these technologies in domain-specific applications.

KEYWORDS

Activity-Centric Computing, Systematic Literature Review, Multi-tasking

ACM Reference format:
Jakob E. Bardram, Steven Jeuris, Paolo Tell, Steven Houben, and Stephen Voida. 2019. Activity-Centric Computing Systems. In *Proceedings of Communication of the ACM, , ??? 2019 (CACM),* 8 pages.
DOI: 10.475/123_4

1 INTRODUCTION

Mobile, ubiquitous, social, and cloud computing have brought an unprecedented amount of information, digitised resources, and computational power—spanning many different devices—to users today. Correspondingly, an increasing amount of work and leisure activity is taking place in this distributed digital computing environment. For example, in a hospital, the medical record and bio-signals of patients are digitised and accessed by multiple stationary, mobile, and wearable devices. At home, digital and social media, email, photo libraries, and the like are accessed on a wide range of devices including laptops, smartphones, TV sets, and other internet-connected appliances. However, this rapid increase in the diversity and volume of both computational devices and digital content quickly introduces corresponding organizational challenges, leading to digital clutter. Many people feel overwhelmed and burdened by organizing and retrieving their digital resources, which includes handling, organising, and finding information—a problem commonly referred to as information overload. Moreover, handling multiple and often concurrent tasks while coordinating with other individuals adds an additional level of complexity.

Despite the overwhelming success of new devices and cloud-based information-sharing infrastructures, the evolution of the user interface models that people use to interact with these innovations and the representations with which they organize electronic information on these platforms has not kept pace. Although it is much more common for users to access information through the browser or on a mobile device today than in the past, most contemporary user interface models are still fundamentally grounded in the personal computer metaphor, as part of which electronic resources are defined by the applications used to manipulate them and "filed" using a desktop metaphor (files, folders, and application windows). This application- and document-centric model leads to a fragmentation of a person's information. For example, information related to a specific work project is often scattered across multiple files, local folders, cloud folders, and across different applications such as email, instant messaging, local and cloud-based document editors, web browsers, and social media channels/communities. Moreover, this information might be scattered across different devices and accessed by multiple users.

While cloud-based technologies allow users to access and share files and documents online and access them across different devices—including cloud-dedicated devices like the ChromeBook—such technologies have overwhelmingly maintained use of the files-and-folders model for presenting resources and applications to users, even when new capabilities such as tag-based (e.g., Gmail) and graph-backed (e.g., Microsoft 365) information management schemes

*Corresponding author.

CACM,

DOI: 10.475/123_4

are emerging. Moreover, the introduction of cloud-based services have for most users just added yet another set of services, applications, and accounts for them to handle, and has in practice added yet another layer of information fragmentation and overload to the picture. Thus, even though touch-based phones and tablets look and feel different, the personal computing model, with its focus on applications (or apps), is still in many cases the dominant means for working with electronic information.

Researchers have argued that these problems call for a fundamentally new abstraction for interaction between people and computers. In this paper, we review a specific approach in which 'activities' become a new computational abstraction around which interaction occurs. An activity is an ongoing effort in a person's life towards a goal. For example, an activity can be a work project, writing a research paper, implementing a feature in software, designing a new product, planning an event, treating a patient, or preparing for a vacation. Activities reflect goals that people want or need to achieve in the real world, and a real-world 'activity' can be represented in the computer as an abstraction of computational data, resources, tools, applications, services, etc. which are needed in order for users to perform this activity.

There are different approaches to realising *Activity-Centric Computing* (ACC) systems. (See the sidebar for two conceptual models proposed for ACC). However, a common theme in these approaches is to mitigate information fragmentation and overload by integrating resources (e.g., information), services (e.g., applications), devices, and users into an activity 'bundle' that ties these four layers together. A representation of computational activities is illustrated in Figure 1. For example, in a software development project, a debugging activity could encapsulate: (i) a number of source code files, unit tests, and test documentation [resources], (ii) a source code editor, a debugger, a terminal window, and a bug reporting system [services], (iii) a twin-display debugging setup and several different smartphone configurations for testing [devices], and (iv) the tester with the two developers who are working on this feature [users].

The idea of ACC is not novel. In fact, many researchers who studied the original personal computer model argued early on that computing systems should provide high-level support for activities. In 1983, Liam Bannon and colleagues observed that "*[c]urrent human-computer interfaces provide little support for the kind of problems users encounter when attempting to accomplish several different tasks in a single session*" [1]. They proposed moving away from computing environments built around applications and files as first-class computational constructs, focusing rather on the higher-level *activities* that people perform on computers.

Since then, quite a lot of research has been done on ACC technologies, ranging from research on user interface management technologies to more fundamental distributed middleware and operating system components to support ACC systems. In order to provide a historical and comprehensive overview of the state-of-the-art in ACC research, this paper presents a systematic review of the research literature on ACC systems and technologies and provides an outlook of their potential and the main implementation challenges in applying these approaches to contemporary and emerging computing environments.

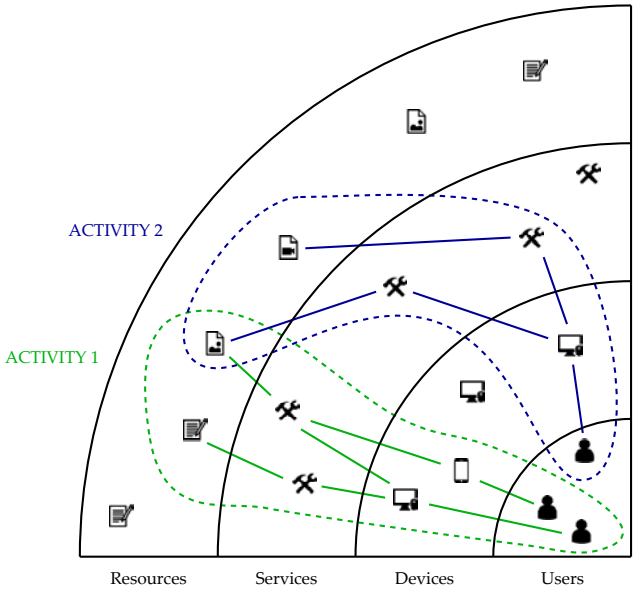

Figure 1: The four layers in computing considered during the design of Activity-Centric Computing systems.

2 A REVIEW OF ACC SYSTEMS

A three-step procedure was followed to identify a comprehensive corpus of ACC research papers from the computing literature, which were further processed for data extraction. First, the authors—all of whom have contributed substantively to the ACC research domain–identified an initial set of publications (N=38) that we agree accurately represents core ACC research. Second, we applied a backward snowballing technique, adding all articles cited by all of the papers in our initial set (N=984). Third, after pruning out all duplicates, we screened all retrievable publications identified in the second step to focus on only those publications presenting *"technologies with a design motivated by the idea of supporting computational activities"* as described earlier. This process yielded to the selection of 101 primary studies and the identification of 68 unique technologies[1]. 58.4% of these papers refer to what we call 'activity' as 'activity', whereas 35.6% refers to 'tasks', and the remainder use other terms, such as 'project'.

The following coding schemes emerged during the data extraction process. Each primary study was labeled with tags related to 'motivation' and 'system type'. Motivation was extracted by analyzing what kind of challenge(s) each paper stated that it was addressing, whereas system type was extracted by analyzing the technological contribution(s). Disagreements and ambiguities in coding were resolved in meetings involving all authors.

Figure 3 shows the coding schemes and distribution of both motivation and system type contributions. Note that each paper may be labeled with multiple tags. For example, the article "Activity-based computing for medical work in hospitals" [4] presents an application, a middleware infrastructure, and a smartspace system.

[1]To the CACM editors and reviewers: We would like to include, or link to, this list of 101 papers. Either by putting a URL into the paper or by adding supplementary material to the ACM Digital Library. Please advise on how to best do this.

SIDEBAR – Conceptual Models for Activity-Centric Computing

The goal of ACC is to replicate the multifaceted and complex nature of human activities in the "real world" in a computational representation. ACC systems do not provide another application, service, or collaboration tool, but rather integrate existing tools across devices, people, services, and resources in a manner that reflects the real-world activity being done. The design challenge in ACC is to create activity representations that are simple, yet flexible enough to accommodate different levels of rigidity [6, 14]. To achieve this, different conceptual models for ACC systems have been proposed, of which the Activity-Based Computing (ABC) and the Unified Activity Management (UAM) models are the most elaborate ones.

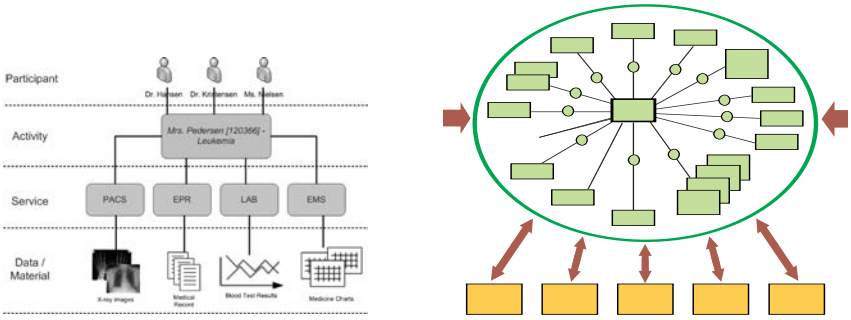

Figure 2: Left: The Activity-Based Computing model. Right : The Unified Activity Management semantic model.

Activity-Based Computing (ABC) – In ABC, a 'computational activity' (or just 'activity') is a computerised representation of a real-world human activity [4]. The purpose of the computational activity is to reflect the human activity and to provide access to resources relevant to its execution. The ABC approach was developed to support hospital work and can be used to model the work done as part of treating patients. As illustrated in Figure 2 (left) a computational activity aggregates and links services, resources, documents, and users that are relevant to the real-world activity of treating Mrs. Pedersen for leukemia. Among other things it gives access to the patient's medical records, medicine charts, and medical images. Access to these materials is mediated by the respective computer systems involved: the electronic patient record system (EPR); the electronic medication system (EMS); and the picture, archiving, and communication system (PACS). Hence, ABC extends computational support 'upwards' from the level of application and document to the level of the overall activity. In the ABC model, this is called 'Activity-Centered Resource Aggregation', which is the first of six core design principles:

- **Activity-Centered Resource Aggregation** – Aggregation of relevant resources, services, applications, documents, data, and users in a one logical bundle. This principle supports information and task management.
- **Activity Suspension and Resumption** – Suspending an activity means its state is stored and removed from the active workspace, while resuming an activity restores it. This principle supports multitasking and interruptions in work.
- **Activity Roaming** – Activities are stored in an infrastructure and hence can be accessed from multiple devices. This allows suspending an activity on one device and resuming it on another device. This principle supports mobility across multiple devices.
- **Activity Adaptation** – When an activity roams (migrates) from one device to another, it adapts to the runtime and resources available on the local device. This principle supports mobile code execution which can take advantage of technical resources like processing power, memory, network, and display size.
- **Activity Sharing** – Activities are per default shared and can be accessed, used, and modified by all users who are 'participants' of the activity. This principles supports collaboration, including access control.
- **Activity Awareness** – Computational activities are always representations of real-world activities and these representations need to build and maintain an 'awareness' of—i.e., knowledge about—this real-world context. This principles supports context-aware adaptation to the users' (work) context.

Unified Activity Management (UAM) – Developed by IBM Research, UAM specifies a semantic model as a unified model for integrating formal business processes with the informal collaboration needed to accomplish business objectives [17]. In UAM, an 'activity description' articulates the actors (people) and roles involved, the resources used (tools, artifacts, people), the results produced, the events it is bounded by, and its relationships to other activities (such as sub-activities or dependent activities). All the people involved can see the activity descriptions and they can modify and extend the descriptions. An 'activity' is metadata, i.e. the glue tying together system resources around the generic semantics of an activity. In a reference implementation of UAM, activity descriptions were implemented in semantic web technology using RDF and OWL. Figure 2 (right) summarizes UAM in which activity representations are managed in an RDF-based Activity Metadata Repository that integrates information from various external services like email, calendars, etc.

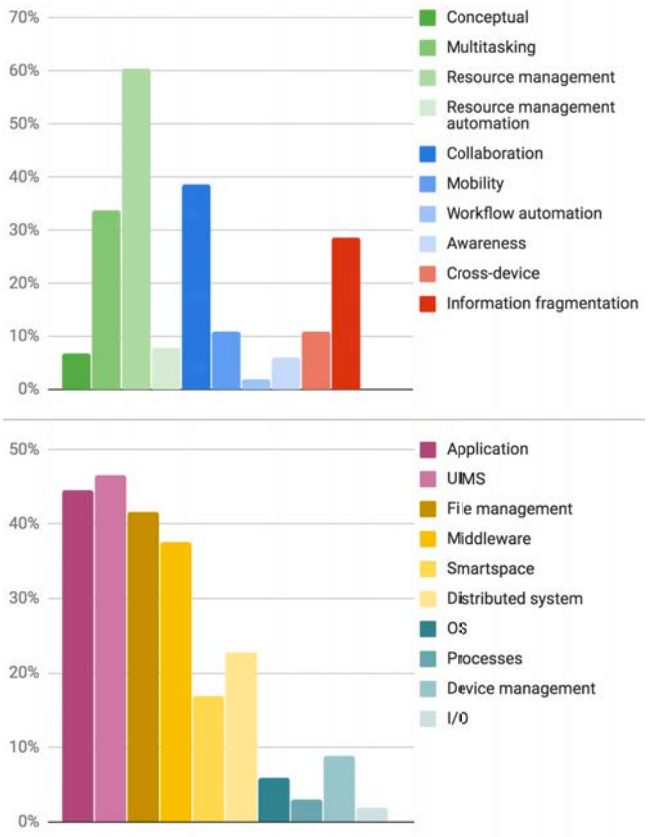

Figure 3: The coding schemes and distribution of 'stated motivation' and 'system type contribution' for all 101 ACC papers.

Figure 4 shows a historical distribution of identified design motivations over time. From this overview, we can identify three ACC waves in the literature: an initial wave in the the 1980s, motivated by the Bannon et al. paper; a second wave in the 2000s; and a recent third wave beginning in 2012 and continuing today. Note that the decline shown beginning in 2015 is a methodological issue; since this review was completed in 2017 and is built from referenced papers, the collection of papers is by nature backward-looking and historical.

In terms of motivation for incorporating support for computational activities, we can identify three broad areas:

Green – motivated by the belief that activities are a better representation of how humans think and/or to provide support for task switching, improved resource management, and automating the overhead of task management.

Blue – motivated to provide support for collaboration, mobility, process optimization, and awareness (of the workspace, task, people, and resources).

Red – motivated to address information fragmentation, including information fragmented across devices.

Based on the data, we can derive that a significant part of ACC research has addressed multitasking (34%), resource management (60%), and collaboration (39%), especially during the 1st and 2nd wave. On the other hand, little research at this stage focused on resource management automation (8%), workflow automation (2%), or awareness (6%). During the 2nd wave, support for collaboration, mobility and awareness (blue) was given increased focus, and in the 2nd and 3rd waves, research was increasingly motivated by the challenges of information and device fragmentation (red).

In terms of system types, we can also identify three broad areas:

Magenta – end-user oriented applications and user interface technology.

Yellow – middleware, file management, and distributed system support

Cyan – low-level operating system support, processes, and I/O.

From the figures, we can see that the majority of papers have focused on end-user applications (45%), user interface management systems (UIMS) (47%) (magenta), and middleware technologies (yellow)—especially file management (42%) and middleware frameworks (38%). Less focus has been directed toward more low-level issues (cyan) like operating systems (6%), processes (3%), and I/O (2%).

From the review, we can identify a set of common topics and technologies, which we unpack as examples of core ACC research contributions.

2.1 Multitasking

Many (34%) ACC systems were motivated by providing support for multitasking, which also represents some of the earliest research in this space. These systems enabled multitasking by supporting suspension of the current activity and resumption of another. This focus recalls the original study by Bannon et al., who argued that a "workspace system should support digression while providing [...] easy return to previous activities" [1]. This implies that people can pause their work on one activity and simply save the entire state of the activity including the configuration of applications, files, windows, and other resources. Afterwards, they can easily switch to another activity, thus, loading the configuration of files, documents, applications, and collaborative tools associated with that activity.

One of the first ACC technologies was the 'Rooms' system presented by Xerox PARC in 1987 [7], which was directly motivated by the Bannon et al. study. Even though this study was based on observations of users interacting with a command-line interface (Unix), similar problems of limited support for multitasking were also observed in the graphical user interfaces developed at Xerox PARC. In Rooms, separate windows associated with the same task could be collated into distinct 'rooms,' and users could switch between these rooms in order to switch tasks. In many ways, Rooms was the predecessor of the 'virtual desktop' systems we know today. Kimura is a more recent (2001) example of an ACC system focusing on supporting multitasking by augmenting the entire office environment [15]. In contrast to Rooms, which limits interaction to the desktop monitor, Kimura leveraged interactive peripheral displays on the walls of the office, allowing users to switch between activities while maintaining a peripheral awareness of other activities in the background.

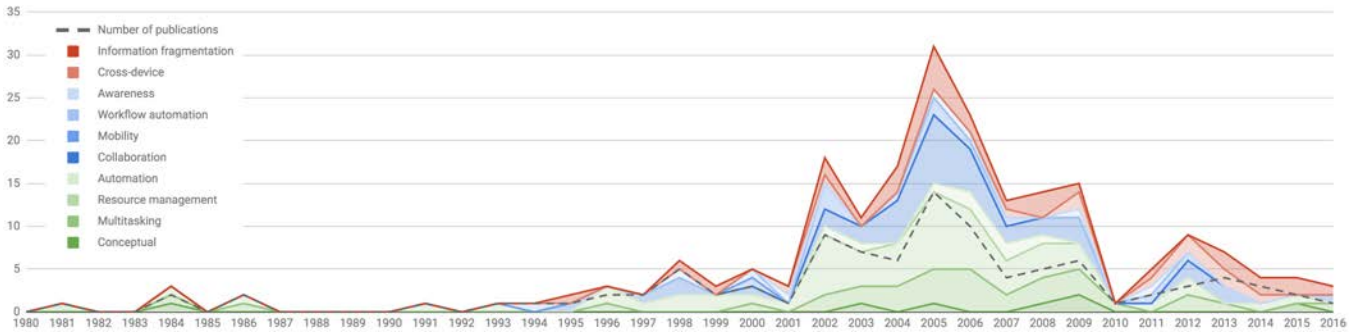

Figure 4: Distribution of motivation for ACC papers in a historical outline.

Figure 5: ActivityBar for Windows XP.

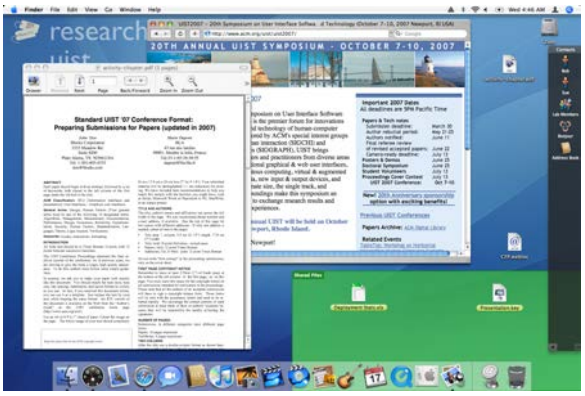

Figure 6: Giornata for MacOS.

2.2 Window Management in Desktop Interfaces

As the user interface in all contemporary OSs (macOS, Windows, Linux) materialized around the desktop metaphor using overlapping windows, icons, menus, and a mouse pointer (also known as the *WIMP* paradigm), it was evident that this model provided limited explicit support for human activity, including multitasking. Therefore, many (47%) ACC systems have provided models that integrate support for activities into the user interface.

For example, the ActivityBar [2] (Figure 5) suggests replacing the Windows XP Taskbar with an 'ActivityBar' that gives direct access to switching among activities. Each activity groups multiple application windows with associated resources, such as documents, spreadsheets, webpages, etc. This approach was later extended to also support sharing and collaborative awareness in the 'co-Activity Manager' system [8] and the entire temporal activity lifecycle in Laevo [11]. Similarly, Microsoft Research (MSR) has proposed a number of ACC extensions to Windows, including the TaskGallery [20] and ScalableFabric [19] window management systems, as well as 'Colletta', which is an extension of the Windows UI that supports lightweight management of the user's activities through tagging [18]. On the MacOS, Giornata [22] (Figure 6) provides support for multitasking through virtual desktop management, tagging of activities, lightweight file management using the desktop surface, and collaborative awareness of the co-workers relevant to each activity.

2.3 Automation of File and Resource Management

Managing multiple files and resources across multiple activities and multiple applications has proven to be a significant challenge in all OSs. For example, keeping track of files related to a specific customer case across folders, email, applications, and cloud-based services is inherently cumbersome. Activity-centric resource and file management technologies have been proposed to address these problems and—as is evident from the magnitude of the corresponding columns in Figure 3—have been central themes in ACC research.

However, even the act of managing computational activity representations incurs some overhead. One approach that has been proposed for further minimizing this cost (but that has been relatively lightly explored, according to our review; see also Figure 3) is in augmenting activity representations with *automation* to automatically handle some of this organizational work on the user's behalf.

For example, by logging interactions with applications used in knowledge work (e.g., email, word processing, spreadsheets, and internet browsers), both the UMEA [12] and TaskTracer [5] system automatically organize resources (e.g., documents, folders, URLs, and contacts) into computational activities. This classification is used in, for example, file management interfaces where an open file dialog box opens by default in a folder associated with the current activity, and quick access is provided to files most likely needed as

part of ongoing work. Similarly, Mylar [13] uses a degree-of-interest model to capture activity contexts in an integrated development environment (IDE) by monitoring the interactions of a programmer with source code. These activity contexts are managed in a 'task list' view which can be used to filter the IDE to only show those elements relevant to the selected task (e.g., implementing a feature or working on a bug fix).

2.4 Collaboration and Awareness

Collaboration is core to human activity and a number of ACC systems (39%) have targeted support for collaborative activity. Activity sharing aims to enable people to work on the same digital activity representations and its resources, without the need for using any 'external' or 3rd party collaboration tool, application, or system. Instead, support for collaboration support is simply built into ACC system support. Additionally, because collaboration and cooperation practices differ across individuals and teams and can change over time, ACC systems have supported different collaboration styles, ranging from full real-time synchronous cooperation within an activity to simpler ways of packaging and sending an activity to other users to support asynchronous collaboration. ACC collaboration mechanisms have also experimented with providing a flexible way for people to define access rights, roles, and the shared context for each activity they are using. These collaborative features have also been used to define and enforce complex organizational work, facilitating the kinds of coordination offered by other workflow-based collaborative systems.

For example, the Activity Explorer [6] and the Unified Activity Management [16] systems developed by IBM Research support the notion of 'activity-centric collaboration,' which aims to support collaboration via activity models, defined as a logical unit of work that incorporates all the tools, people, and resources needed to get a job done. In contrast to prior personal information management systems, the IBM approach had an explicit focus on supporting collaboration by suggesting a unified activity model for *business processes* across people and organizational boundaries. Activity-centric support for collaboration was implemented as part of the IBM Lotus Workplace groupware system. In a hospital domain, the 'Activity-Based Computing' (ABC) system provided support for the extensive collaboration related to medical treatment of hospitalized patients [4]. The ABC system demonstrated the role of activities in fostering both co-located and remote collaboration, and supported scenarios ranging from a co-located team meeting between doctors and nurses to remote video conferencing between, for example, a radiologist in the radiology department and a physician during a ward round.

2.5 Interactive Surfaces and Cross-Device Interaction

Recently, we have witnessed an explosion in the variety and popularity of mobile and ubiquitous computing devices such as smartphones, tablets, whiteboards, tabletops, and game consoles. Traditional cross-device interaction has been accomplished through sending files or documents from one device to another, using available on-device tools to show or use the document. However, this 'basic' cross-device operation does not support moving a complex work context from one device to another seamlessly. In the 3rd wave of ACC research, researchers have thus proposed that ACC can help manage this complexity by using the notion of a device-agnostic 'activity' to bundle together resources accessible across multiple devices and facilitate configuration of these device ecosystems to suit the needs of specific real-world activities.

The ReticularSpaces system [3] (Figure 8) suggests a uniform user interface across multiple interactive surfaces (tablet, wall, tabletop) that allows users to access and collaborate on shared resources, organized into activities. For example, during a software development stand-up meeting, all requirement documentation, software architecture descriptions, and source code for a particular feature under review are available across all devices. Similarly, the ActivitySpace [10] system allows users to synchronize files across tablets, smartphones, and desktop devices by using the notion of an 'activity' as the means for switching among different collections of content. Finally, the electronic laboratory bench (eLabBench) [21] (Figure 7) provides an example of how resources for a biology experiment can be bundled together and made accessible on an interactive lab bench during experimental work inside the lab.

Figure 7: The electronic laboratory workbench (eLabBench) (from [21]).

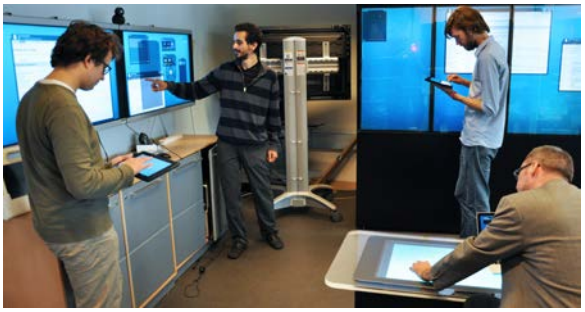

Figure 8: ReticularSpaces: Collocated activity sharing across multiple devices in a smart space environment (from [3]).

3 OUTLOOK AND FUTURE CHALLENGES

Research on Activity-Centric Computing (ACC) has been ongoing since the early 1980s and has achieved much in terms of demonstrating how support for multitasking, mobility, collaboration, and cross-device interaction can be incorporated into computing platforms as well as end-user applications across different domains. Based on a thorough review of 101 papers, we found that ACC has proposed conceptual and technological models to better support window management, file management, workflow management, distributed systems, interactive smart space technology, and cross-device / ubiquitous computing. As such, ACC as a research theme, cuts across several computer science disciplines and offers a potentially valuable series of approaches for addressing the contemporary and significant problems of information fragmentation and information overload.

However, our review also revealed a set of limitations to ACC. First, most research has focused on end-user applications (45%) and user interface management (47%), and less on more basic technologies like how to incorporate ACC into operating systems, file management, distributed computing, and networking technologies. At the same time, the development of ACC applications presented in the research literature has been cumbersome exactly due to this lack of underlying technological support. Thus, more basic research on the lower-level technological components supporting ACC is needed—especially investigations of how support for ACC can be incorporated into or exposed by mainstream operating systems. As an example, one of the most pervasive examples of this kind of missing support from our literature review relates to the need for ACC systems to support suspension of the current activity and resumption of another. Because activity models are stateful, each activity must maintain state information in a persistent way, allowing the state of that activity to be saved (during suspension) and restored (during resumption) at a later time. Enabling a full-stack stateful activity management system has proven to be one of the major challenges in ACC since this requires access to detailed runtime state information spanning the entire computer stack; from end-user applications to the window manager's layout and on down to the underlying file, networking, and process-level state—information that is not readily available in contemporary operating systems (like Windows and macOS) nor from most applications.

Second, from a conceptual point-of-view, a notable barrier to the adoption of ACC technologies is the fact that ACC systems require either end users or ACC systems to manage the computational representations of activities—work that is "invisibly" delegated to the end users in current, application-centered computing environments. The manual management of 'activities'—i.e., the manual creation of a computational activity and organization of its associated resources (such as files and users)—introduces an extra overhead to information management, similar to managing files in a hierarchical folder structure. However, these activity representations may more closely replicate the multifaceted clusters of digital resources, services, and users that map to an individual's discrete real-world tasks, making this organizational work easier, more meaningful, or more memorable [14] than are our current, fragmentation-prone interfaces. An alternative approach is to liberate users from this kind of manual custodial work by relying on content extraction and pattern recognition to automatically organize computing resources into indexed activities. This solution also comes with a cost, however; users must give up some degree of control in the definition of their digital activities, which might lead to mismatches between computational and cognitive representations of activities. These systems might also be semi-automatic, providing specific options or possibilities, without being fully prescriptive. For example, the physical location of a user and their device(s) could be leveraged to filter possible activities or to only show activities that were previously used at that physical location. Striking the right balance among these approaches in future ACC systems will be essential to encourage adoption.

Third, despite the fact that most major computer science companies (e.g., IBM, Microsoft, Apple, and Google) have contributed to research on ACC or experimented with ACC research systems, we have still seen a relatively limited impact of this research on the software architecture of shipping consumer platforms; that is, resource organization at the level of the operating system or task management at the level of the window manager. Even with nearly thirty years of research and development into the benefits of ACC approaches, the application- and document-centered interaction paradigm continues to reign. We have found a few notable examples of success stories: IBM has incorporated computational activity representations into its Lotus Connections suite of enterprise collaboration tools; KDE's *Plasma* desktop environment uses multifaceted activity representations to enhance a typical virtual desktop-driven computing workspace; and Mylar [13] has been introduced as Mylyn[2] in the popular Eclipse IDE as a task-focused interface for programmers. However, for each of these success stories, there are similar examples of systems that did not make it into the mainstream—for example, Apple's application-agnostic OpenDoc platform and Microsoft's proposed (and cancelled) WinFS relational filesystem. This underslines the challenges involved in opening up and redesigning the underlying computational architecture to more completely support ACC systems. And it emphasizes the importance of clearly articulating the benefits that end-users stand to gain by investing time and effort to learn and adopt activity-centered interaction paradigms.

Looking forward, this review has helped us to enumerate exciting open research areas within ACC. Given the explosion in the number of devices and their heterogeneity and interconnectivity, ACC is a strong candidate for a computing paradigm that can help address these complex challenges in service of a more coherent user experience. Limited research has been conducted on cross-device management (9%) and smartspace technology (17%) in the ACC domain, and here there is still much work to be done.

Currently, a major shift towards cloud-based computing is taking place and all major software companies are investing in infrastructures for cloud-based computing. As we have argued and demonstrated earlier [9], cloud-based technologies provides an excellent platform for ACC; it provides the ability to share, distribute, and synchronize heterogeneous resources in realtime across multiple users, devices, and locations. However, as mentioned in the introduction, the current state-of-art of cloud-based computing is to provide services similar to local resources like cpu power, files, and

[2] https://www.eclipse.org/mylyn/

applications. If these were aggregated into cloud-based 'activities', a solid foundation for enabling ACC would be available. Recently, Microsoft have announced its 'Microsoft 365' environment, which have support for resource aggregation, suspend / resume, and cross-device coordination via constructs called 'Sets' and 'Graphs', all of which seems promising building blocks for supporting ACC. In general, we would argue that higher-level support, such as ACC, would be central to the success of a scalable user experience in future development of cloud-based computing.

Furthermore, applying ACC principles, concepts, and technologies to the development of end-user applications in industry is potentially beneficial for many different domains. The research literature reviewed here points out a few areas that have been well-explored to date — information work, medical work, and software development — but many other domains would likely benefit from having direct computational support for domain-specific activities.

In summary, going forward, ACC still presents a variety of important and challenging research topics for researchers and practitioners in many different computing fields—from basic infrastructure to end-user interfaces and applications—to address.

REFERENCES

[1] Liam Bannon, Allen Cypher, Steven Greenspan, and Melissa L Monty. 1983. Evaluation and analysis of users' activity organization. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. ACM, 54–57.

[2] Jakob Bardram, Jonathan Bunde-Pedersen, and Mads Soegaard. 2006. Support for Activity-based Computing in a Personal Computing Operating System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 211–220. https://doi.org/10.1145/1124772.1124805

[3] Jakob Bardram, Sofiane Gueddana, Steven Houben, and Søren Nielsen. 2012. ReticularSpaces: Activity-based Computing Support for Physically Distributed and Collaborative Smart Spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2845–2854. https://doi.org/10.1145/2207676.2208689

[4] Jakob E Bardram. 2009. Activity-based Computing for Medical Work in Hospitals. *ACM Trans. Comput.-Hum. Interact.* 16, 2 (June 2009), 10:1–10:36. https://doi.org/10.1145/1534903.1534907

[5] Anton N Dragunov, Thomas G Dietterich, Kevin Johnsrude, Matthew McLaughlin, Lida Li, and Jonathan L Herlocker. 2005. TaskTracer: A Desktop Environment to Support Multi-tasking Knowledge Workers. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*. ACM, New York, NY, USA, 75–82. https://doi.org/10.1145/1040830.1040855

[6] W. Geyer, M. J. Muller, M. T. Moore, E. Wilcox, L.-T. Cheng, B. Brownholtz, C. Hill, and D. R. Millen. 2006. Activity Explorer: Activity-centric collaboration from research to product. *IBM Systems Journal* 45, 4 (2006), 713–738. https://doi.org/10.1147/sj.454.0713

[7] D Austin Henderson Jr and Stuart Card. 1986. Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-based Graphical User Interface. *ACM Trans. Graph.* 5, 3 (July 1986), 211–243. https://doi.org/10.1145/24054.24056

[8] Steven Houben, Jakob E Bardram, Jo Vermeulen, Kris Luyten, and Karin Coninx. 2013. Activity-centric Support for Ad Hoc Knowledge Work: A Case Study of Co-activity Manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2263–2272. https://doi.org/10.1145/2470654.2481312

[9] Steven Houben, Søren Nielsen, Morten Esbensen, and Jakob E Bardram. 2013. NooSphere: An Activity-centric Infrastructure for Distributed Interaction. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. ACM, New York, NY, USA, 13:1–13:10. https://doi.org/10.1145/2541831.2541856

[10] Steven Houben, Paolo Tell, and Jakob E Bardram. 2014. ActivitySpace: Managing Device Ecologies in an Activity-Centric Configuration Space. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces*. ACM, New York, NY, USA, 119–128. https://doi.org/10.1145/2669485.2669493

[11] Steven Jeuris, Steven Houben, and Jakob Bardram. 2014. Laevo: A Temporal Desktop Interface for Integrated Knowledge Work. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 679–688. https://doi.org/10.1145/2642918.2647391

[12] Victor Kaptelinin. 2003. UMEA: Translating Interaction Histories into Project Contexts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 353–360. https://doi.org/10.1145/642611.642673

[13] Mik Kersten and Gail C. Murphy. 2006. Using Task Context to Improve Programmer Productivity. In *Proceedings of the 14th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. ACM, New York, NY, USA, 1–11. https://doi.org/10.1145/1181775.1181777

[14] Alison Kidd. 1994. The Marks Are on the Knowledge Worker. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*. ACM, New York, NY, USA, 186–191. https://doi.org/10.1145/191666.191740

[15] Blair MacIntyre, Elizabeth D Mynatt, Stephen Voida, Klaus M Hansen, Joe Tullio, and Gregory M Corso. 2001. Support for Multitasking and Background Awareness Using Interactive Peripheral Displays. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 41–50. https://doi.org/10.1145/502348.502355

[16] P Moody, D Gruen, M. J. Muller, J Tang, and T P Moran. 2006. Business activity patterns: A new model for collaborative business applications. *IBM Systems Journal* 45, 4 (2006), 683–694. https://doi.org/10.1147/sj.454.0683

[17] Thomas P. Moran, Alex Cozzi, and Stephen P Farrell. 2005. Unified Activity Management: Supporting People in e-Business. *Commun. ACM* 48, 12 (Dec. 2005), 67–70. https://doi.org/10.1145/1101779.1101811

[18] Gerard Oleksik, Max L Wilson, Craig Tashman, Eduarda Mendes Rodrigues, Gabriella Kazai, Gavin Smyth, Natasa Milic-Frayling, and Rachel Jones. 2009. Lightweight Tagging Expands Information and Activity Management Practices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 279–288. https://doi.org/10.1145/1518701.1518746

[19] George Robertson, Eric Horvitz, Mary Czerwinski, Patrick Baudisch, Dugald Ralph Hutchings, Brian Meyers, Daniel Robbins, and Greg Smith. 2004. Scalable Fabric: Flexible Task Management. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. ACM, New York, NY, USA, 85–89. https://doi.org/10.1145/989863.989874

[20] George Robertson, Maarten Van Dantzich, Daniel Robbins, Mary Czerwinski, Ken Hinckley, Kirsten Risden, David Thiel, and Vadim Gorokhovsky. 2000. The Task Gallery: A 3D Window Manager. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, 494–501. https://doi.org/10.1145/332040.332482

[21] Aurélien Tabard, Juan David Hincapié-Ramos, Morten Esbensen, and Jakob E Bardram. 2011. The eLabBench: An Interactive Tabletop System for the Biology Laboratory. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. ACM, New York, NY, USA, 202–211. https://doi.org/10.1145/2076354.2076391

[22] Stephen Voida, Elizabeth D Mynatt, and W Keith Edwards. 2008. Re-framing the Desktop Interface Around the Activities of Knowledge Work. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 211–220. https://doi.org/10.1145/1449715.1449751