Center for Pervasive Computing
Report Series

Publication
CfPC-2003-PB-56

An Initial Comparison between Aura and Activity-Based Computing

Author(s): Henrik Bærbak Christensen and Jakob E. Bardram
Center for Pervasive Computing
Department of Computer Science
University of Aarhus
Denmark
hbc@daimi.au.dk / bardram@daimi.au.dk

Version: 1.0

Status: Open

Abstract

In this report we outline an initial comparison between the Aura project's architectural framework for task level computing, researched and developed at Carnegie Mellon University, and the Activity-Based Computing Infrastructure, researched and developed at Center for Pervasive Computing, University of Aarhus.

Contents

1 Introduction

The Aura project at Carnegie Mellon University shares many of the same goals as in the Activity-Based Computing project at University of Aarhus. In this report, we will highlight some of these goals and describe how similar concepts have emerged. More important, however, we will describe how the different domains the two projects are created within have lead to different design decisions. It is our hope that we can share lessons learned in the future.

In order to keep the report short, we assume that the reader has knowledge of two imporant papers about Aura, namely

- “Project Aura: Towards Distraction-Free Pervasive Computing” David Garlan, Dan Siewiorek, Asim Smailagic, and Peter Steenkiste. IEEE Pervasive Computing, special issue on ”Integrated Pervasive Computing Environments”, Volume 1, Number 2, April-June 2002, pages 22-31.
- “Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments” Sousa, J.P., Garlan, D. Proceedings of the 3rd Working IEEE/IFIP Conference on Software Architecture 2002, Montreal, August 25-31, 2002. To appear

We also assume a basic knowledge of the paper :

- “Supporting Human Activities — Exploring Activity-Centered Computing”, Henrik Bærbak Christensen and Jakob Bardram, Proceedings of the Fourth International Conference on Ubiquitous Computing UBICOMP 2002, Goteborg, Sweden, September 29 - October 1, 2002. Lecture Notes in Computer Science 2401, Springer Verlag.

2 Task Level Computing

Both the Aura and the Activity-Based Computing (ABC) project share the goal of supporting computing for highly mobile users that interact with computing resources of varying capabilities. As stated in Sousa et al.'s paper:*Ideally, a ubiquitous computing infrastructure would allow users to move their computational tasks easily from one environment to another* [15, §1]. In an environment with mobile users, human attention is the scarce resource [14, slide 5], [1]. Therefore this movement of computing task must ideally be made without distracting the user.

The word *task* is important as the Aura project identifies a problem in present day interaction with computing devices, namely that *Humans interact at too low a level with computers*[14, slide 9].

The Aura project introduces the concept of *Task-driven computing* [14, slide 10], [16], where a user's high level intent is captured in tasks: *. . . , it is crucial that the system maintain some representation of user intent* [12, p. 28].

The ABC project has come to similar conclusions. The ABC project is part of the Pervasive Healthcare Project [13] within the Center for Pervasive Computing research initiative [7] at Department of Computer Science, University of Aarhus, Denmark. The Pervasive Healthcare project's aim is to research how pervasive computing technology can enhance healthcare work for clinicians. It is a project with participation from both software industry and, importantly, clinicians from Aarhus County Hospital. Our research methods include ethnographic observations of clinical work as well as experimental systems development methods such as scenarios-based design, future workshops, role-playing games, design workshops and evaluation experiments.

The main point of healthcare is obviously not to do computing but to treat patients; computing is a means to an end. Thus our aim is to make computing a non-intrusive support in the patient treatment: human attention must not be distracted by computing.

Our study of healthcare work [6, 4, 5, 3, 2, 9] and our workshops with clinicians has lead us to the conclusion that the best way to support the mobile, hectic and often interrupted, work of clinicians is to let the computing infrastructure understand the high level, human, activities that nurses and physicians are engaged in: prescribing medicine, giving medicine to patients, doctor's conference, etc. It is these *activities* that must be able to migrate to new computing devices when the clinicians move around, because they embody the clinicians' working context.

2.1 Task/Activity as first class entities

Healthcare activities draws upon functionality from many different applications, as shown in figure 1 (taken from [10]) where the activity: prescribe medicine for Mr. Hansen; is a single, "atomic", activity from the physician's point of view. From the point of view of the computing system, however, it is simply many unrelated applications that are running at the same time.

We have formulated this as a key proposal: *To support users with their physical work activities, the computing system must understand the concept of an activity and handle it like a first class object.* [10, §3.2]. This is very similar to the corresponding definition in Sousa et al.: *users tasks become first class entities that are represented explicitly* [15, §1].

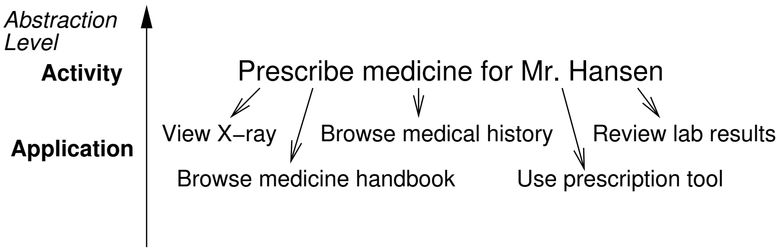

Figure 1: A single activity involves many applications

2.2 Task/Activity Definition

Aura tasks are handled by Prism, that define key characteristics [12, p. 28]:

- Explicit representation of user tasks as coalitions of abstract services.
- Context observation that lets the infrastructure configure tasks in a way that is appropriate for the (execution) environment.
- Environment management infrastructure that assists with resource monitoring and adaptation.

Similarly, the ABC activity is defined as *the digital equivalent of a physical activity; for instance the activity of prescribing medicine for a given patient in healthcare can be mirrored by a "prescribe medicine activity" that embody all relevant computational state for that activity: identity of the patient, of the doctor, time and date, medical record data, lab results, etc., as well as used applications, views and user interface interaction state.* [10, § 3.3].

Thus, task and activity are basically the same abstraction at both the conceptual as well as technical level:

> ***Conceptual:*** *a computational equivalent of human intent and action.*
>
> ***Technical:*** *An explicit, abstract, representation of the execution state of a set of running applications or services.*

2.3 Infrastructure support

Both Aura and ABC defines an infrastructure/architectural framework that supports the definition, storage and migration of tasks/activities between devices. The respective architectures are outlined in[Fig. 1] [15] and [10, Fig 2.] respectively. They share a lot of commonality.

2.4 Context Observers/Location Awareness

Both Aura and ABC acknowledge the importance of context awareness and location tracking. In Garlan et. al.'s paper, the "Fred" scenario outlines how Aura is able to autonomously migrate Fred's slide preparation task to his PDA as he leaves his office [12, p. 23]. A similar scenario involving Fred moving from home to his office is outlined in [15].

In ABC, we have envisioned and designed components into our ABC framework for handling context information as well as abstractions over location monitors (briefly mentioned in [10]). Specifically, we have experimented with RFID devices, BlueTooth, and contact-less JavaCards, and have used these in our workshops to monitor the movement and position of nurses and physicians. As explained below, this has enabled us to experiment with simple forms of automatic inference of user intent.

2.5 Proactivity/Activity Discovery

Both Aura and ABC acknowledges the perspective of proactivity. In Garlan et. al.'s paper, the "Fred" scenario outlines how Aura prepares a projection computer for an anticipated slide show [12, p. 23], based on knowledge of Fred's calendar and the location tracking system.

An inference system has been developed and tested in the ABC project, based on expert system technology [11]. Christensen outlines how some simple clinical activities can be inferred based upon tracking persons and items in a hospital [8]. Specifically, rules can be defined that trigger automatic creation of a "give medicine to patient - activity" when the ABC infrastructure detects that the medicine tray for patient X, patient X himself, and a nurse is in close proximity, and the time is around the typical time when medicine is given to patients. The created activity embodies all information regarding the patient in question, the nurse to log into the electronic patient record (EPR) system, and the medicine in the tray. Based upon this information even the GUI state of the EPR can be inferred: bring up the medicine schema with the proper patient, centered on the proper date and medicine.

However, in one important aspect our proactivity concept differs from what is outlined in the Aura papers. We deal with that later.

2.6 Heterogeneous Devices

Both Aura and ABC acknowledges the requirement that task/activities must be able to be instantiated on heterogeneous devices: *When a user moves from one environment to another, Aura attempts to reconfigure the new environment so that the user can continue working on tasks started elsewhere* [12, p. 27], which is much in line with our ideas of "public computers": *If a person's work activity is associated with a computational activity, the infrastructure is required to be able to provide access to the person's activities swiftly on any public computer in his/her vicinity.* [10, §2.3].

It is stated in the WICSA paper that you have some experiences with migration between radically different environments. We have the intent to experiment with heterogeneous devices, but the focus of our workshops—to study the activity concept in realistic situations—have forced attention to other aspects, and presently we have migrated activities between devices with similar characteristics.

3 Additional Aspects in ABC

In the ABC project we have also considered aspects that we have not found discussed in detail in Aura.

3.1 Collaborative aspects

In a hospital many activities involves more than one clinician. Collaboration is more the rule than the exception. Physicians communicate with radiologists, nurses, and the patient in order to define a treatment. We found it natural and useful to view the activity concept as one that can serve as basis for collaboration. We have experimented with collaborative activities in our developed infrastructure and evaluated them in workshops. The basic idea is that an activity may invite other people to join it and then the activity becomes similar to a 'shared desktop': what I see is what you see. Presently, we are in the process of publishing some initial results from this work, but still many issues remain open.

3.2 User Authentication

Security is important in a public place like a hospital. Strangers walking in from the street should not be granted access to patient record. However, traditional "type user name and password" login simply will not work in the hectic daily life of clinicians—it is much too slow. For instance, a nurse typically gives medicine to patients 40 times a day. If we take our metaphor of public computers available everywhere seriously (in contrast to carrying a personal computer) then even this simple action would involve logging in and out at least 40 times.

We have therefore experimented with proximity-based login using the location monitor components of our infrastructure: *We find that a system of proximity-based authentication is very interesting in this light. I.e. a user is authenticated to the infrastructure by proximity to a public computer* [10, §2.3]. A proximity-based proposal uses the tag worn by clinicians to speed up the login-in process. However, questions remain open about how to achieve the "right" level of security. We are presently investigating different kinds of biometric devices to try to strike the proper balance between speed and security.

4 Open Issues

While Aura and ABC share the same vision towards supporting users at the task level by seamless migration of their computing activities, there are still a lot of open issues. We think, this is a fertile ground for discussions between the two projects. Below we identify some that we find key issues.

4.1 What defines User Intent?

Both Aura papers, [12, 15], states that tasks capture high level intent. The question arises, however, what or who defines the intent? Sousa et al. [15, §2.3] states that given a more sophisticated Context Observer *the less Prism has to rely on explicit indications from a user concerning their intentions.*

There are actually two points made in this statement. First, that users may explicitly state intentions. Second, that context awareness can provide all (?) or part of the user's intention.

Concerning the first point: if Aura users have to indicate their intent explicitly, then this requirement is in conflict with the Aura premise, namely that human attention is the scares resource.

We have discussed this issue with clinicians at our workshops. Even the relatively simple action of *naming* an activity when it is created is often too distracting. An extreme scenario is the physician that is called for immediate help with a patient suffering heart-attack, rushing to the nearest computer to get vital information only to be met by a friendly dialog-box: "Please enter a name for the activity". While this is of course an extreme situation, it does show that the problem is not trivial.

In our prototype, we have resorted to default names for newly created activities and the option to rename it later. This passes at our workshop but probably only because our role-plays lack the scalability problem: a scenario takes half an hour and involves perhaps 3–5 different activities that are easily handled mentally. But we do not think it scales well to, say, 20–30 activities.

Even so—defining a name is only a clue for humans as to what the intention is. If our infrastructure needs more elaborate information regarding user intent then naming is not nearly enough. And were do we then get reliable information?

This leads to the second point, namely that context awareness can define user intention. Our experience is that it is very difficult to correctly infer a satisfactory complete set of attributes to define user intent uniquely from environment monitoring and context awareness—even in what first appears as obvious cases. And, that it may not even be interesting. Of course, we speak from our knowledge of the particular domain healthcare; things may be different in other domains.

As an example, one of our workshops had the theme: prescription of medicine. We initially envisioned that some environmental triggers could define that a physician is most likely to want to initiate a "prescribe medicine for patient X" activity. For instance, when the physician is on the morning ward round and near a patient's bed. However, in practice physicians prescribe medicine at odd hours and in all sorts of places: at the morning conference, in the aisle outside the patient's ward and actually seldom when the patient is nearby. Engaging in a medicine prescription activity is triggered by sources too complex to capture except by explicit indication by the user.

4.2 Discovery versus Execution

The Fred scenario mentions Aura taking action based on inferred activity. Fred is rushing to the conference room, thus Aura downloads files and software to the computer in the conference room and turns the projector on. Aura infers this from knowledge of Fred's calendar and his movement.

But, what if Aura infers wrongly? What if Fred rushed out of the office because his wife phoned that their daughter had just been hospitalized after falling from a swing? Fred would probably want to keep the mobile phone open for keeping in touch and is *not* interested in a dialog box telling him that he has to defer his calls because his Aura enabled phone is busy downloading power-point slides.

At one of the first workshops, we presented our vision of helping nurses to *automatically* record medicine given to patients in the electronic patient record simply based on location tracking of medicine trays, nurses, and patients. We found it quite smart that this was done simply when placing the medicine tray on the patient's bed table.

However, the clinicians were very upset with this idea. First of all, they pointed out that medical decisions were now made by the computing system, not by trained, clinical, staff. Second, the actual action to take is not uniquely defined by the external triggers we kept track of, such as time schedules, and location of people and things. Indeed, complex human decision making based upon knowledge of the patient and the situation. Third, even the same context triggers only define a space for user intent, not an exact point. Even given the pretty comprehensive context triggers: Nurse Berg is near patient Anderson; the 12 o'clock medicine for Anderson is not yet given; it is around 12 o'clock; Anderson is in is her bed; the medicine tray of Anderson is on the bed table near Anderson; we could not infer whether Berg wants to document that all medicine has been given to Anderson or just some of it (unless we track each individual tablet!). We can infer that nurse Berg probably wants to do something about medicine recording for Anderson but not precisely.

This problem is also raised as an issue in Sousa et al.'s paper §4: *. . . systems are notoriously poor at capturing user intent automatically.*

Our proposal was thus to 1) infer a set of likely user intents 2) define activities for each in the set and 3) present these suggested activities *non-intrusively* for the user. The inference issues are discussed in [8]. We have so far not published anything on the user interface issue, but we believe strongly in the idea of a "humble" computing system that quietly propose helpful activities instead of imposing the systems view upon the user. The infamous Microsoft paper-clips was an example of the latter. This allows the users to ignore the computer proposals without distraction; but also to access them when they come in handy.

4.3 Intent or Light-weight Activities?

Humans have intentions with their tasks. Does computational activities have intent that must be modeled? Initially, we thought so, but we are not so sure anymore that computational activities need to model human activities accurately to be usefull. In Sousa et al.'s paper [15, §4] we see this issue raised: *. . . Aura should prove useful even with no deeper knowledge of the task . . .*

We had a hefty debate after a workshop about whether our activities should accurately mirror human tasks. On one hand, if the computing infrastructure has accurate knowledge of defined tasks, then transferring tasks from one clinician to another could be supported directly by the infrastructure. This happens all the time in a hospital, during the shifts when for instance the evening nurse takes over from the day nurse. Thus instead of talking to the next nurse on guard (potentially forgetting some important tasks), the nurse leaving could simply transfer the list of pending tasks to the next nurse.

We found a lot of problems with this seemingly appealing idea. First, our clinicians did not like this idea. Human communication is important, and speech is much faster than keyboard exercises on the computer. Second, our activity infrastructure drifted towards a workflow system. Workflow systems dictate how work is done; we in contrast wants to support current work practices where interruptions are not an evil but key tools for communication and learning.

At present, our perception is that a computational activity is a way to checkpoint and reestablish composite state in a set of applications, and what the users' intent is with these "computational nudgets" is left for the user to deal with. We have termed this that our activities are "light-weight". The infrastructure just supports fast activity switching, seamless migration to new devices, and ability to transfer computational activities to other persons as *part of* handing over responsibility and *not* as the mechanism to do so.

This does not rule out that inferred intent can create activities, though. This is still interesting as outlined in the previous section.

4.4 Who defines state?

Taking an architectural and software engineering perspective there is also an important question, namely what is required by programmers wanting to write services for task level / activity-based computing infrastructures?

In Sousa et al.'s paper examples of XML formats are shown along with a description of how services creates and parses such descriptions during checkpoint and establish.

This has the evident advantage that it is generic and different concrete services may be instantiated to support migration of activities across devices with varying resources. We have had similar ideas but at present use untyped Java objects that of course are less generic.

The downside, however, is that state check-pointing and parsing is entirely left to the programmer. This may be OK for a word processor but we found it error-prone and cumbersome when dealing with the medicine schema which is a highly complex Swing GUI. It became even more painfully when we dealing with a collaborative activity which essentially checkpoints and distributes state descriptions to all other participating users every time one user makes a change.

We therefore think that it is important that we find some techniques that relieve the burden on the programmer when he has to program the generic state handling part of a new service to be deployed on the infrastructure. We have some ideas but they are in their initial stage yet.

5 Summary

We conclude that we share many of same visions, have come to many of the same concepts and ideas regarding supporting activities as first class entities and have also come to similar proposals for computational support[1].

Aura has a much wider scope that our ABC project, and deals with problems at many more levels than our work does. We have limited resources and have concentrated on the task/activity level. But we of course face the same challenges as do Aura when it comes to resource adaptation in the concrete device.

Our project has a strong point because of our experience in a very complex domain where the technology is very suitable; a domain that shed light on many of the problems that we have but also is a major source of inspiration and helps defining the right questions to ask. We can also see that we have first hand experience with some of the issues that are raised in some of the Aura papers.

[1] It should perhaps be noted that our idea of “activities” was born as a result of our workshops in late 2001. It was only after this phase that we became aware of the Aura project.

References

[1] Aura Project. http://www.cs.cmu.edu/˜aura.

[2] J. E. Bardram. Plans as Situated Action: An Activity Theory Approach to Workflow Systems. In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work*. Kluwer Academic Publishers, 1997.

[3] J. E. Bardram. Designing for the Dynamics of Cooperative Work Activities. In *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, Seattle, Washington, USA, 1998. ACM Press.

[4] J. E. Bardram. Scenario-based Design of Cooperative Systems: Re-designing an Hospital Information System in Denmark. In *Group Decision and Negotiation*, volume 9, pages 237–250. Kluwer Academic Publishers, 2000.

[5] J. E. Bardram. Temporal Coordination. *Computer Supported Cooperative Work*, 9:157–187, 2000.

[6] C. Bossen. The parameters of Common Information Spaces: the Heterogeneity of cooperative Work at a Hospital Ward. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work CSCW2002*, New Orleans, Louisiana, USA, Nov. 2002. ACM.

[7] Center for Pervasive Computing. www.pervasive.dk.

[8] H. B. Christensen. Using Logic Programming to Detect Activities in Pervasive Healthcare. In *18th International Conference on Logic Programming ICLP 2002*, pages 421–436, Copenhagen, Denmark, Aug. 2002. LNCS 2401, Springer Verlag.

[9] H. B. Christensen, J. Bardram, and S. Dittmer. Theme One: Administration and Documentation of Medicine. Report and Evaluation. Technical Report CfPC-2001-PB-1, Center for Pervasive Computing, Aarhus, Denmark, 2001. www.pervasive.dk/publications.

[10] H. B. Christensen and J. E. Bardram. Supporting Human Activities — Exploring Activity-Centered Computing. In *Proceedings of Fourth International Conference on Ubiquitous Computing, UbiComp 2002*, Göteborg, Sweden, 2002.

[11] E. Friedman-Hill. Jess, the Rule Engine for the Java Platform. http://herzberg.ca.sandia.gov/jess/.

[12] D. Garlan, D. Siewiorek, A. Smailagic, and P. Steenkiste. Project Aura: Towards Distraction-Free Pervasive Computing. *IEEE Pervasive Computing*, 1(2):22–31, 2002.

[13] Pervasive Healthcare. www.healthcare.pervasive.dk.

[14] M. Satyanarayanan. Research Challenges in Project Aura. Slides from Key note at HPDC, Aug. 2000.

[15] J. P. Sousa and D. Garlan. Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. In *WICSA*, 2002.

[16] Z. Wang and D. Garlan. Task-Driven Computing. Technical Report CMU-CS-00-154, School of Computer Science, Carnegie Mellon University, May 2000.