



**Workshop on  
Computer Support for  
Human Activities and Tasks  
Vienna 2004**

**Open Issues in Activity-based Computing**

*Jakob E. Bardram  
Henrik B. Christensen  
Department of Computer Science  
University of Aarhus*



**Where do we come from ?**

**Project *Pervasive Healthcare***  
**Based upon studies of**  
***Clinical Work in Hospitals***

**Initial hypothesis:**

***Pervasive computing technologies are ideal  
for supporting clinical work.***

**Ideas**

- context awareness
- composite devices

Shared Material




Mobility







 **AARHUS UNIVERSITET**

## Activity Based Computing

**Basic Problem:**

- Human think in terms of tasks/activities
- Computers think in terms of applications and files
- Prevailing operating systems based on desktop metaphor

**ABC: Activity Based Computing**

- lessen the gap between humans and computers

CSHTA 2004 Bardram and Christensen 8



# Present prototype

**Telepointer**

**Activity Services**

**Activity Bar**

T	Ord. / Sep.	Preparat	giving	9-10-2003				19-10-2003			
				12	17	22	9	12	17	22	
F	13-10-2003	Berodual 4 mg					100	200			
F	13-10-2003	Furosid 40 mg, tbl									
F	13-10-2003	Kalocid 750 mg.									
F	13-10-2003	Paracet 1 g, tbl					120	100	20	200	

CSHTA 2004

Bardram and Christensen

9



## Open Issues

Observed in healthcare domain



# Human Intent?

## How do we define or describe human intent?

### Two major sources

- automatically inference via context observation
- human interaction

### Human interaction ?

- *Even the relatively simple action of **naming** an activity when it is created is often too distracting in health care*
- *and what do we use to label/denote an activity?*
  - *we need this to find it later...*
- ‘Activities are rarely specified up front’ (Victor)



# Automatic inference

## Context observation defines intent?

- inferring activities unambiguously based on context awareness is very hard (“accuracy property”)
  - except perhaps in very well defined environments
- Example: medicine tray on bed table
  - just login nurse ?
  - show medicine schema ?
  - document medicine given ?
- We don’t know ...

### *Our proposal*

- propose instead of act
- non-intrusive UI





## Automatic inference

### Ubicomp 03

- large sensor arrays, modeling, data collections
- to turn on a lamp

**but the problem is solved: a contact !**

Perhaps instead of inferring 'cooking act' the user just hits the friendly 'cooking button'?

- comprehensible, accurate !
- adaptability ?



## Intent or not?

**The question is, however:**

- *Is accurately capturing intent necessary?*
- is that necessary for it to be useful?
- or is it just the technical part of it: a state snapshot?
- and if so, how do we identify 'snapshot' so we know what it means?
- *Clinicians did not like computational activities as means of transferring responsibility and information during shifts...*



## Activity Lifecycle

### When does one activity suspend and another start?

### What delimits one activity from the next?

- Nurse pours medicine for Mr. Hansen (activity 1)
- Nurse gives medicine to Mr. Hansen (activity 2)
- ... but the first turns fluidly into the next !!!
  
- nurse is pouring medicine, colleague asks a question
  - should she suspend, and create a new one?
  - or will she just bring up the proper service – but now it is in a 'wrong activity'?



## Activity Browsing and Managing

### How do we manage a large set of activities?

- workshops got away with 2–3 numbered activities
- but – physicians handle
  - 10-15 patients in ward
  - 10-15 patients in out-patient clinic
  - administrative tasks
  - radiologist and other conferences
  - research
  
- how do we organize, initiate, terminate these in a way that does not make into a task all by itself?



*“Design for people to bring meaning, not for technology to reflect meaning”*

Salvador / UbiComp'03



## Open issues

How do we programmatically define state?

```
public interface StatefullApplication {  
    public String getServiceName();  
  
    public void enableStateManagement(boolean b);  
    public void activityActivated(int act_id);  
  
    public void setApplicationState(ApplicationState state);  
    public ApplicationState getApplicationState();  
    public void setComponentState(String id, ComponentState state);  
    public ComponentState getComponentState(String id);  
  
    public void setActiveUser(String id);  
    public void logout();  
    public void bringToFront();  
    public void refresh();  
}
```

Full state

$\Delta$  state

*ApplicationState is basically just a hashtable*



### **However, this support is next to “no support” for the programmer of the services:**

- the state object for a medicine schema is *very* complex
- that is – maintenance is a cumbersome and error-prone process

### **Compare RMI middleware technologies**

- IDL and stub generation, made for developer
- run-time support by the ORB