



# Tracking task context to support resumption

Damien Clauzel

American University of Paris

HCI Engage 2006



# Contents

## What do we want to do?

- Goals

- Our work environment

## What do we need to do it?

- Needs

- Hosting application requirements

## How do we do it?

- Possible approaches

- Data collection methods

## Conclusion

## Ideas & suggestion



## What do we want to do?

We want to follow the user activity:

- what **exactly** is doing the user?
- why is he doing a specific action?
- how is he doing it?

We want to be able to manipulate the system:

- storing and restoring context
- activating functions and widgets



## Our work environment

- heterogeneous servers and workstations
- target learners are 12-14 years old children
- we improve existing e-learning plat-forms
- we create various modules for psychological (attention & motivation) and activity management



## What do we need to do it?

We need to know what the user is doing

⇒ activity tracking: high level (search, writing a letter...)

⇒ action tracking: low level (key press, widgets used...)

We need to be able to remotely control the plat-form

⇒ communicating software ⇒ common interfaces

⇒ *smart* applications



## Hosting application requirements

1. the position of the caret in the last selected text area
2. the coordinates of the pointer
3. the coordinates of a click event, and which application, area and widget were having the focus
4. the coordinate, geometry and status (active, inactive, folded, maximized, z-level) of the windows
5. the status of the menus
6. the time spent on the current focus, the overall task and the activity
7. the time schedule of the user
8. ...



## How do we do it?

Two main possibilities:

- data gathering from operating system  $\Rightarrow$  not enough (?)  
 $\Rightarrow$  technical problems & few data
- modifying the plat-form & applications  $\Rightarrow$  how far can we go?  
 $\Rightarrow$  means having tracking ready softwares



## Screen mapping

The idea is to track cursor and graphical events (windows moving, menus unfolding, etc.) by doing a  $(x; y)$  coordinates mapping.

The coordinates and actions are linked to graphical elements which are associated to functions.

<b>advantages</b>	<b>drawbacks</b>
generic solution	doesn't allow mistakes in activity watching
works everywhere	hard to follow dynamic and customisable interfaces
transparent to user	window manager based solution
	cannot deal with text input



# Spywares

This method relays on keylogger and files and process watching daemons for monitoring the user activities: processes, application running, files opened, windows list, focus tracking, text typed, etc.

The idea is to track at the operating system level the activity of the user.

<b>advantages</b>	<b>drawbacks</b>
can collect various data	security issues
transparent to user	OS based solution



## Smart applications

This solution consists of having customized applications which will send an event when a functionality (menu entry, icon, widget. . . ) is activated.

<b>advantages</b>	<b>drawbacks</b>
lot of various information	softwares must be <i>tracking ready</i>
transparent to user	

# Conclusion

- it is difficult to collect even simple data
- the choice of the solution depends of the computer environment
- any solution always need a complex development

## Ideas & suggestions

- screen mapping can be sufficient for very simple tracking
- spywares approach should be excluded for security reason
- smart applications are the most promising solution