Adapting the Multi-Desktop Paradigm
Towards a Multi-Context Interface

Sven Schwarz, Malte Kiesel, Ludger van Elst
DFKI GmbH, Kaiserslautern

MRC 2008
Delft, The Netherlands
9 June 2008
Overview

► Szenario
  - Knowledge work
  - Multi-tasking

► Approach
  - Apply Multi-Desktop Paradigm
  - Extensions to traditional Multi-Desktop

► Implementation

► Summary
Knowledge worker‘s (digital) work

► Basically, knowledge workers
  - work with documents
  - use office applications

► Particularly, knowledge workers
  - have to deal with multiple tasks “in parallel“
  - are often interrupted during work

► Task switching force workers to
  - stop current task
    ▪ close current documents/applications
  - start/resume another task
    ▪ open new/former documents/applications
Multi-tasking is expensive

- Tasks in parallel
  - tasks *never* finished
  - interleaved task execution
  - => halt + resume tasks

- Task switching overhead
  - halt + store current state:
    close windows, write note about state
  - remember/recall other task’s state:
    find folders, URLs
  - recover other state:
    open docs
  - resume other state

Goal: Reduce task switching overhead!
Overview

► Szenario
  - Knowledge work
  - Multi-tasking

► Approach
  - Apply Multi-Desktop Paradigm
  - Extensions to traditional Multi-Desktop

► Implementation

► Summary
Multi-desktop paradigm is a good start
Multiple desktops are used to organize daily work in 2 alternative ways:

1. **tool oriented:**
   each desktop provides a special set of applications to suite a special class of tasks
   - communication desktop: mail + chat
   - hacking desktop: eclipse, javadoc
   - music desktop: iTunes, …
   - news and web desktop: firefox

2. **task oriented:**
   each desktop contains the material and information needed to accomplish one specific task
   - booking trip to MRC’08 conference
   - preparing slides for MRC’08
   - designing and discussing Ralf’s dashboard
   - supervising diploma thesis Jan Haas
Overview

► Szenario
  - Knowledge work
  - Multi-tasking

► Approach
  - Apply Multi-Desktop Paradigm
  - Extensions to traditional Multi-Desktop

► Implementation

► Summary
Extensions to traditional Multi-Desktop

► Arbitrary, increasing amount of desktops
  - One (new) desktop for each (new) task  =>  many desktops

► Meta-data for desktops
  - Annotate desktops with context information (automatically)

► Sophisticated user interfaces
  - Visualize / cluster / search / filter / switch desktops

► Persistent storage of desktop state
  - Store / restore application windows + open documents
Choosing and managing 1000s of desktops needs alternative interaction metaphors:

**Clustering, indexing, searching** for desktops gets necessary
Display and group differences of relevant desktops
Overview

► Szenario
  - Knowledge work
  - Multi-tasking

► Approach
  - Apply Multi-Desktop Paradigm
  - Extensions to traditional Multi-Desktop

► Implementation

► Summary
The PIMO (Personal Information Model) is a model of the user’s individual concepts of his knowledge world.

- **PIMO (Personal Information Model)**
  - Ontological model of the user’s world view
    - classes: Persons, Projects, Topics, …
    - instances of these classes

- **PIMO concepts are used to annotate desktops**
  - Context elicitation estimates for every PIMO concept its relevancy for the current context.
  - As each desktop has an assigned context, a desktop is described with contextually relevant PIMO concepts.

- **Tools to manage your PIMO**
  - soon: PIMO Editor [http://pimoeditor.opendfki.de/](http://pimoeditor.opendfki.de/)
User observation feeds context elicitation

User Observation Hub

- userobservation.opendfki.de

- open-source platform used by multiple researchers
- one shared user action ontology
- distributes observed user actions to registered listeners
- collects data from multiple observation sources (extensible):
  - Mozilla plugins “Dragontalk” (DFKI): observe email + web browsing
  - File System Observer (DFKI)
  - PAS logger (L3S): observe window management
  - Real-Time Document Image Retrieval with LLAH (Osaka Prefecture University): recognize/observe a paper document with a web camera
Desktops are tagged automatically by user observation and context elicitation (desktop = context)
“MyDesk“ adapts the multi-desktop paradigm and realizes a context switching interface

http://mydesk.opendfki.de
Diploma thesis, Moritz Plößl

- Desktop switching (Windows, C#)
  → Visibility-Trick (Hide/Show)

- Timeline of desktops (like ALT+TAB)

- Manual 2-dim. layout of desktop

- Manual/autom. tagging of desktops

- Searching/filtering desktops

- Persistency of desktop state
  → Store/restore open windows
Problems (1)

- Versioning of documents is not yet supported
  - Only document’s location (URL) stored
  - Always the newest version is used
  - No problem with static documents (new=old version)

- Dynamic documents are problematic
  - The recovered application may show a newer version than last time!
  - This may or may not be what you expect!
Problems (2)

- Explicit context switching + explicit separation of contexts
  - makes things easy for context-sensitive assistance
  - helps humans to remember and resume former tasks
  ⇒ (+) removes some part of the context switching overhead
  ⇒ (−) does not remove all of the context switching overhead

- Very small tasks…
  - … with a typically execution time < 2 minutes …
  - Examples: email check, chat response, enter calendar entry
  - Spending an additional desktop does not make sense
  - The overhead for handling an additional desktop is too expensive
  - For these mini tasks, “implicit context“ calculus seems better
Evaluation of features

► Context Identification
  - Evaluation data: Log NOPs + context of NOPs (switches with MyDesk)
  - Hypothesis: context can be identified by context elicitation

► Context-Switch Detection
  - Evaluation data: Log NOPs + context switches (MyDesk)
  - Hypothesis: context switches can be detected by context elicitation

► Context-Similarity
  - Evaluation data: 2-dimensional desktop map (MyDesk)
  - Hypothesis: euclidian distance between desktop icons ≈ context similarity from context elicitation

► Context-Sensitive Tools
  - Evaluations: Ralf's Dashboard, ALOE, …

► Enhanced Multi-Desktop Paradigm (MyDesk)
  - Evaluation data: Log NOPs (close/open window op.) + context switches
  - Hypothesis: Some pairs of close-open NOPs become obsolete
Overview

► Szenario
  - Knowledge work
  - Multi-tasking

► Approach
  - Apply Multi-Desktop Paradigm
  - Extensions to traditional Multi-Desktop

► Implementation

► Summary
Summary

► Szenario
- Office worker, digital knowledge work
- Multiple tasks in parallel
  ▪ interruptions, task-switches
  ▪ resuming tasks requires remembering former task state

► Approach
- Apply Multi-Desktop Paradigm
  ▪ multiple virtual workspaces (desktops)
  ▪ switching tasks = switching desktops
  ▪ desktops carry information about task state (open docs)
- Extensions to traditional Multi-Desktop Paradigm
  ▪ arbitrary amount of desktops (1 desktop for each task)
  ▪ persistent storage of desktop state (store + restore)
  ▪ meta-data for desktops (automatically)
  ▪ sophisticated user interfaces
Adapting the Multi-Desktop Paradigm
Towards a Multi-Context Interface

Sven Schwarz, Malte Kiesel, Ludger van Elst
DFKI GmbH, Kaiserslautern

MRC 2008
Delft, The Netherlands
9 June 2008