Developing Adaptive Groupware Applications Using a Mobile Component Framework (DACIA)

Radu Litiu and Atul Prakash
University of Michigan, EECS
Outline

- Motivation and Goals
- DACIA Architecture
- Building Adaptive Groupware Applications
- Related Work
- Summary
Problems

- Flexibility and adaptability in CSCW systems (Roseman93, Bentley95, Shen92)

- Variability and heterogeneity
  - User and application demands
  - Hardware and network variability

- Application and user mobility (Belloti96, Chung96 – X client migration)
  - Support for offline users
Adaptation Need - SPARC Collaboratory

An application is a graph of connected components.

Possible changes:

- Execute the computation on the client machine
- Store computed images instead of raw data
- Add/remove modules
Mobility
DACIA* Features

- Dynamic application reconfiguration
- Component mobility
- Persistent connectivity between components
- Support for offline users – application parking

*Dynamic Adjustment of Component InterActions
DACIA Architecture

Engine (mechanism)
- Communicate between hosts
- Manage connections between components
- Relocate components
- Reconfigure the application

Monitor (policy)
- Monitor performance
- Make reconfiguration decisions
- Implement application-specific reconfiguration policies
PROCs*

- Communication through ports
- Low communication costs
- Component mobility – state transfer
- Persistent connectivity

*Processing and Routing Component
Component Mobility (I)
Component Mobility (II)
Application Parking

- Intermittent connectivity
- Applications participate to collaboration on the user’s behalf while the user is disconnected
Dynamic Application Reconfiguration

- Change connections between components
- Change components’ location
- Load new components

An adaptive application: multi-party communication
Reconfiguration Mechanisms

- Specialized monitors

- Programming API:
  - `connect [hostname] [portnumber]`
  - `connectProcs [sourceProcID] [sourcePortNo] [destProcID] [destPortNo]`
  - `disconnectProcs [sourceProcID] [sourcePortNo]`
  - `move [procID] [hostname]`
  - `start [procID]`
  - `startMonitor`

- Command-line interface

- Graphical interface
Related Work

- Extensible architectures: Fitzpatrick96, Lee96
- Context-aware applications: Harter99
- Cooperative buildings: Streitz98
- Code mobility & mobile agents: Telescript, Obliq, Sumatra, Tacoma, Aglets
- Distributed component architectures: CORBA, Scout, FarGo, Rover
Summary

- DACIA - a framework for building adaptive groupware applications
- Dynamic reconfiguration
  - Improve the performance
  - Customized and extensible configurations
- Application and user mobility
- Persistent connectivity
- Application parking
Current and Future Work

- PDA porting
- Policies and algorithms for application reconfiguration
- Formalism for specifying components and composition rules
- Deployment and experimentation
- Security infrastructure