

Challenges in Using a Mobile Component Framework to Develop Adaptive Groupware Applications

Radu Litiu and Atul Prakash
University of Michigan, EECS

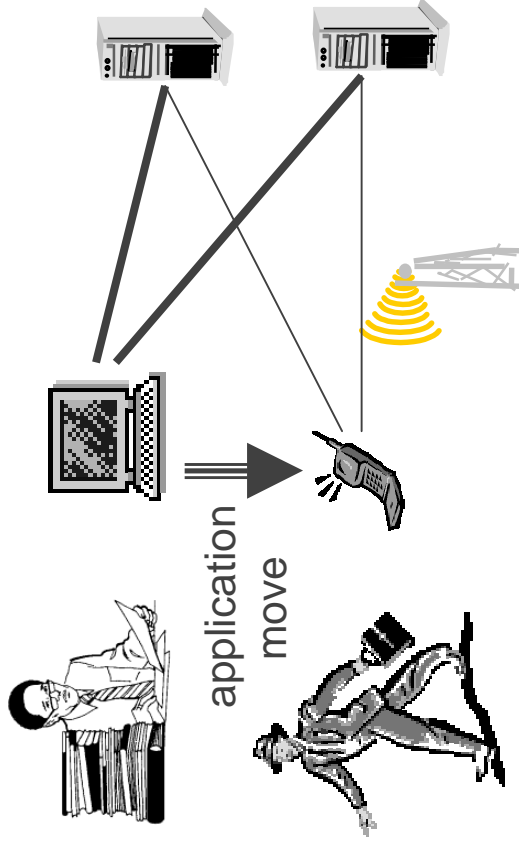


Outline

- Problems and Design Goals
- System Architecture
- Building Adaptive Groupware Applications
- Lessons Learned and Open Issues
- Summary

Problems

- Variability and heterogeneity
 - User and application demands
 - Hardware and network variability
- Flexibility and adaptability
- Application and user mobility
 - Intermittent connectivity
 - Support for offline users

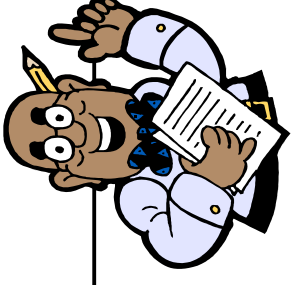




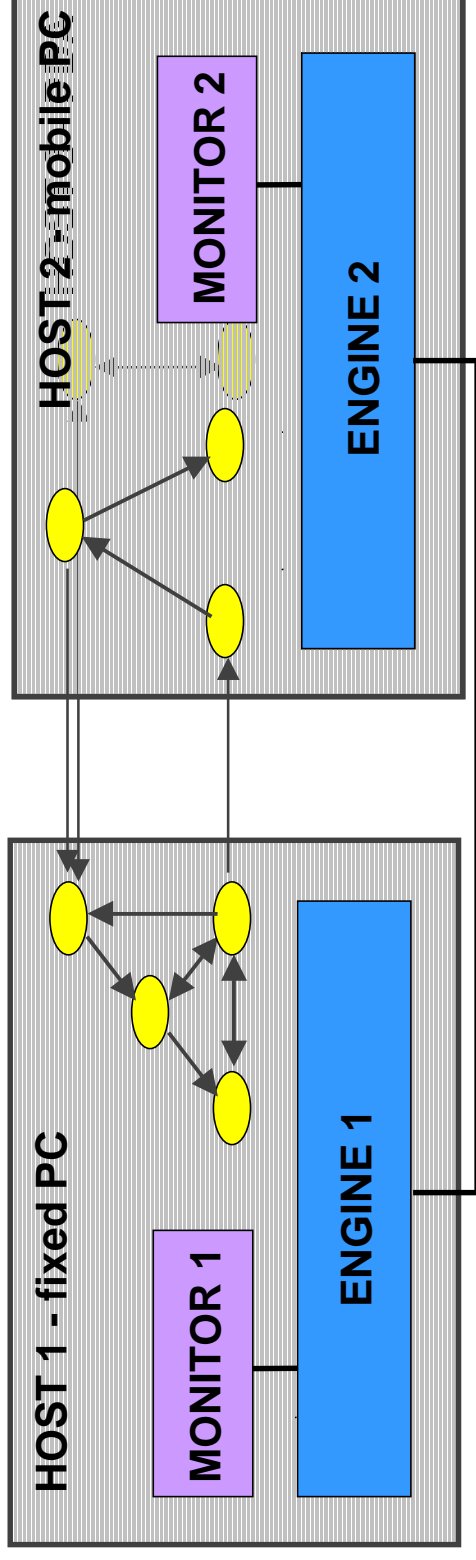
DACIA* Features

- Component-based framework
- Dynamic application reconfiguration
- Component mobility
- Persistent connectivity between components
- 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 (Processing and Routing Component)

- Communication through ports
- Component mobility – state transfer

Component Mobility (I)

Component Mobility (II)

DACIA: application layout for host saturn

File View Engines Procs Help

saturn

seoul

449016 Chat

sanjuan

402016 Chat

Connections :
Socket[addr=seoul.eecs.umich.edu/141.213.10.]

Local PROCs :
Remote PROCs :
402016 Chat:1, 2 ports, host: sanjuan
0 -> 449016:1
1 -> 449016:0
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0

```

X TERM2
Remote PROCs :
449016 Chat:1,
0 -> 402016
Engine > Can't send
print
Connections :
Socket[addr=seoul.eecs.umich.edu/141.213.10.147,port=34192,localport=6301]
Socket[addr=seoul.eecs.umich.edu/141.213.10.147,port=34192,localport=6301]
Local PROCs :
402016 Chat:1,
0 -> 449016
1 -> 449016
Remote PROCs :
449016 Chat:1,
0 -> 402016
1 -> 402016
Engine > Warning: *
print
Connections :
Socket[addr=seoul.eecs.umich.edu/141.213.10.147,port=34192,localport=6301]
Socket[addr=seoul.eecs.umich.edu/141.213.10.147,port=34192,localport=6301]
Local PROCs :
Remote PROCs :
402016 Chat:1, 2 ports, host: sanjuan
0 -> 449016:1
1 -> 449016:0
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
Engine >
  
```

DACIA: application layout for host sanjuan

File View Engines Procs Help

sanjuan

402016 Chat

seoul

449016 Chat

Connections :
Socket[addr=saturn/141.213.10.100,port=6301]

Local PROCs :
402016 Chat:1, 2 ports, host: sanjuan
0 -> 449016:1
1 -> 449016:0
Remote PROCs :
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0

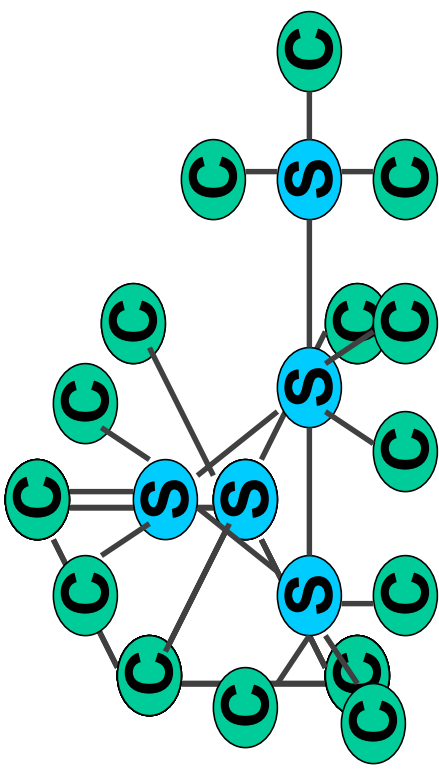
```

Chat
Menu
Enter a new message:
Send
Latest message:
OUT: Hello, chat user from seoul
IN : Where are you located ?
OUT: I am on saturn.
OUT: I am moving to sanjuan right now.
IN : OK, I am waiting...
Engine > print
Connections :
Socket[addr=saturn/141.213.10.100,port=6301]
Local PROCs :
402016 Chat:1, 2 ports, host: sanjuan
0 -> 449016:1
1 -> 449016:0
Remote PROCs :
449016 Chat:1, 2 ports, host: seoul
0 -> 402016:1
1 -> 402016:0
Engine >
  
```

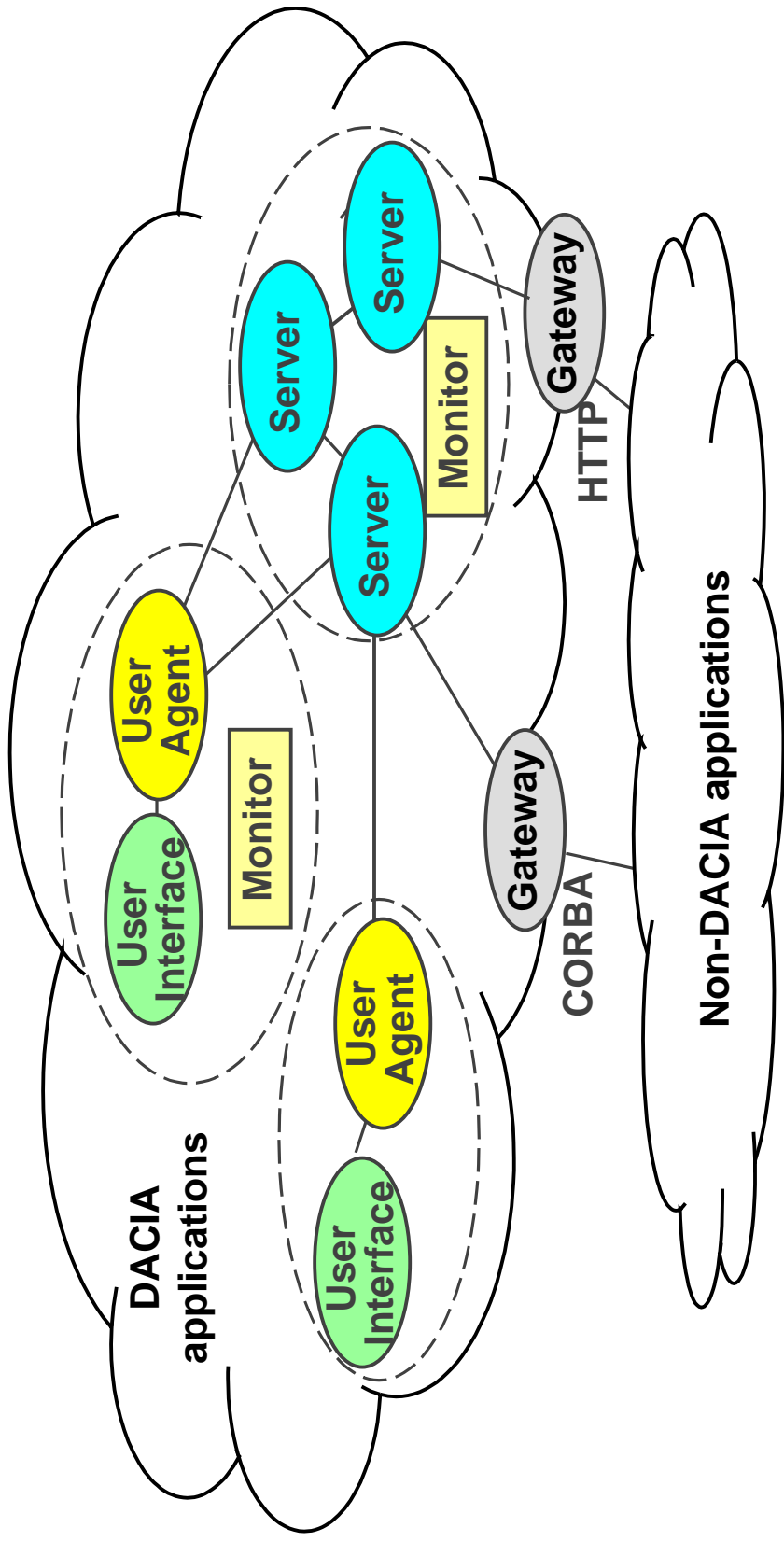
Dynamic Application Reconfiguration

- Change connections between components
- Change components' location
- Load new components
- Mechanisms
 - Specialized monitors
 - Programming API
 - Command-line interface
 - Graphical interface

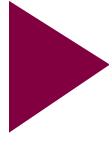
An adaptive application:
multi-party communication



Structuring Groupware Applications



- Types of components
- Separate *User Interface* and *User Agent* code
- Horizontal/vertical decomposition

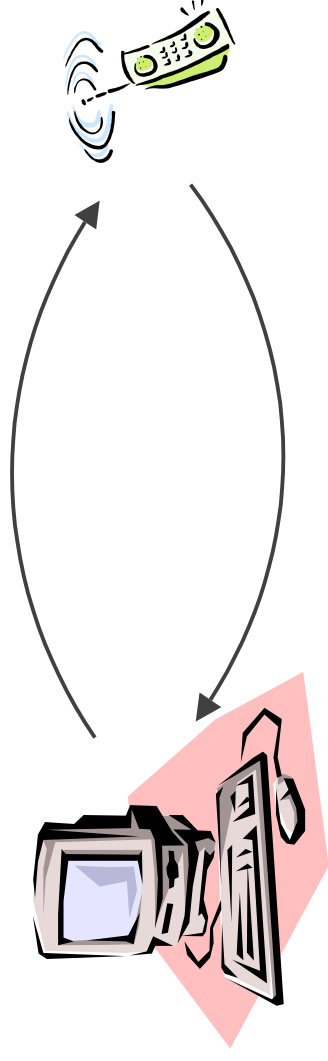


Division of Functionality and Coordination

- Separate mechanisms from adaptive policies
- Measure application performance
- Distributed coordination
 - Multiple administrative domains
 - Cooperation vs. competition

Support for Multiple Devices

- Context-aware applications
- Move components between different types of devices
- Hardware-dependent components
 - All-in-one
 - Multiple versions for one component
- State transfer





Security

- Host security
 - Malicious components
 - Programming errors
- Component security
 - Execution integrity
 - Data integrity
 - Secrecy



Summary

- Component-based framework for building adaptive groupware applications
- Dynamic reconfiguration
- Application and user mobility

Open issues:

- Application decomposition and coordination
- Support or multiple devices
- Security