

Facilitating Agent Messaging on PDAs

Sasu Tarkoma

Helsinki Institute for Information Technology,
University of Helsinki

Mikko Laukkanen

Sonera Corporation

MATA'02 FOURTH INTERNATIONAL
WORKSHOP ON **MOBILE AGENTS** FOR
TELECOMMUNICATION **APPLICATIONS**

Contents

- Introduction
- MicroFIPA-OS
- Internal Performance
- External Performance
- Conclusions

Introduction

- Agent Technology and middleware on PDAs
 - ◆ Agent-based services for nomadic users using PDA devices?
 - ◆ EU IST 5th framework project CRUMPET
- Challenges
 - ◆ Resource-constrained environment, mobility
 - ◆ Communication over slow and unreliable wireless links
- Agents rely on asynchronous messaging
 - ◆ Well-suited to mobile and wireless environments
- Requirements:
 - ◆ Environment: Scalability to different configurations.
 - ◆ Messaging: Support for disconnected operation, minimize latency and overhead, reliability.

Background and Related Work

- FIPA-OS is an Open Source agent toolkit based on the FIPA specifications
 - ◆ The platform consists of AMS,DF,ACC and agents.
 - ◆ Agent programming is based on tasks.
- MicroFIPA-OS is based on FIPA-OS and PersonalJava
 - ◆ PJAVA widely supported, available for different environments, compatible with JDK 1.1, and the PersonalProfile
 - ◆ CLDC and MIDP were found to be too limiting
- JADE-LEAP, lightweight agent platform
 - ◆ JADE for small devices
 - ◆ Focus on agent container, not full platform on PDA

MicroFIPA-OS

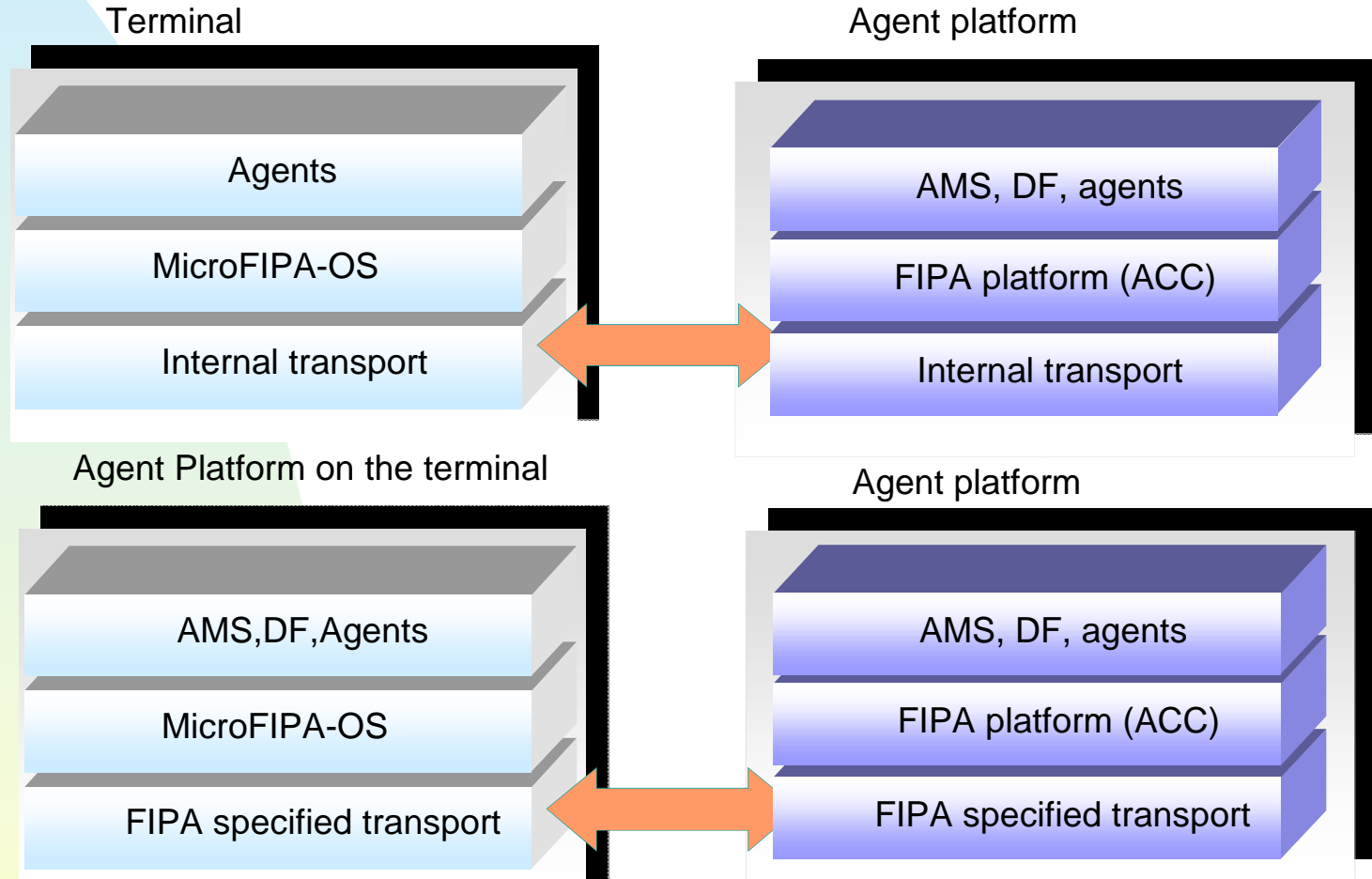
- We have previously examined FIPA-OS 1.3.0 with minor modifications on PDAs (ATAL-2001)
- MicroFIPA-OS was derived from FIPA-OS based on these observations
 - ◆ Supports the same API and interfaces: AMS,DF
 - ◆ Thread and resource pools
 - ◆ Non-mandatory XML parsing
 - ◆ Minimal mode API
 - ◆ Modifications to communication stack
 - ◆ New conversation and messaging layers
 - ◆ Nomadic Application Support
- Current focus on internal and external messaging

Messaging

- FIPA-OS uses RMI for internal communication and a centralized ACC for external communication
 - ◆ Each agent has its own transport stack
- MicroFIPA-OS agents share the transports
- RMI is not efficient in wireless environments
 - ◆ New default protocol is based on sockets.
 - ◆ HTTP-based external protocol for interoperability
 - ◆ Transports may implement different buffering and delivery policies




Deployment

- We have two deployment models:
 - ◆ Peer-to-peer
 - ◆ Agent platforms use standards compliant protocols to communicate with each other
 - ◆ Client-server
 - ◆ Browser-clients, proxy-based approaches
 - ◆ Partial platforms: agents execute on the client, support for disconnections
 - ◆ MicroFIPA-OS provides support for both models by allowing the execution of AMS and DF on the client and interoperability with the HTTP transport.
- The internal performance measurements were done using an agent platform on the PDA (AMS,DF,test agent)





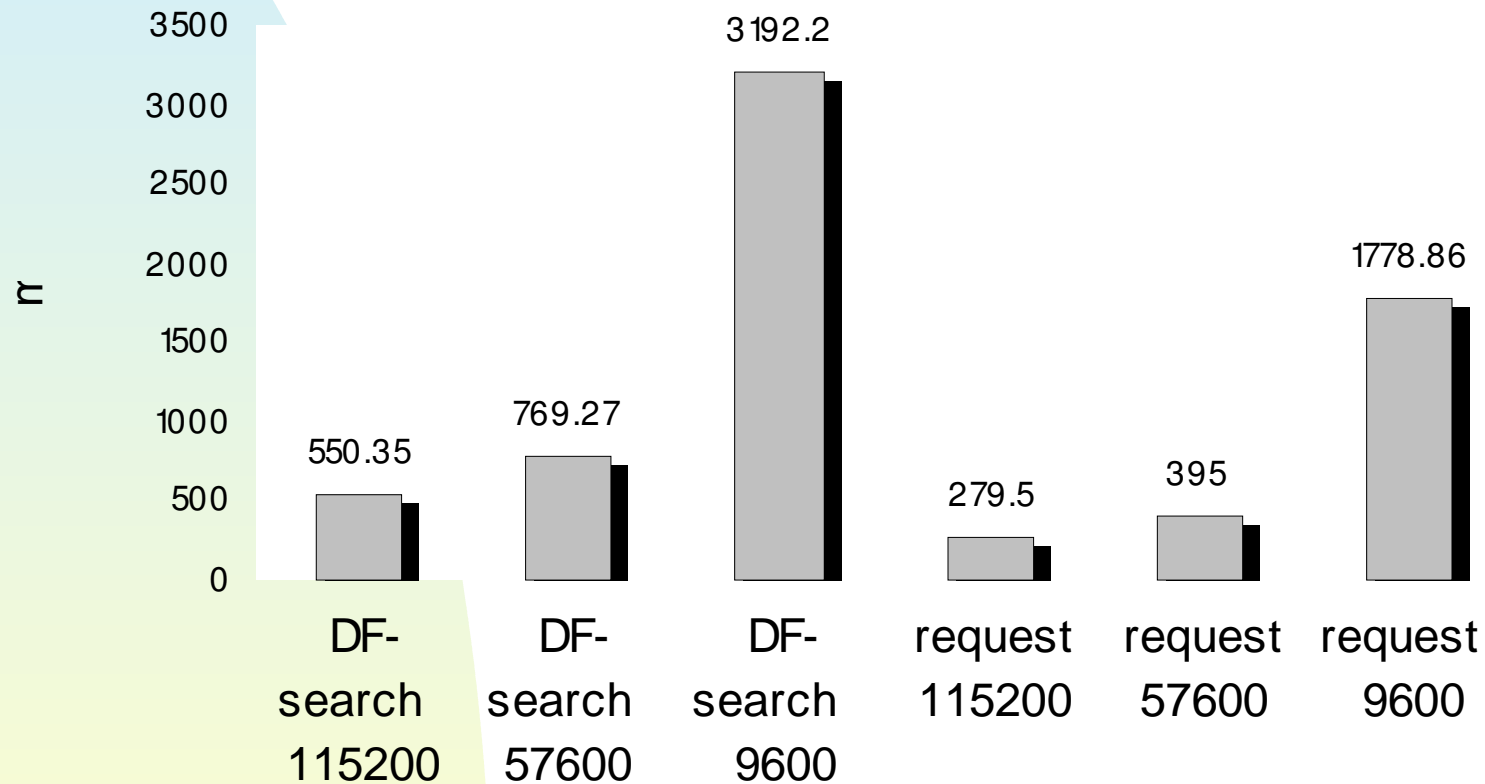
Environment

		-		
	Casio Cassiopeia E 115		Compaq iPAQ H3630	Laptop
CPU	131MHz MIPS		206MHz StrongARM	500MHz
OS	Windows CE 2.11		Linux 2.4.3	Linux 2.2.16
JVM	Sun Pjava 1.0		Sun JDK 1.1.8	Sun JDK 1.1.8
Caffeine Mark	27		120	679

Internal/Local Performance

	Casio Cassiopeia E-115	Compaq iPAQ H3630	Laptop
Startup (ms) ams+df+test agent	16574.5 +/- 1509	2206.5 +/- 1.58	439.2 +/- 32.1
DF-search (ms)	7765.6 +/- 568	932.8 +/- 211	153.51 +/- 11
FIPA-request (ms)	861.3 +/- 484.2	144 +/- 22	18.1 +/- 9.6
FIPA-request minimal (ms)	161.2 +/- 15.7	37.3 +/- 0.48	10.2 +/- 0.42
Runtime footprint	3 MB	4.1MB	3.9 MB

External Performance (iPAQ)



Discussion

- Task and Conversation Management are more efficient than during our initial benchmarks using FIPA-OS 1.3.0 on PDAs
 - ◆ Local FIPA-Request is approx. 23 and DF-search 4 times faster (E-115).
- They still represent considerable overhead for low-end PDAs
 - ◆ DF-search can be considered to be too slow on E-115
- One solution is to use the minimal API and still use ACL for agent communication
- The overall system may benefit from accessing complex external resources
- With wireless and slow links external communication should be minimized
 - ◆ Local processing required



Conclusions

- We have presented our experiences with running an agent execution environment on two different PDAs
 - ◆ It is possible to execute AMS and DF on PDAs
- The internal and external performance measurements indicate that the price of portability (Java, agent code) is an increased demand for resources
- Agents targeted at PDA devices need to be relatively simple in order to gain reasonable performance
- Distribution of computationally heavy components on fixed network hosts relieves performance bottlenecks, however the communication cost must also be taken into account

Future Work

- Wireless Middleware
 - ◆ Multimode communication
 - ◆ Event Service
- Modelling Communication Context
- Reconfiguration of client systems