



TC_Adhoc@FIPA25

Scope: Enhance interoperability between FIPA agents operating in ad-hoc environments

Chair: Michael Berger, Siemens

Co-Chair: Heikki Helin, Sonera



Problem statement

- In an ad-hoc network, devices use wireless technologies to communicate in a peer-to-peer (i.e., device-to-device) fashion. Thus, **agents** on two mobile devices, originally created **on different platforms**, need to **discover each other and build dynamically ad-hoc compounds** which allow each agent to communicate with the other.
- Because of **limited memory resources**, the device may **not be able to host all functionalities** requested by FIPA. Therefore, the management of agents and their communication under these constraints must be handled.



Focus

- Definition of possible agent platform fragments, which can form dynamically a compound.
- Definition of mechanisms and protocols for agent platform fragments to build, release, join and leave compounds.
- Usage of existing approaches which provide support on different levels (e.g., Bluetooth ad-hoc networks, JXTA P2P approach).



TC Ad-hoc History and Goals

- 
- A vertical red arrow with horizontal crossbars at the top and bottom, pointing downwards, is positioned on the left side of the slide, spanning the height of the main content area.
- FIPA24 (Feb. 2002, Lausanne)
 - Discussion of ad-hoc proposal to 2nd CfT of TC Gateways
 - TC Ad-hoc started, accepted workplan f-wp-00020 (16 month program of work)
 - TC Ad-hoc 1st CfT published
 - FIPA25 (May 2002, Vancouver)
 - Presentation and discussion of technical contributions to 1st CfT of TC Ad-hoc
 - FIPA27 (Oct. 2002)
 - Deliver first draft of preliminary specification
 - FIPA29 (April 2003)
 - Making specification as experimental



TC Ad-hoc meetings at FIPA25

- Monday, 14:00 - 18:00, Oak1
- Tuesday, 9:00 - 12:30, Oak1
- Thursday, 14:00 - 18:00, Oak1

- 4+1 technical proposals as result of 1st CfT:
 - Olga Ratsimor et al., UMBC, Baltimore
 - Jamie Lawrence; MLE - Media Lab Europe, Dublin
 - M. Berger and M. Watzke; Siemens AG, Munich (reviewed draft from FIPA24)
 - Celeste Campo; Universidad Carlos III de Madrid
 - Makoto Okada, Fujitsu



Proposal from UMBC

- Lightweight version of mandatory platform on each device
 - Introducing new elements to establish „alliances“
 - Cache: storing advertisements from other platforms
 - Advertisement: broadcast announcements to neighborhood
 - Forwarder: forwarding announces
 - Policy manager: device / application announcing and caching
 - Discovery protocol: first ask DF, than ask Cache, *cast a request
-
- Discussion:
 - Solution on how to alliance -> Announcement broadcasting and policy dependent caching
 - Cache Manager in addition to existing DF
 - Advertisement as agent or AMS change -> not decided
 - Caching: explicit services / agents or DF addresses
 - Should the definition of alliances go to FIPA? -> not decided



Proposal from MLE, Dynamic Interaction Group

- Assumptions:
 - Use any existing service discovery protocol
 - Discover and maintain routes for message transport; may provide multi-hop routes
 - Small ad-hoc nodes (resource constraints, mostly host a single agent)
- Proposed changes
 - remove AMS and DF as mandatory
 - Introduce a Discovery Agent
 - Leased directory entries in AMS and DF
 - notifications from AMS and DF upon directory updates
 - addition of a lite-AMS



Proposal from MLE, Dynamic Interaction Group

- Platform: current FIPA definition
- Fragment?
- liteAMS: agent and platform management
 - -> Agent management will always be needed, white pages directory is not needed -> lite AMS for management only
- Discovery Agent
 - replace the DF at each fragment (advertising directories and agents)
 - discovers other fragments and their agents
 - notifies local agents of discovered directories and agents
 - controls local activation of DF and AMS



Proposal from MLE, Dynamic Interaction Group

- Fragment becomes a Platform
 - DA starts AMS and DF (when device capable, DF required)
 - DA registers all agents with the AMS
 - Lite AMS will shutdown? Suspend? Sync? Remain?



Proposal University of Madrid

1.) DF fragment on each device

2.) Analysis of service discovery in the fixed network:
SLP: Service Location Protocol, IETF

User Agent for service discovery

Service Agents advertising their location and attributes

Directory Agents store information about services

Operation modes:

a) DA collects service info, UA unicasts the requests to DA

b) no DA, UAs repeatedly multicast their request, SAs listen to these requests, unicast responses to UA

Jini: Sun, HW and SW are Java Objects

DA (Jini Lookup Service, lease times, ...),

SAs announce services to DA by unicast, UAs request DA



Proposal University of Madrid

Salutation: focus is large sets of devices and services

DA (Salutation Manager), discovery for applications, services, and devices

SSDP: for UPnP, minimal protocol for multicast-based discovery

Optional DA (Service Directory)

SAs send announcement message by multicast or broadcast

Alternatively, announcement by unicast to DA

Discovery by asking the DA or multicast request message

- Problems with all these proposals:
 - central server for mobile adhoc networks not suitable
 - use of multicast or broadcast is power consuming



Proposal University of Madrid

3.) Introduction of new Service Discovery Protocol (PDP)

- push and pull solutions
- DEAPspace algorithm
 - push solution, services will be broadcasted
 - every device has DA, knows all services around
- Goal of PDP: devices announce its services only when others request that service -> broadcast to all devices which are interested on new services
- each device has list of services and timeout of availability
- PDP request, PDP reply



Proposal from Siemens + FIPA24

- **Announcing**
 - Broadcast of an ACL-based agent platform announcement message
 - Broadcast based on HTTPMU protocol (UDP Multicast of HTTP messages)
 - Specified by UPnP Forum Technical Committee
 - Message in a single UDP packet
 - ACL message and envelope represented in XML (FIPA XC00071, FIPA XC00085 specifications)
- **Dissolution**
 - Try and error (just find out the disconnection in case of trying to use a service from another platform); deregister the local DF entries referring to the remote DF
 - Define a clear defederation by a special message as optional polite way to a single platform or to all (optional single inform)



Proposal Fujitsu (made during meeting)

- 1st step: search for device
 - Usage of UDP
 - polling on fixed port number
 - URI = machine name + connport + IP
 - Usage of URI 020
 - Gnutella uses 80
 - WinMX uses 96
 - the devices must have a known IP address already
- 2nd step
 - send query, getting description file back
 - every device has a cache storing the information



Discussion at FIPA25



Discussion

- Proposals on different levels
 - -> Structure and order proposals needed, common view

Comparison/Common things in the proposals:

Siemens	UMBC	MLE
<i>Components on device:</i>		
DF?, AMS	DF (+cache), AMS	DA, AMS-lite (DF), AMS, DA
<i>Service discovery:</i>		
DA on AMS	Protocol, Advertisements	DA
<i>Platform discovery; underlying mechanism (acc address):</i>		
Broadcast	Broadcast/caching	?
HTTPmu/UDP		JXTA, ...
<i>Add-ons:</i>		
	Policies	



Discussion

- -> First architecture definition

	Caching, profiling, ...		
<i>ACL</i>	Proactive (broadcast)		Reactive (DA)
<i>Discovery</i>	SDP	JXTA, Jini, UpnP, ...	
		Httpmu, broadcast, ...	
<i>Physical</i>	BT	IP	IrDA
		WLAN	LAN



Discussion

- Library of discovery protocols:
 - Need to specify an "interface" to a set of discovery protocols
 - E.g., service number in BT, some text information in SLP, etc.
 - At agent level we shouldn't consider (at all) how the actual platform and service discovery is done / implemented
- Discussion about setting up a DF federation
 - 4 possibilities identified at FIPA24: (1) asking for other DFs, (2) announcing my DF, (3) announcing my DF, getting some back, and (4) asking directly for a service by broadcast (CFP for all)
- Discussion about optional DF
 - DF definitely not needed in all platforms; agents in a platform w/o DF can use remote DF to announce their services/search other services -> Decision from TC X2S: DF optional!! FIPA25, TC Ad-hoc



Results

- Two separate issues:
 - definition of fragments and compounds
 - service and platform discovery (take in consideration that DF is optional now!!)
- Differentiate between platform and service discovery
 - Goal: deal with service discovery, hide platform discovery
- Decision: we discuss service discovery at the agent level
- Don't care for multi-hop sending of messages
- Writing a white paper about possible solutions handling agents ad-hoc environments



White Paper content and responsibilities

- 0.) Device discovery (BT, WLAN, IrDA)
- 1.) Platform discovery
 - a) ACC / platform description (<URL>, <serviceID>, <JiniServiceID>) **every, according to every mechanism in 1b)**
 - b) how can we use the following discovery mechanisms:
 - UDP (HTTPMU, own) **FJ, Sie**
 - BT **UMBC**
 - Jini **MLE**
 - UpNP **FJ**
 - JXTA **MLE**
 - Salutation **Son**
 - Routing / MANET **MLE**
 - SCP / SSDP / Chord **Sie**
 - all others **all**



Document

- 2.) Service discovery (on agent level)
 - 2a) message format **Son + all**
 - Definiton of ACL broadcast / multicast (+ implications to X2S)
 - pull; push, subscribe (just one predicate?)
 - 2b) Infrastructure **Sie + all**
 - DA, policy manager, AMSlite, ontology, ACC ?...
 - Discover Service (e.g. DF), or Agent or just a protocol, leasing
 - Definition of fragments
 - Implications to X2S
 - 3.) caching, profiling / policy, security (hand over to WG security) **UMBC, QML**
- > 24.5.02 template Siemens; 12.7.02 chapter 1, 2, 3 optional



FIPA25, Adhoc attendees

- Jamie Lawrence, MLE, jamiel@mle.media.mit.edu
- Olga Ratsimor, UMBC, oratsi2@cs.umbc.edu
- Heimo Laamanen, Sonera, Heimo.Laamanen@sonera.com
- James Odell, JOA, email@jamesodell.com
- Kari Koivuniemi, Minutor, kari.koivuniemi@minutor.fi
- Heikki Helin, Sonera, heikki.helin@sonera.com
- Bobbin Teegarden, Bot Inc., teegs@earthlink.net
- Makoto Okada, Fujitsu Labs, okadamkt@flab.fujitsu.co.jp
- Michael Berger, Siemens AG, m.berger@siemens.com
- Stefan Poslad, QueenMaryUnivLondon,
Stefan.Poslad@elec.qmul.ac.uk