Organizing Ad Hoc Agents in Smart Environments

Colloquium Computer Graphics & CAD CAM
W.Pasman
March 25, 2004
Overview

Problem description
Existing solutions
Service matching solution
Organizing agents: efficiency, context awareness
Example of service matching
Agent negotiation in detail
Problem Description

Large Smart Ad Hoc Environments:

- Environment is full of agents (autonomous pieces of software) trying to offer services
- Ad-Hoc: Agents can appear, move or disappear at any time
- Agents all speak different languages (ontologies)

How does the user find the agent he needs?
Existing approaches

• Menu or list
  - does not scale to large environments
  - cumbersome if available
    services not fixed
• Simple script-like natural language interface
  - user has to remember keywords for services, eg
    “turn on - master bedroom light”
• Centralized natural language interpreter
  - highly complex, inflexible central conversion
  - conversion to semantics is highly task-dependent
    and does not fit with central approach
Our approach: Service Matching

- Free Natural Language
  + no keywords to remember
- Interpretation at individual Agent level (distributed),
  + clear task frame
  + proven robust technology for free natural language
- two-step understanding
  + first ask agents if they UNDERSTAND and CAN HANDLE a request. Activate only one if multiple understand the request.
Context Awareness

Broadcast request to ALL agents is not a good idea:
- completely overload the system (millions of services?)
- “your coffee is served in Sydney”?

Solution: Context Awareness to every agent
- All agents know and can communicate about task- and location related agents
Related Agents Graph
knowledge distributed over agents
Using Relations for Service Matching

Example Use:
• user in kitchen,
• Just turned on the light
• now asks the Service Matcher
Agent Negotiation in (some) Detail

1: GUI event

PersonalAgent

2: AttemptHandling(“coffee with...”)

ServiceMatcher

3a: ((all ?x (= ?x (aRelatedAgent ?y ?z))))
3b: AttemptInt(“coffee with..”)

Other agents
In system

4: NOT UNDERSTOOD
5: AttemptInt("coffee with...")

6: Slot Filling/ Parsing

ServiceMatcher

CoffeeMachineNLI

7a: ((iota ?a (ItemInStock sugar ?s)))

7b: ((iota ?p (CanMakeCoffee ?p 10 )))

HKGroceryStorage

HKFridgeContent

CoffeeMachine
CoffeeMachineNLI

8: (ItemInStock water ?w)
   (ItemInStock coffeepowder ?c)

9: (= w 3)
   (= c 6)

10: OpenGUI

11: (= p 0.9)

12: (= s 3)
   (= s 0)

HKGroceryStorage

HKFridgeContent

...
13: Interpretation {
    Msg(receiver=CoffeeMachine "MakeCoffee amount:1")
    understanding 1.0 executable 0.9 }

Or

Interpretation {
    Msg(receiver=CoffeeMachineGUI
    "OpenGUI sugar 2 unit 0 amount 0 type 'coffee' milk 0")
    understanding=1.0 executable 0.8 }

14: Interpretations

Other drink machines
ServiceMatcher

15: Execute(Interpretation ....)

CoffeeMachineNLI
17a: OpenGUI sugar 2 unit 0 amount 0 type 'coffee' milk 0

17b: MakeCoffee amount 1

18: MakeCoffee amount 1
DEMO

(if time left...)

March 25, 2004