User Focus Management in Agent Worlds

W.Pasman

W.Pasman@twi.tudelft.nl

CACTUS Plenary Meeting, June 12 2003, Den Haag

Overview

1. Research Question
2. General Approach
3. Architecture
4. Methodology

Research Question

Nearby future: User between huge amount of agents, and he doesn’t know which agent(s) can help him with his task

Question: How can we support user in selection of, and navigation through agents?

General Approach

1. Natural language seems suited for such uncertain situations, allowing users to describe their needs in a flexible way.

2. Design a mechanism to select the agent that can handle that needs best and that can negotiate further with user if necessary.

3. Connect user to selected agent.

4. Detect when user is finished, stuck, having an interrupt etc and handle accordingly.

About Direct Interaction

Direct interaction: agents are commanded by manipulating physical objects

Problems:

1. Agents are not always reachable (ceiling light)
2. User does not really care how he is helped (“please light the room”)
3. Service often involves multiple physical objects (“please light this vase”, “let’s see the Matrix”)

4. Services often not naturally linkable to an object (the travel planner, the currency converter, Microsoft Office)

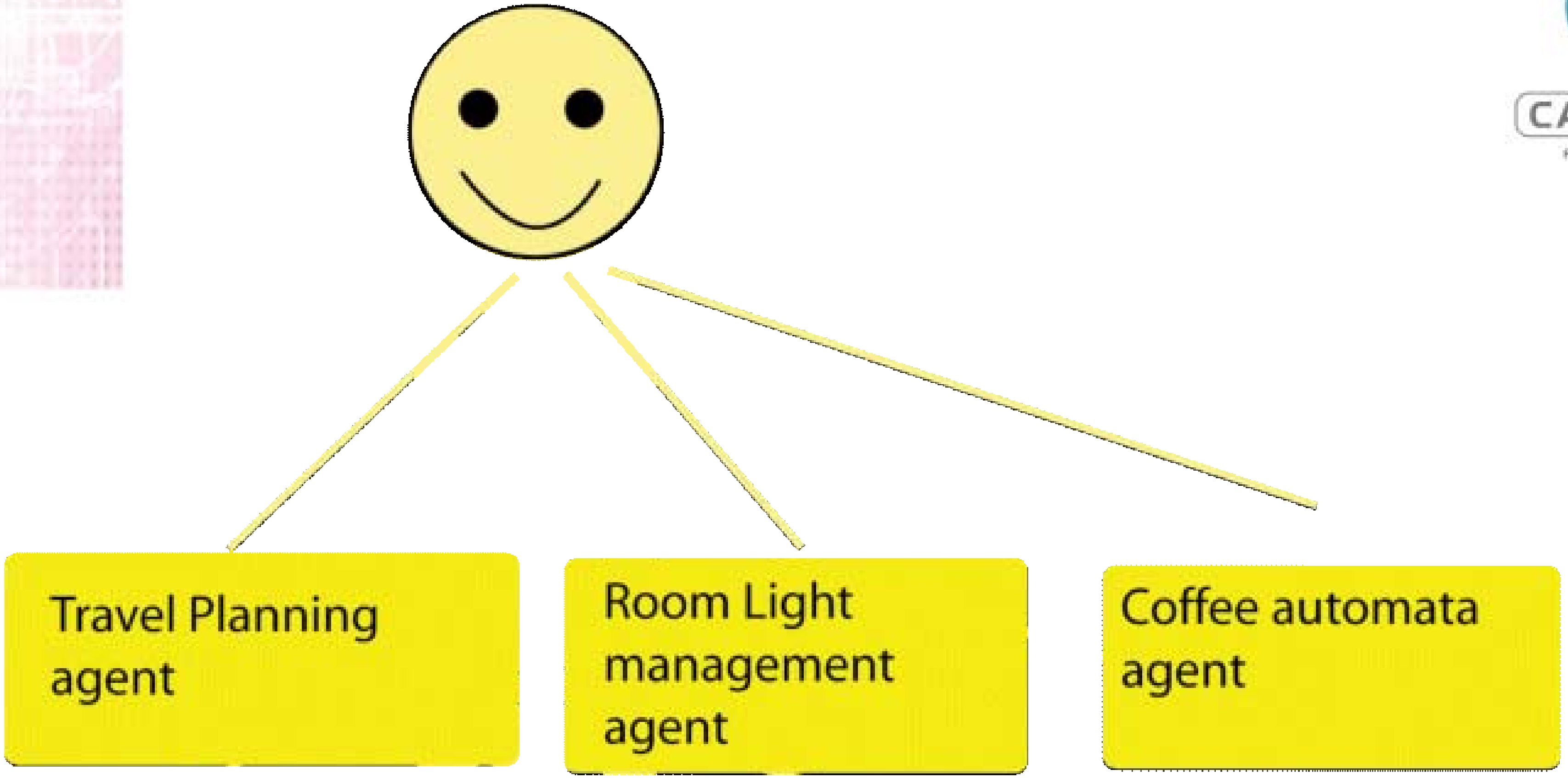
5. Coupling agents to some physical objects and not to others probably frustrates the user

-> Direct interaction IS convenient, fast etc but only **after** the user has learned the metaphores

Architecture

@1: Natural Language (NL)

- Every NL-enabled agent can inspect the NL query from the user and assess how suited it is to handle it.
- The agent can consult context (history, user profile, gaze direction…) for assessment.
- Agent returns user to calling agent (focus manager?) when the conversation runs out of its focus



Travel Planning agent

Room Light management agent

Coffee automata agent

travel, route,
from, to,
<date>,<time>,
arrive, depart,
....

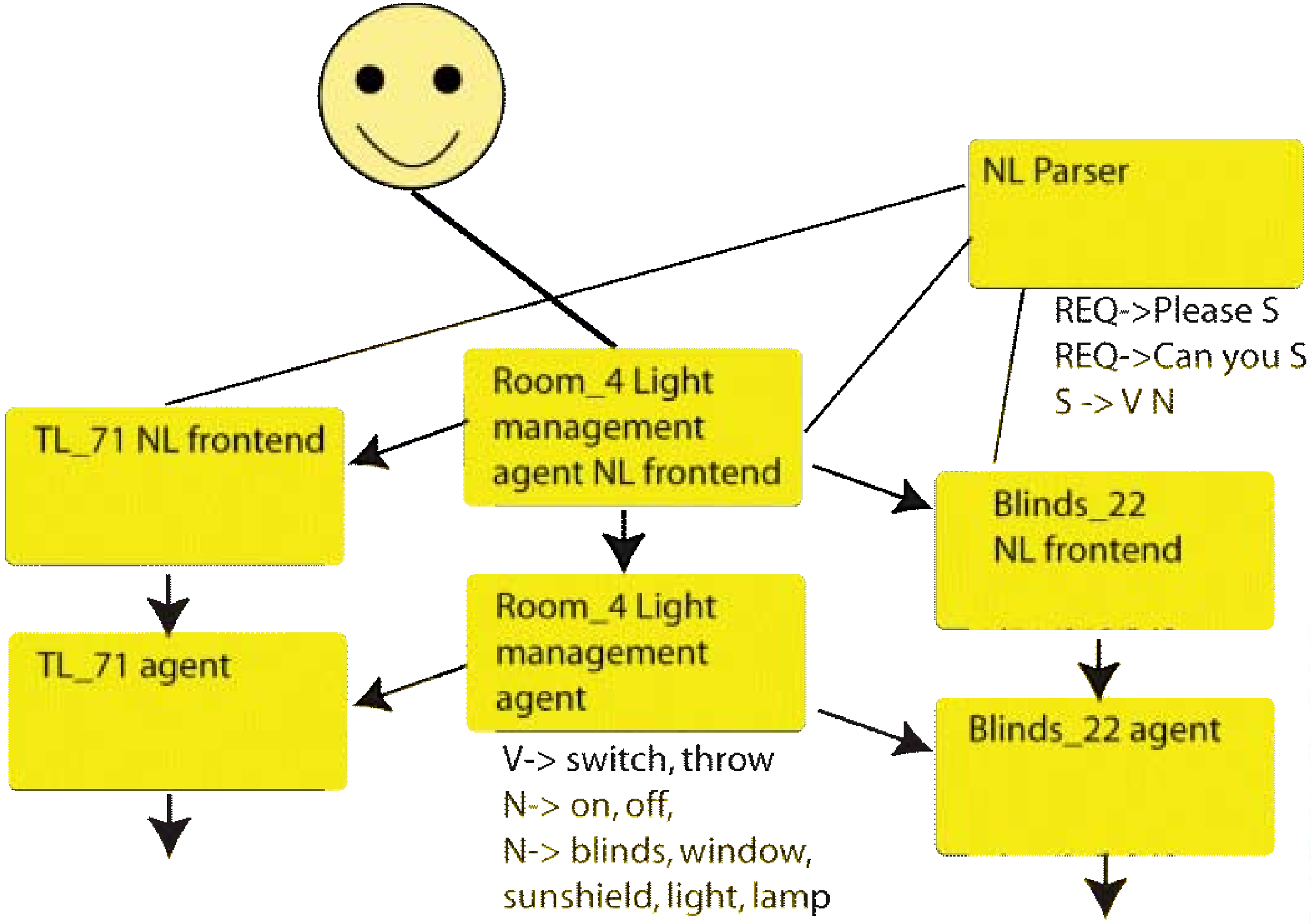
lamp, light,
brighter, dimmer,
switch, on, off,
blinds, window,
sunshield
...

Coffee, tea, soup,
hot, cold, sugar,
cream, cup, can,
price, payment, coin,
...

NL parsing to speech acts

“Please light this painting” --> REQUEST(action:light object:mondriaan3)

General parser structure available to convert text to speech acts, so that each agent has to add only task specific words & semantics



NL Parser

REQ->Please S
REQ->Can you S
S -> V N

Room_4 Light management agent NL frontend

TL_71 NL frontend

V-> switch, throw
N->TL, light

Blinds_22 NL frontend

TL_71 agent

Room_4 Light management agent

Blinds_22 agent

V-> switch, throw
N-> on, off,
N-> blinds, window, sunshield, light, lamp brighter, dimmer,
...

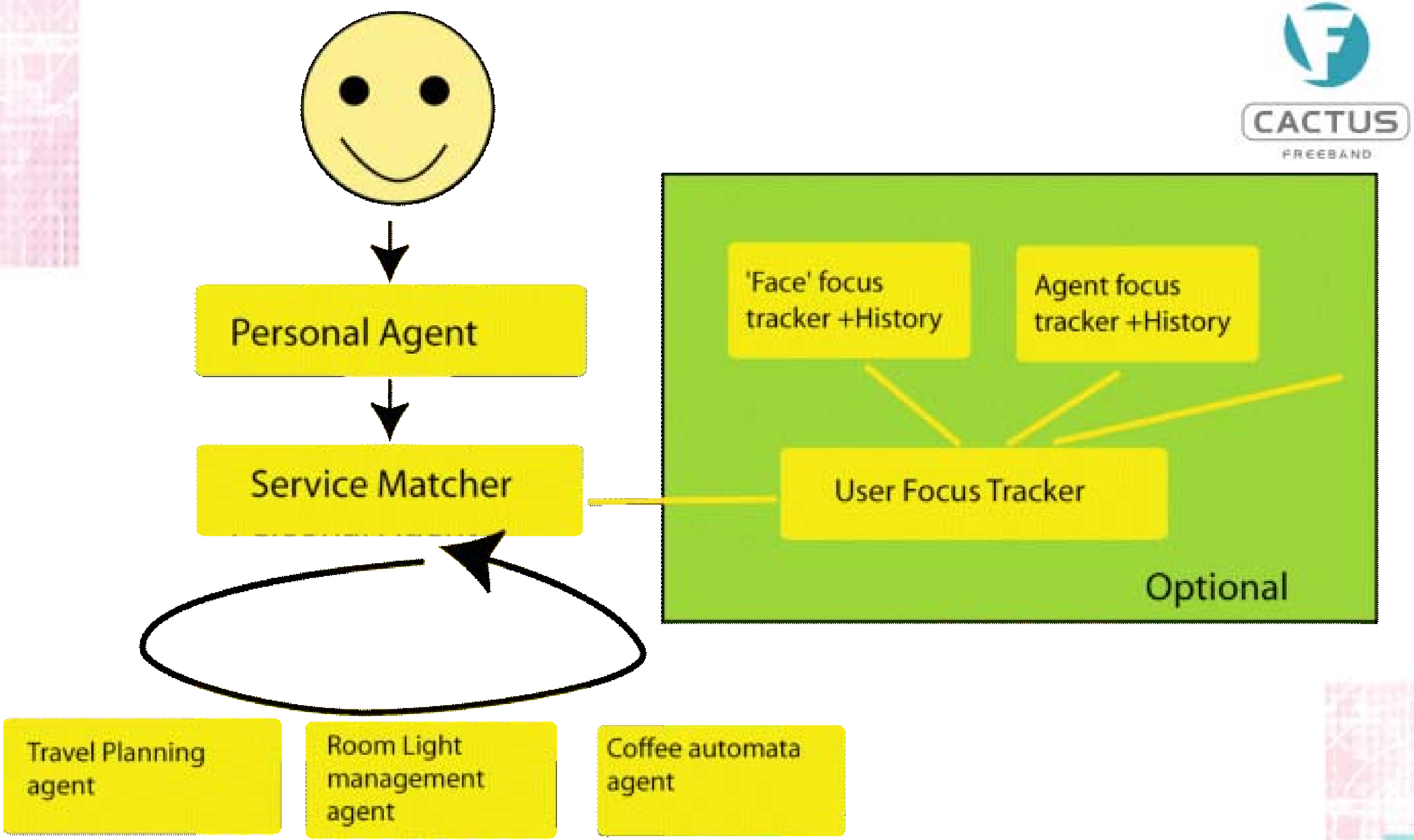
Service Matcher Agent

Coordinates search for the best agent

1. Query all NL active agents about confidence
2. Connect agent with highest confidence to the user. Ask user if more agents are highly confident. Enable more agents if no agent is very confident. Ask more user input if search takes too long or no more agents available.

Personal Agent/ iDEA

- Maintains connection between agent(s) and user
- Default target for interaction, both hardware and software.
- Personal agent has to ensure responsiveness of the personal hardware
- Personal hardware is minimal device for communication, especially if environment has insufficient or ‘hidden’ interface possibilities.



travel, route,
from, to,
<date>,<time>,
arrive, depart,
....

lamp, light,
brighter, dimmer,
switch, on, off,
blinds, window,
sunshield
...

Coffee, tea, soup,
hot, cold, sugar,
cream, cup, can,
price, payment, coin,
...

Methodology

Questions:

1. Is natural language effective for selecting a service?
2. Do various mechanisms to improve the service selection work?
3. Can our system detect and handle shifting focus properly?

@1: Is NL effective for selecting a service?

- User test
- Comparison with selection from a text list
- Distinguish thinking/reading from interfacing efford?
- Maybe need to put user in realistic scenario? Text intro, video intro, really outdoors, video game, …?

Minimal needs @1

Keyboard+text display
Personal Agent
Service matcher
sufficient number of services with NL interface

- How many services is ‘sufficient’ (20-30?)
- Ad-hoc changing services?
- Services don’t need to work. But that may frustrate users?

@2: Do various mechanisms to improve the service selection work?

- User test
- Comparison to situation without improvemt

Needs:
Focus tracking mechanisms
Agents using focus information

This may be lot of work, probably WoZ?

@3: Can our system detect and handle shifting focus properly?

- User Test
- Task involving multiple, tightly related services

Needed:

- Closely related but different services
- Focus shift detection mechanisms in agents or higherlevel/script agents

How about parallel tasks? Should we do work on that?