

# Service Matching in Ad Hoc Agent Systems

**Cactus Plenaire Vergadering/Mini Conference**

**W.Pasman**

**3 December 2003**



1

# Overview

## 1 State of Art in Service Matching

Smarthome HAL: commercial system

Chu-Carroll: Call Routing

CSLR Phoenix parser

Context handling in Story Understanding

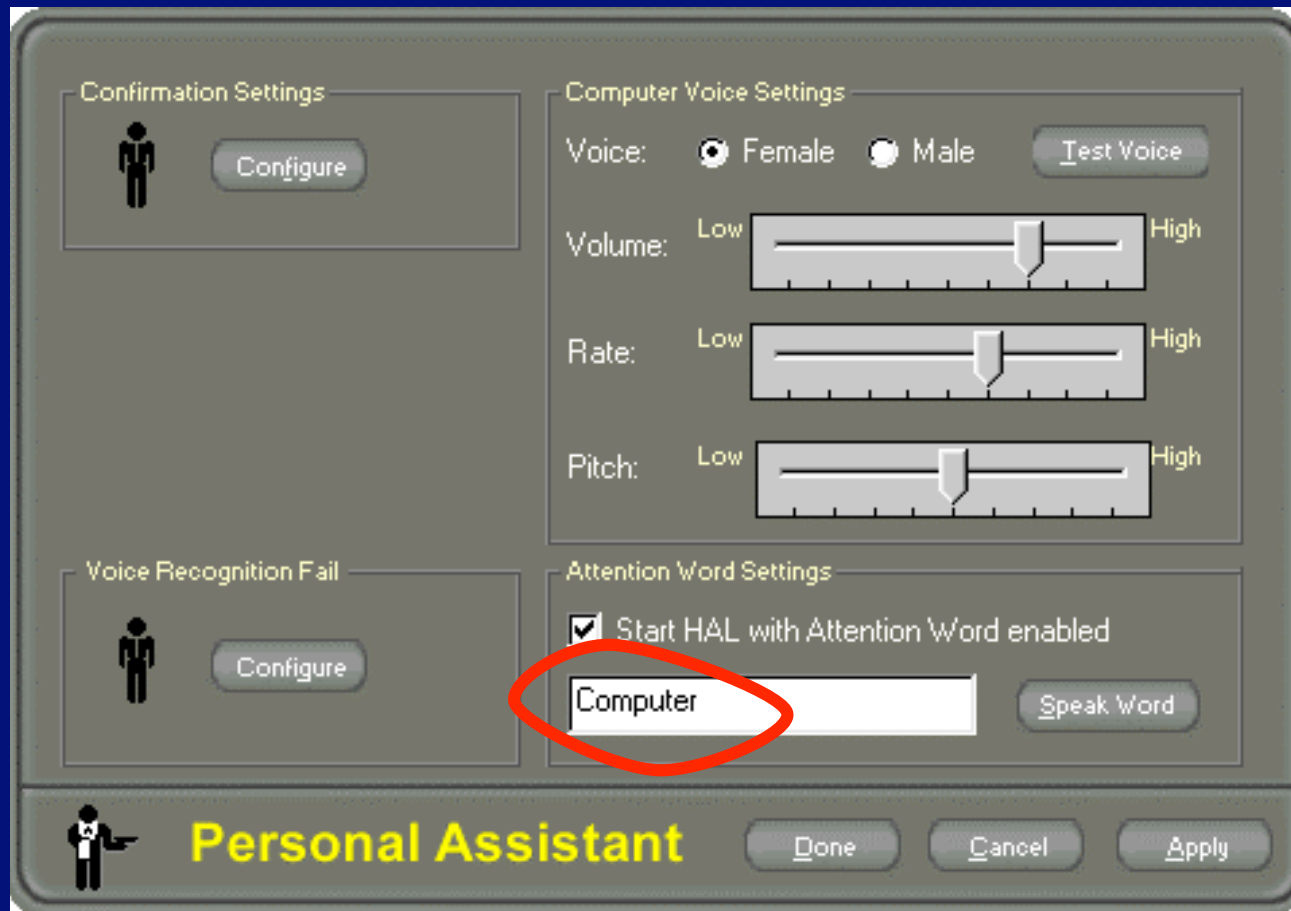
## 2 Cactus Testbed Architecture

Organizing agents

Service Matching

# Smarthome HAL

Voice/Script based system, X10 powerline communication  
Control via RC/mic/(mobile)phone



# X10 Device ID

HAL will automatically control the device(s) that have a matching X-10 code.  
Select the House Code and Unit Code for the device.

House Code: A01 Unit Code: 1

**Device Wizard** Cancel < Back Next >

HAL will allow you to turn appliances on and off, dim lights, open and close doors, and more!  
Select the type of action this device performs:

On/Off (e.g. lights and appliances)  
 Dimmable (lights)

Open/Unlock (powered doors/locks)

**Device Wizard** Cancel < Back Finish

Special  
controls

Give the device a name

You use this name to address the device



Enter a descriptive name for this device such as 'Front', 'Porch', 'Light', then select the type of device.

Location: LIVING

(optional) ROOM

(optional)

Device: LIGHT

Type: X10

LIVING ROOM LIGHT

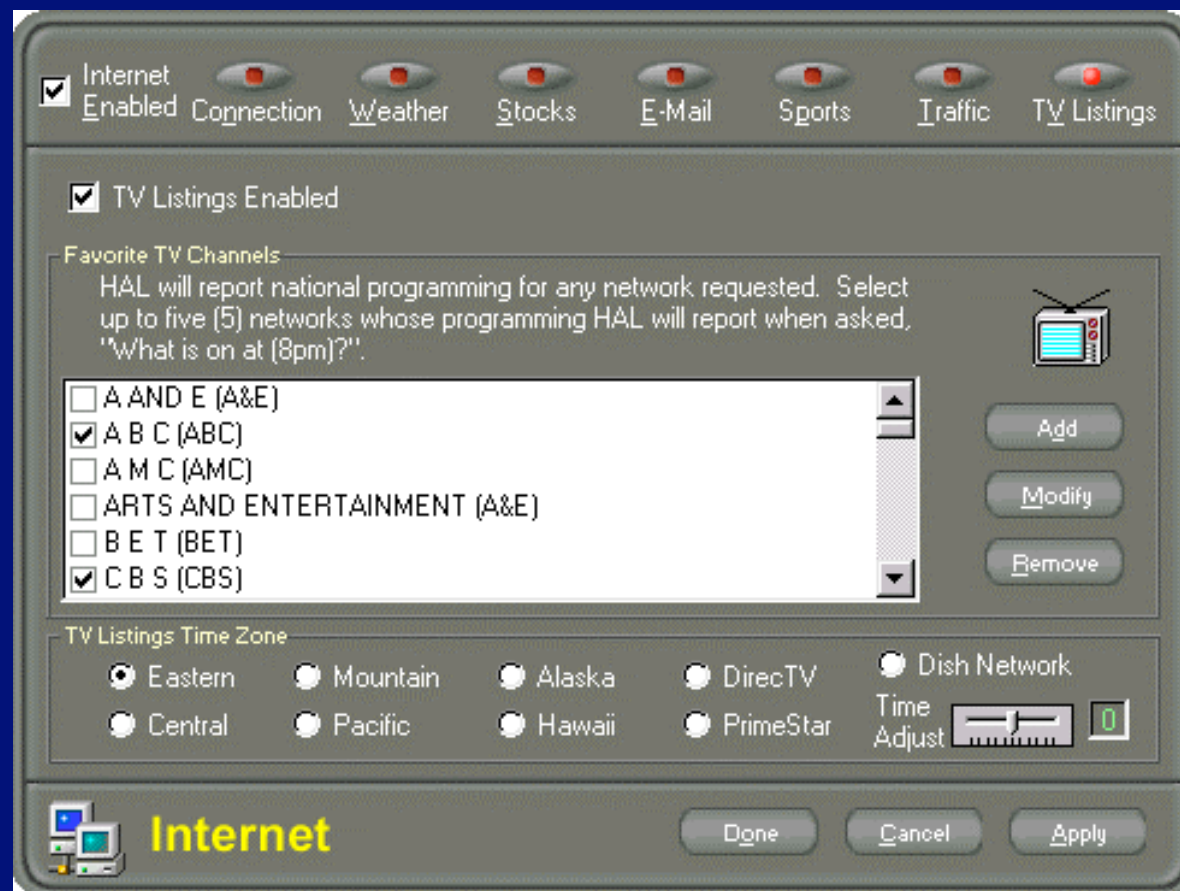
 **Device Wizard**

Special agents (weather, sports, stocks,..)

Focusing command eg 'Select TV Listings'

After that you can ask 'what's on TV tonight' etc.

Agents have config screen

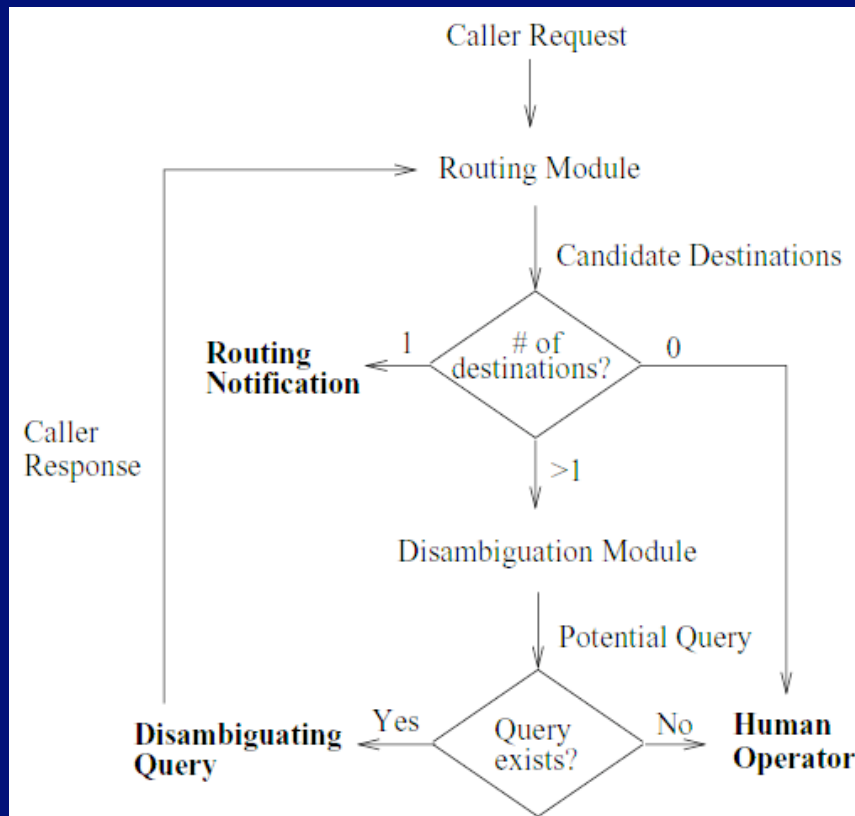


# 'Complex' scripts with command and precondition



# Call Routing

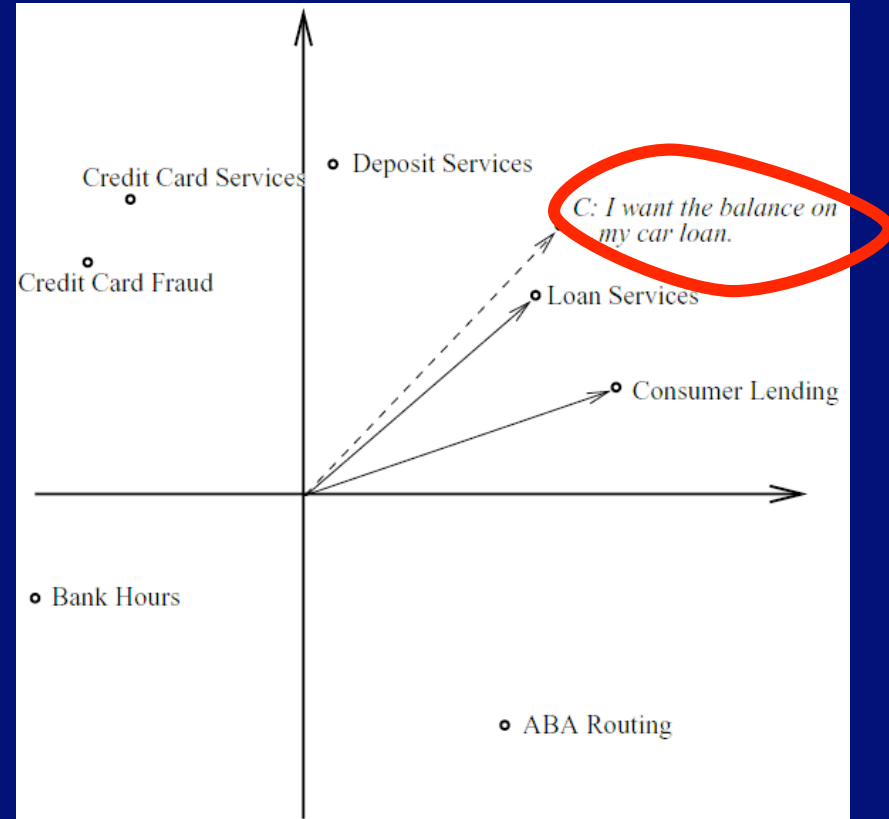
Setting: large company (eg bank) wants to forward incoming calls automatically to right department



(Chu-Caroll '99)



- 1 map incoming call to a vector
- 2 Determine closest existing service vector
- 3 Make disambiguation query if more than 1 vector close



Implementation: Corpus + statistics

Corpus: 4500 calls to 23 departments of bank

Found 757 n-grams used in first utterance.

Create ~20dim vector spaces spanning the services

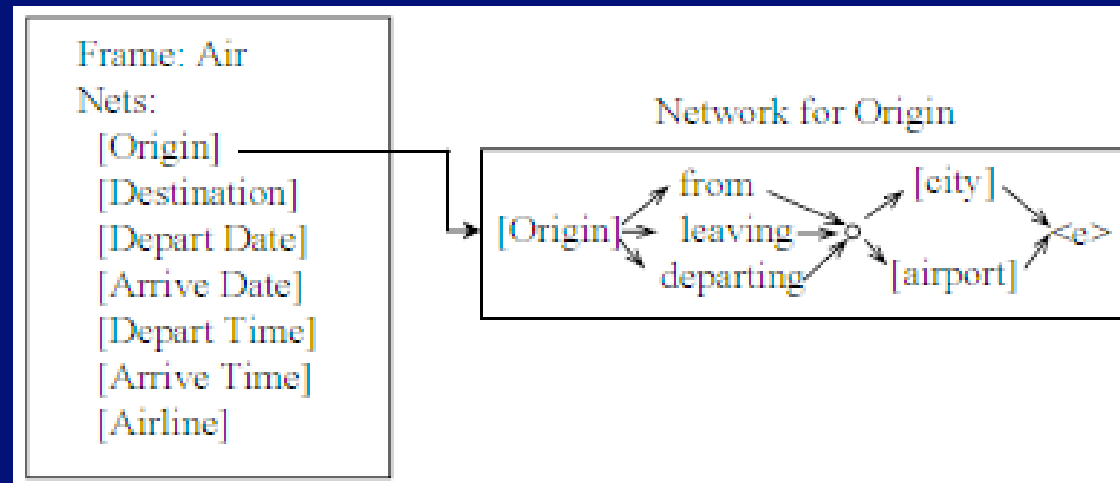
Sing Val Decomposition of ngram+service matrix needed.

# Phoenix Parser

Frame-based  
Parsing:

Frames with  
Slots

Skip un-recognised  
inputs



Input: Show fares of flights from Denver to Boston on United

Parse:

Frame: Air  
  
[Field]( show [\_fares](fares of flights))  
[Origin]( from [City]( Denver ))  
[Destination]( to [City]( Boston ))  
[airline]( on [AirlineName]( United ))

Extract:

Frame: Air  
  
Field: fares  
Origin.City: Denver  
Destination.City: Boston  
AirlineName: United

Early approach (Constantinides):

All parses are scored

Maximize number of covered words

Minimize number of fragments

Maximize number of filled slots in a frame

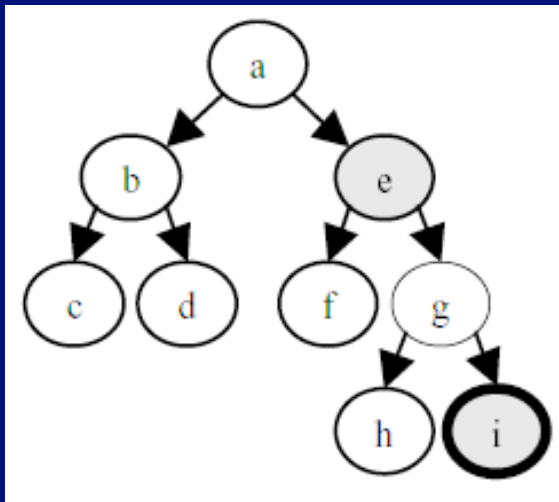
-> Not suited for many different services

Later improvements (Rudnicky):

Slots are pre-sorted to their 'power to bring solution'

Task tree to guide the dialog,

For instance trip=fly+hotel+car. top node (a) tries to gather all data, (e) is currently active getting a plane and (i) is trying to find out the departure date.

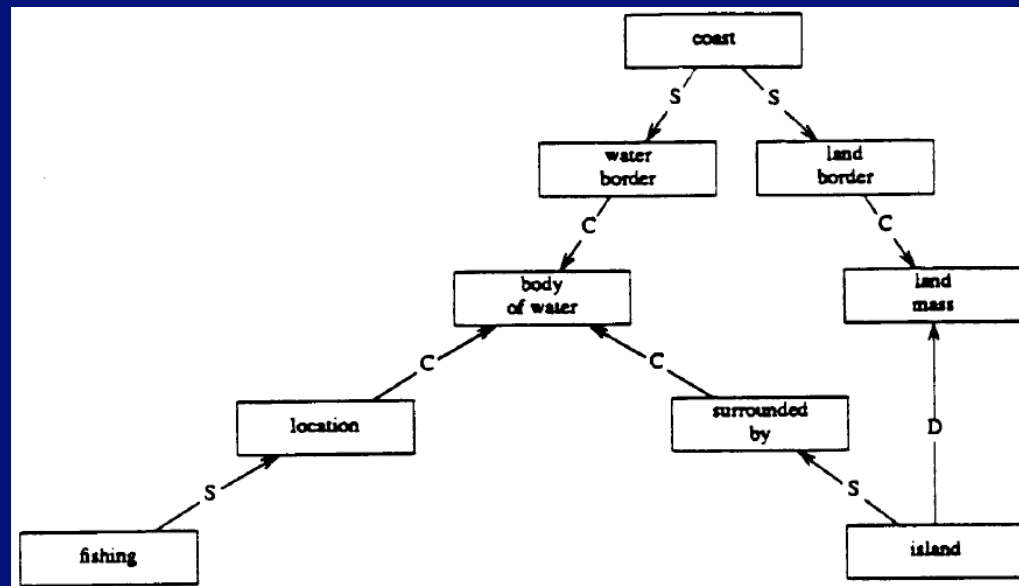


# Story Understanding

Several approaches.

Marker passing approach inspiring for Cactus system

Example (Norvig). User talks about fishing and island. Body of water is common parent --> notion of location, surrounded by and body of water are 'inferred'.



Various types of links are used to derive the right relations. Eg constraints, subclass-of, slot-of, view (alternative interpretation), quantifier.

Larger nets allow more extensive use of context.

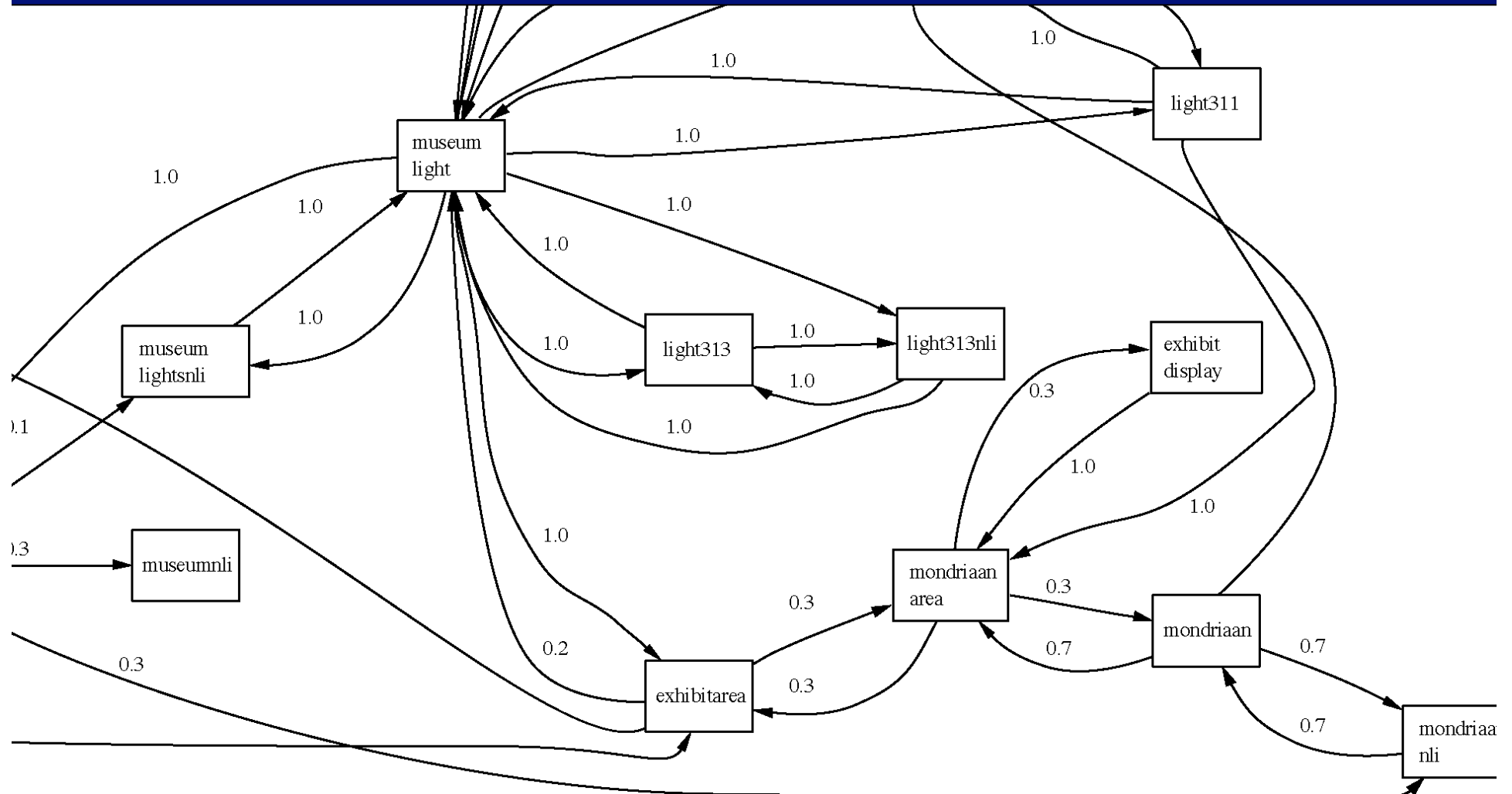
However large nets cause huge number of such shared parents

One approach: only certain type of link combinations are relevant.

Later research indicates that statistics (by adding probabilities to edges) can greatly help searching for relevant relations

# CACTUS. Organizing ad-hoc agents

Every agent knows its place: task- location- and user- relations, and the relevance of the relation



# Some details

Agents change the context by changing the relations in the agent world.

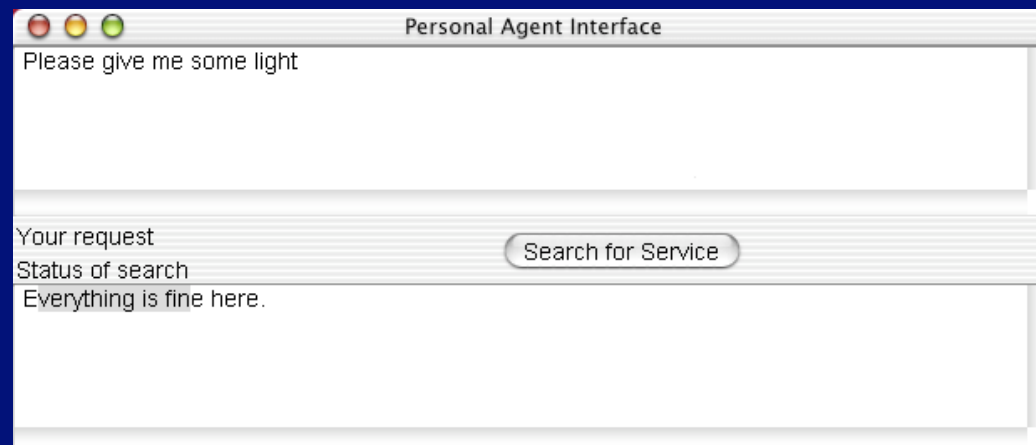
Every agent also knows how the area in the physical world it 'represents'



# Service Matching

**Service Matching:** search agent matching user's need

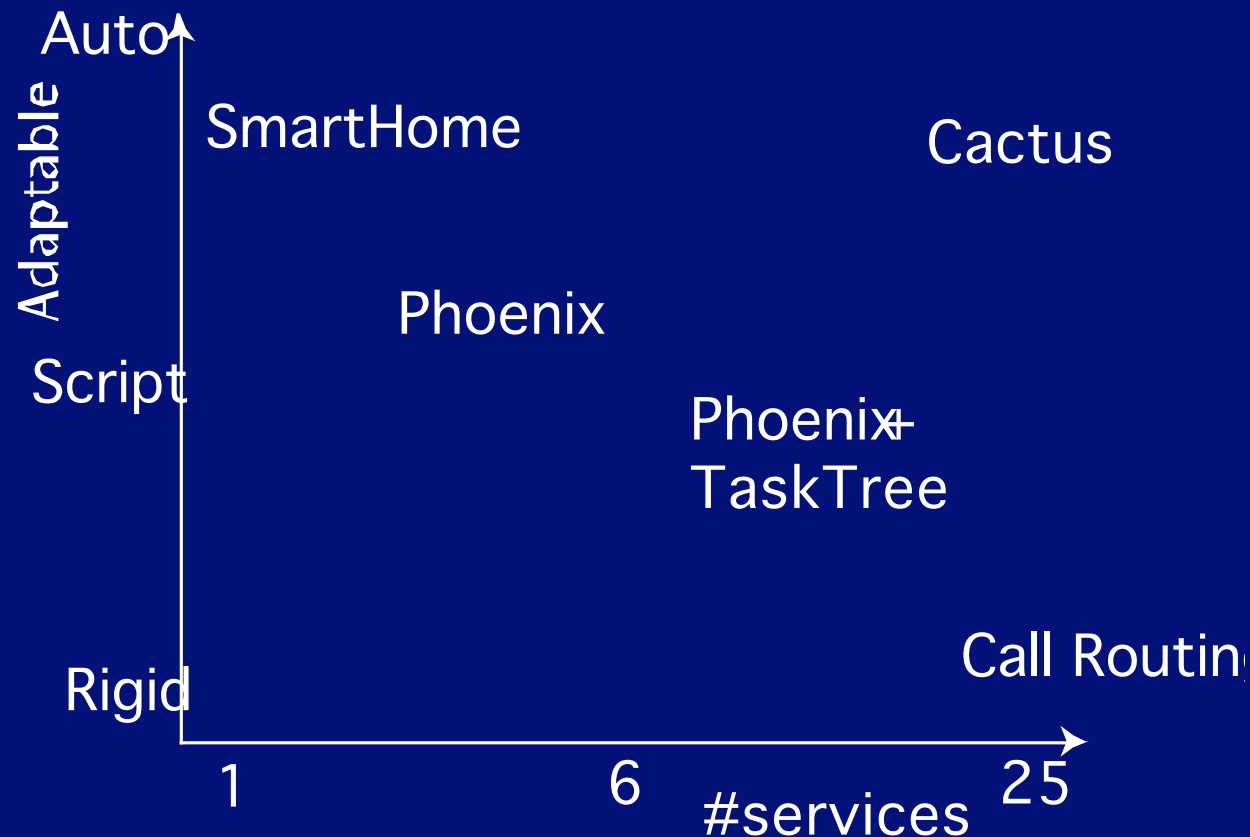
- Some agents in the agent world understand NL
- Context-dependent search using agent's context info
- Using Natural Language request from user
- Searching agent space for agent understanding request
- gradual extension of search space
- User's request is also used to start up the service (no repeat of the request necessary)



# Conclusion

We are working on and improving

- distributed natural language understanding
- highly adaptable systems with large number of services



# Literature

Smarthome (2003). Welcome to SmartHome. [smarthome.com](http://smarthome.com).

Carpenter, B., & Chu-Carroll, J. (1999). Vector-Based Natural Language Call Routing. *Journal of Computational Linguistics*, 25(30), 361-388.

Norvig, P. (1987). A unified theory of inference for text understanding. Ph.D. dissertation, Department of Computer Science, University of California at Berkeley.

CSLR (1991). The Phoenix Parser User Manual.

Constantinides, P. C. (1999). Scoring techniques for Phoenix Parses. Available as part of the Phoenix documentation.

Rudnicky, A., & Xu, W. (1999). An agenda-based dialog management architecture for spoken language systems. *IEEE Automatic Speech Recognition and Understanding Workshop*, I-337.