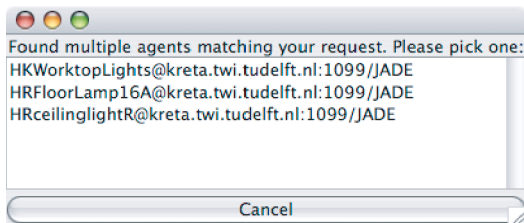
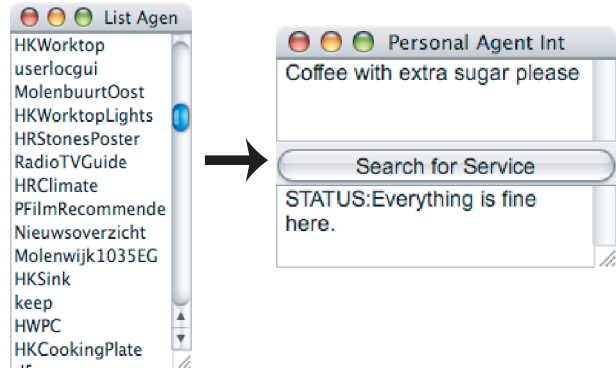


User-Service Matching in Smart Ad-hoc Environments

W.Pasman, w.pasman@ewi.tudelft.nl
Cactus Project, Faculty of EWI, Delft University of Technology, The Netherlands.

Problem: Finding the Appropriate Service(s)

In future intelligent environments, the user will have difficulties to find the service/agent that he needs. Selecting an agent from a menu does not scale for larger numbers of agents. Also, in ad hoc situations the list may change all the time. Our goal is to make such large ad-hoc agent systems manageable for the user. The proposed service matching agent allows the user to express his needs in natural language, and his request is then matched to the agent that thinks it is best able to handle that request.



User negotiation after user asked 'light on please', as it is unclear which light he means.

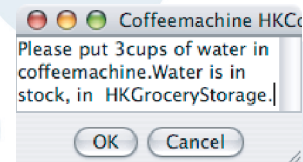
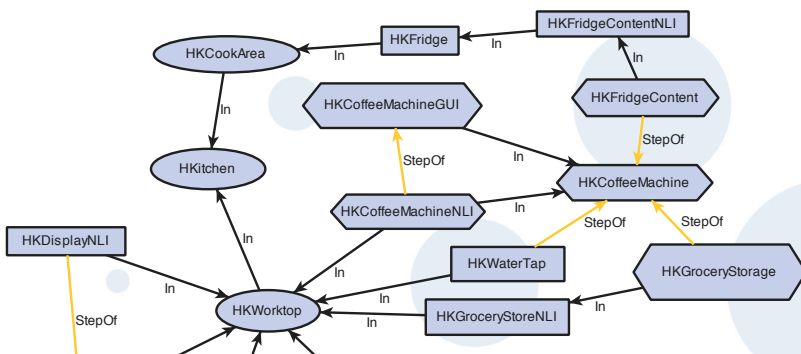
Avoiding obtrusive agents

To avoid a waterfall of services after a request, the service matcher negotiates with agents (and user if necessary) about potential interpretations of the request. Agents initiate action towards the user only after authorization by the service matcher to do so.

Ad Hoc Robust Solutions

All agents know how they are related with other agents: their task relations, location relations and relations with the user. This enables context dependent searches through agent space, avoiding broadcasting user requests to all agents. These relations are also used for other tasks such as autoconfiguration and user-following GUIs.

Agents are aware of ad-hoc aspects and flexible. They negotiate to see if a good offer can be made to the user, and try to get help from the user if appropriate. Alternative services can be looked for in a location aware way, if the standard service fails.



Small part of agent relations in the demo system around the coffee machine. The NLI agent attempts to translate user requests into agent messages. The GUI agent interfaces with the user to get additional information.

Coffeemachine attempting to get refilled. Note that it checked if and where water is in stock.