WOA 2004

A Personal Agent Supporting Ubiquitous Interaction

Giovanni Cozzolongo, Berardina De Carolis,
Sebastiano Pizzutilo
{cozzolongo, decarolis, pizzutilo}@di.uniba.it

Intelligent Interfaces –

Dipartimento di Informatica Bari (DIB) - Italy

Scenario:

"After breakfast Wendy organizes her tasks in the "to do list", setting when and where executing them. (i.e. *buy food before coming back home*)" Her tasks will be managed by her **personal agent** that will remind her about things to do. Beside activating this reminding function, Wendy can delegate task execution to agent.

To be effective and unobtrusive, task remind or execution should be triggered according to context and user preferences.

After working the user goes home driving her car. Since in the todolist there is the task "buy food", it will be reminded using the display of the car. When he comes near the supermarket, the agent advertises the user in a more evident way, using messages and sounds.

Wendy can choose to listen to the agent and buy food or to ignoring it. In this case the system, on the basis of the assigned autonomy level, will suggest the chance of buying food on line. It will ask the user for confirmation, and will execute the task accordingly.

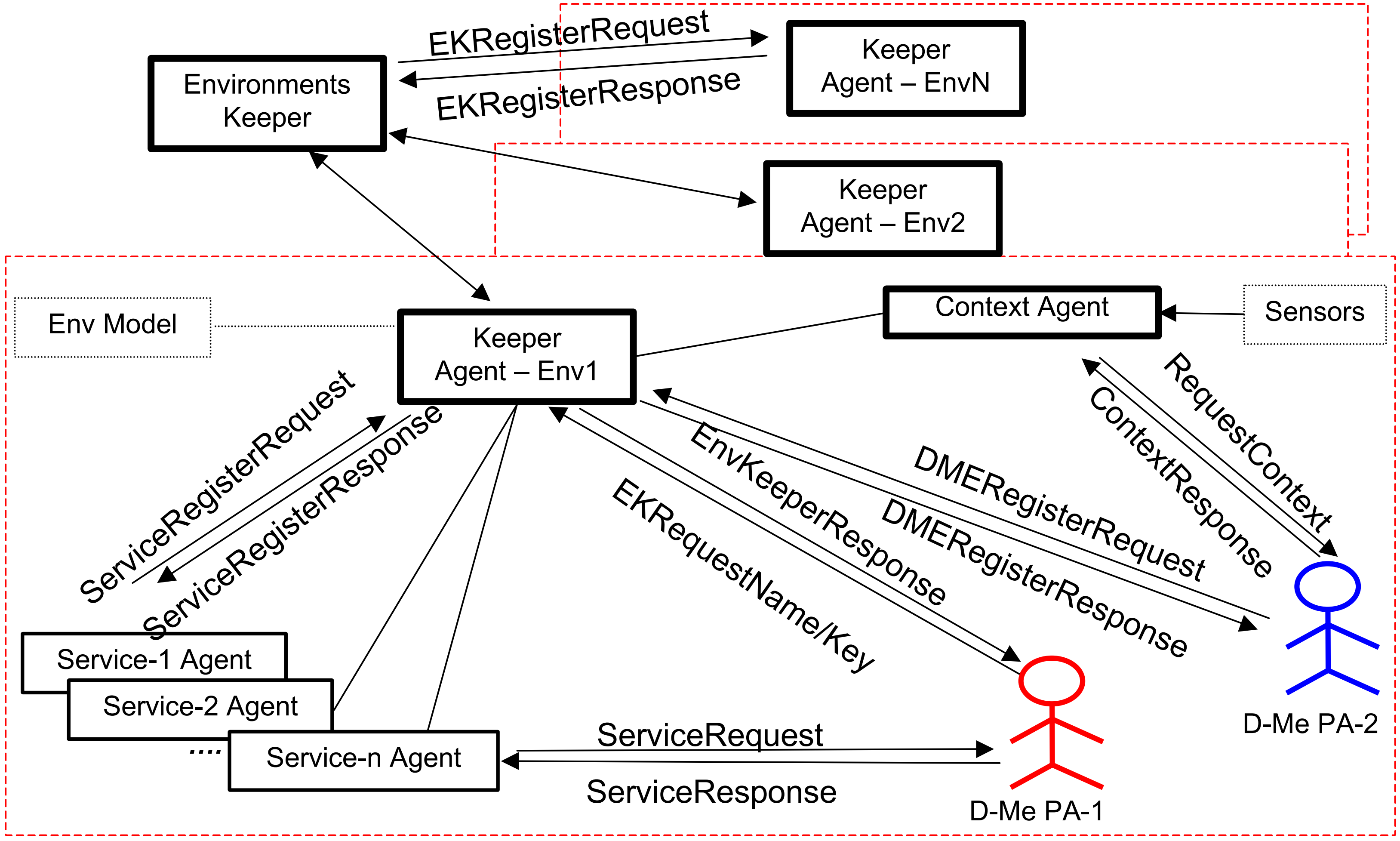
Requirements

To realize this kind of scenario we need:

- ***Communication protocols***
- ***Service discovery***
- ***Context awareness***
- ***User adaptivity***
- ***Shared ontologies and semantics***

D-Me (Digital - Me) System

Architecture of the System ...



… Architecture of the System …

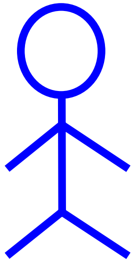
- **Keeper agent**:
  - coordinates agents.
  - *white and yellow pages*, registering agents to the system using the appropriate protocol
  - every environment has a local *Keeper Agent,* for local agents
  - the *Environment Keeper* coordinates inter-environment communication.
  - every agent should register with his Keeper Agent
  - every Keeper Agent should register with an Environment Keeper
  - service discovery is used to join unknown Environments
  - services can be searched *by name* or *by keyword*

… Architecture of the System

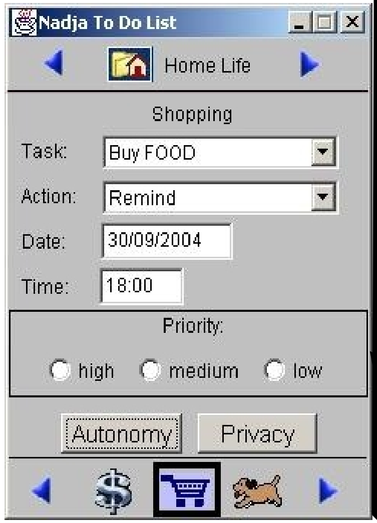
- **Service Agent:**
  - agent that provides a service.
- **Context agent:**
  - provides information about the context
- **D-Me Personal Agent:**
  - represent the user in the environment
  - interact with other personal/service agents.

Agents communicate using *request/response* protocols, specified as a sequence of ACL messages;

D-Me Personal Agent ...



User



Nadja To Do List

Home Life

Shopping

Task: Buy FOOD

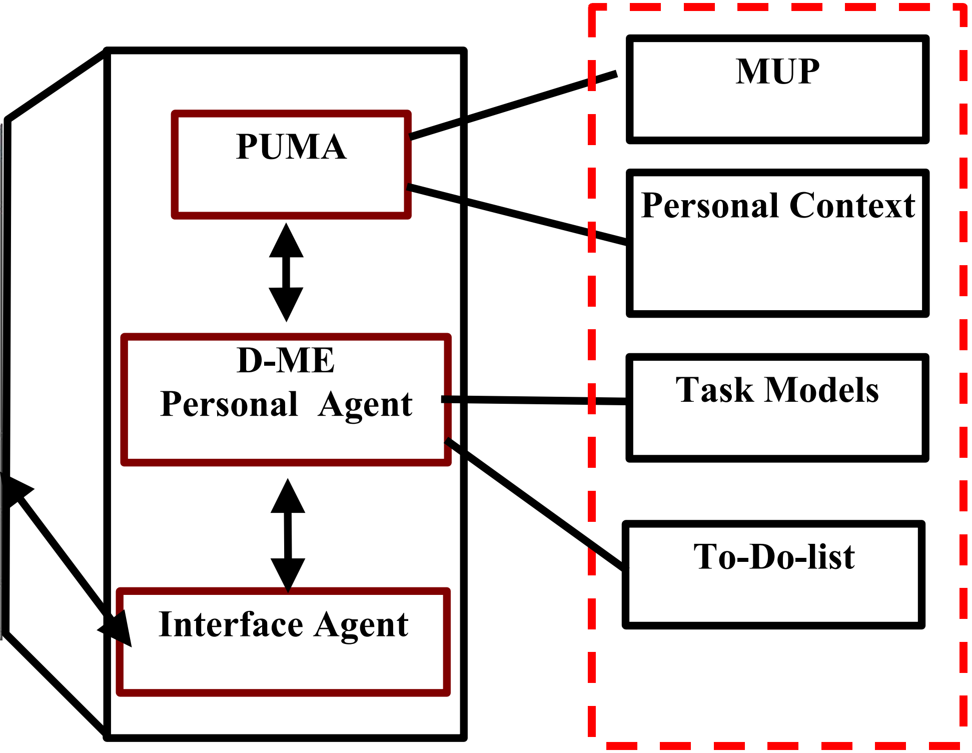
Action: Remind

Date: 30/09/2004

Time: 18:00

Priority:

○ high ○ medium ○ low



TO DO LIST D-Me User Side KBs

…D-Me Personal Agent…

The PA is modelled as a BDI agent

cyclic behaviour that checks whether some intention (task to execute) can be executed, given its beliefs (user and context profile) and desire (goal)

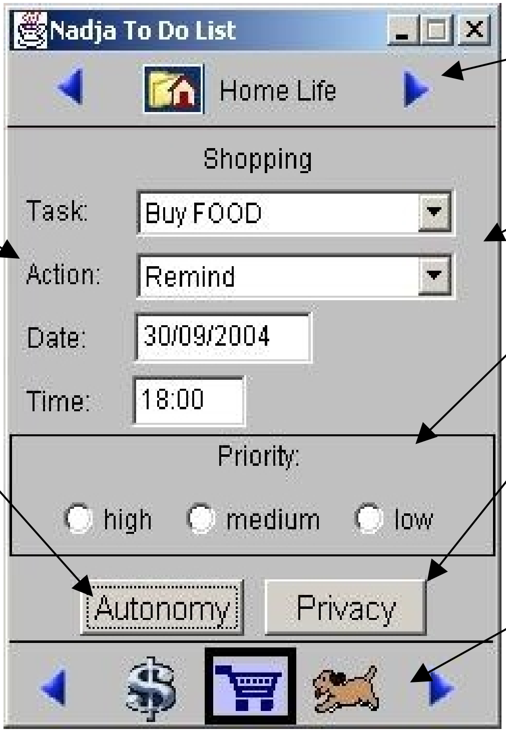
Desires:

- execute totally or in part tasks
- create new tasks if required
- get user related information relevant for adapting task execution
- get context related information relevant for contextualizing task execution
- Personalized communication

In presence of smart environment it activates the execution of user requested tasks passing relevant information for personalizing the results

To-Do-List

the **To-Do-List** interface allows users to insert his task, as an agenda



Task: Buy FOOD

Action: Remind

Date: 30/09/2004

Time: 18:00

Priority: ○ high ○ medium ○ low

- an action to perform
- the autonomy level

```
<Autonomy
slotName="autonomy"
execution="high"
communication="high"
personalData="middle"
resourcesExploitation="low" />
```

- the scope (Homelife, Business, Travel)
- Task name, date, time
- the priority
- the privacy settings: allow/deny the provision of personal information
- the environment in which executing the task

```
<Task slotName="taskDefinition" id="1" name="buy" key="food action="remind"
"date=3009041800" belongingScope="homelife" environment="all" p-env="supermarket"
priority="high" what="food list" when="before” whenevent=”going back home"
remindBefore="1800"/>
```

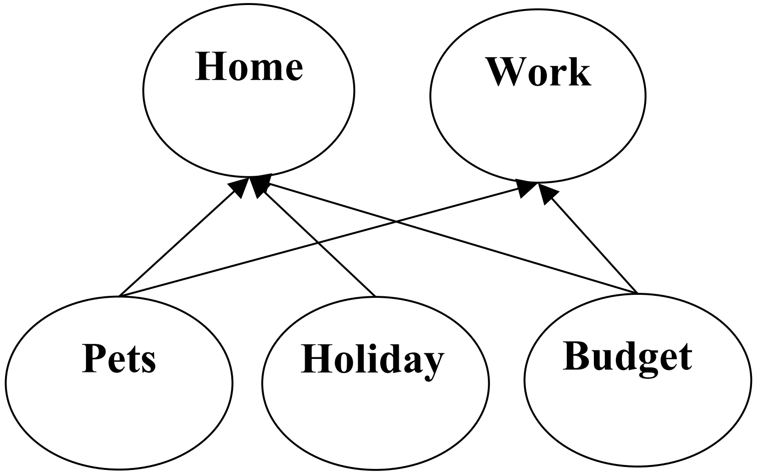
Autonomy Level

- Execution autonomy
  - booking on line
- Communication autonomy
  - sending emails
- Personal data diffusion Autonomy
  - name, address, credit card number, phone number ...
- Resource autonomy
  - battery, display resolution,money,time,ecc...

Personal User Modeling Agent (PUMA)

Mobile approach to user modeling:

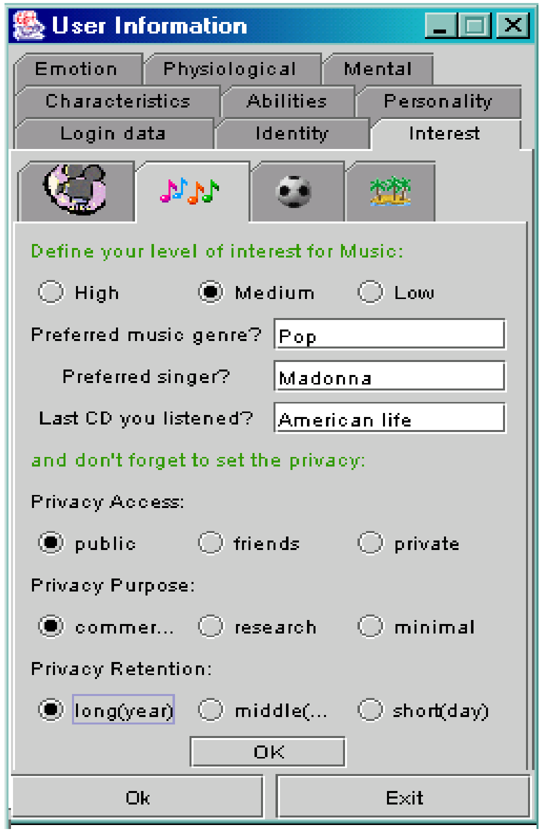
- ***User brings his/her profile on a personal device***
- ***Hierarchical user model***
- ***Nucleus-Satellite relationship between users main-personal device (PC - PDA).***
- ***PUMA manages the relation between user' devices.***
- ***PUMA stores in the device the profile relevant to the situation***
- ***Situational Statements (UbisWorld)***



PUMA – Situational Statements

- Managing user data linked to situational information
- Supporting privacy

```
<SituationalStatement version="Full_0.1">
        <content>
                <subject><UbisWorld:Nadja /></subject>
                <predicate><UserOL:buying cookies /></predicate>
                <predicate-range><UserOL:normal,specialoffer,3x2/>
                </predicate-range>
                <object>special offer <object>
        </content>
        <restriction>< location>supermarket<location></restriction>
        <meta>
                <owner><UbisWorld:Nadja /></owner>
                <privacy><UbisWorld:friends /></privacy>
                <purpose><UbisWorld:commercial /></purpose>
                <retention><UbisWorld:short /></retention>
                <viewer><UbisWorld:X-Supermarket /></viewer>
                <evidence>not-specified</evidence>
                <confidence>high</confidence>
        </meta>
</SituationalStatement>
```



Define your level of interest for Music:

○ High ● Medium ○ Low

Preferred music genre? Pop

Preferred singer? Madonna

Last CD you listened? American life

and don't forget to set the privacy:

Privacy Access:

● public ○ friends ○ private

Privacy Purpose:

● commer... ○ research ○ minimal

Privacy Retention:

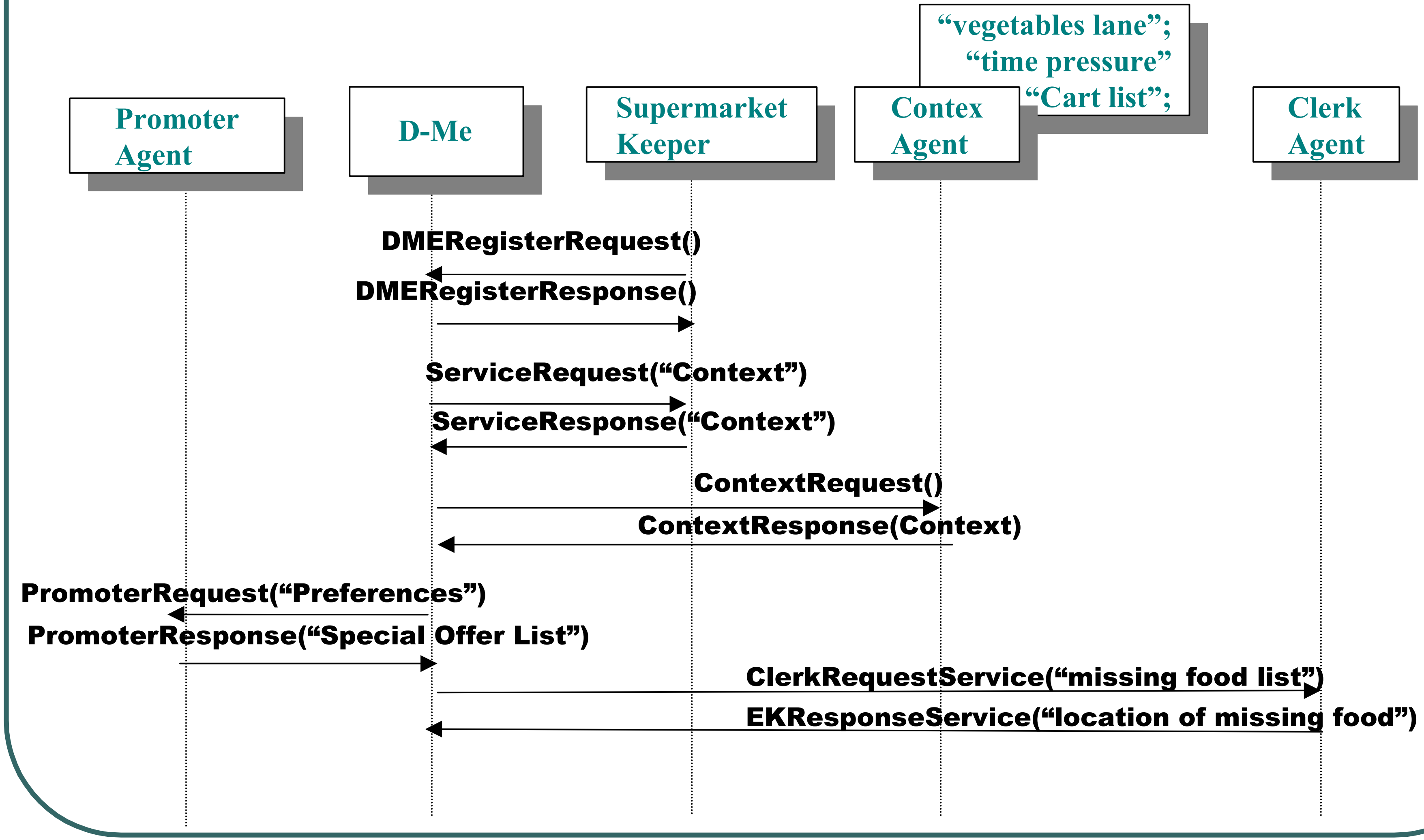
● long(year) ○ middle(... ○ short(day)

Context Information

- Environment
  - static features (scope, physical feature, type of environment, ecc..)
  - dynamic features (es: noise level, light, tagged objects...)
- User
  - dynamic features (emotional state, location, time pressure, ecc...)
  - device employed (type of display, capacity of displaying images...)

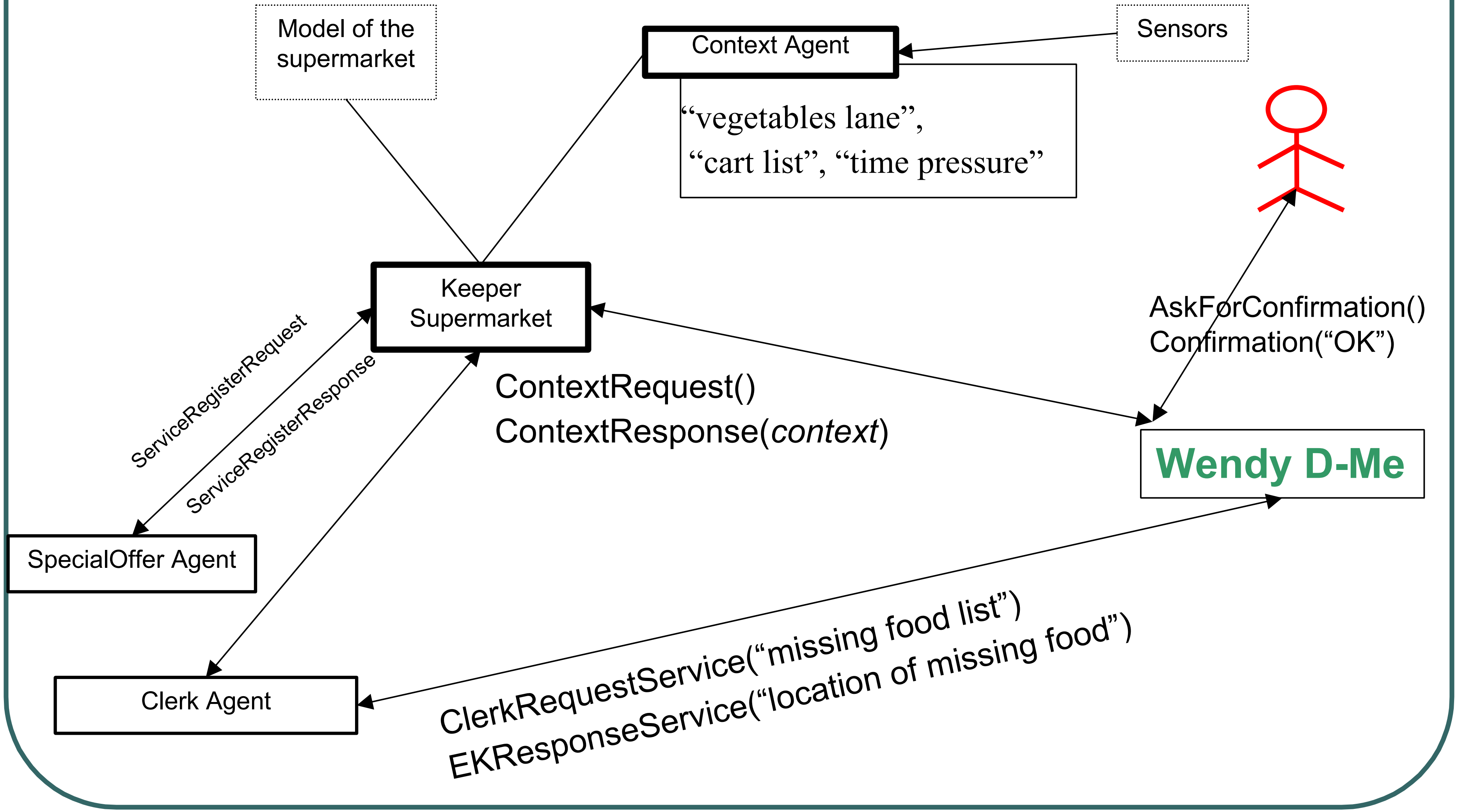
The context information are collected and managed by the context agent. In our prototype sensors are simulate.

Example



"vegetables lane";
"time pressure"
"Cart list";
Promoter Agent
D-Me
Supermarket Keeper
Contex Agent
Clerk Agent
DMERegisterRequest()
DMERegisterResponse()
ServiceRequest("Context")
ServiceResponse("Context")
ContextRequest()
ContextResponse(Context)
PromoterRequest("Preferences")
PromoterResponse("Special Offer List")
ClerkRequestService("missing food list")
EKResponseService("location of missing food")

Example



Model of the supermarket

Context Agent

Sensors

"vegetables lane",
"cart list", "time pressure"

Keeper Supermarket

ServiceRegisterRequest

ServiceRegisterResponse

ContextRequest()
ContextResponse(*context*)

AskForConfirmation()
Confirmation("OK")

Wendy D-Me

SpecialOffer Agent

Clerk Agent

ClerkRequestService("missing food list")
EKResponseService("location of missing food")

Discussion – Open Issues

- Evaluation
- Service discovery
- Shared Ontology
- Learning
- Group Modeling
- …