SAT – Final Article: Tackling Smart Home User Requirements with Agent Technology

Lauri Kainulainen

University of Tampere
Department of Computer Sciences
Computer Science / Interactive Technology
SAT – Task 3
March 2006

University of Tampere
Department of Computer Sciences
Computer Science / Interactive Technology
Lauri Kainulainen: SAT – Final Article:
Tackling Smart Home User Requirements with Agent Technology

Contents

1. Introduction

Home automation research aims at bringing new smart services into the modern home. Smart home research is by all means not a new subject. Numerous projects like Pluto, MySmartHome and Home Automation Inc. have existed for quite a while. One of the intelligent home suppliers – “Smarthome” – has been operating through it's online-shop since 1995. There has also been quite a lot of academic research done on home automation.

Regarding computer science, smart homes provide an interesting research field as they contain many popular domains of modern research. These domains include such topics as “*tangible interfaces”, “ubiquitous computing”, “ambient intelligence”* and *“speech interfaces”*.

Despite all the efforts wide spread success of intelligent homes is yet to come. One of the major problems is that the home automation solutions are mostly based on a technological “push” rather than a consumer “pull” [Mäyrä et al., 2005]. Although the technology is almost at our grasp, successful deployment of home automation requires research on other areas – such as *sociology*, *industrial design* and *usability* – in addition to the directly computer science related fields like *proactive computing* and *multimodal interfaces*.

The problem with the technological push can also be seen in many parts of modern research. Multiple papers provide innovative ways to bring proactive computers into our homes, but very few try to find out if they are actually welcome. Although user requirements for smart homes have been researched (see [Röcker, 2004] and [Mäyrä et al., 2005] for example), the tests seem to be separated from actual implementations or designs. In addition to this home environments are especially hard to analyse and comprehend. Studies have shown that homes are not thought of as simply places of dwelling, but instead they're seen more as a 'state-of-mind' [Mäyrä et al., 2005].

In this paper I propose a multi-agent architecture that responds to the elicited user requirements and provides a system that can live up to the challenges put forth by the home environment. I will rely heavily on previous research on the subject and try to combine expressed ideas and learnt lessons. The system architecture will be described on a high level and will only provide an abstraction. No implementation work, nor proof-of-concept has been done. The main aim is to design a non-intrusive system that could address the needs of the users without major drawbacks.

2. Why Agents?

Many of the challenges put forth by the home environment suggest that software agents might be the correct approach. Firstly homes are unique and tend to change over time as new household items are bought, old ones are sold or the house is renovated. This brings a requirement for modularity and flexibility on the system.

The system should also be 'on' all the time. Occasional crashes are only approved if the system can recover from them independently and hopefully without the user noticing. This introduces the requirements for stability and autonomy.

A smart home system should also respond to the actions of the user and learn her daily routines. The term proactivity is often associated with intelligent environments to mean that the system is able to suggest actions or directly perform them without the user's explicit command.

An autonomous software agent is, as defined by Stan Franklin and Art Graesser [1996], "*a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.*" When we add the normal characteristic of cooperation – that is often associated with agents – to the definition, we see that the nature of software agents work well in smart home environments: autonomy, intelligence, awareness and cooperation are good qualities for an intelligent entity in a smart home. In addition to this the modularity that agent-based planning enforces makes it easier to design and describe such systems.

2.1. Related Work

Many multi-agent based research pursuits have been made into the the home automation domain. Researchers at the University of Texas in Arlington built a smart home system in their project MavHome (Managing an Intelligent Versatile Home) [Cook et al., 2003]. The system utilizes a hierarchy of agents: the top ones are in charge of the big picture, while the ones at the bottom do the real physical modifications based on what they are told. This system allows seamless connectivity among components and easier development of the underlying techniques.

Another example of agent technology use in home automation is the UMASS Intelligent Home Project [Lesser et al., 1999]. This project also uses a hierarchy of agents as the top-most abstraction of the architecture. The hierarchy is however a lot more horizontal than in MavHome: instead of a direct chain-of-command, the agents coordinate resource usage and actions together. The project gives many innovative points to resource management. I will come back to these later.

3. The Requirements

I will use the clearly defined requirements provided by the project AMIGO (Ambient Intelligence for Networked Home Environment) [Röcker et al., 2004]. The requirements were gathered with a cross-cultural study at six different sites in five different European countries. The researchers used a three stage, scenario-driven approach. During the first stage the participants gave quantitative feedback on different smart home scenarios. In the second part a story was presented to the participants, who then gave qualitative feedback on the ideas presented. The third and last stage of the requirements gathering involved free discussion over smart homes in general.

After the elicitation of the requirements, the AMIGO team created six different design guidelines and prioritized them according to how important they were seen by the participants. In the following chapters I will describe these six guidelines and present related research that confirms them.

The AMIGO project has already provided a design based on their research [Georgantas, 2005]. This design is done with a service-oriented methodology, is fairly complex and goes into close detail. The project aims to bring four different domains together: consumer electronics, mobile computing, personal computers and home automation. This brings a huge overhead to design as the team has to please the players on all the four teams. The AMIGO project is mainly geared towards creating a networked home in which different entities from all the four domains can work together. This paper however tries to find a complete system that addresses user requirements, not to create a network standard that could connect the appliances of tomorrow. While AMIGO is obviously a huge project designed for real life, this paper is just a pure play with different ideas.

3.1. First Priority: Control, Safety and Non-intrusion

The first category of elicited requirements were seen as the most important ones and will always surpass the others. In this category, five different requirements were discovered. As the first requirement the participants wanted to always remain in total control of the system. This requirement has also been discovered in research made by Battarbee and Kuusela [2005] from the Hypermedia laboratory of Tampere University, where it was expressed that people, not computers, should be in control of their lives.

Other requirements included non-functional concepts such as safety, security and privacy. The smart home system should also provide a clear added value in comparison to the normal home, should not replace direct communication between people and should maintain the comfort of home.

3.2. Second Priority: Help with the Information Burden

A tool to help with the overload of information was the second most favoured requirement category. People follow several different information sources daily from newspapers to RSS-feeds. In addition to this many types of information is needed at arbitrary times, such as recipes when deciding what to cook, or TV-timetables when deciding what to watch. The smart home system should provide this information depending on the user and the conditions. The need for strong context awareness is evident in the background.

3.3. Third Priority: Reduce Housekeeping Chores and Prevent Accidents

In addition to helping occupants cope with their household chores, the AMIGO participants expressed that the system should also be cost and energy saving, and it should integrate and combine the functionality of appliances.

Intelligent cleaner robots, like the iRobot Scooba (http://www.irobot.com/sp.cfm?pageid=128) and Karcher RC3000 RoboCleaner (http://www.best-vacuum.com/karcher-robocleaner.html) are already available on the consumer market. But putting these robots to work independently, does not meet the given requirement about integration and cost/energy saving. Resource management is also needed, because many household chores require the same things like electricity and hot water. A dishwasher and a washing machine running at the same time could otherwise lead to a cold shower on the occupant's part.

In addition to resource management, temporal planning is also required – it could prove chaotic to have both the vacuum cleaner robot and the scrubber robot (like Scooba) work at the same time in the same room. Temporal planning also implies the understanding of different time contexts – it's better to have the cleaner robot doing its chores when everyone has gone to work, instead of cleaning and rumbling around during night-time.

3.4. Fourth Priority: Follow-me Content and Safety

AMIGO participants valued assistance in organizing their lives both in the house and between home and work. This introduced the need for “follow-me” content and the need for pervasive, always available information. Inside the house this could for example mean the possibility of favourite songs and music following the user as she walks from one room to the other. Between the home and workplace this could mean having the possibility of exporting the play lists on a mobile device, which could then be taken with you.

Follow-me content was discovered to be one of the most common characteristics associated with smart homes [Meyer and Rakotonirainy, 2003]. The indoor follow-me concept can also mean a certain way of context awareness on the system's part. For example if someone wants to engage in a video conference with Bob, only the screen in the room in which Bob resides is activated for alarming him. Another example of follow-me content is the scenario expressed by Meyer and Rakotonirainy in which electronic picture frames change their content based on who is in the room currently.

This category also repeated the non-functional requirements about safety and privacy. The users wanted follow-me content, but they didn't want that content to be followed by someone else. Also the fear of hostile, outside hackers or viruses taking control of their homes was expressed by requiring a good level of security.

3.5. Fifth Priority: Context-aware Assistance with Home-organization

This category is more related to the general ease of use than the others. Requirements such as user recognition when she stands in front of the entrance and the automatic adjustment of lights and window shades according to the context were expressed. These requirements again cry for context awareness on the part of the smart home system. Proactive actions and understanding of the usage patterns are required to fulfil these needs.

3.6. Sixth Priority: Communication and Privacy

The last category is about staying in touch with your friends and family. To see each other while communicating was the preferred way. But this category also implied again the need for privacy – communication should be made easier, but only when the user so wishes.

4. The System

Although no details of a possible hardware implementation is presented, the basic idea of the system is to have a main frame computer that contains the critical parts of the system. The main frame is stored in an isolated and secure place, and acts as a hub for communications, meaning that it connects to the internet, telephone lines and TV antennas. As described by Spinellis [2003] this setup brings three advantages: when all the outside connections are terminated to the same destination wiring becomes easier, the generated noise by the system won't trouble anyone and the system can be physically secured from burglars and other evildoers.

In addition to these advantages, having a main frame that acts like a hub for communication between the devices in the house, and between the house and the outside world enables the use of features that are common, but rarely seen in home environments, such as in-house printer sharing, shared internet firewall protection and wireless access, and backup storage for personal files. An UPS-unit may also be used to ensure that the system survives a power shortage.

Critical appliances such as burglar alarms and smoke detectors should however be capable of working on their own without help from the main frame. Optional communication with the smart home system could however provide additional value. For example the smoke detector could inform the user through the mobile phone that it has called the fire department while the user was away.

4.1. The Basics: Hardware Abstraction and Context Awareness

In their paper on context aware homes Meyer and Rakotonirainy [2003] propose three critical components for a smart home system. The first component is a hardware abstraction layer that abstracts the communication between the underlying devices and the software components.

The second minimum component is a context manager which gets data from the hardware abstraction layer and organizes that data into context information that can be easily used by other components that require context awareness.

The last component Meyer and Rakotonirainy claim to be needed is a privacy manager that makes sure that only the minimum amount of needed information leaves the privacy domain.

The two first components seem like a good foundation for a multi-agent system: hardware abstraction eases the addition and control of different devices and is a basic part of any modern operating system, and the context manager eases the programming of the other software components and makes them more suitable for a general purpose use. The need for a separate privacy manager however seems a bit redundant: clear permissions on what a component or a user is allowed to access can be realized on a per-component-basis, with a coordinating agent or simply with user-group-permissions, like the ones in UNIX-like operating systems.

The diagram in figure 1 depicts the flow of context information on a high level of the system. The hardware abstraction layer provides raw sensor data for the context manager which in turn notifies the software agents of the changes that they are interested in. For example an agent responsible for the front door might be interested if the context manager gets information about the motion sensor spotting something there. After getting the information the front door agent could directly tap into the raw data that the front door camera is giving and process it in the preferred way. This direct feed could also be used by a user interface agent to display the video footage to the user. The communication can also go the other way. For example if the front door agent identifies the user standing in front of the door, it can tell the context manager to update it's information.

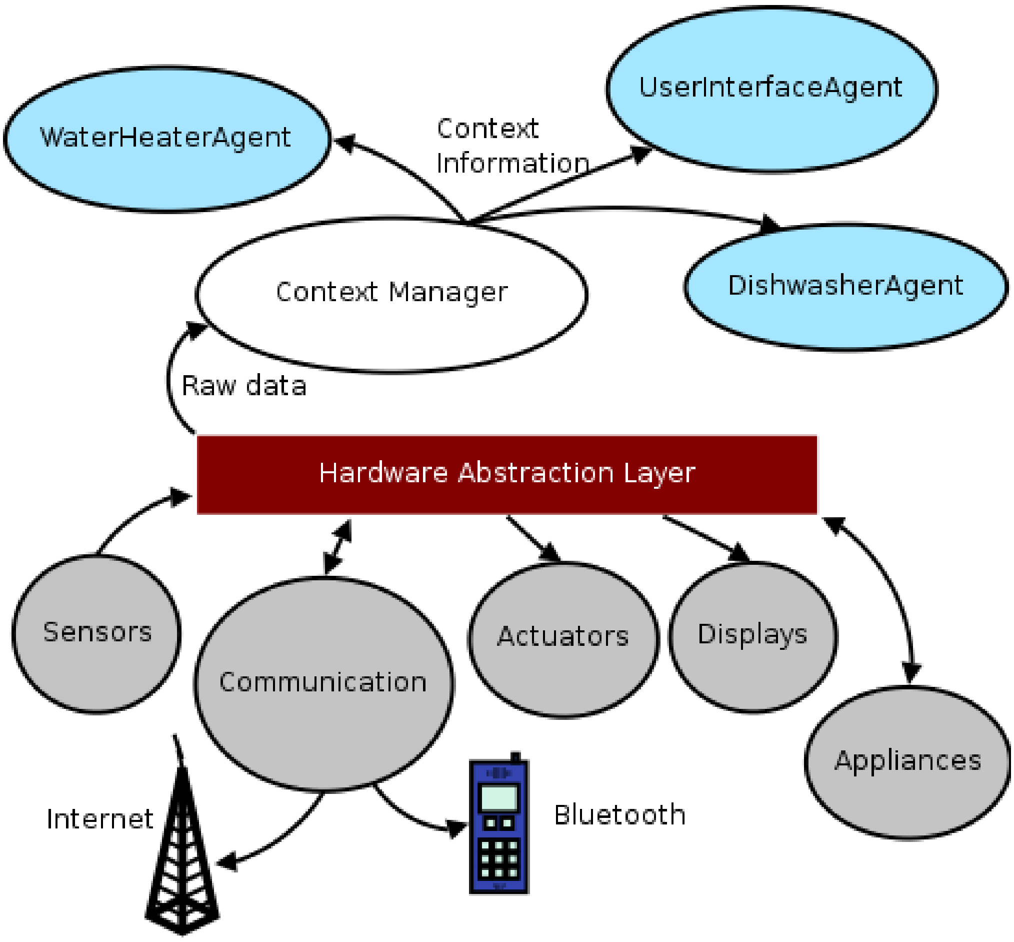

Figure 1: HAL provides the Context Manager raw data which is converted into usable context information.

The context manager should also be aware of the context of the user and her capabilities. For example if the occupant is hearing impaired, the manager should realize that no audio should be used for notification, or if the user is discussing with his friends in the living room and would not like to be disturbed, the system could use non-intrusive ways to notify him, such as dimming the lights a bit.

The requirements about indoor follow-me content can be satisfied with the context manager. This way agents in charge of different follow-me contents can simply specify a target user. The context manager will then make sure that the output is directed to a appropriate device near the user. The agents could also target certain spaces inside the house. For example during a party the occupant probably won't want the music to follow him in to the bathroom while the guests in the living room are left with silence.

The downside in this design is the complexity of the context manager. However the difficulty of context management suggests that it should be better to have one component handle context and provide easy-to-use information to the other components. The total abstraction of the context makes the development of other components a lot simpler. For example the possibility for an audio agent to target a user instead of first finding out where that user is and then following her movements, makes the implementation of the agent a lot more easier.

4.2. The Agents

The actual functionality of the system is provided by the agents. The context manager and the hardware abstraction layer provide a good foundation for the agents live in. An autonomous agent is hardly useful by itself. In order to cooperate with others the agent must have a way to find them. For this an agent directory is needed. In my design I propose that the agent directory is integrated to the context manager, because the information about available agents is context information in itself.

Now that the agents can find each other, a common way of cooperation has to be found. The resource management framework introduced in the UMASS Intelligent Home Project could provide a good start for cooperating agents. The UMASS home is controlled by a horizontally organized group of agents that are capable of communicating and agreeing on resource usage. The framework still has some rough corners, but the principle has been shown to work [Lesser et al., 1999]. For example if the dishwasher agent would like to do it's chores, it has to ask the agent in charge of hot water to provide it some. If the occupant is taking a shower at the same time, the hot water agent may decide to deny the request and then the dishwasher will try again later.

One excellent idea in the UMASS project was the use of noise as a resource. It provides a nice way to regulate the disturbance caused by all the household appliances.

Using similar resource management tactics as proposed by the UMASS project, the agents are capable of saving energy, cutting cost and acting more efficiently. Thus meeting the requirements of the third level in the AMIGO project.

4.2.1. Coordination and Control

User control was on the first level of requirements. The requirement decreases the need for strong proactivity. Instead of having a very proactive system – like MavHome [Cook et al., 2003] – that does most of the chores for the users, the weaker proactive system only suggests different tactics and acts without user intervention only when the user has approved such actions. This style of proactivity is also the preferred one found in the Morphome study [Mäyrä et al., 2005]. Weak proactivity also gives the power to the user, because she has the power to accept or deny all actions. This is possibly the most important feature, that makes the user feel in control [Battarbee and Kuusela, 2005]. The actions themselves can be continuous jobs, like heating, that won't terminate until the user explicitly requires them to, or quick actions like opening the front door lock, that are executed quickly.

Exceptions to the rule of weak proactivity are situations that can be interpreted as hazards. For example a fire detector could inform the fire department automatically when it discovers a fire, or the shower could be shut down if it is running and nobody is near the bathroom.

To ensure that the system stays in line and agents don't go around doing their chores without user intervention, one agent is needed to oversee the others. This agent is the coordination agent. All other agents require a permission from the coordination agent to do their chores. Small tasks mostly including communication between agents, such as resource coordination, can be handled without intervention from the coordination agent.

In addition to controlling the other agents, the coordination agent is also responsible for the proactivity by storing user preferences and detecting usage patterns. If the agent notices that the occupant likes to close the curtains when she starts to watch the television, it follows the principles of weak proactivity and asks the user next time she turns on the TV if she wants the curtains to be closed automatically.

As depicted in the example of figure two, the other agents ask for permission to carry on their actions from the coordination agent, which in turn asks the user using an appropriate user interface agent near the occupant. This illustrates the special status of the coordination agent in the otherwise horizontal hierarchy of agents. In the figure the washing machine agent has just finished it's job and informs the coordination agent and the context manager about the change. The coordination agent decides that the user wants to be informed about this and seeks the best interface to communicate with the user. After finding the best route to the occupant, the appropriate user interface agent is instructed to notify about the washing machine. Depending on the location of the occupant, the user interface chosen could be graphical, audio or even a mobile telephone.

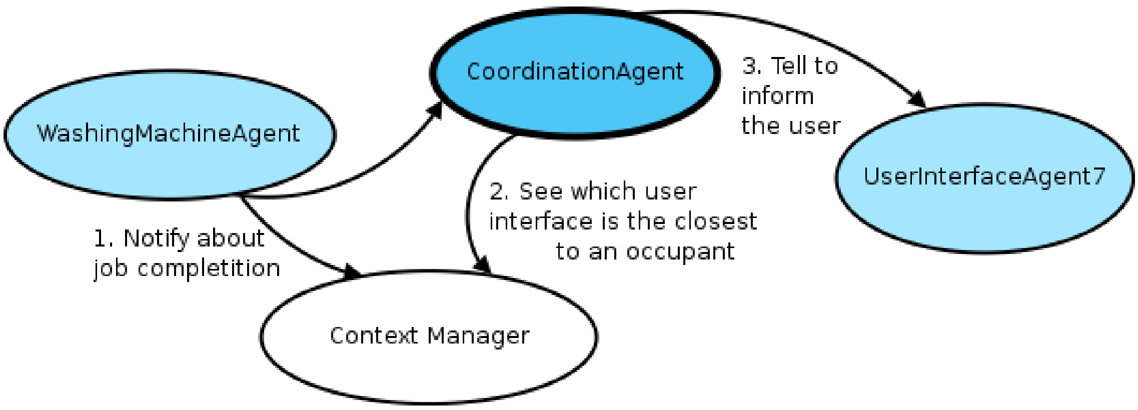

Figure 2: The coordination agent controls the action

The occupant may also automate certain actions through the coordination agent. It will then command the other agents to carry out these specified actions when the correct time comes. This way the coordination agent is capable of working autonomously and acts as a kind of a butler of the house: it commands the other servants (agents) and communicates with the master (occupant) about her plans.

5. The User Interface Agent

One smart home contains multiple user interface agents. These agents can reside on the devices they are monitoring and can be reached via a network connection. They monitor the context manager for changes that interest them and read their own sensors for input. The agent inside the touch screen in the kitchen could for example receive a notification that someone has entered the room and start listening for input on the screen and through an embedded microphone. Another UI agent could reside in the television in the living room and be operated via a remote.

Multiple user interfaces scattered around the house and designed to be used by everyone, present a challenge to privacy: using information from the context manager and it's sensors, the user interface agent may try to guess who the user communicating with it is, but the information might not be totally correct. Trying to automatically display a personal user interface to the person in front of the display might be too risky. One false guess might result in a total lack of trust in the system.

Because privacy was such a heavy requirement, the system should provide a solid and secure way to recognize someone through the user interface agents. The default way could be normal, traditional passwords, but the system could also use more advanced methods such as fingerprint recognition. A basic user interface should be available to anyone, but more critical commands require secure identification and permissions to carry them out.

Different user interface agents may have different ways to provide authentication. For example a user interface agent in a mobile phone may approve a simple PIN-based authentication, while the agent inside the occupants laptop might rely on a stronger pass phrase.

Having the coordination agent as a kind of a local gate keeper inside the house helps security, because the coordination agent can keep track of the approved user interfaces, partly preventing malicious agents from getting access to the system. All wireless sensitive information should be encrypted and a procedure has to be invented to prevent malicious third party agents from acting as a coordination agent and commanding the others. One possible way could be that agents only rely on the coordination agent that has been specified on the context manager. This way the context manager could act as a kind of a trusted third party between the agents.

An example of user interaction with the system can be seen in figure three in which the occupant contacts the smart home system with a mobile phone through GPRS. Because the connection comes from outside the house, the user interface agent in the phone requires authentication for all actions. In this scenario the occupant wishes to start the coffee machine. After the authentication, the user interface agent contacts the coordination agent that asks the coffee maker agent to start the machine. The coffee machine agent needs water and asks the water agent for the resource. The water agent informs that it can provide water and the coffee maker agent starts the process. After the coffee maker agent informs that it has started the action, feedback is given to the user.

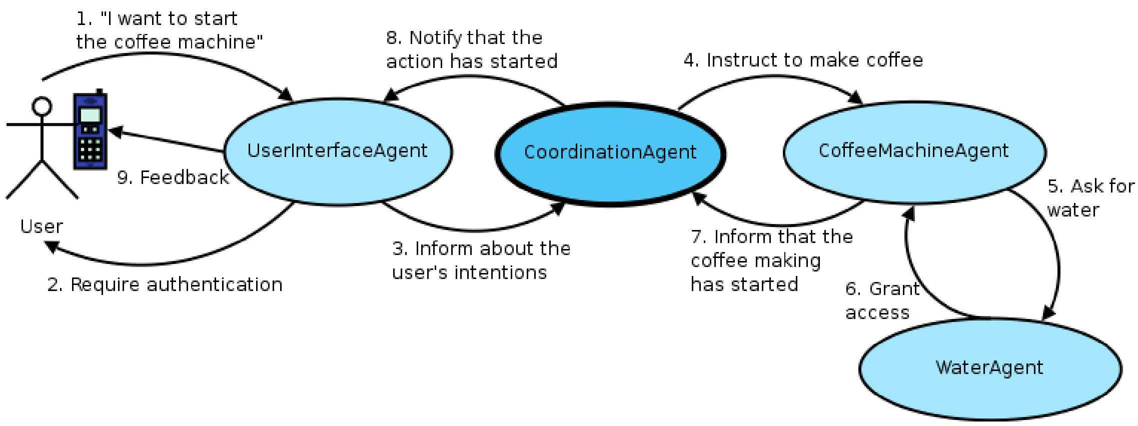

Figure 3: An example on user interaction and the interaction between agents

5.1.1. Agents to Meet the Other Requirements

Helping with the information burden of today, the smart home system could contain an information retrieval agent that collects important information relevant to the occupants. The information retrieval agent may also provide general information such as weather forecasts and traffic information to everyone and then seek more personal information such as emails, calendar events and selected RSS-feeds when the user has logged in. This of course means that the information retrieval agent has to be able to save the preferred information sources of all the users.

Fourth level requirement about follow-me content between the house and the workplace could be realized – for example – with an information retrieval agent that is capable of exporting the relevant content on separate devices or providing access to the content through the network.

Third level requirements about easing household chores can be achieved with agents that coordinate the operation of robotic devices such as cleaning robots. Accident prevention is possible by having agents monitor the potentially dangerous devices. For example the agent in charge of the kitchen oven could receive notification that the user has left the kitchen, while the oven is empty and on. If she won't be back in five minutes the agent could notify the coordination agent, which would ask the user if the oven should be shut down. Having the oven on for fifteen minutes without anything inside, could be declared as a hazard and the oven agent would then be authorized to shut down the oven automatically.

Video conferencing is already a quite common technology. This could easily used in the smart home system. The output and input could be directed to the right room with the help from the context manager. Video could be used if the user so wishes and without video the system could default to audio.

5.2. Questions to Consider

The first phase in smart home design is the selection of sensors that monitor the house. While from a technical point-of-view more sensors could look better, the occupant of the house might not agree. The high prioritized and frequently mentioned need for privacy should be taken into account already at this stage. Meyer and Rakotonirainy propose the use of smart sensors that contain embedded chips and are capable of processing their own input data. This way the amount of critical private data can be cut already at this phase [Meyer and Rakotonirainy, 2003]. While this may sound technically quite secure, I would argue that the resident of the house still feels like being watched. Probably even more so, because the camera provider advertised that the sensors were “smart”. Instead of using intrusive ways to monitor the house, the system should survive using multiple less intrusive sensors.

While video cameras have many advantages, such as motion detection, surveillance and possibility to use simple gestures for natural input, the idea of having a video camera in your bedroom might not sound like a good idea to most users. Because the home is a very private domain for it's occupants, relying too heavily on video input might create an Orwellian nightmare for the occupants. Placing one video camera outside the front door and perhaps one in the living room can however be seen as justified and provide nice features such as seeing who is at the door from anywhere in the house. Also providing physical ways such as privacy shutters to turn off the recording might help in conveying a sense of control and privacy to the occupants.

Less intrusive ways of retrieving context-information and input are available. Using simple front door keys that contain a RFID tag make it possible to personalize the keys and that way loosely identify the user entering the house. Normal passive motion detectors are an easy and cheap way to detect presence inside and outside the house. Many normal household items can also be modified to provide context-information: intelligent pot plants and multiple different items, such as the ones designed at Samsung labs [Park et al., 2003], have already been proposed.

Another problem is finding a common communication language and protocol between the agents. The UMASS project uses a communication protocol called SHARP for the resource coordination [Lesser et al., 1999]. This didn't however meet all the demands for an ideal solution and it's highly unlikely that it'll meet the needs of the architecture presented in this paper. Using a standardized agent communication language such as FIPA-ACL or KQML could provide a robust solution and ease the development of third party components, but the all-purpose nature of these languages could also bring some unnecessary overhead to the system. Because of the multiple different communication protocols, such as bluetooth, TCP and GPRS, present in a smart home, the agent platform should abstract the protocol.

6. Conclusion

I have presented a multi-agent smart home architecture that could address the requirements found by the AMIGO project. The technology is already available and no large obstacles should hinder the implementation. Some research is needed to find an optimal agent communication model and develop it to support the resource management between the agents.

Overall agent technology seems to be a solid solution for a smart home system. The characteristics of autonomy, cooperation, intelligence and awareness are appropriate and needed qualities. The use of a centralized context manager and hardware abstraction eases the development of third party applications, which can then be utilized dynamically by the smart home agents.

A system designed through the actual user requirements, such as the system I've presented here, is the only way to get intelligent technology into the households. In the end it's not important how the system is implemented as long as it works like the users want it to.

7. References

[Battarbee and Kuusela, 2005] – Katja Battarbee and Kristo Kuusela, Interacting with Proactive Technology, In: Frans Mäyrä and Ilpo Koskinen (eds.), *The Metamorphosis of Home.* University of Tampere, 2005, 71-85

[Cook et al., 2003] – D.J. Cook, M. Huber, K. Gopalratnam ja M. Youngblood, *Learning to Control a Smart Home Environment,* Innovative Applications of Artificial Intelligence, 2003, available as: http://ranger.uta.edu/~holder/courses/cse6362/pubs/Cook03.pdf

[Franklin and Graesser, 1996] - Stan Franklin and Art Graesser, Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents, *Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages*, Springer-Verlag, 1996, Also available as: http://www.msci.memphis.edu/~franklin/AgentProg.html

[Georgantas, 2005] - Nikolaos Georgantas (ed.), Specification of the Amigo Abstract Middleware Architecture, *Available as http://www.hitech-projects.com/euprojects/amigo/deliverables/Amigo_WP2_D2.1_v10%20final.pdf, 2005*

[Lesser et al., 1999] - Victor Lesser, Michael Atighetchi, Brett Benyo, Bryan Horling, Anita Raja, Régis Vincent, Thomas Wagner, Ping Xuan and Shelley XQ. Zhang, The UMASS Intelligent Home Project, *Proceedings of the third annual conference on Autonomous Agents*, 1999, 291 – 298, ACM Press

[Meyer and Rakotonirainy, 2003] - Sven Meyer and Andry Rakotonirainy, A survey of research on context-aware homes, Proceedings of the Australasian information security workshop conference on ACSW frontiers, **Volume 21**, 2003, 159 – 168

[Mäyrä et al., 2005] – Frans Mäyrä, Tere Vadén and Ilpo Koskinen, Introduction: Living in Metamorphosis – the Whys and Hows of Proactive Home Design Research, In: Frans Mäyrä and Ilpo Koskinen (eds.), *The Metamorphosis of Home*. University of Tampere, 2005, 7-27

[Park et al., 2003] - Sang Hyun Park, So Hee Won, Jong Bong Lee and Sung Woo Kim, Smart home – digitally engineered domestic life, *Personal and Ubiquitous Computing*, **Volume 7 , Issue 3-4,** 2003, 189 – 196

[Röcker et al., 2004] - Carsten Röcker, Maddy D. Janse, Nathalie Portolan and Norbert Streitz, User Requirements for Intelligent Home Environments: A Scenario-Driven Approach and Empirical Cross-Cultural Study, *Joint sOc-EUSAI conference*, 2004, 111-116, Also available as http://www.hitech-projects.com/euprojects/amigo/publications/roecker_et_al.pdf

[Spinellis, 2003] - Diomidis D. Spinellis, The information furnace: consolidated home control, *Personal and Ubiquitous Computing*, **Volume 7 Issue 1**, 2003, 53 – 69