Implementing scenarios in a Smart Learning Environment

Christoph Burghardt, Christiane Reisse, Thomas Heider, Martin Giersich and Thomas Kirste
University of Rostock
Department of Computer Science
Chair of Mobile Multimedia Information Systems
Rostock, Germany
Email: firstname.lastname@uni-rostock.de

Abstract

Among the scenarios for pervasive learning are teleteaching and project work. The former tries to deliver a lecture to a distributed audience, while the latter tries to enhance the learning effect by concentrated collaboration. In this paper, we want to show how an ambient environment, designed to support teams in a meeting, can be used to support pervasive learning scenarios as well as meetings. The developed applications help the users in both types of scenarios with pro-active assistance, enabling the users to focus on their task, rather than fiddling with the controls of the devices.

1 Introduction

There have been a number of research projects in ubiquitous computing in the past that have researched the future of teaching.

An example is the University of Stanford's Intelligent Classroom. [**?**] This project augments rooms with big interactive display walls. For interacting with the system, a software infrastructure was created and experiments in human computer interaction where conducted.

The Smart Classroom Project [**?**] implemented a distributed teaching scenario. Video streams of remote participants are displayed on a wall of the classroom and the teacher selects remote students via a laser pointer.

Another project is the eClass project at Georgia Institute of Technology [1]. The main focus in this project is annotating meetings, capturing notes and documents and managing them in an electronic library. While supporting the team with automating tasks such as taking notes, this project did not focus on helping users with managing all the installed electronic devices.

While most ambient environment initiatives develop a certain application that fits into an ambient environment scenario, our approach is slightly different. We have developed a smart meeting room [7] for the domain of supporting groups of users. In this paper, we want to show that this generic infrastructure can also support the scenarios of pervasive learning.

1.1 SmartLab Infrastructure

We have a Smart Laboratory as a means of development for ambient environments. The SmartLab is augmented with sensors. Among these sensors are cameras, microphones and an ultrawideband positioning system. Two cameras capture the speaker and the audience. One camera is movable and can focus on certain areas, e.g. the surfaces of the installed projectors. The microphones capture the voices of the audience.

Furthermore a number of actuators is installed. Seven projectors can display information on a number of surfaces. One projector is mounted on a movable head in order to optimize the visibility of the information. A smartboard captures the input of the speaker or notes taken. The room is connected via a home automation bus: All installed lights, sun-blinds and climate control can be controlled via a computer. The lights can be dimmed.

2 Pervasive Learning Scenarios

In our SmartLab we have implemented some of the key applications that are important for a pervasive learning scenario. In this chapter, we want to demonstrate how the installed components work together in different scenarios in order to show how the combination of devices is greater than the sum of its parts.

The room is equipped with multiple sensors and actuators. For displaying content we have installed six projectors along with eight surfaces. In order to be able to reach every surface, one of the projectors is mounted on a moving head.

0-7695-3113-X/08 $25.00 © 2008 IEEE
DOI 10.1109/PERCOM.2008.96

For capturing the actions of the users we installed an ultrawideband positioning system, along with a set of cameras and microphones. This enables us to capture the current position, velocity and direction of the user. The cameras and the microphones enable context acquisition, including simple tasks (presence / no presence) as well as specialised context.

The smart-board captures the input of the user. Together with a projector, the whiteboard can control a computer desktop.

All installed devices are interconnected through a home automation bus. All lights, sun-blinds and surfaces and climate control can be controlled via a computer.

All these devices are interconnected via a network (LAN or WLAN) using our own middleware called Eco. This middleware handles device discovery and cooperation.

We want to demonstrate how the cooperation between the devices is the key to a good experience with a smart environment. In order to evaluate how the installed components work together we will present some scenarios to exemplify the cooperation between different devices.

The first scenario is a teleteaching scenario: the transmission of a lecture to pupils / students far away, e.g. over the internet. A prerequisite for a successful lecture are two or more cameras and a director that operates the equipment, in order to ensure that all important details of the lecture are recorded. Because it is not an option to employ a human director for every smart environment out there, a good cooperation of the devices should be able to ensure a comparable achievement.

Our SmartLab is equipped with two cameras. The first one is movable and has a motorized pan/zoom/tilt. The second one provides a fish-eye view that captures the audience and gives an overview of the situation in the room.

To give the lecturer more freedom, it is important that he does not need to focus on the technical details of the transmission. Instead, the installed technical equipment should enable him to focus on his lecture.

The sensors in the room enable the camera to act smart. By knowing where the person and the important objects are, the camera can pan and zoom the display window. The audience always sees all the details that are interesting to them: the person that gives the lecture and the whiteboard. The person can move through the room, the camera will follow. If the person is near a whiteboard, it tries to capture as much of the whiteboard as possible.

If the lecturer is not near a whiteboard to show the details, the fish-eye camera takes over again. It gives an overview not only of the lecturer but also of some of the audience. This enables a person watching the transmission to get a broader impression of the current focus and some reactions of the audience.

If a question is asked, the camera focuses on the questioner. If a dialog is spoken, then the fish-eye gets in to capture the dialog between the two, without the main camera moving back and forth.

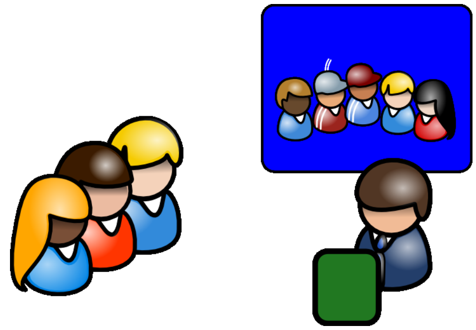

Figure 1. The lecturer gets recorded during his presentation. The people in the other room are "connected" through a projection on the surface.

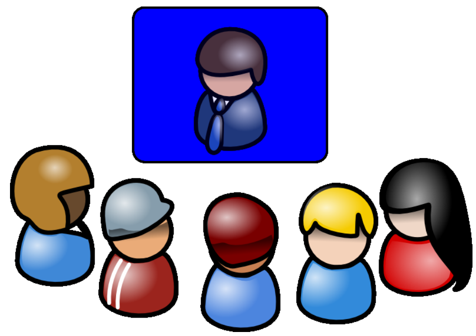

Figure 2. The audience listens to the lecture on a big screen. They can ask questions and are "connected" to the main lecture room with a projection a surface.

To guarantee that the lecture is captured as good as possible, the camera is interconnected with the lighting control and the sun-blinds. To ensure optimal contrast, the light is dimmed. If light dimming does not yield satisfactory results, the sun-blinds are automatically lowered. Again, it is the cooperation of all devices that makes a room "smart".

Another important scenario is group work. Small groups work together on a project for some time. They have to share and compare information to come to a conclusion. Our SmartLab can give assistance for this type of meetings as well:

The projectors can change the lighting colors and conditions of the room. Should the presenter run out of time, we can change the room color from a light green to a red tone to inform him. If a heated discussion develops, we can try to calm down the people with relaxing images or color fades.

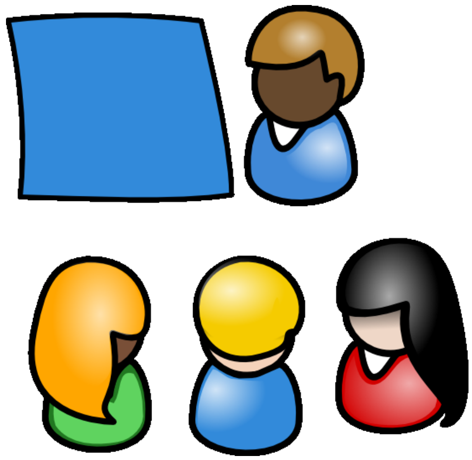

Figure 3. A workshop is a small group of people, focusing on a specific problem.

In the next sections, we will describe some of the underlying technologies for these applications in more detail.

3 Activity and Intention Recognition

For describing human behavior we employ probabilistic models. They specify the probabilities of different possible actions that may be undertaken in certain situations.

For *inferring* the activity of a user from sensor data, we need a specification of *how probable* a certain execution sequence is. Computing the user's current activity from sensor data requires a model that allows to make statements about the likelihood of sensor data given a specific activity. A system can then try to identify the user's current task based on the most likely action sequence with respect to the observed sensor data.

Bayesian Filtering doing this has been successfully used in several projects that aim at supporting user activities in classrooms, meeting rooms, and office environments [5, 2, 4].

Dynamic Bayesian networks (DBNs) are investigated increasingly for the purpose of modelling a user's activities [**?**, **?**].

A Bayesian Network (BN) is a probabilistic graphical model that specifies the relations between a finite set of discrete random variables. Formally spoken, a Bayesian Network (BN) is a directed acyclic graph (DAG) $G = (V, E)$. V is the finite set of vertices and E denotes the finite set of directed edges. Each node in a BN represent a random variable Q with a conditional probability distribution. Edges describe the dependencies of random variables whereas a directed edge from node A to B denotes that the state of B conditionally depends on the state A. DBNs are time-sliced based and in each time slice the possible state of the system is defined in terms of a BN, but conditional dependency can occur between variables of different time slices. This is used to express that the value of a variable at time slice t can depend on a variable at time slice $t - 1$.

For the sake of brevity, we spare the details about how to perform inference and filtering with these models and refer the interested reader to specific literature [**?**].

Our DBN is based on graph-based probabilistic location estimation. It improves the described hierarchical state model [**?**, **?**]. For a complete description of the model, see [6]. Our current research focuses on how to efficiently generate probabilistic models for a wide group of scenarios. Our results, regarding intention recognition for cooperative teams, inference accuracy and speed showed that despite noisy observable sensor data, precise and robust inference is possible[8].

This model allows us to predict the next actions of the user and therefore enables pro-active assistance. The intention analysis and the representation of the intentions as an explicit goal is a prerequisite for scheduling the devices.

4 Multi-Display Management

Multi-Display Environments support collaborative problem solving and teamwork by providing multiple display surfaces for presenting information. Typical examples for such environments are meeting rooms, conference rooms, "mission control centers" and also modern teaching/learning environments.

One difficult task here is the *Display Mapping problem* – that is, deciding which information to present on what display in order to optimally satisfy the users' needs for information. While this task is more or less trivial in single-user, single-display situations, it becomes challenging in multi-user, multi-display settings: Users and displays are spatially dispersed so that the visibility of (semi-) public and private displays varies among users. Also, information needs may vary among users, so that finding the "best" assignment of information to displays becomes a difficult problem.

Current approaches for controlling multi-display environments rely on manual assignment, using a suitable interactive interface, resolving conflicts by social protocols (negotiations). One example is the ModSlideShow system [3], which is designed to manage presentation slides on multiple displays. For assignment of content to displays, meeting participants drag-and-drop presentations from their notebooks to any of the available displays. Another similar system is the PointRight software developed for Stanford's Meyer Teamspace [**?**]. However, manual display assignment has to cope with the following problems:

• **Interest conflicts** between users might be solved faster by computer supported negotiation mechanism: Morris et

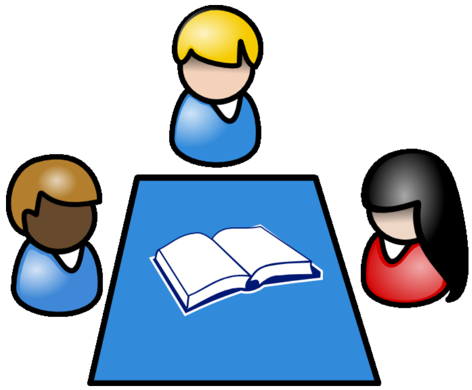

Figure 4. A group examining a problem en detail.

al. [?] have already observed that social protocols do not always suffice for coordinating the use of shared resources, such as display surfaces, in teams – even in relatively simple situations. They suspect that the need for coordination may increase as the number of users, the number of documents, or the number or size of the surfaces increases. Indeed, they advocate the development of specific strategies for automating the negotiation process.

• The need for **dynamic realignment** of Display Mapping is caused by topic changes in the user population – in this situation, the user's focus of attention will be on the changing topic rather than on convincing the display infrastructure to change the topic.

Therefore, an automatic display assignment might be helpful in multi-display environments, specifically in multi-user settings. In our approach, we consider the finding of the "best" assignment of information to displays as a typical optimization problem that can be solved by defining an objective function $q(x)$, the "quality", to maximize and then applying a suitable optimization algorithm to compute $x_{max} = \arg\max_{x \in X} q(x)$.

So the Display Mapping problem gives rise to two subproblems:

- What is a suitable definition for $q(x)$? I.e., what is the objective function to be maximized in order to achieve an optimal (or at least: satisfactory) solution for the Display Mapping problem?
- How should the computation of x_{max} be distributed across the members of an ensemble of displays? – This is especially interesting when dynamic ensembles have to be considered (e.g., portable projectors carried into a meeting room, etc.).

Our goal function q has been defined completely independent from a concrete ensemble of users, displays, documents, and surfaces. It describes the globally optimal behavior for any possible ensemble. Once machinery is available for computing the optimum for q, any ensemble will be able to behave optimally – as far as q is a correct definition of an ensemble's global optimum from the user's point of view. We have also developed an algorithm that is able to automatically compute a display mapping for a set of users and documents (see [?] for more details).

In [?] we have described the experiment we have used for assessing the performance of an automatic display assignment based on q in comparison to a manual assignment. Specifically, we have measured the impact of manual vs. automatic display assignment on the performance of a team in solving a semi-cooperative task. In such tasks, the need for cooperation and joint use of information is not evident from the start, but rather arises while working on the task. We think that this kind of aspect pertains to many team processes, specifically in multidisciplinary teams.

Our user studies have shown that – at least for specific scenarios – automatic display assignment based on q is better than manual assignment. Therefore, it proves that it is *possible* to successfully identify a set of quality criteria for automatic display assignment. Indeed, we have even been able to show that automatic assignment enables teams to solve their tasks in a shorter time, with less conflicts between team members, and with greater satisfaction.

5 Intelligent Room Control

For the communication of devices we implemented a middleware called Eco in our smart lab, as already mentioned in section 2. Devices can register and unregister themselves with this system, as well as request information about which other devices are present. They can even send instructions to other devices. Communication among devices is carried via so-called EcoChannels the devices speak and listen to. Technically, these EcoChannels are realized using multicast addresses.

We are currently developing a distributed, self-organizing room control system for our SmartLab that uses the Eco middleware as a communication infrastructure.[1] This room control system allows devices like notebooks, projectors and canvasses to cooperate spontaneously in order to assist the users with tasks like giving a presentation. Spontaneously means that these devices do not have to know one another. They can be brought into the room, register with the middleware and are able to cooperate right away. An important point is that this system works fully decentralized and is therefore very robust. This kind of approach is beneficial for project work scenarios because it allows a group of users to focus on their collaborative tasks

[1] This system is currently implemented as a simulation, but will use real devices in the future.

rather than connecting their notebooks to projectors, manually adjusting the lighting conditions etc. We will briefly describe this approach in the following section.

For our room control system we adapted an approach proposed by Pattie Maes [?] in the area of robotics. Each device describes the actions it can perform in terms of its preconditions and effects, similar to operators in classical planning. A projector, for example, has the capability of projecting a document onto a canvas if another device sends the document to the projector. The effect of this action is that the document is visible on a canvas. These action descriptions are incorporated into so-called competence modules – active entities that physically reside on the devices, communicating via the Eco middleware. Through communication they determine which of them will become active next. Furthermore, they perform all administrative tasks themselves, e.g. keeping track of the state of the environment (at least of the part that is relevant to them) or synchronizing with other modules.

Each competence module has an activation level. This activation level states how "popular" a module is among other modules, that is, how desirable other entities in the system deem the execution of the module. If the activation level exceeds a certain threshold and all of the module's preconditions are fulfilled, the module becomes executable. This approach assumes that the users' goals are described as states that should be made true by the device ensemble, e.g. *Lamp 5 turned on*, *Lamp 2 turned off* and *Document 1 is shown on canvas 4*. The users' goals are provided by the intention recognition (see section 3). Furthermore, the state of the environment can be detected via sensors (e.g light sensors).

Modules are connected to other modules through links, as shown in figure 5. Just a few links and modules are shown in this figure to preserve readability. Through these links modules can send activation energy to other modules which can help them in fulfilling their goals. They can also take away activation energy from modules that might hinder them. Energy is initially inserted into the network by the context (e.g. sensor data and the users' goals). That is, if a sensor notes that a precondition of a module is fulfilled, this sensor sends activation energy to the module. Likewise, if there is a user goal that corresponds to an effect of a module, this module also receives activation energy because the user would obviously regard the execution of the module as a good idea. The activation energy then spreads throughout the network via the links. Eventually enough energy will accumulate in some module and it will become active, provided that all its preconditions are fulfilled. Through its execution the module puts the system into a new state, thus providing new context information and new energy.[2] After some time the next module will have all of its preconditions fulfilled and possess enough energy to be executed. The simulation stops when all the goals have been fulfilled.

[2]If no module is executed in a simulation cycle, the activation threshold is lowered by ten per cent.

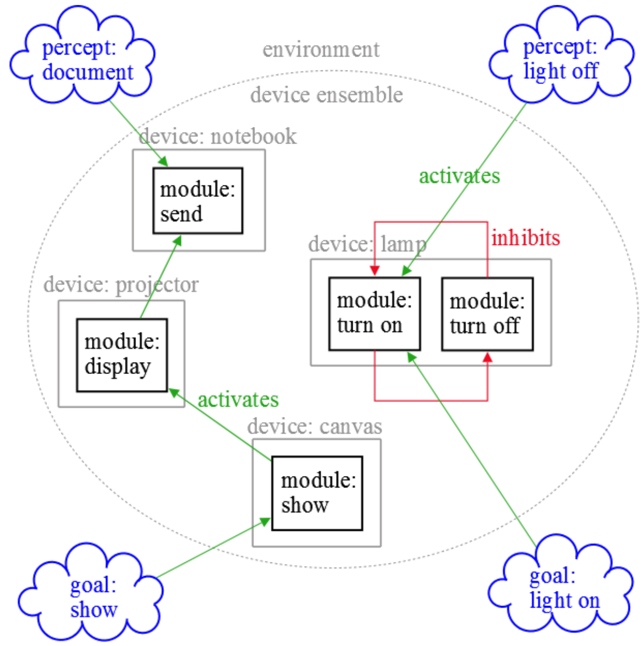

Figure 5. Modules interact through links, the most "popular" modules are executed. Just a few modules and links are depicted for the sake of clarity.

Let us look at a simple example scenario to illustrate this.[3] In this scenario, the room is equipped with four projectors, six canvasses, six lamps, and a notebook hosting a document. The initial state, goal state and the optimal action sequence leading from the initial to the goal state are:

initial state	all lamps off document open on notebook
goal state	lamp 1 on lamp 2 on document is displayed on canvas 3
optimal action sequence	turn on lamp 1 turn on lamp 2 notebook sends document to projector 3 projector 3 projects document onto canvas 3 canvas 3 moves downwards

In 20 test runs with this scenario, the algorithm took six simulation cycles to find a solution, which was the optimal solution in every case. In one run the algorithm found the optimal solution after seven simulation cycles.

[3]More complex scenarios have been implemented, but discussing them is beyond the scope of this paper.

6 Conclusion

The SmartLab is our prototype for an ambient environment. It is suitable for implementing smart meeting room scenarios as well as pervasive learning scenarios. We are evaluating new concepts to enhance the collaboration of devices and users. We developed a middleware which separates the single steps into separate modules. We are currently working on increasing convenience for the users. Our main goal is to enhance the sensors and to adapt the currently installed devices better to new scenarios in order to broaden the possibilities of use and improve flexibility.

7 Acknowledgement

This research is supported by the German National Science Foundation (DFG, GRK1424).

References

[1] G. D. Abowd, J. Brotherton, and J. Bhalodia. Classroom 2000: a system for capturing and accessing multimedia classroom experiences. In *CHI '98: CHI 98 conference summary on Human factors in computing systems*, pages 20–21, New York, NY, USA, 1998. ACM Press.

[2] H. Bui. A general model for online probabilistic plan recognition. In *IJCAI '03: Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 1309–1315, 2003.

[3] P. Chiu, Q. Liu, J. S. Boreczky, J. Foote, D. Kimber, S. Lertsithichai, and C. Liao. Manipulating and annotating slides in a multi-display environment. In *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction*, 2003.

[4] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 838–845, Washington, DC, USA, 2005. IEEE Computer Society.

[5] D. Franklin, J. Budzik, and K. Hammond. Plan-based interfaces: keeping track of user tasks and acting to cooperate. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 79–86, New York, NY, USA, 2002. ACM Press.

[6] M. Giersich, P. Forbrig, G. Fuchs, T. Kirste, D. Reichart, and H. Schumann. Towards an integrated approach for task modeling and human behavior recognition. In J. Jacko, editor, *Human-Computer Interaction, HCII 2007*, volume Part I, pages 1109—1118, Heidelberg, June 2007. Springer Verlag.

[7] M. Giersich and T. Kirste. Effects of agendas on model-based intention inference of cooperative teams. In *Proc. 3rd Int'l. Conf. on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007)*, White Plains, NY, USA, Nov. 12–15 2007.

[8] M. Giersich and T. Kirste. Effects of Agendas on Model-based Intention Inference of Cooperative Teams. In *Proceedings of CollaborateCom2007 - The 3rd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, White Plains, NY, USA, November 12 - 15 2007. IEEE Xplore.

[9] T. Heider, M. Giersich, and T. Kirste. Resource optimization in multi-display environments with distributed grasp. In *Proceedings of the First International Conference on Ambient Intelligence Developments (AmI.d'06)*, pages 60 – 76, Sophia Antipolis, France, September 19 - 22 2006. Springer.

[10] T. Heider and T. Kirste. Automatic vs. manual multi-display configuration: A study of user performance in a semi-cooperative task setting. In *The 21st BCS HCI Group conference (HCI 2007)*, Lancaster, UK, September 2007.

[11] B. Johanson, A. Fox, and T. Winograd. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing*, 1(2):67–74, 2002.

[12] B. Johanson, G. Hutchins, T. Winograd, and M. Stone. PointRight: Experience with Flexible Input Redirection in Interactive Workspaces. In *Proc. ACM Conference on User Interface and Software Technology (UIST2002)*, pages 227–234, Paris, France, 2002.

[13] P. Maes. Situated Agents Can Have Goals. In P. Maes, editor, *Designing Autonomous Agents*, pages 49–70. MIT Press, 1990.

[14] M. R. Morris, K. Ryall, C. Shen, C. Forlines, and F. Vernier. Beyond "social protocols": multi-user coordination policies for co-located groupware. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pages 262–265, New York, NY, USA, 2004. ACM Press.

[15] K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, 2002.

[16] D. J. Patterson, L. Liao, D. Fox, and H. A. Kautz. Inferring high-level behavior from low-level sensors. In *Proceedings of UBICOMP 2003: The Fifth International Conference on Ubiquitous Computing*, October 12-15 2003.

[17] D. J. Patterson, L. Liao, K. Gajos, M. Collier, N. Livic, K. Olson, S. Wang, D. Fox, and H. A. Kautz. Opportunity knocks: A system to provide cognitive assistance with transportation services. In *Ubicomp*, pages 433–450. Springer, 2004.

[18] Y. Shi, W. Xie, G. Xu, R. Shi, E. Chen, Y. Mao, and F. Liu. The Smart Classroom: Merging technologies for seamless tele-education. *IEEE Pervasive Computing*, 2(2):47–55, 2003.