

Smart Classroom: Bringing Pervasive Computing into Distance Learning

Yuanchun Shi, Weijun Qin, Yue Suo, Xin Xiao

1 Introduction

In recent years, distance learning has increasingly become one of the most important applications on the internet and is being discussed and studied by various universities, institutes and companies. The Web/Internet provides relatively easy ways to publish hyper-linked multimedia content for more audiences. Yet, we find that most of the courseware are simply shifted from textbook to HTML files. However, in most cases the teacher's live instruction is very important for catching the attention and interest of the students. That's why Real-Time Interactive Virtual Classroom (RTIVC) always plays an indispensable role in distance learning, where teachers and students located in different places can take part in the class synchronously through certain multimedia communication systems and obtain real-time and media-rich interactions using Pervasive Computing technologies [1]. The Classroom 2000 project [2] at GIT has been devoted to the automated capturing of the classroom experience. Likewise, the Smart Classroom project [3] at our institute is focused on Tele-education. Most currently deployed real-time Tele-education systems are desktop-based, in which the teacher's experience is totally different from teaching in a real classroom. For example, he/she should remain stationary in front of

Yuanchun Shi
Key Laboratory of Pervasive Computing, Tsinghua University, Beijing 100084, China, e-mail:
shiyu@tsinghua.edu.cn

Weijun Qin
Key Laboratory of Pervasive Computing, Tsinghua University, Beijing 100084, China, e-mail:
qinweijun99@mails.tsinghua.edu.cn

Yue Suo
Key Laboratory of Pervasive Computing, Tsinghua University, Beijing 100084, China, e-mail:
suoy@mails.tsinghua.edu.cn

Xin Xiao
Key Laboratory of Pervasive Computing, Tsinghua University, Beijing 100084, China, e-mail:
xiaoxin00@mails.tsinghua.edu.cn

a desktop computer instead of moving freely in the classroom. The goal of Smart Classroom project is to narrow the gap between the teacher's experience in Tele-education and that in the traditional classroom education, by means of integrating these two currently separated education environments together. Our approach was to move the user interface of a real-time Tele-education system from the desktop into the 3D space of an augmented classroom (called Smart Classroom) so that in this classroom the teacher could interact with the remote students with multiple natural modalities just like interacting with the local students. However, to provide this type of distance learning in large scale, there still remain some barriers:

- **Desktop-based teaching metaphors are not natural enough for most teachers.** Most current RTIVC systems are desktop-based, i.e. the users should remain stationary in front of a desktop computer and use the keyboard or mouse to operate the class or to interact with others. This metaphor is particularly unacceptable for the teachers, because their experience here is of much difference from that in a real classroom. In a real classroom, they can move freely, talk to the students with hand-gesture and eye contact and illustrate their ideas conveniently by scribbling on the blackboard. Many teachers involving in Teleeducation, when talking with us, complained that this divergence of the experience makes them uncomfortable and reduced the efficiency of the teaching and learning activity.
- **Lack of adequate technologies to cope with large-scale access.** Most Teleeducation schools simply adopted commercial videoconference products as the operating platform for RTIVC, where all clients should connect to a centered MCU (Multi-Point Controlling Unit) and data initiated from one client is replicated (sometime maybe mixed) and forwarded to all other clients by MCU. However these systems are not scalable, for the maximum user number (usually ten more) is rigidly limited by the capacity of the MCU. So, today, most teleeducation schools can only operate RTIVC classes on a small student number base. A possible approach to address the scalability issue is leveraging the IP Multicast technology, where no central data-replicating node like MCU is required. But the current state of the art of IP Multicast is still not fully matured.
- **Lack of adequate technologies to accommodate students with different network and device conditions in one session.** Most current RTIVC systems have rigid requirements on the network and device conditions of the clients. Clients with inferior conditions either could not join the session or could not get smooth service quality. On the other hand, clients with superior conditions could not fully take advantage of their extra capabilities. Since handheld devices such as Pocket PC and Smart Phone, and wireless network such as GPRS and WiFi are becoming more and more popular, it is a natural demand to allow people to accept lifelong education through these devices, while they will inevitable possess diversified capabilities and network connections.

The Smart Remote Classroom project at our institute is a long-term project to overcome the above-mentioned difficulties in current practice of RTIVC and building an integrated system as an exemplar for the next generation real-time interac-

tive distance learning in China. The aim of this project is to develop new pervasive computing technologies in the classroom to provide practical natural convenient multimodal interfaces and context-aware applications to assist local teacher, and to develop large-scale remote e-learning interactive application to enhance the class activity between local teacher and remote students.

The rest of the chapter will be organized as follows: Sect. 2 introduce the overview of Smart Classroom and gives a typical use experience class scenario. Sect. 3 presents key multimodal interfaces and context-awareness technologies for building sensor-rich classroom environment. Large-scale real-time collaborative distance learning technology is investigated in Sect. 4 in order to provide augmented learning experience and interactions between local class and remote students. In Sect. 5, we introduce the infrastructure called Smart Platform to integrate various components to create virtual class and a conclusion is made finally in Sect. 6.

2 Smart Classroom: Sensor-Rich Smart Environment

In this section, we briefly introduce the layout of Smart Classroom which was settled in Key Laboratory of Pervasive Computing, Tsinghua University and give a typical user experience scenario in distance learning.

2.1 The Layout of Smart Classroom

The Smart Classroom is physically built in a separate room of Pervasive Computing Lab in Tsinghua University. As Fig. 1 shown, several video cameras, microphone arrays are installed in it to sense human's gesture, motion and utterance. According to the characteristic of invisibility in pervasive computing environment, we deliberately removed all the computers out of sight. Two wall-sized projector displays are mounted on two vertically crossed walls. According to their purposes, they are called "Media Board" and "Student Board" separately. The Media Board is used for lecturer's use as a blackboard, on which prepared electronic courseware and lecturers' annotation are displayed. The Student Board is used for displaying the status and information of remote students, who are part of the class via Internet.

The classroom is divided into two areas, complying with the real world classroom's model. One is the teaching area, where is close to the two boards and usually dominated by lecturer. The other is the audience area, where is the place for local students. Why are both remote students and local students supported in this room? The answer is simple, that we're complying with the philosophy of Natural and Augmented. Natural means we'll obey real-world model of classroom as much as possible to provide lecturer and students the feeling of reality and familiarity, which leads to the existence of local students. Augmented means we'll try to extend beyond the limitation imposed by the incapability of traditional technology, which

is the reason for remote student. Here is a snapshot of Smart Classroom Prototype as Fig. 2.

2.2 A Typical User Experience Scenario in Smart Classroom

The following is a typical user-experience scenario happened within the Smart Classroom.

Multiple persons enter the room through the door. At the door, there is an audio-visual identification module identifying the entering person's identity through facial and voice identification. If the person is identified as lecturer, he is granted the control right of the Smart Classroom. Besides, he takes a badge embedded with location sensor. The visual motion-track module tracks the lecturer's motion in the room. Once he steps into the teaching area, he will be able to use gesture and voice command to exploit the Smart Classroom to give lessons. Persons in the Smart Classroom other than lecturer are deemed as local students. When the lecturer is in the teaching area, he can start the class by just saying, "Now let's start our class." The Smart Classroom starts Microphone Array Agent to capture and recognize his voice command, and then launches necessary modules such as Virtual Mouse agent, Same View agent (which will be talked about later). Lecturer loads prepared electronic courseware by utterance like, "Go to Chapter 1 of Multimedia course". In the meanwhile, The Smart Cameraman Agent was activated to focus on the lecturer

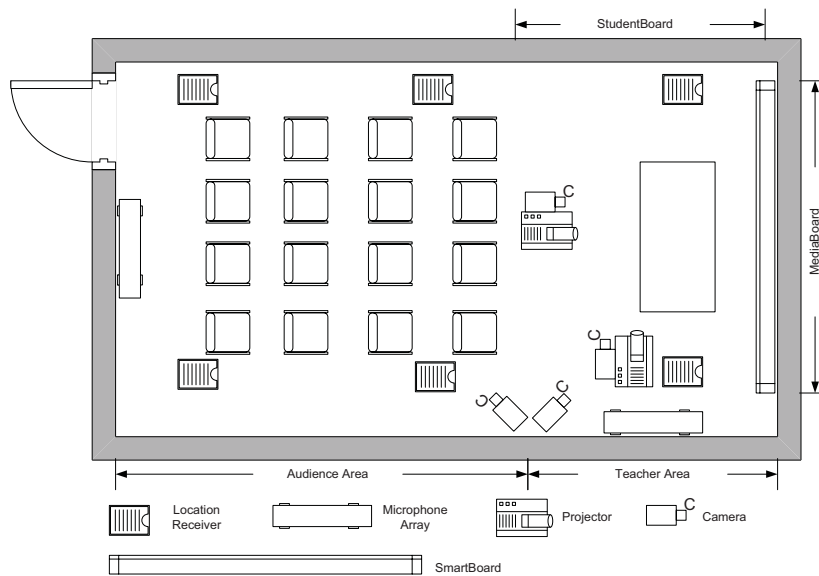


Fig. 1 The Layout of Smart Classroom

based on his current location and show the close-up view. The HTML-based courseware is then projected on the wall display. Lecturer can use hand-motion to stimulate the Virtual Mouse agent to annotate on the electronic board. Several type of hand gestures are assigned corresponding semantic meanings, which cause several operations like highlighting, annotating, adding pictures, remove object, executing links, scrolling pages etc, on the electronic board. Lecturer can also grant speech right to remote students by finger pointing or voice command, like "Suo, please give us your opinion". On the Student Board, remote students' photos and some information as name, role, speech right etc are displayed. When a remote student requests for floor, his icon on the Student board twinkles. Once the lecturer grants the floor to a specific remote student, his video and audio streams are synchronously played both in the Smart Classroom and on other remote students' computers.

3 Key Multimodal Interface and Context-Awareness Technologies

In this section, we present the key multimodal interface and context-awareness technologies in Smart Classroom which merge speech, handwriting, gestures, location tracking, direct manipulation, large projected touch-sensitive displays and laser pointer tracking which aim to provide enhanced experience for both of teachers and students in Smart Classroom and detect the basic contextual information to create context-aware applications to facilitate human to achieve their task.

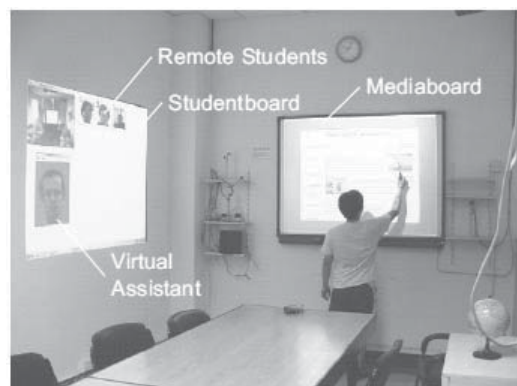


Fig. 2 A Snapshot of Smart Classroom Prototype

3.1 Location Tracking

In the Smart Classroom Project, the aim of location tracking is to capture the target's indoor location context using location sensors and positioning technologies to provide location-based services. In the scenario of the Smart Classroom, the lecturer's location is captured from the taken badge sensor and calculated from multiple signal receivers. Thus, the Smart Classroom system can provide context-aware services based on the lecturer's location. For example, the Smart Cameraman Agent made of camera pan and tilt can focus on the lecturer based on his precise position when the lecturer is speaking inside the teaching area. For supporting location-aware computing in indoor smart environments, the location sensing/positioning system not only need to provide objects' precise location, but also should own such characteristics as: isotropy and convenience for portability. In this subsection, we present an indoor location sensing system, Cicada. This System is based on the TDOA (time difference of arrival) between Radiofrequency and ultrasound to estimate distance, and adopts a technology integrating Slide Window Filter (SWF) and Extended Kalman Filter (EKF) to calculate location. Consequently, it not only can determine the coordinate location within 5cm average deviation either for static objects or for mobile objects, but also owns a nearly omni-directional working area. Moreover, it is able to run independently, mini and light so that it is very easy to be portable and even embedded into people's paraphernalia.

3.1.1 Working Principle and Framework

Being somewhat similar to the physical principle of Cricket [4], Cicada is also based on the TDOA between RF (radiofrequency) and ultrasound. Being different from Cricket, Cicada is of active mode. The framework of Cicada is shown in Fig. 3.

As Fig. 3 shows, Cicada consists of three main parts: CBadge, CReader and Location Server. CBadges are carried by users or attached to those objects located, which emit RF and Ultrasound at the same time periodically. And each RF corresponding to each CBadge modulates the CBadge's ID. CReaders are deployed on the fixed location of a building, e.g. ceilings, whose coordinate location are known by beforehand measurement. Because the propagation speeds of RF and Ultrasound are different, the TDOA between them from a CBadge to a CReader is direct proportional to the distance between them, where the coefficient is the velocity of sound in air (neglecting the propagation time of RF). According to the theory, a CReader can infer the distance from a CBadge, and then report it to a dedicated computer, Location Server, through a serial port cable. The Location Server collects all distance and calculates out location. Applications can acquire location from the Location Server as its clients.

The close-up of CBadge and CReader is shown in Fig. 4. A RMB coin and a ruler (the unit is cm) are nearby. The CBadge has a rechargeable 3V lithium battery attached on its back. The omni-directional ultrasound transmitter of CBadge is composed of 5 ultrasound sensors being mutually orthogonal. The CReader has two

ports: one is USB, which connects to the serial port of a computer by a USB-to-COM Switching; and another is parallel port, which connects to a computer's parallel port. The parallel port is unplugged unless CReader is re-programmed. And when Cicada runs, the USB port must connect to the Location Server (being plugged) all the time.

3.1.2 Positioning Algorithm

After acquiring the distance set from CReaders, how to infer objects' coordinate location is concerned with the positioning algorithm. In Cicada, the positioning computation includes two phases: distance filter and location calculation.

- **Distance Filter**

Due to ultrasound's reflection, obstruction and diffraction from wall, furniture and instruments etc, the indoor multi-path effect occurs very often. That is, quite a few of distances is invalid because they do not traverse along a LOS (line-of-sight). Before the distances go into location calculation, the invalid value of them must be filtered out carefully. Here Cicada adopts a method called Slide Window Filter (SWF).

The data received from CReaders includes CReader's ID, CBadge's ID, distance between the CReader and the CBadge, and the timestamp when the CBadge emits its signal, which is quaternion $[r, b, d, t]$. If the CReader and CBadge are given, the data from CReaders is a set of pairs $[t_i, d_i], i = 1, 2, \dots, L$, according to time's ascendant order.

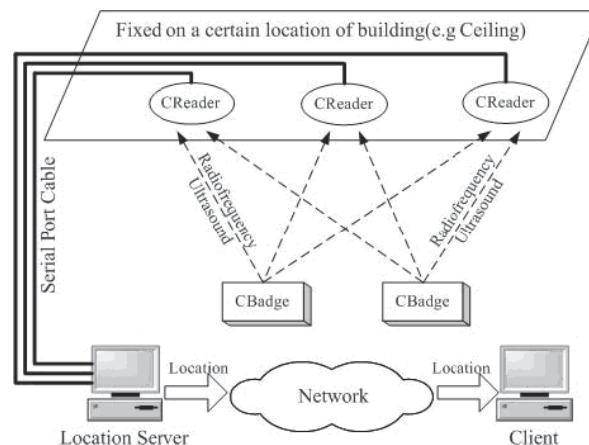
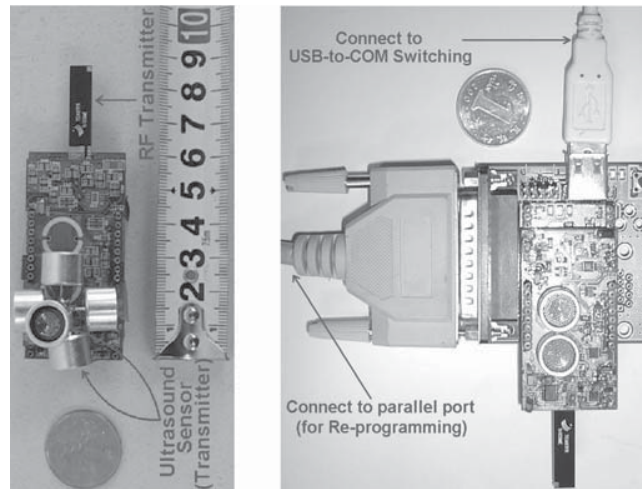


Fig. 3 Working Principle and Framework of Cicada

Define a distance tuple D as the triple $D_i = [t_i, d_i, v_i]$, where $v_i = (d_i - d_{i-1}) / (t_i - t_{i-1})$. A slide widow is a cyclic queuing to stores recent distance tuple for each pair of CReader and CBadge. In slide widow, what the rear points at is the distance tuple received most recently, and what the front points at is the distance tuple received earliest, which was dequeued just now. Define the average velocity of slide widow as $\bar{V} = \frac{1}{n} \sum_{i=1}^n v_i$. When a new distance tuple $N = [t, d, v]$ is received, if $v \leq \min\{\alpha\bar{V}, V_{max}\}$, it will be added into the slide window; otherwise it will be rejected as an invalid distance. Here V_{max} is the maximal velocity of objects' moving, and α represents the maximal acceleration. For example, in our actual applications, V_{max} is set to 1.5m/s, which is a typical moving velocity for human indoors, and $\alpha = 1.5$.

- **Location Calculation**

On location's calculation method, after comparing with an intuitive method, Linear Equations Method (LEM), we adopt the Extended Kalman Filter (EKF) as Cicada's location calculation algorithm. Kalman Filter [5] is an optimal estimation method for a linear dynamic system perturbed by Gaussian white noise, while Extended Kalman Filter [6] is the Kalman Filter extended for non-linear system. EKF is composed of circular iterations, and each of iterations is called a time-step, which consists of two phases: prediction and correction, and each time-step can guarantee that the error covariance between the estimated value and actual value is minimal. Moreover, since EKF works on time domain, it owns low computation complexity, against those working on frequency domain, such as Wiener filter. On location calculation, Cicada adopts the EKF based on position-velocity model (PV model).



(a) CBadge

(b) CReader

Fig. 4 The CBadge and CReader in Cicada

We evaluate the position precision performance of Cicada system as Table shown in the Smart Classroom. From the result, the average distance deviation for tracking the moving object is under 26cm.

Table 1 The Position Precision of Cicada

PARAMETER	VALUE
Average Deviation of X-axis	5.7cm
Average Deviation of Y-axis	3.8cm
Average Deviation of Z-axis	23cm
Average Planar Deviation	7.3cm
Average Distance Deviation	26cm

3.2 Direct Manipulation based on Laser Pointer Tracking

Large displays are widely equipped in smart environment these days. In the Smart Classroom, large displays are commonly used for displaying coursewares on the surface of walls or Smart Boards. However, traditional interaction devices which are designed to suit desktop screen, such as mice, keyboards, have various limitations in such environments. In this subsection, we present a novel human-computer interaction system, known as the CollabPointer, for facilitating interaction with large displays in Smart Spaces. A laser pointer integrated with three additional buttons and wireless communication modules is induced as input device in our system and three features distinguish the CollabPointer from other interaction technologies as follows:

- First, the coordinates of the red laser point on the screen emitted by the laser pointer are interpreted as the cursor's position and the additional buttons on it wirelessly emulate a mouse's buttons through radio frequency. It enables remote interaction at any distance.
- Second, when multiple users are interacting, with two-steps associating methods described in this subsection, our system can identify different laser pointers and support multi-user collaboration.
- Last but not least, the laser pointer emits its identity through radio frequency during interaction. The system receives it and treats different users separately.

3.2.1 Design of CollabPointer Device

A common laser pointer integrated with three additional buttons and wireless communication modules is introduced as the interaction device in our system. In ad-

dition, to receive the RF signal emitted by the laser pointer and transmit it to the computer, a new hardware called the Receiver is also introduced.

- **The laser pointer**

Fig. 5 shows the architecture of it. There are totally three additional buttons on the laser pointer, On/Off button, Right button and Left Button. Their functions are described as Table 2.

Table 2 Functions of Buttons

BUTTONS	FUCTIONS
On/Off Button	(1)Emitting a laser beam (2)Broadcasting the user's
Right Button	Wirelessly emulating mouse's right button
Left Button	Wirelessly emulating mouse's left button

On/Off button: This button is a switch, not only for turning on the laser pointer but also for broadcasting the user's identity through wireless communication modules. If the button is down, the laser beam is emitted and the ID of the laser pointer is broadcasted through radio frequency at the same time. The system receives the ID through the Receiver. According to its ID, we assign different users with different access priorities. For example, an administrator has more power than a guest.

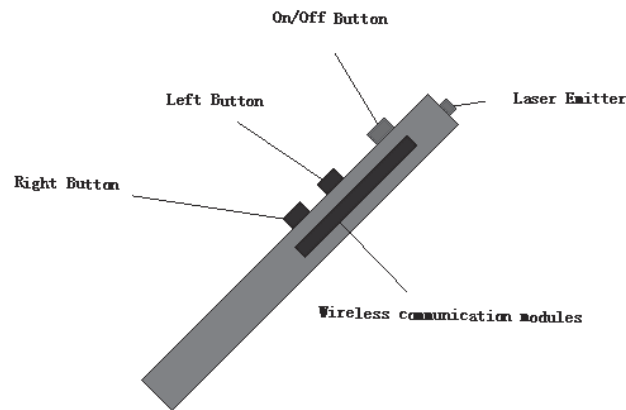


Fig. 5 Architecture of Laser Pointer

Right Button: This button wirelessly emulates a standard mouse's right button. The state of it is transmitted to the computer through radio frequency. The computer receives it via the Receiver and matches it to a standard mouse's right button. When multiple users are interacting simultaneously, users are ranked according to their identities and the computer is controlled by the user who has the highest access priorities. For example, when a teacher and a student are pressing the Right Button simultaneously, the system responds to the teacher's action while ignores the student's.

Left Button: It is nearly the same as the Right Button. The only difference is that it emulates a mouse's left button.

- **The Receiver**

It receives the radio frequency emitted by the laser pointer, decodes it, and then transfers it to the computer through USB interface. According to the three buttons integrated on laser pointer, the computer explains the received signal as the laser pointer's ID, the Right Button's state or the Left Button's state. Figure 6 is the prototype of the Receiver.

First, it decodes the radio frequency emitted by the laser pointer. Next, the IC 74LS00 is used to transit the electrical signal to logical zero, which serves as an interrupt input for the microcontroller. Finally, and also the most important, the microcontroller is programmed to communicate with the computer through USB interface. Phillip's PDIUSB12 IC servers as a bridge between the microcontroller and the computer.

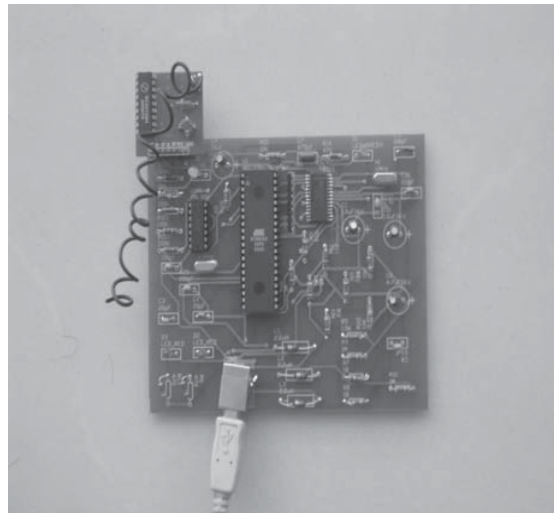


Fig. 6 Prototype of the Receiver

3.2.2 Interaction Mode

For a single user, the laser pointer emulates a standard mouse and can be utilized to interact at a distance. For multiple users, the system can identify a various laser pointers and afford seamless and parallel collaboration among several users. In addition, we can obtain each user's identity while he is interacting. According to it, different people are treated differently.

- **Single User**

In this mode, the coordinates of the laser spot on the display are mapped to the position of the cursor, which means that the user can directly utilize the laser pointer to manipulate the cursor. The Right/Left button on the laser pointer wirelessly emulates a mouse's right/left button. Using this new interaction device, people can "point and click" wirelessly at any distance.

A computer-vision-based module called Laser2Cursor was utilized to locate the laser point's position and mapped it to the position of the cursor. Laser2Cursor embodies a number of ideas not seen in previous work on laser pointer tracking. First, we have developed a training process to improve the system's adaptability. By learning the background of the image captured by the cameras and parameters such as color segmentation and motion detection thresholds, our system automatically adapts to new environments. Second, to improve the system's robustness, we integrate multiple cues such as color, motion, and shape in the laser spot detection. Because most people's hands are unsteady, when a person aims a laser pointer, the spot's exact position usually jitters. We use this characteristic as an additional cue to detect the spot. Next, a Kalman filter smoothes the spot's trajectory, which tends to be irregular and rough.

In addition, the laser pointer broadcasted the user's ID via radio frequency while it is working. The system receives it through the Receiver and treats users separately. For example, we assign different users with different accessing priorities. In Smart Classroom, where the system has been implemented, teachers and students are assigned different accessing priorities. With teachers' laser pointer, the user can view all files on the computer; while with students' one, he can just view his own document.

- **Multi-User**

When multiple users are interacting simultaneously, associating laser spots on the screen with corresponding laser pointers is the basis for collaboration. As shown in Fig. 7, we achieve this aim by two steps. The first is to associate laser strokes with corresponding laser pointers and the second is to associate laser spots with laser strokes.

3.2.3 Use Case in Smart Classroom

We have utilized VC++6.0 to develop an application named M-Drawing to enhance collaboration for a number of users. When it runs, a totally transparent window

covers the screen. Multiple users can draw on it with laser pointers simultaneously and strokes appear different colors according to different users. Fig. 8 is the scenario that two people are discussing about a presentation slide.

3.3 Speaker Tracking based on Microphone Array and Location Sensing

Practically, the teacher sometimes had to take wireless microphone to speak if the class was held in a large classroom, however it's uncomfortable to wear such a strange equipment. In the Smart Classroom Project, the speaker tracking technology was investigated to capture the speaker's voice without taking any equipment

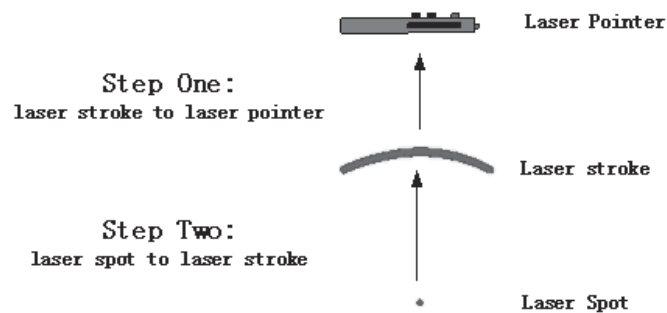


Fig. 7 Associating Laser Spots to Laser Pointers

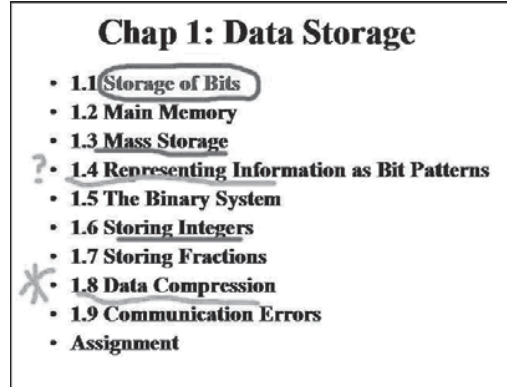


Fig. 8 Two Users are Discussing.

on the speaker's clothes using microphone array and location tracking device Cicada mentioned in Section 3.1. The goal of Cicada is to detect the location of the participants in the meeting room and determine their identities. A small badge containing distinctive user ID is attached to each participant, so every participant has a unique ID and can be distinguished from the others. With the badges, all the participants can be tracked simultaneously and precisely. Four microphone arrays fixed to the walls are used to capture the audio feature of the speaker. Each microphone array has eight omni-directional microphones in a horizontal line with an inter-sensor separation of 8 cm. Audio signals obtained from the microphone arrays are processed to estimate the TDOA between different microphone channels. These two modalities are subsequently integrated by a speaker association algorithm for the purpose of identifying where and who the current active speaker is. The following subsections will detail our system.

3.3.1 Capturing Audio Feature using Microphone Array

Precise person localization, which has become a basic technology in a smart classroom, is useful in a variety of domains, including experience recording, activity analysis, and camera steering. Meanwhile, the reliable identification of the speaker (e.g. teacher), pointing who is speaking, can provide additional information for videoaudio indexing and retrieval tasks in experience recording.

The task of speaker localization and identification poses three basic problems: locating all the participants in the smart classroom, distinguishing the speaker from other silent participants, and determining the speaker's identity to provide the basic context information in smart classroom environment.

Our system uses four microphone arrays to capture the audio feature of the speaker. As shown in Fig. 9, four microphone arrays are placed on different walls of the room at the height of 150 cm. The X, Y plane of microphone array coordinate system is parallel to the ground. Since the arrays are placed at approximately mouth level of the users, we can make the simplifying assumption that sound sources are on the microphone array plane.

The audio feature is based on estimates of the TDOA between different microphone channels in the array. Several practical TDOA estimation algorithms have

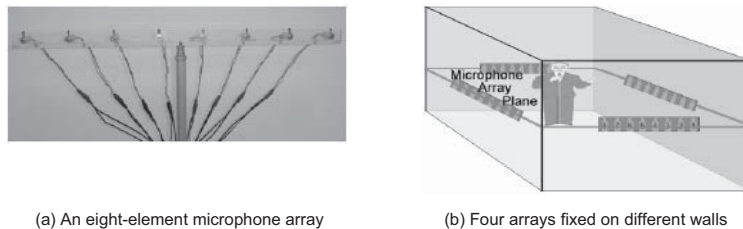


Fig. 9 Deployment of Microphone Array in Smart Classroom

been explored in [7] and [8], and we choose to use the PHAT-GCC (Phase Transform Generalized Cross-Correlation) method, which is relatively robust in the room environment. Consider two of the microphones m_i and m_j in an array, and let x_i and x_j respectively be the signal recorded at corresponding microphone. Then x_i and x_j can be modeled respectively as:

$$\begin{cases} x_i(t) = h_i(t) * s(t) + n_i(t) \\ x_j(t) = h_j(t) * s(t - \tau_{ij}) + n_j(t) \end{cases} \quad (1)$$

where $s(t)$ denotes the source signal, $h_i(t)$ is the acoustic impulse response between the sound source and the microphone i , $n_i(t)$ is the respective additive noise (assumed to be zero mean and uncorrelated with the source signal), and τ_{ij} presents the relative delay of the signal.

Then the PHAT-GCC function can be expressed as

$$R_{ij}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{X_i(\omega)X_j^*(\omega)}{\|X_i(\omega)X_j^*(\omega)\|} e^{j\omega\tau} d\omega \quad (2)$$

where $X_i(\omega)$ is the Fourier transforms of $x_j(t)$. The relative time delay τ_{ij} is estimated as the time lag with the global maximum peak in the PHAT-GCC function

$$\hat{\tau}_{ij} = \arg_{\tau} \max R_{ij}(\tau) \quad (3)$$

For any pair of microphones, the TDOA can determine a hyperboloid where the sound source is located. Theoretically, by grouping microphones into pairs and estimating the TDOA of each pair, the source location can be found at the point where all associated hyperboloids intersect. Normally, the sound is assumed to be in the far field of the array, and then a hyperboloid can be simplified to a cone, which indicates the DOA (Direction of Arrival) of the source. However, the simplification can introduce significant estimation error when the aperture of the array is large or when the source is relatively near the array. Since we can acquire precise location information of all the participants in the room from Cicada system, microphone arrays do not need to work out the speaker location directly. Instead we can simply use TDOA to associate the sound source with one of the participants. In this way, we can avoid the estimation error introduced by simplification, and improve the performance of the system.

3.3.2 Integration with Microphone Array and Location Tracking

Cicada and microphone arrays are integrated by a fusion algorithm to associate the sound source with one of the participants in order to determine who the current speaker is. We assume that each of the participants in the meeting room carries a CBadge provided by Cicada system, so every participant will have a distinctive ID and can be precisely tracked.

For each participant i tracked, we obtain a 3D position, X_i , which represents the location (x, y, z) of the CBadge in Cicada coordinate system. Since Cicada and microphone array coordinate system are aligned, (x, y) is approximately the participant's mouth position in microphone array plane.

Let s_j denote the two dimensional position of a participant in the microphone array plane, and let m_{i1} and m_{i2} respectively be the position of the microphones in pair i . If the participant s_j speaks, then the TDOA between the pair of microphones can be calculated as

$$\tau_i(s_j) = \tau_i(m_{i1}, m_{i2}, s_j) = \frac{\|s_j - m_{i1}\| - \|s_j - m_{i2}\|}{c} \quad (4)$$

where c is the sound propagation speed. We set up the associated statistical model as follow

$$p(\hat{\tau}_i | s_j) = N(\hat{\tau}_i; \tau_i(s_j), f(\sigma_\tau)) \quad (5)$$

where $N(-; \mu, \sigma)$ is a Gaussian density with mean μ and standard deviation σ , and $\hat{\tau}_i$ indicates the actual TDOA estimated by microphone pair i . σ_τ is the standard deviation obtained from a training TDOA dataset of a speaker at a known location, while $f(\sigma_\tau)$ is a linear function of σ_τ .

We can add up the probability density values from all pairs of microphones for the participant s_j , and normalize by m to get a better estimate

$$P(s_j | \hat{T}) = \frac{p(\hat{T} | s_j) P(s_j)}{\sum_{j=1}^n p(\hat{T} | s_j) P(s_j)} \quad (6)$$

The participant with the highest posterior probability is selected to be the speaker. The prior probability can also be set to be unequal according to the application scenario. For example, in a classroom, the teacher is the person that is normally speaking. Therefore, higher prior probability value can be assigned to the teacher.

We suppose that there is no simultaneous speech in the room, so each speech segment belongs to one specific participant. Then, the system needs to associate the sound source with a certain participant only at the beginning of each speech segment. In order to detect the speech activity, we adopt a simple and efficient speech detector, constructed by calculating the signal zero-crossing rate. A speech detector is implemented on each microphone channel respectively. If more than half of all the channels detect speech activity, the system will start the speaker association process.

3.4 Context-awareness in Smart Classroom

context-awareness enhances human-centric, intelligent behavior in a smart environment; however, context-awareness is not widely used due to the lack of mature design pattern to support context-aware applications. In the Smart Classroom, we

demonstrate a context-aware application called Smart Cameraman which is used to switch the live-video scene at the remote side based on the situational contexts, such as the teacher's gesture, context. Firstly, we present a formal context model, which combines first order probabilistic logic (FOPL) and web ontology language (OWL) ontologies, to provide a common understanding of contextual information to facilitate context modeling and reasoning about imperfect and ambiguous contextual information and to enable context knowledge sharing and reuse. At then, we introduce a probabilistic context reasoning method to deduce high-level knowledge from the low-level uncertain data from sensors, activators, or software agents.

3.4.1 Context Modeling Combined with FOPL and Ontology

A well-defined context model is an important key to access the context in any context-aware system [9]. For instance, Henricksen et al. [10] investigated the unified modeling language (UML) and entity-relationship (ER) modeling approach to represent context structures and properties. Gu et al. [11] presented an ontology-based context model to derive high-level contexts from low-level context data. The important context sources are captured from embedded sensors in a smart space environment which give uncertain, imperfect data. However, most of ontology-based context models fail to represent uncertainty, while logic-based context models fail to describe semantic relationships between context entities [9]. Here, the fundamental ontology-based and logic-based context models are combined in a first-order probabilistic logic to represent the basic context structure and construct a probabilistic inference mechanism, which combines the expressive power of first-order logic with the uncertainty context reasoning of probabilistic theory [13]. This shared understanding of specific domains gives context modeling which uses ontology and semantic web services to describe the concepts and relationships of context entities in smart environment [12].

first order probabilistic logic (FOPL) is used to represent the basic context structure which combines first order logic and probabilistic models in the machine learning community. The definitions of terminology, including Field, Predicate, Context Atom, and Context Literal are presented in the following.

- $Field \in F^*$, where a *Field* is a set of individuals belong to the same class, e.g., $Person = \{Qin, Shi, Suo\}$, $Room = \{Room526, Room527\}$
- $Predicate \in V^*$, where a *Predicate* indicates the relationship among the entities or the properties of an entity, e.g. *location*, *co-locate*.
- $ContextAtom \in A^*$, where *ContextAtom* is represented as $predicate(term, term, \dots)$ in which a *term* is a constant, a variable, or a function followed by a parenthesized list of terms separated by commas with a predicate acting on the terms. For example, $location(Qin)$ indicates *Qin*'s location, and $co-locate(Qin, Suo)$ indicates that *Qin* and *Suo* are located in the same place.
- $ContextLiteral \in L^*$, where *ContextLiteral* is represented as the form of $contextAtom = v$ in which *contextAtom* is the instance of *ContextAtom* and

v indicates the status of *contextAtom* or the value of the terms. For example, $location(Qin) = Room527$ indicates that *Qin*'s location is *Room527*.

Thus, the context knowledge is represented as the form of $Pr(L_1, L_2, L_3) = c$, where $L_1, L_2, L_3 \in ContextLiteral$, which indicates the concurrent probability of several *ContextLiterals*. For example, $Pr(location(Qin) = Room527, location(Suo) = Room527) = 0.76$ indicates that the probability of the fact that *Qin* is located in *Room527* while *Suo* is located in *Room527* equals 0.76. The structures and properties of this basic model are described in an ontology language to define the conceptual contexts in a rich semantic level. The basic context structure is represented using web ontology language (OWL). Two OWL classes are defined as *PriorProb* and *CondProb* as in the approach of Ding et al. [14] for representing probabilities. A prior probability of a context literal is defined as the instance of class *PriorProb*, which has the two mandatory properties *hasContextLiteral* and *hasProbValue*. A conditional probability of a context literal is defined as the instance of class *CondProb*, which has three mandatory properties *hasContextLiteral*, *hasProbValue* and *hasCondition*. An example of an FOPL statement is represented by an OWL expression in Fig. 10.

The purpose of the ontology-based context model is to formalize the structured contextual entities in smart spaces by making use of the ontology methodology to define the concepts and relationships of the context elements. The context ontology is divided into a core context ontology for general conceptual entities in the smart space and an extended context ontology for the domain-specific environment, e.g. the classroom domain. The core context ontology defines very general concepts for the context in the smart space that are universal and sharable for building context-aware applications. The extended context ontology defines additional concepts and vocabularies for supporting various types of domain-specific applications.

The core context ontology investigate seven basic concepts of user, location, time, activity, service, environment, and platform, which are considered as the basic and general entities existed in Smart Space as shown in Fig. 11. Part of the core context ontology is adopted from several different widely-accepted consensus ontologies, e.g. DAML-Time, OWL-S, etc. The instance of Smart Space consists of classes of *User*, *Location*, *Time*, *Activity*, *Service*, *Environment* and *Platform*.

```
Pr(L1|L)=0.76
<owl:CondProb rdf:ID="Pr(L1|L)">
  <hasCondition>L</hasCondition>
  <hasContextLiteral>L1</hasContextLiteral>
  <hasProbValue>0.76</hasProbValue>
</owl:CondProb>
```

Fig. 10 FOPL statement represented in an OWL expression

- *User*: As user plays an important and centric role in smart space applications, this ontology defines the vocabularies to represent profile information, contact information, user preference and mood which are sensitive to user current activity or task.
- *Location, Time and Activity*: Note that the relevancy among location, time, and user's activity facilitates the validation of inconsistent contextual information because these contexts might be sensed by various sensors with different accuracies. For example, it's obvious that the sensed context, Dr. Qin is having a meeting in the Room526 at 10:00 a.m., meets conflicts with another sensed context, Dr. Qin is having a meeting in the different Room527 at 10:00 a.m. By checking the activity's location from Dr. Qin's schedule, we can confirm which one is correct.
- *Platform and Service*: The platform ontology defines descriptions and vocabularies of hardware devices or sensors, and software infrastructure in a smart space. The service ontology defines the multi-level specifications of services that the platform provides in order to support service discovery and compo-

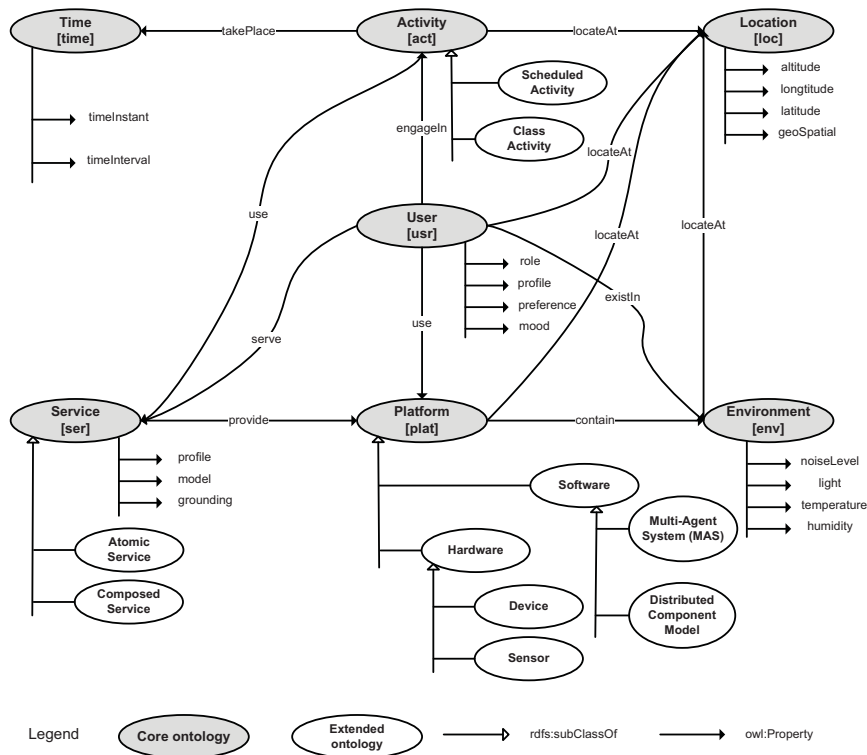


Fig. 11 Context Ontology

sition. In the Semantic Web community, OWL-S is adopted as a standardized framework to describe service in general.

- *Environment*: The environment ontology defines the context specification of physical environment conditions that user interacts with, such noise level, light condition, humidity and temperature, etc.

The extended context ontology extends the core context ontology, and defines the details and additional vocabularies of which apply to various different domains. For instance, in classroom domain, User has subclasses such Teacher, Local Student, and Remote Student who play the different roles and control rights in the class activities. The advantage of extended context ontology is that the separation of domain reduces the scale of context knowledge and burdens of context processing for pervasive computing applications, and facilitates the effective context inference with the limited complexity [14].

3.4.2 Probabilistic Context Reasoning

The context reasoning domain uses a rule-based inference mechanism which uses knowledge-based model construction (KBMC) to deduce high-level, new context knowledge from low-level detected facts. Within the FOPL framework, context rules are defined in as the form of $Pr(L_h|L_{b1}, L_{b2}, \dots) = c : \neg L_{C1}, L_{C2}, L_{C3}, \dots$, which means that the probability of L_h is c for the constraints L_{C1}, L_{C2}, L_{C3} and conditions L_{b1}, L_{b2} . Note that L_{C_i} denotes only the context fact and the others denote arbitrary Context Literals. For instance, in the classroom scenario, the statement $Pr(TeacherStatus(Teacher) = talking | Speaking(Student) = false) = 0.7 : \neg IsBlackBoardTouched(Room527) = false$ denotes the rules that when the blackboard of Room527 has not been touched, the probability that the teacher is talking equals 0.7 for the condition that the student is silent.

After representing the context rule, an additional property element *rdfs:dependOn* is defined to capture the dependency relationship between the datatype and the object properties in OWL. The importance of the FOPL-based context rule is that probability and Bayesian Network tools can then be used to reason with uncertain context. The benefit of the *rdfs:dependOn* element is used to translate the resource description framework (RDF) graph into the Bayesian network's direct acyclic graph (DAG). Each node of the DAG represents a ContextLiteral, with directed arcs between nodes representing causal dependencies between ContextLiterals. The scale of the DAG is reduced by constraints, including valid syntax rules, the independence hypothesis of a causal set that extends the causal independence definition[11], the average distribution hypothesis of the residual probability, and the conditional independence hypothesis. These present the generation of the unnecessary nodes in the net so as to minimize the scale of the DAG and ascertain the exclusivity of the answer distribution. Therefore, the constraints and hypotheses control the inference complexity within an acceptable range.

3.4.3 Case Study in Smart Classroom

For the smart classroom project, a smart cameraman module was designed to change the live-video scene to a situational context according to the class activity in a classroom by switching an array of cameras. By making use of context-aware services, remote students were able to focus their attention on relevant scenes on the client side. In this case, the context-awareness provided by the middleware captures the contextual information relevant to the user's activity and provides class activity clues to the smart cameraman module. The middleware also delivers customized video to remote students with various qualities due to the various capabilities of their computer or systems, such as display screen size or network bandwidth.

The smart cameraman scenario uses four types of context rules in the class activity:

- **Teacher Writing on the MediaBoard:** When the teacher is writing comments on the MediaBoard, the smart cameraman module may select a close-up view of the board, as shown in Fig. 12(a).
- **Teacher Showing a Model:** When the teacher holds up a model, the smart cameraman module may zoom in on the model as shown in Fig. 12(b).
- **Remote Student Speaking:** When a remote student is speaking, live video of the student may be delivered to other remote students.
- **Other:** In all other situations, the smart cameraman module may select an overview of the classroom shown in Fig. 12(c).

A predefined probability between 0 and 1 is attached to the context rules using basic structure of the first-order probabilistic logic partially shown in Table 3. When an event (e.g. the teacher writing on the mediaboard) occurs, the concurrent probability distribution of the camera's status is reconstructed according to the context rules, so that the camera tracks the live focus. A case generator shown in Fig. 12(d) has been developed to simulate a variety of situations and contextual information to test the functionalities of the smart cameraman module.



Fig. 12 (a) Teacher writing on MediaBoard; (b) Teacher showing a model; (c) Teacher having a discussion with local students; (d) Case generator

Table 3 Examples of context rules defined in the smart cameraman scenario

Context Rules	FOPL Formula
CR1	$Pr(IsStatus(Camera) = CLOSEUP_VIEW Action(Teacher) = WRITING, IsStatus(MediaBoard) = TRUE) = 0.8$ $:\neg OnTouched(MediaBoard) = TRUE$
CR2	$Pr(IsStatus(Camera) = CLOSEUP_VIEW Action(Teacher) = SHOWING) = 0.15$ $:\neg OnTouched(MediaBoard) = TRUE$
CR3	$Pr(IsStatus(Camera) = CLOSEUP_VIEW Action(Teacher) = SPEAKING) = 0.05$ $:\neg OnTouched(MediaBoard) = TRUE$

4 Large-Scale Real-Time Collaborative Distance Learning Supporting

4.1 Totally Ordered Reliable Multicast

Large-scale interactive applications have demanding requirements on underlying transport protocols for efficient dissemination of real-time multimedia data over heterogeneous networks. Existing reliable multicast protocols failed to meet these requirements due to following reasons: 1) most protocols presume the existence of multicast fully-enabled network infrastructure, which is usually not the case for current Internet; 2) protocols that are able to support multiple concurrent data sources only have limited scalability; 3) few of them have implemented an end-to-end TCP-friendly congestion control policy.

In TORM protocol, Multicast is used wherever applicable, and Unicast tunnels are created dynamically to connect session members located in separated multicast “islands”(refer to Fig. 13, where Reliable Multicast Server is called RMS for short and Reliable Multicast Client is called RMC for short). High scalability is achieved by organizing members of a session into a hierarchical structure, but in contrast to most of existing tree-based protocols, any number of concurrent sources are allowed to exist in a session. In order to support interactive applications that involve multiple

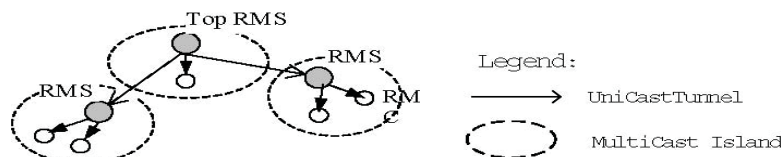


Fig. 13 Architecture of TORM

users cooperatively operating based on a shared state, TORM also incorporates two serialization algorithms, i.e., distributed one and centralized one, to ensure totally ordered message delivery through all members. Another eminent feature of TORM is a novel congestion control scheme that is able to response to congestions in a timely fashion and can fairly share bandwidth with other competing network flows. Here real-time measurements of reception rate of all receivers, packet losses, and variations of Round-Trip-Time are used as feedbacks for sender rate adjustments. Finally, network heterogeneity is fully addressed in TORM by partitioning receivers with different bandwidth capacities into homogeneous “domains” so that receivers behind bottleneck links are not likely to slow down the entire session. Trans-coding (AMTM) can also be applied on the border of domains to provide receivers with data in different qualities that best match their capabilities.

To testify the efficiency of TORM, we compared the loss recovery performance of TORM with SRM/Adaptive through software simulation (more specifically, with NS2). The recovery latency and duplicate-request number of these two protocols were tested with star, 2-clusters and random topology. The result shows that TORM has better performance than SRM as a reliable multicast protocol for large scale, real-time and interactive applications.

4.2 Adaptive Multimedia Transport Model

Adaptive Content Delivery (ACD) [15] bears the best efficiency among all possible types of Adaptive Multimedia Delivery schemes [16], in which application-layer semantics of the delivered data is coupled with underlying transport mechanisms. In Smart Remote Classroom project, the live instructing content and recorded courseware are in a compound multimedia document format. We developed AMTM to provide differentiated services for the delivery of this format of data to dynamically trans-code the multimedia data according to the variation of capabilities of each user as mentioned above without data redundancy.

The Multimedia Compound Document Semantic Model in AMTM can describe the data organization, PQoS (Perceived QoS) and transforming of compound documents with the notions of Media Object, Content Info Value and Status Space of Transforming respectively. A compound multimedia document is parsed as a structural description of embedded Media Objects as is shown in Fig. 14 (a). Fig. 14 (b) illustrates a typical transforming status tree, where the original media is labeled with Status0, and converted into the new node Status11 through operation1 with parameter1, while sub-node Status13 is transformed to newer nodes, namely Status21, Status 22, Status 23 and so on. The AMTM itself is used to abstract the process of adaptive delivery as finding the optimal resource allocation scheme and associated transformation plan for the embedded Media Objects by searching in the Status Space of Transforming. This policy is implemented in the RMS of TORM.

4.3 SameView: Real-Time Interactive Distance Learning Application in Smart Classroom

SameView is an application developed for real-time interactive distance learning based on the proposed TORM and AMTM platform. Fig. 15 shows a snapshot of a SameView client.

4.3.1 Interaction Channels

SameView provides a set of interaction channels for the teacher and students to efficiently achieve the goal of teaching and learning.

- Mediaboard**, which is a shared whiteboard capable of displaying multimedia contents in HTML format. The teacher can show slides for the class on the board. Moreover, he can add annotations or scribble on the slides on the fly. All actions the teacher makes on the board, such as jumping between slides, scrolling the slides and writing on the slides, will be reflected on each student's client program. When permitted by the teacher, a student can also take control of the Mediaboard, for example, write down his solution to a problem issued by the teacher.

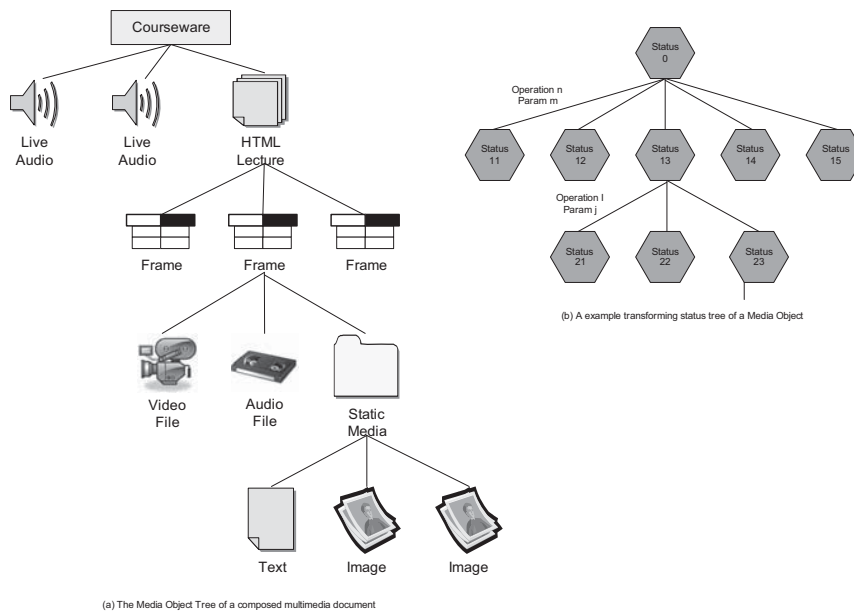


Fig. 14 Basic Notions in AMTM

- **Audio/Video.** The students can hear and see the audio and video of the teacher side. In addition, Student can also broadcast his audio and video to the others, when permitted by the teacher, for example, when the teacher asks him to give comments on a topic.
- **Chat.** In addition, teachers and students can communicate with text messages.

4.3.2 Session Management

The users participating a class through SameView will play different roles in the class. The possible roles are Chair, Presenter, Audience and Anonymous. The following chart shows their respective privileges.

Table 4 Privileges of Possible Roles in Class

Role	Change other user's role	Action on Mediaboard	Broadcast Audio/Video to others	User Number with this Role	Listed on Participant List
Chair	Allowed	Allowed	Allowed	1	Yes
Presenter	NA	Allowed	Allowed	≥ 0	Yes
Audience	NA	NA	NA	≥ 0	Yes
Anonymous	NA	NA	NA	≥ 0	No

The teacher usually plays the role of Chair, while students are always initially assigned to the role of Audience. As a class going, the teacher can dynamically change the role of a student as necessary, for example, to invite the student to give comments on a topic. If there is more than one participant broadcasting Audio/Video at the same time, only the one who speaks loudest will exceed.

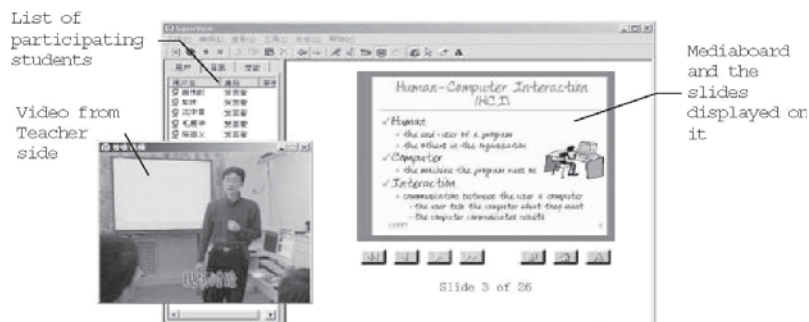


Fig. 15 A Snap of the SameView Client

4.3.3 Class Recording

SameView can capture the exact process of a class by recording everything happened on all interaction channels, such as the slides the teacher showed, the annotations made on the slides as well as the live video and audio, into a structured compound document. The recorded information in the document keeps synchronized in the time-line and a SameView Player program is provided to play back the document. This way, students can review the class any time they want. Actually, this recorded document is also a good type of courseware for E-Learning. In addition, we also provide a post-edit toolkit for the teacher to edit the recorded document as necessary, such as to correct some mistakes made on the class, or to add some tags and indices to the document which can help the student to use the document more efficiently. Fig. 16 shows the interface of the post-editing tool.

5 Smart Platform: Multiagent-Based Infrastructure

As an important notion in the vision of Pervasive Computing, smart environment is defined as an enhanced physical space where people can get the services of computer systems without approaching to computers or using the cumbersome keyboard or mouse to interact with them. Typically a smart environment is a distributed system that involves many sensors, perception devices, software modules and computers. To develop and support such a complicated system, some type of software platform is a must.

Smart Platform is just designed and developed as a solution for this software platform of smart environment which is used as infrastructure of Smart Classroom to integrate all the software components. In this section, we describes the features

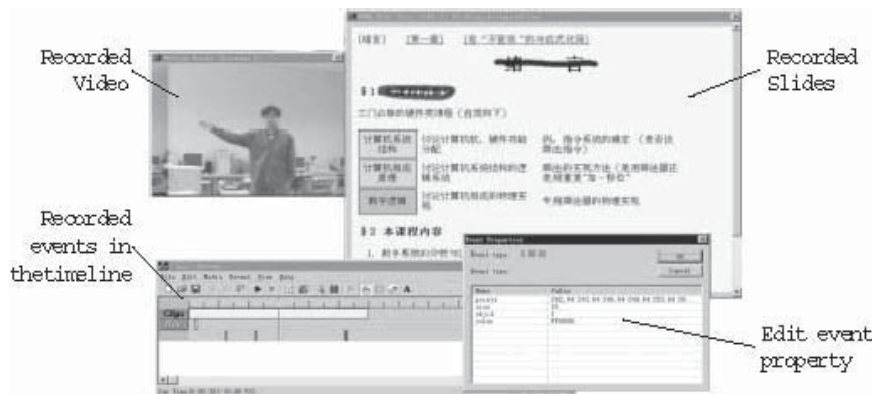


Fig. 16 Post-editing Tool for the SameView Recorded Class

and the architecture of Smart Platform, and provides a description of how to develop an agent on this platform.

We just assume a general familiarity with software engineering practices and programming language concepts. Some familiarity with the basics of the XML language is also helpful, as XML serves as the basis communication language throughout the whole platform.

The Smart Platform is modeled as a Multi-Agent System, in which the basic unit is agent. Each agent is an autonomous process that either contributes some services to the whole system or uses the service of other agents to achieve a specific goal. As a software platform, Smart Platform implements a runtime environment for the agents and an agent develop kit, which include a programming interface and some debug tools, for the developers.

5.1 Architecture

Smart Platform runs upon networks-connected computers in a Smart Space. Smart Platform masks the boundary of the involved the computers and provides a uniform running environment and highly structured communication model for the software modules run on the platform. This figure shows the architecture of the whole Smart Platform.

The runtime environment is composed of three kinds of components, which are Agent, Container and Directory Service(DS).

- An Agent is the basic encapsulation of the software modules of the systems.
- Each computer participating in the runtime environment will host a dedicated process called Container, which provides system-level services for and maintains the agents that run on the same computer. In some sense, Container is the mediator between agent and DS. It makes the low-level communicate details transparent to agent developers and provides a simple communication interface for agent.
- There is one global dedicated process called DS in the environment. The DS mediate the “Delegated” communications between agents and provide services such as agent query, dependency resolution.

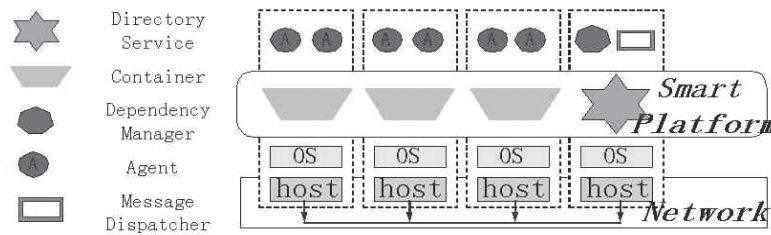


Fig. 17 Architecture and Components of Smart Platform

5.2 The Features and Design Objectives

5.2.1 Dynamically Find and Join

In the traditional way, if we want to access a service on the network, we must have the prior knowledge of the access address of the service, i.e. the IP address of the server and the port of the service on the server. While typically there will be dozens of agents in a decent Smart Space system, it is so tedious if we have to manually interconnect every agent. Instead, in the “find and join” mechanisms, Smart Platform uses IP multicasting to dynamically discover and assemble the computation environment, eliminating the need of manual configuration.

5.2.2 The combination of Delegated Communication and Peer to Peer Communication

In Smart Platform, we provide two different communication modes, “Delegated” and “Peer to Peer”. In “Delegated” mode, all the messages between agents are mediated by a special runtime module called DS. Agents post and accept messages without caring who is the other party. DS will forward the messages to the proper agent according to the contents of the message. Thus, the agents are significantly decoupled with each other. However, in this mode, the DS is burdened with a heavy load, so it is not suitable for the case where agents need high volume of data exchange. On the other hand, in “Peer to Peer” mode, the connection is established between agents directly, this is for the case where two agents need high bandwidth and real time communications. Here DS also plays an important role for helping the two agents to setup and maintain the communication channel.

Agents are free to select the proper communication mode according to their specific requirements.

5.2.3 Automatically Manage and Resolve of the Agent Dependencies

In a multi-agent system, agents must collaborate with each other. One agent may use the services provided by another agent. We called this relationship as agent dependence. The topological structure of the dependencies between agents may be a tree or even a net. The Smart Platform can manage and resolve those dependencies.

When each agent joins the computation environment, it must announce the services it provides and the services it depends. Smart Platform will store this information in a persistent storage. If an agent starts up asking for a service and the agent, which provide this service happens not to be in running, the Smart Platform will use the stored knowledge to locate the agents and automatically launch the agent. This feature is called “Agent Dependency Resolution”.

By this mechanism, it is no longer needed to manually start all the agents in a Smart Space system. Just start some core agents in a system; the whole system can be put into a determinate state.

5.2.4 XML based ICL (Interagent Communication Language)

ICL is the definition of the syntactic structure, and the semantic to some extent, of the messages between agents. We choose the XML as the basis of the syntax of the ICL in Smart Platform. The inherent advantages of XML benefit the Smart Platform in some aspects. (1) The extensibility and its ability to flexibly describe almost all kinds of data makes the ICL can be easily extended. (2) As the one of standard of technology of Internet, it eases the inter-operation of the Smart Platform with other heterogeneous systems. (3) There are many software libraries for the processing of XML in both industry and academe, which ease the development of the Smart Platform.

6 Conclusion

In Smart Classroom Project, we developed a set of key technologies for real-time interactive distance learning and make a new paradigm of real-time interactive distance learning with pervasive computing technologies: 1) Integrating various kinds of sensors (e.g. location tracking, microphone array), multimodal interaction technologies (e.g. speech recognition, direct manipulation, speaker tracking) and context-aware applications in order to bring ambient intelligence into a real traditional classroom environment; 2) Unifying the face-to-face education and tele-education with the Smart Classroom, which in one hand give a consistent teaching experience to teachers and in other hand reduce teachers's workforce needed, for the teacher do not need to give the same class for the on-campus students and remote students separately. 3) Able to accept large-scale users to access the virtual classroom simultaneously with different network and device conditions. 4) The class can be recorded and turned into a piece of courseware for E-learning.

References

- [1] Satyanarayanan, M.: Pervasive computing: vision and challenges. *IEEE Personal Communications*. (8), pp. 10–17. (2001)
- [2] Abowd, G.D.: Classroom 2000: An experiment with the instrumentation of a living educational environment. *IBM System Journal*. (38), pp. 508–531. (2000)

- [3] Shi, Y.C., Xie, W.K., Xu, G.Y., Shi, R.T., Chen, E.Y., Mao, Y.H., Liu, F.: The smart classroom: merging technologies for seamless tele-education. *IEEE Pervasive Computing*. (2), pp. 47–55. (2003)
- [4] Miu, K.L.: Design and implementation of an indoor mobile navigation system. (2002)
- [5] Welch, G., Bishop, G.: An introduction to the kalman filter. In: *Tutorial of SIGGRAPH*. pp. 1–81. (2001)
- [6] Grewal, M.S., Andrews, A.P.: *Kalman filtering: theory and practice*. Prentice-Hall Inc, New Jersey, USA. (1993)
- [7] Knapp, C.H., Carter, G.C.: The generalized correlation method for estimation of time delay. *IEEE Transaction on Acoustics, Speech and Signal Processing*. (4), pp. 320–327. (1976)
- [8] Brandstein, M.S., Silverman, H.F.: A practical methodology for speech source localization with microphone arrays. *Computer, Speech, and Language*. (11), pp. 91–126. (1997)
- [9] Strang, T., Linnhoff-Popien, C.: A Context Modeling Survey. In: *Proceedings of UbiComp Workshop on Advanced Context Modeling, Reasoning and Management*. (2004)
- [10] Henriksen, K.: A framework for context-aware pervasive computing applications. (2003)
- [11] Gu, T., Pung, H.K., Zhang, D.Q.: A service-oriented middleware for building context-aware services. *J. Network and Computer Science*. 18, 1–18. (2005)
- [12] Guarino, N.: Formal ontology and information systems. In: *Proceedings of FOIS*. pp. 3–15. (1998)
- [13] Poole, D.: First-order probabilistic inference. In: *Proceedings of IJCAI*. pp. 985 – 991. (2003)
- [14] Ding, Z.L., Peng, Y., Pan, R.: Soft computing in ontologies and semantic web. (2005)
- [15] Zhang, H.J.: Adaptive content delivery: a new research area in media computing. In: *Proceedings of the 2000 International Workshop on Multimedia Data Storage, Retrieval, Integration and Applications*. pp. 13–15. (2000)
- [16] Clark, D., Tennenhouse, D.: Architectural considerations for a new generation of protocols. In: *Proceedings of SIGCOMM*. (1999)