Intelligent Conference Room System Model Based on Agent Driver

Jiang Jiang1,*, Jianyong Wang2, Xueliang Zhang3
1Guangdong Power Grid Co., Ltd., Guangzhou Guangdong 510699 China
2Guangdong Electric Power Information Technology Co., Ltd., Guangzhou Guangdong 510062 China
3Guangzhou Yueneng Information Technology Co., Ltd., Guangzhou Guangdong 510062 China

*Corresponding author: houyao_jy@126.com

Abstract. The traditional conference room system in the function realization is relatively single, more and more can not meet the growing needs of people. This paper designs and implements an intelligent conference room system based on agent driver. The intelligent conference room adds conference room management function. Add user rights, status management, information publishing and conference room locking functions to the prototype conference room system. At the same time, the system adds intelligent video playback function. Through red5 streaming media server and Xuggle Codec, the video information of different formats is converted into RTMP format information and sent to all clients. This realizes the video file sharing and playing function. The system also realizes the access function of holographic monitoring system. Through the cross search between the intelligent conference room system and the database of the monitoring system, the user can customize the camera list function. The system uses proxy server to realize the seamless connection between the intelligent conference room system and the monitoring system, so as to realize the access function of the monitoring system. The experimental data show that the intelligent conference room can improve the recognition of conference information. This provides a reference for the design and implementation of adaptive speech control intelligent conference room.

Keywords: Agent Driven, Intelligent Conference Room, Shared Play, Monitoring Access

1. Introduction
In the process of pervasive computing, the computer system will seamlessly integrate into every user's daily life and provide them with information and services anytime and anywhere [1-2]. In this open and dynamic environment, intelligent information processing entities must realize the functions of knowledge sharing, reasoning to the surrounding environment and service interoperability [3].

At present, many research architectures have developed their own prototype systems and architectures, such as using context aware handheld devices to create personalized guidance for
museum visitors [4-5]. When users enter the office, they can dynamically migrate the applications of mobile devices (such as mobile phones or PDA) with limited resources and low processing power to nearby desktop computers with rich resources and strong processing power. However, this kind of information migration is time-consuming and laborious [6]. Because there is no global control and coordination mechanism for these devices, each device is basically in the "information island" state, so it can only provide weak support for knowledge sharing and reasoning, which is difficult to meet the requirements of freely exchanging information and sharing knowledge with intelligent devices as the carrier in the intelligent space environment [7-9]. In order to realize intelligence in pervasive computing, we should improve the efficiency of data processing and reasoning ability of previous systems. Through a new model of smart meeting room (SMR), this paper discusses how to realize knowledge sharing, context reasoning, detect and solve the problem of context inconsistency in the intelligent space environment.

2. The structure of SMR

R101 is an intelligent conference room, its entrance is equipped with RFID sensor, which can check attendance of participants. When Tom enters the conference room, these sensors inform the context broker Center (CBC) of R101 that Tom has entered the conference room by acquiring the information on his mobile phone, and then CBC adds the fact to the knowledge base.

When Tom sits down, the agent in his mobile phone will find the CBC of R101 through the Bluetooth device, and then inform the CBC of the owner's personal information control strategy, so that the mobile phone and the agent can establish communication by using handshake protocol (such as establishing trust after verifying the agent's identity). The following control strategy describes what Tom requires the agent center to do: (1) inform CBC whether to allow other devices to access his context information (such as allowing access to his location and role, not allowing access to the phone number he dials). (2) When his context information changes, CBC should inform other agents (such as his location information to his personal agent at home). (3) Allow other agents to get his context information (for example, when he is in the conference room, all other agents in the conference room can get his context information).

After receiving Tom's personal information control strategy, CBC creates a profile for him. When CBC processes his related contextual knowledge information, it will abide by the rules and constraints of the profile. For example, under the above control strategy, Tom's description information will guide the agent to: (1) obtain and infer the context information of Tom's location and action; (2) inform Tom's personal agent at home when Tom's context information changes; (3) agents in the conference room share his context information.

Knowing that Tom's mobile phone is now in R101, and there is no contrary fact, the agent infers that Tom is also there. In addition, because R101 is a part of the computer building and the computer building is a part of the school, the agent infers that Tom is in the computer building and also in the school. These conclusions are added to the agent's knowledge base. According to its description information, CBC contacts Tom's personal agent [10]. Because Tom has entrusted his personal agent to manage today's work schedule, his personal agent knows that Tom will make a meeting report in the school during this period, and the personal agent will inform CBC of R101 of the fact.

After receiving the information about Tom's intention, CBC of R101 will inform projector agent and lighting control agent of R101 of this information. After a few minutes, the projector agent downloads the slides he wants to use from Tom's home page. The light control agent dims the light, and Tom can start the meeting report.

3. Shared ontology for pervasive computing

Owl is a semantic web language used by computer applications. As a part of semantic web, owl is developed by W3C. Owl is a representation language for defining and instantiating ontologies. It not only displays information for users, but also helps computers understand and process information.
To realize knowledge sharing and context reasoning, we designed a set of shared ontology (so4pc) for pervasive computing. The advantage of SO4PC is that it provides a shared ontology for the developers of pervasive computing, which brings together many useful words from different consensus ontologies. We believe that SO4PC can not only be competent for SMR, but also help developers who are not familiar with knowledge representation to quickly build ontology applications without redefining ontology, so that they can concentrate more on system implementation.

Ontology referenced by SO4PC includes Friend2Of2A2Friend ontology (FOAF), DAML2Time, spatial ontology of OpenCyc, RegionalConnectionCalculus, RCC (RCC), COBRA2ONT, MoGATUBDI ontology and Rei policy ontology. SO4PC consists of two parts: SO4PC core and SO4PC extension.

SO4PC core ontology set defines general vocabulary, which is applicable to different pervasive computing applications. The SO4PC extended ontology set expanded from the core ontology defines extra vocabulary to support special types of applications, and provides examples for future ontology expansion. Ontology set consists of words representing concepts, which are related to people, Agent, belief-desire-intention (BDI), actions, strategies, time, space and events. SO4PC contains 9 different ontology files. Figure 2 shows ontology files and their interrelationships.

4. Contextual reasoning

SMR supports two types of context reasoning: (1) the reasoning of establishing context sharing model; (2) the reasoning of maintaining context consistency model. The first type of reasoning includes using reasoning to explain awareness information, and the second type includes using reasoning to detect and solve inconsistent information.

4.1. Methods of context reasoning

Context reasoning is very important for every context aware system. The function of context awareness system depends on the support of context reasoning. In the early system, the typical context reasoning was realized by using Java and C++. These object-oriented languages support code reuse on API, but they lack expressive power for knowledge representation. When using these languages for context reasoning, the non declarative representation of logical reasoning often makes code modification and error checking difficult. In order to solve these problems, we design a logic reasoning method based on rules. The advantages of this method are as follows:

(1) An explicit representation of context reasoning rules helps to separate high-level reasoning logic from low-level function implementation. Through this partition, developers can modify or replace some components of context reasoning without rewriting a large number of programs;

(2) The rule-based approach allows many defined logical models of general concepts to be mapped directly to context reasoning implementations, such as time and space. This kind of logical reasoning related to general concepts is often used in context reasoning;

(3) When context interpretation rules are explicitly expressed, meta reasoning technology can be used to detect and solve the inconsistency of context interpretation.
4.2. Time reasoning
When describing events, time is a central concept. In SMR, it can include the statistics of meeting attendance, the realization of behavior, the sending and receiving of information between agents, the change of service and internal state of equipment, etc. It takes time to describe these events, so CBC can deduce the time sequence relationship between them. These logical conclusions for time reasoning are based on SO4PC-based time ontology.

4.3. Spatial reasoning
Spatial information is an important part of context information. The logical inference of the position information of an object is called spatial inference. In SMR, spatial reasoning is not only to know the location of physical objects, but also to use geospatial data and ontology to obtain new spatial knowledge that cannot be obtained directly by sensors.

In SMR, SO4PC spatial ontology is the core of spatial reasoning. These inferences assume that all spatial information can be used to represent SO4PC spatial ontology. Based on ontology semantics,
the following logical inferences are supported: (1) Given geographical locations X and Y (which are examples of spc: GeographicalSpace class), infer whether Y contains X or whether X contains Y; (2) Given two geographical locations X and Y, infer whether they are related. (3) If Y contains X in space, deduce the geographical position of Y according to the geographical position X with defined longitude and latitude.

Figure 4 Reasoning with spatial ontology

Inference infers the spatial attributes of a location that cannot be obtained directly from the sensor. For example, from the information of personal PDA read from GPS equipment, CBC can infer some basic information about the position of this person (such as latitude and longitude); (2) Spatial reasoning can be used to detect inconsistent information about personal position. For example, according to the defined ontology, room A and room B are not connected in space, but two different information obtained by CBC indicates that the same person is in both room A and room B in the same time interval. Using spatial reasoning, it can be inferred that a person cannot appear in two unconnected rooms at the same time. Therefore, it is concluded that the location information is inconsistent. Thirdly, by coordinating the granularity of user location information, CBC can use spatial reasoning to protect their personal information. For example, based on a set of defined granularity rules of user location information, CBC can generalize a person's location information, which can be shared by Agent that the person does not trust, thus sharing necessary information and protecting the personal privacy of the person at a proper granularity.

4.4. Detect inconsistencies

OWL language can be used to detect inconsistent information, that is, context ontology modeling, and ontology reasoning can be used to detect inconsistent information. OWL is a representational language, which can represent different aspects of context. This includes the objects in the context, the relationships between these objects, and the constraints on these relationships. Once the ontology is defined, the observed and acquired context information can be represented as the data instance of the ontology. Given an ontology and some case data, ontology reasoning can be used to infer whether the context described by case data is inconsistent with the model defined by ontology.

Assume the following situation: Tom enters the conference room (R101). This conference room
detected the Bluetooth device in his mobile phone. Since there is no other evidence to the contrary, the agent infers that Tom is also in the conference room and informs CBC. A moment later, Tom left the conference room in a hurry, but left his cell phone there. After leaving the conference room, Tom returned to the office (R201). When he entered the office, the voice recognition equipment installed in his house recognized his voice and informed CBC that Tom was in his office. According to spatial ontology knowledge (as shown in fig. 4), R201 and R101 are two independent rooms in space. According to this information, and there is no evidence that Tom has left any room, CBC concludes that there are inconsistencies in Tom's location information.

4.5. Using logical reasoning to solve inconsistency

Information inconsistency is caused by the source agent that provides the information. Because these agents may have used inaccurate sensor data or defective domain inference to interpret context, the information they get may be inconsistent. The case described in Section 4.4 is a typical example, which shows that the information inconsistency is caused by the limitation of source agent detection and inaccurate reasoning. We developed a logical reasoning method based on hypothesis to solve the inconsistency of information. In this approach, contextual CBC measures the credibility of the assumptions used by agents to arrive at their conclusions to get the most reliable assumptions.

We use logical reasoning system to realize modeling and reasoning about hypothesis. In the implementation, CBC has an explicit model about the hypothesis, which is used by other agents to explain the context. When CBC detects inconsistent information, it dynamically obtains hypotheses by communicating with other agents, and then uses logical reasoning based on hypotheses to solve the problem of inconsistent information.

In the implementation, the reasoning system is designed as follows

(1) It is assumed that all CBSCS use the same ontology. CBC does not need to detect and solve the problem of inconsistent information caused by inaccurate ontology specification;

(2) All agents share a common ontology with CBC. This general ontology includes vocabulary and related semantics used to describe context and model reasoning hypothesis;

(3) When CBC requires agents to prove their trust in context, all agents should not intentionally provide error information to deceive CBC. In other words, there are no lying agents, and all agents cooperate with each other;

(4) CBC is only responsible for solving all kinds of inconsistent information, and it has enough relevant knowledge to solve these inconsistencies. The internal logic inference of CBC defines this knowledge first.

The reasoning method is described in Section 4.4 When detecting Tom's inconsistent location information, CBC queries the corresponding sensor to prove its hypothesis: (1) query monitor S1 to get tom's cell phone (moto) (2) query the sensor S2 to get the noise of R201. Through the query results of S1, CBC knows that Tom's mobile phone has 8 calls not answered, and no calls have been made in recent time. Through the query result of S2, CBC knows that the noise level of R201 is 50db, which belongs to the normal range. In this way, CBC confirms that the assumption of S2 is more reliable, and the inconsistency problem is solved.

![Figure 5 Solving inconsistencies with hypothesis based reasoning](image-url)
5. Conclusion

The SMR model demonstrates the following feasibility: (1) using OWL ontology can make distributed agents share knowledge; (2) logical reasoning of context information; (3) using policy language to control their privacy. However, there are still many problems to be solved to implement the SMR model, including the measurability of knowledge sharing in a distributed and dynamic environment. The performance and time complexity of context reasoning, as well as the user interface problems related to the control policy of editing and maintaining user privacy, are discussed when a large number of detection data appear. It will further expand context awareness support, including tracking the location of absentees, tracking the availability of portable projection devices, etc.

References

[1] Niu Xin. Research on the Design of Intelligent Conference Room. Henan Science and Technology, 2018, 1: 40-41
[2] Chen Han. Environment Intelligent Control Technology of Conference Room and Conference Hall. Intelligent Building, 2008 (11): 49-52
[3] Cai Qingong. Improvement of Lighting System for Multi Function Conference Room in Hotel. Electronic World, 2020, No. 583 (01): 159-160
[4] Fan Gongrui, Huang Siqi, Liu Wensheng. Audio and Video System Design of Multi-function Hall. Intelligent Building, 2015, No. 182 (10): 71-76
[5] Zhong Guojian. Application and Development of AI in Smart Park -- Exploration and Construction of Intelligent Conference Room. China Public Security, 2019, 9: 39-41
[6] Huang Yuanyuan, Chen Chaofan, Cheng Zhigang. Design and Implementation of Intelligent Conference Room Reservation System Based on Internet of Things. Computer Knowledge and Technology, 2018, V.14 (20): 68-70
[7] Lu Xiuqiong. Small Intelligent Multimedia Conference Room Solutions. Digital Communication World, 2016, 3: 106-107
[8] Wang Xiaocheng. Discussion on System Integration Design of Multimedia Intelligent Conference Room. Neijiang Science and Technology, 2013 (01): 164-165
[9] Hong Yuan, Yin Dandan. Application and Value of Intelligent Voice Technology in Adaptive Speech Control Intelligent Conference Room. Intelligent Building, 2016, 12: 35-36
[10] Wen Xianli. Application of Mobile Platform Technology in the Design of Intelligent Conference Room Reservation System. Information Technology and Informatization, 2019, 2016, 42 (019): 30-31