SLAP: a Location-aware Software Infrastructure for Smart Space

Hongliang Gu, Yuanchun Shi, Guangyou Xu, Weisheng He, Baopeng Zhang
Key Laboratory of Pervasive Computing, Ministry of Education
Department of CS, Tsinghua University, Beijing, P.R. China
{ghl02@mails, shiyc@, xgy-dcs@mail, hws99@mails, zbp02@mails}.tsinghua.edu.cn

Abstract

In this paper, aiming at the deficiencies of Open Agent Architecture (OAA) on accommodating the location-aware computing, we propose an improved software infrastructure for Smart Classroom (a kind of Smart Space): SLAP (Smart Location-Aware Platform), which can not only offer an architecture and mechanism supporting the location-aware computing, but also cater for a more efficient communication performance than OAA on the wireless network.

1. Introduction

A Smart Space [1] (or Intelligent Environment), which integrates large numbers of distributed hardware and software into a physical space, including lots of positioning sensors (or location-sensors), and mobile computing units, is an important application field of the ubiquitous/pervasive computing. As a component of the Smart Space, the software infrastructure of Smart Space (SISS), which takes charge of coordinating and managing numbers of hardware and software modules, is the underlying and crucial part of the Smart Spaces.

The location-aware computing is a significant characteristic of the ubiquitous computing. In our project, the location-aware computing means that the applications or services can accord to the location of located-objects [2] to modify their own behaviors non-intrusively to adapt to the users' purpose.

Our project, Smart Classroom [3], is a Smart Space on the field of tele-education. In the past, Smart Classroom, which focuses on implementing a multi-modal Human-Computer-Interaction (HCI), being hardly related to the location-aware applications, adopts the Smart Platform[4], a ramification of Open Agent Architecture (OAA [6]), as its SISS. At that time, OAA is competent. However, with joining in of mobile computing units and positing system, and more and more location-aware applications emerging, when the supporting the location-aware computing becomes one of the SISS' necessary functions, OAA becomes hardly competent for the role. For this reason, we, on the basis of the improvement of OAA, develop a new SISS of Smart Space: Smart Location-Aware Platform (SLAP), which can not only offer an architecture and mechanism supporting the location-aware computing, but also cater for more efficient performance of communication than OAA on wireless network.

The contents below are laid out as follows: Section 2 gives an overview of the location-aware computing in Smart Classroom. Section 3 introduces the working principle of OAA, and analyses the deficiencies of OAA on supporting the location-aware computing. Section 4 presents our solution to location-aware SISS: SLAP. Section 5 gives the relevant experiment. Section 6 contains our conclusion.

2. Location-aware computing components in Smart Classroom

In Smart Classroom, the location-aware computing consists of three components: position system, Location Server and location-aware applications, which all as the modules run on the SISS. The position system is the Cricket [5], in which each unit knows its own geometric coordinate location and then sends its location to the Location Server by a wireless network. The applications demand that the system can accord to their locations (or spatial relationship) to establish or cut off their communication. For example, the Smart Board (a large-sized touch screen), once a PDA enters its service scope, needs the PDA to send it the user's interactive activities so that it can change the display in synchronization with the PDA. That is, only when the communication's spatial condition that the modules declare is met, the communication (or interaction) between them can arise, which is one of the requirements of location-aware computing on the SISS.

Proceedings of the Third IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (SEUS'05)
0-7695-2357-9/05 $20.00 © 2005 **IEEE**

3. Analysis on deficiencies of OAA

3.1. Working principle of OAA

AS a SISS, OAA is a multi-agent system from SRI, in which every module (including applications, services) is encapsulated into an agent. The communication data between agents is named as message. The message communication adopts Publish-Subscribe mechanism (so-called *delegated computing* in OAA). In this mechanism, on the one hand, the agent which wants to handle a certain kind of message, registers the name of message on *Facilitator*, which is called subscribing message; on the other hand, the agent which tends to send messages for other agents to handle, only sends Facilitator the messages tagged with message's name possibly subscribed by other agents, which is called publishing message. Facilitator, as a forwarding center, takes charge of according to messages' name to dispatch messages to every agent which subscribed the messages.

3.2. Deficiencies of OAA on location-aware computing

Though the Publish-Subscribe mechanism of OAA acquires the advantage of inter-module loosely coupling, it has also the following conspicuous characteristics:

1) Publish-Subscribe freely, inducing difficulty in controlling in some conditions.

In this mechanism, the message-sending agent can publish its messages utterly freely. What time it publishes, how many messages it publishes are all determined by the agent itself. Considering the following situation in Smart Classroom: the Location Server (also as an agent) wants to accord to a certain policy to obtain a user's location from the position agent (the agent is dedicated to obtaining the location from the Cricket card and sending out the location), namely to update its location data. For example, according to a certain updating policy at a certain time, the Location Server needs the position agent carried by a person to publish its location messages at the frequency of twice per 1 second, and another carried by a device to publish its messages only once per 5 minutes, for the device's location much more seldom varies than the person. However, this updating policy is difficult to be carried out in this mechanism, for the right of publishing message is whole in control of the senders, namely the position agents. Additionally, those published location messages which the target agent (the Location Server) does not need, not only occupy the bandwidth but also aggravate the contention in the wireless network, which causes a longer latency and a lower efficiency.

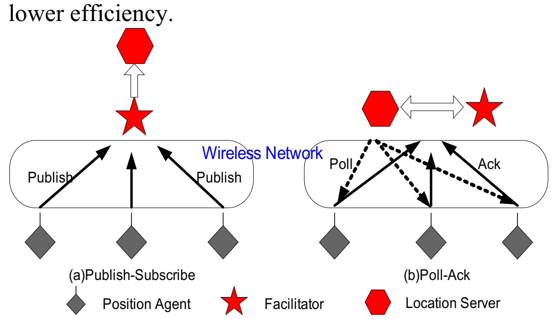

Figure 1. Communication between the Location Server and the position agents

2) The messages are irrelevant to the application's location.

In OAA, the system only accords to the message's name or type to dispatch. In this mechanism, the message's communication between agents is from-start-to-finish. That is, after subscribing a certain kind of messages, an agent will keep on receiving the messages until it unsubscribe the message. This period is not interrupted or recovered by the varying of its location. In OAA, it is difficult to be implemented that the system accords to an agent's location to determine what messages and when to send to it, which is just what the location-aware computing needs. Therefore, the pure Publish-Subscribe mechanism of OAA, which is irrelevant to location, doesn't satisfy the requirement of location-aware computing any longer.

4. Our solution

Aiming at the deficiencies of OAA on location-aware computing discussed above, we present our solution: Smart Location-Aware Platform for Smart Classroom (SLAP). Here we first introduce our inter-agent communication mechanism solving the deficiencies of OAA, then bring in the architecture of SLAP, which involves some details of implementation.

4.1. Poll-Ack mechanism

As discussed above, on the communication between the location-server agent and the position agents, the Publish-Subscribe mode renders a low performance. Aiming at this issue, we present a new message communication mode, called Poll-Ack mechanism. This mechanism is described as follows:

As Figure 1(b) illustrates, the master agent (i.e. the Location Server) emits a broadcast/multicast called

Poll message, which indicates the next transmitting agent. During the emitting of Poll, those agents, including the servant agents and the other agents (e.g. Facilitator) wanting to obtain the message from the servant agents, will hear the Poll message. On receiving the Poll message, the servant agent (i.e. the position agent) whose name is indicated in the Poll message, will reply an acknowledgement called the Ack message, which carries the location information, in a fixed time. Those agents, including the master agent and the agents wanting to obtain the message from the servant agent, will get the Ack message .When the master agent receives the Ack message it expect, it begins next Poll-Ack cycle. Otherwise, if having not received the expected Ack message during a certain maximal period of time, the master agent either repeats the Poll or skips it in the next cycle according to its certain policy.

4.2. Subscribe-Publish-Notify mechanism

As mentioned above, the cause that the messages are irrelevant to the location and Facilitator does not care the agent location's varying, induces another deficiency of OAA on location-aware computing. Aiming at this issue, we bring forward a location-dependent message mechanism: Subscribe-Publish-Notify. The working principle is narrated as follows:

First, when an agent subscribes a group of messages, it must simultaneously declare the spatial condition of the messages. Only when the spatial condition is met, the relevant messages are really published to the agent. Those spatial conditions are ultimately submitted to the Location Server.

Then, the Location Server accords to the varying of agents' location to emit the notifications to Facilitator, which are to explain whether and when the message group's spatial condition is met. Facilitator accords to the notifications to filter the relevant messages' publishing. If being necessary, Facilitator will forward the notifications to the target agents, so as to notify them to take the relevant activities, e.g. sleep, wake up, or prepare for receiving the message stream etc.

4.3. Architecture of SLAP

Here we give a detailed description of the architecture of SLAP, which involves our other accessorial improvements on OAA besides the two mentioned above. The architecture is shown in Figure 2.

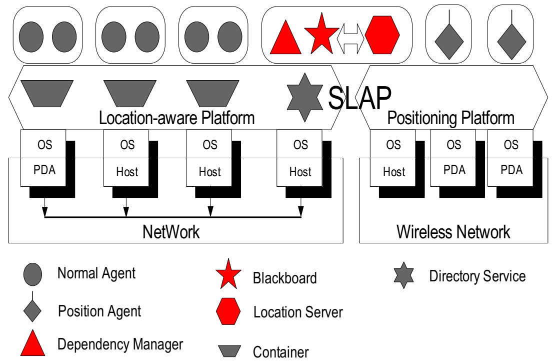

Figure 2. The architecture of SLAP

In general, SLAP, acting as a middleware between OS (Operation System) and applications (running as agents), is composed of two parts, corresponding to two categories of inter-agent communication mechanism. The right part, called the positioning platform, which is in the situation of wireless network connecting the position agents and the Location Server, adopts the Poll-Ack mode. And the left part, called the location-aware platform, which is used for communication between the non-position agents (the normal agents), adopts the Subscribe-Publish-Notify mode. Some components in SLAP are specified as follows:

The dependency manager takes charge of the maintenance and auto-setting-up of the service dependent relation between agents. The Blackboard is a message center, playing the role of Facilitator to take charge of dispatching all messages with the collaboration of the Location Server. And the Location Server takes charge of the collection of objects' locations and disseminating of all notifications. Those components all run as global dedicated processes respectively. The containers act as the mediators under the agent layer, used for shielding the heterogeneous OS. Since employing the same OS, the PDAs of position system have no container. The directory service is used for the discovery of services in SLAP.

5. Experiments

To evaluate the performance of SLAP compared with OAA, we conduct a simulating experiment. This experiment is to measure the messages' delivery latency between the Location Server and the position terminals (where the position agents run) in wireless network. We adopt a PC outfitted with a wireless network adaptor (Lenovo Freeblue LSC-600C [7], complying with IEEE802.11b), as the Location Server, and take 5 or 20 laptops equipped with the same wireless network adaptors, as the position terminals. The experiment is divided into 2 groups, one of which is a PC with 5 laptops, another of which is a PC with 20 laptops. In each group, both the Publish-Subscribe mode of OAA and the Poll-Ack mode of SLAP are implemented to acquire the comparable results. In every laptop, a process simulating position agent emits messages, with a fixed size of 512 bytes, according to Poisson distribution with the expectation equal to average message rate per second. In the PC, a process takes charge of receiving all messages, and computing their arrival delay (or access delay). Another process, also running in PC, emits broadcasting messages with a fixed size of 256 bytes, at a fixed frequency. That process is used to simulate the Polls. Over 10 minutes' recording for each group simulation, we got the statistical data at last. The result is shown in Figure 3.

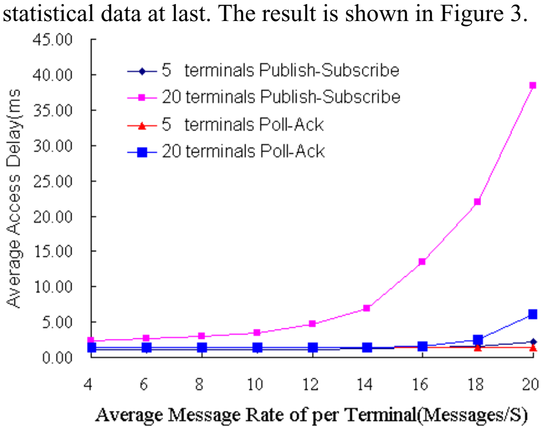

Figure 3. Performance of SLAP versus OAA

As Figure 3 shows, in comparison with the Publish-Subscribe mode of OAA, the Poll-Ack mode of SLAP reduces average access delay. Either more terminals or higher average message rate of per terminal leads to more obvious improvement. The Poll-Ack mode retains an approximate equal-time delay for per-message, except that there is an abrupt increase at the message rate of around 16 messages /S in the latter group of 20 terminals. I guess the cause should be that, at that point the throughput of the Poll-Ack mode is near to saturation. And we also deem that the performance improvement of SLAP mainly results from the decreasing of collision and backoff in the network. Further, we can reason out, if a Poll message becomes shorter relative to a normal message, a better improvement could be achieved.

6. Conclusion

In this work, on the basis of OAA, we present a new SISS, SLAP, which mainly presents two improvements: the Poll-Ack and the Subscribe-Publish-Notify inter-agent communication mode. The former is more suitable for the position system environment than OAA, due to the flexibility to control and higher efficiency which the simulating experiment proves. The latter makes the messages relevant to the agents' location and the inter-agent communication in control of their spatial relationship, which OAA can hardly afford. Both the advantages and the compatibility with OAA make SLAP a promising SISS supporting the location-aware computing.

Acknowledgements

This research is supported by NCET(Program for New Century Excellent Talents in University, China) and NSFC(Natural Science Foundation, China)

References

[1] http://www.nist.gov/smartspace/

[2] B. Schilit, N. Adams, and R. Want: 'Context-aware computing applications', IEEE Workshop on Mobile Computing Systems and Applications, IEEE CS Press, 1995, pp. 85-90

[3] Y. C., Shi, et al.: 'The smart classroom: merging technologies for seamless tele-education', IEEE Pervasive Computing, IEEE CS Press, 2003, 2, (2), pp. 47-55

[4] W. K. Xie, et al.: 'Smart Platform: A Software Infrastructure for Smart Space (SISS)', Proc.4th Int'l Conf. Multimodal Interfaces (ICMI2002), IEEE CS Press, 2002, pp. 429–434.

[5] Nissanka B. et al.:'The Cricket Compass for Context-Aware Mobile Applications', Proc. 7th annual int'l Conf. on Mobile computing and networking, ACM Press, 2001, pp.1-14

[6] Adam Cheyer, David Martin: 'The Open Agent Architecture', Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publisher, 2001,4, (1-2), March, pp.143-148

[7] Lenovo,Inc., web site: http://www.lenovonetworks.com/