

Towards delivery of situational information in distributed environments

Chengqiang Sun (孙成强), Yongseong Cho, Jongsun Choi, Jaeyoung Choi
(*School of Computer Science and Engineering, Soongsil University, Seoul 156-743, Korea*)

Abstract: In the distributed environment, robots should be able to provide users with adaptive services automatically according to the situational information changing dynamically which is obtained from both users and their environments. The workflow depends on situational information obtained from physical environments and provides context-aware services automatically based on the information retrieved. And the workflow in the business processes and the distributed computing environments have supported the automation of services by connecting tasks. The workflow needs to specify ubiquitous situational information as state-transition constraints. However, the delivery and use of sensor data in the workflow is a difficult problem for the robot system. In order to bridge the gap between applications and low-level constructs and to acquire raw situational information for the execution of the context-aware workflow in the robot systems, this paper presents an approach which can achieve the sensor data transmission between a sensing server and a robot system easily.

Key words: publish/subscribe; robot service; workflow; situational information; distributed environments

CLD number: TN919.72

Document code: A

Article ID: 1674-8042(2013)03-0282-04

doi: 10.3969/j.issn.1674-8042.2013.03.017

In the near future, sensor networks will be ubiquitous around us and sensors will produce large quantities of heterogeneous data. Thus, robots should interact with multiple sensor networks that belong to many different domains. Situational information can be gathered from users and their environments. An example is the use of sensor data to provide raw situational information in the workflow. Through the executing workflow process, robots can cooperate with people to solve daily tasks. This means that these kinds of robots have to work in challenging and uncontrolled environments. Because the workflow executed by the workflow engine in the robot system depends on the dynamic states of the physical world (as in context-aware workflow^[1]), external data (the situational information) is often gathered by sensing systems. Let us assume that this external data is completely accurate and there is no inaccuracy due to the quality of the used data when executing the workflow. Based on such external data, well-defined decisions are taken within the workflow execution.

For executing workflow, robots should extract useful information from these dynamic environments, especially the distributed ones. Derivation of situational information such as location, status and actions of users, devices and network by sensing physical environment is necessary to provide users

with context-aware services.

In order to derive the sensor data from the physical world, and to provide more accurate raw situational information for the execution of the workflow in the robot system from the sensor network, this paper presents a delivery approach based on publish/subscribe paradigm^[2]. Here, we will discuss about the proposed delivery approach and demonstrate that it can deliver situational information to any service' users completely.

1 Foundations and related work

There have been many studies to enable robot applications to integrate sensor data into the workflow in the distributed computing environment. The publish/subscribe model has gained a lot of attention and been applied to a large variety of protocols and applications. Based on this, we carefully reviewed the existing communication technologies.

Java message service (JMS)^[3] is a kind of Java message oriented middleware^[4] technology for sending messages between two or more clients. There are many JMSs such as ActiveMQ^[5] and SonicMQ^[6]. They are message brokers which implement JMS API. They require an extra component, the message transfer agent (message broker) which leads to the reduction in performance and reliability

* Received date: 2013-04-12

Foundation item: The MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) Support program (NIPA-2013-H0301-13-2006) supervised by the NIPA (National IT Industry Promotion Agency)

Corresponding author: Jaeyoung Choi(choi@ssu.ac.kr)

also makes the system more difficult and expensive to be maintained as a whole. Also, there is no synchronization function in JMS. ZeroMQ^[7] is an asynchronous messaging library aiming at scalable distributed or concurrent applications. Although supported by many kinds of patterns such as publish/subscribe pattern that connects a set of publishers to a set of subscribers, user applications still have something to be handled more programmatically. ETRI data distribution service (EDDS) is developed by the electronics and telecommunications research institute (ETRI) for the requirement that the sensor data derived from the physical world should be disseminated in a timely and reliable manner to all interested collaborative entities in many cyber-physical systems (CPS). It presents a novel publish/subscribe-based middleware architecture called real-

time data distribution service (RDDS)^[8]. Another research related to the publish/subscribe architecture is the Ganglia^[9], which is a scalable distributed monitoring system for high performance computing systems such as clusters and grids.

2 Architecture of proposed delivery approach

As discussed in the previous section, many robot service applications need to collect and integrate data generated from a large and physically dispersed contingent of sensor networks continuously.

In this paper, the communication architecture of the proposed scheme based on publish/subscribe model satisfies the requirement, as shown in Fig. 1.

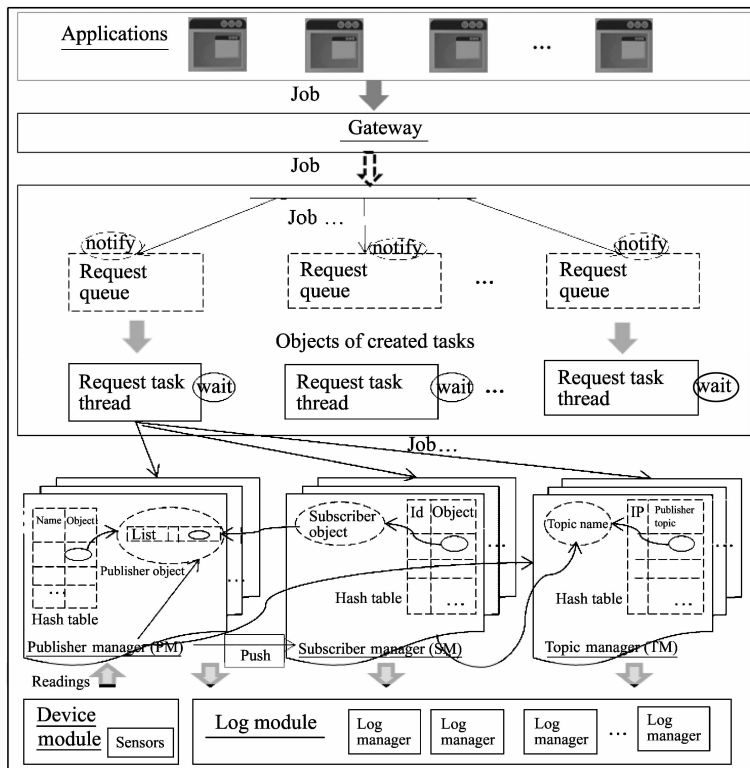


Fig. 1 Publish/subscribe-based communication architecture

According to the aforementioned key technologies and composing knowledges, we designed the communication architecture based on publish/subscribe paradigm for the proposed scheme. In the aspect of function, the proposed communication architecture can be divided into 7 parts including gateway, request processor, publisher manager, subscriber manager, topic manager, log module and device module, as shown in Fig. 1.

The gateway is the entry for user applications to communication with the communication scheme. All the users' command will be transferred through

it while the request processor contains a number of request queues and task threads. For the thread safe, only the publisher manager, topic manager, and specified number of subscriber managers can be processed in one task thread. Both the publisher manager and the subscriber manager have a hash table to preserve the created objects. Furthermore, the topic manager is used to map topics between the subscriber's side and the publisher's side. Once the mapping is successful, data delivery occurs between them. This procedure will be done through multicast. Both the data delivery and operation with user

applications would be recorded in the log module. It collects all the information during the communication and writes all the collected record into files.

3 Design of packet definition and interoperability protocol

The proposed communication approach depends on reliable multicast technology whatever in discovering subscribers or delivering sensor data. A reliable multicast protocol enables the publishers to detect lost and out-of-order messages and take corrective actions to result in the gap-free and in-order

stream. According to the interoperability protocol, the packet format definition is designed in Fig. 2.

As shown in Fig. 2, the whole field is divided into 5 parts and each of them has its special meaning. The first byte points out the operation code during the interactive session of the user application with this communication scheme. For the next, both the size of sequence number and the length of packet are fixed into 4 bytes. The data field is used to represent the date when the packet is published. Due to the variety of operation codes from the interoperability protocol we designed, each operation code would be met by a specified format.

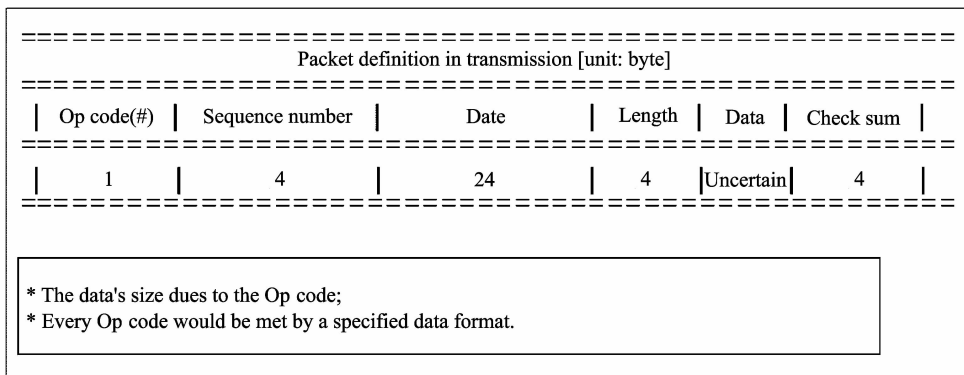


Fig. 2 Packet definition architecture in transmission

The proposed communication scheme supports the interoperation with user applications which can control it through the command or by invoking the method provided in our package. Based on publish/subscribe paradigm, the appropriate interoperate protocol developed can be adapted for any user application. We developed the following interoperation protocol, as shown in Table 1.

Table 1 Supported operation codes and their meanings in the packet transmission

Op code	Description
0X01	Publisher register
0X02	Publisher delete
0X03	Publisher send data
0X04	Subscriber register
0X05	Subscriber suspend
0X06	Subscriber resume
0X07	Delete subscribe from one publisher
0X08	Delete subscriber from multiple publishers
0X09	Delete subscriber from all publishers

4 Data flow between publisher and subscriber

As described in the abstract, a context-aware workflow language (CAWL)^[10] document designed

for a specific scenario must be translated and executed to provide an adaptive service for a user's situation. For this purpose, we need a process to manipulate the situational information aggregated from a sensor network, as shown in Fig. 3.

It is possible to understand the whole communication happening between the sensing server and the robot system in Fig. 3. After both the robot system and the sensing server complete the binding with the proposed communication scheme, the sensing server periodically sends a packet containing topics to the network via multicast. Then, the robot service user will show the interests for its interested topic while the topic manager is responsible for checking the compatibility of topics in both sensing server's side and robot user's side. Once the mapping of topic between the sensing server's side and the robot user's side is successful, the delivery of sensor data from the sensor network will be established directly. Furthermore, the workflow engine will receive the related situational information through the gateway using the local socket communication. The context mapper extracts the types and values from the objectified entities aggregated from the situational information through the proposed communication approach. And then it would compare those types and values of entities from both the user-defined CAWL document and the sensor network.

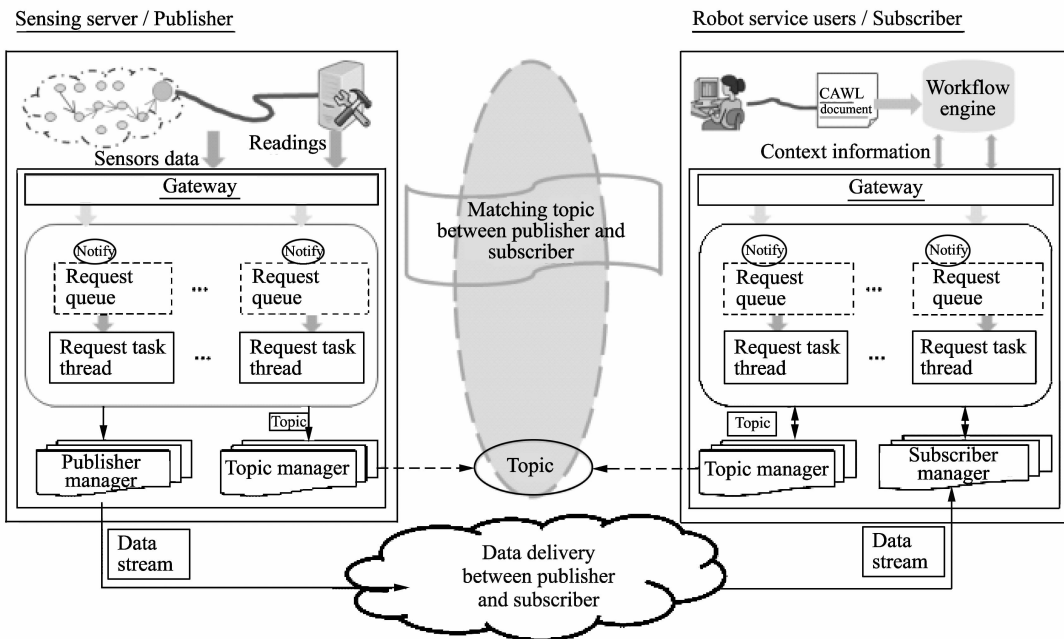


Fig. 3 Data flow between sensing server and robot service user

5 Conclusion and future work

In this paper, we introduced a delivery approach based on the publish/subscribe model for the handling of the sensor data collected from the sensor network and publishing of the sensor data to the robot system for the executing workflow. We also presented the mechanism how the sensor data is published by the sensing server to the robot system assuring that the mapping of the topics both on publisher's and subscriber's side are successful. Moreover, the packet definition in transmission and interoperability protocol with the user application is designed to satisfy the communication requirements.

In near future, we plan to explore the correlation among the distributed sensors and automatic discovery mechanism in establishing communications. To achieve high performance in data delivery, the real-time and low-latency can be another keypoint to study.

References

- [1] Han J H, Cho Y Y, Choi J Y. Context-aware workflow language based on web services for ubiquitous computing. In: Proceedings of International Conference on Computational Science and Its Applications (ICCSA 2005), Singapore, 2005, 3481: 1008-1017.
- [2] Eugster P Th, Felber P A, Guerraoui R, et al. The many faces of publish/subscribe. *Journal of ACM Computing Surveys*, 2003, 35(2): 114-131.
- [3] Lee H H, Tseng C H. A software framework for Java message service based Internet messaging system. In: Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2003: 161-165.
- [4] Curry E. Message-oriented middleware// Mahmoud Q H. *Middleware for communications*. John Wiley & Sons, Chichester, England, 2004: 1-28.
- [5] Snyder B, Bosanac D, Davies R. *ActiveMQ in Action*. 1st ed. Manning Publications Co., USA, 2011.
- [6] Maheshwari P, Pang M. Benchmarking message-oriented middleware: TIB/RV versus SonicMQ. *Concurrency and Computation: Practice and Experience*, 2005, 17(2): 1507-1526.
- [7] Hintjens P. *ZeroMQ-messaging for many applications*. O'Reilly Media, USA, 2013.
- [8] Kang W C, Kapitanova K, Son S H. RDDS: a real-time data distribution service for cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 2012, 8(2): 393-405.
- [9] Massie M L, Chun B N, Culler D E. The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing PC*, 2004, 30(5/6): 817-840.
- [10] Choi J S, Cho Y Y, Choi J Y. The design of a context-aware workflow language for supporting multiple workflows. *Journal of Korea Society for Internet Information*, 2009, 10(6): 145-157.