DReTS: A Distributed Real-Time Tele-Operation System

Myoungjin Kim
Dept. of Computer Engineering
Konkuk University
Seoul, Korea
tough105@konkuk.ac.kr

Hanku Lee, HyungSeok Kim, Sung-Ryul Kim
School of Internet & Multimedia Engineering
Konkuk University
Seoul, Korea
{hlee, hyuskim, kimsr}@konkuk.ac.kr

Abstract— It is not easy to design and implement real-time models in uncontrolled distributed environments and to support well-defined interfaces from real-time systems to external systems. We introduce a distributed real-time system, integrated with tele-operation technology able to collect and process data (location information, image information and information about equipment in remote places) generated from remotely distributed construction sites in real time and operate equipment related to the sites from a long distance away. The main purpose of this paper is to study how to support real-time applications, to design distributed real-time processing techniques for real industrial applications, and to implement and develop a TMO-based Distributed Real-Time Tele-Operation System (DReTS) Model with less strict real-time constraints in ubiquitous environments.

Keywords – Distributed system; real-time system; Tele-Operation system; TMO(Time-triggered Message-triggered Object)

I. INTRODUCTION

We can access geographically distributed remote instruments, equipments, databases, human resources, high-performance computers, etc. as if accessing local resources from a long distance away. Though this accessibility is very stable and secure, it brings us another side: How can a huge amount of data created by construction equipments, sensors, and computing devices be well-synchronized in ubiquitous environments? With conventional programming methods it is very hard to implement real-time models in uncontrolled distributed environments, to design interactions between real-time systems and external systems, and to support well-defined interfaces from real-time systems to external systems. [1]

In this paper we propose a distributed real-time tele-operation system, called DReTS, in order to solve the problems mentioned above. First, we suggest a method to support the development of real-time application programs much more easily in distributed environments. Secondly, we propose a distributed real-time model, based on the TMO [2], so that distributed real-time work-processing may be carried out more easily in the integrated control system. In other words, work-scheduling function, data acquisition technique, synchronization and thread management are implemented by using TMO. Thirdly, we design and implement the DReTS, integrated with TMO agents, performing works in the remote monitoring systems. [3]

II. RELATED WORK

The Time-Triggered Message-Triggered Object (TMO) is a concrete syntactic structure and execution semantics for economical reliable design and implementation of RT systems [2, 4, 5, 6]. TMO is a high-level real-time computing object. It is built in standard C++ and APIs called TMO Support Library (TMOSL). Its member functions are executed within specified time windows. TMO is also a high-level distributed computing object. TMOs interact among themselves via remote method calls. Remote method calls are made in forms that are essentially the same as those of calling conventional object methods.

TMO contains two types of methods, time-triggered methods (SpM) and conventional service methods (SvM). The SpM executions are triggered upon reaching of the RT clock at specific values determined at the design time whereas the SvM executions are triggered by service request messages from clients. Moreover, actions to be taken at real times which can be determined at the design time can appear only in SpM’s. As in other RT object models, the TMO incorporates deadlines and it does in the most general form. Basically, for output actions and methods completions of a TMO, the designer guarantees and advertises execution time-window bounded by start times and completion times. Real-time Multicast and Memory Replication Channel (RMMC) is an alternative to the remote method invocation for facilitating interactions among TMOs. Use of RMMCs tends to lead to better efficiency than the use of traditional remote method invocations does in many applications, especially in the area of distributed multimedia applications which involve frequent delivery of the same data to more than two participants distributed among multiple nodes.

III. ANALYSIS OF DRETS MODEL

With the fast development of the Internet and ubiquitous computing environments, it is no longer necessary for remote instruments, computing resources, and human resources (i.e. engineers and researchers) to be located in the same place and at the same time. That is, it should be
possible for engineers and researchers to access remote instruments and computing resources from a long distance away. However, we need to support real-time controls and the timing characteristics on these geographically distributed and real-time applications without pain during the development.

In general classic real-time systems have been designed with respect to time-restriction for single processors. Implementing and debugging of real-time controls and timing characteristics cause pain during the development of distributed real-time applications with conventional real-time methods. Thus it is difficult to adapt the classic real-time systems into distributed environments such as ubiquitous systems.

In this paper we propose a TMO based real-time system to handle and control various data (sensor data, visual data, equipment status data, etc.) generated from remotely distributed construction sites in real time.

Goals set to design such a model as above are as follows: The first one is to suggest a method to support the development of real-time application programs much more easily in distributed construction site environments. Secondly, the second one is modeling the system through the TMO Object Model so that distributed real-time work-processing may be carried out more easily in the integrated control system. In other words, work-scheduling function, data acquisition technique, synchronization and thread management are implemented by using TMO. The third one is to suggest a framework to make it easy to develop distributed real-time application programs based on TMO agents performing works in the remote monitoring system and distributed construction sites.

A. DReTS Model

Since the DReTS Model can analyze and remote-monitor the real-time data acquired from various kinds of equipment (sensor equipment, image acquisition equipment and construction equipment) in construction sites, its system can provide engineers or remote-managers with cooperative-work environments to carry out their work much more easily, remarkably reducing their job performance time.

The framework suggested makes it possible to extend into models applicable to a large number of fields where the actual circumstances of sites can be monitored and the relevant equipments are operated from a long distance away, such as road management, bridge management and pollutant management, by extending its function in addition to the distributed real-time remote control system, through which construction fields can be managed.

B. Feature of DReTS Model

- It provides development environments to easily develop distributed real-time application programs based on TMO.
- It provides the easiness of expanding, reducing and modifying the system and the easiness of managing external communications between threads and objects by inheriting the features of TMO object extensibility.
- In consideration of application methods and functions, etc. based on the framework suggested, it is possible to apply to various other fields through modifications of TMO objects.

C. Architecture

Figure 1 depicts the architecture of the proposed TMO-based Distributed Real-Time Tele-Operation model. One of the main issues for the proposed model is to apply the easy-to-use TMO to real-time applications that are usually hard to design and implement with conventional programming methods. The proposed model is divided into 3 domains: RD (remote domain), ICSD (Information Convergence Server domain), and CSD (Control System domain).

D. RD(Remote Domain)

The remote domain (RD) is to collect remote data and to monitor remote instruments. RD consists of the Time Sever TMO (TST) and Working TMOs (WTs). TST gives the timing characteristics to WTs. The video, audio, and sensor data with the timing characteristics are transferred via sockets (TCP/IP) to Information Convergence Server Domain (ICSD).

WTs are synchronized by the time characteristics supplied by TST. The time characteristics supplied by TST are more suitable to the proposed model than those supplied by the Internet or GPS time services since TST is closely located to other WTs and this locality avoids the network latency that makes it hard to synchronize real-time applications.
E. ICSD (Information Convergence Server Domain)

The Information Convergence Server Domain (ICSD) is to manage information convergence sever in order to help data communication between RD and CSD (Control System Domain). In real time ICSD manages data communication between RD and CSD (Control System Domain) via TCP/IP-based client / server model. Moreover, it periodically handles collected data (with time characteristics from RD) and control-messages (from CSD) to be safely and precisely transferred.

ICSD should keep waking up, be started prior to other domains, and wait for collected data (with time characteristics from RD) and control-messages (from CSD). When collected data from RD is arrived, ICSD immediately transmits the data to CSD. Also, when control-messages from CSD are arrived, it immediately transmits the messages to the specific WT where the messages are heading in RD.

Servers in ICSD can storage a large amount of data from the remote domain and can provide the secure management of data from the remote domain to the interfaces.

F. CSD (Control System Domain)

Using CSD, remote engineers can monitor and control the entire system in real time. CSD is to provide user interfaces to check the status of the whole system, to monitor sensors and equipments in RD, and to manage control-messages to control remote instruments in real time.

IV. DESIGN OF THE DRETS

A. Framework

In this section we describe several designs issues caused by using TMO APIs for RD and ICSD in detail.

That is, RD and ICSD explained in the model are designed with object model based on TMO. The Distributed Real-Time Tele-operation Application Framework is implemented using TMO toolkit. [2]

ICSD consists of 4 TMOs: Sensor_Request_TMO, Monitoring_TMO, Command_TMO, and Time_Server_TMO. Sensor_Request_TMO requests Sensor_SvM in Sensor_TMO to transmit collected sensor data. Monitoring_TMO requests Sub_Thread to capture and transmit visual data. Command_TMO transmits control messages to Motion_SvM in Device_TMO. Time_Server_TMO transmits server system time to Time_Set_SvM in Time_Set_TMO.

RD consists of Sub_Thread and 3 TMOs: Time_Set_TMO, Device_TMO, and Sensor_TMO. All TMOs can be easily and securely synchronized via time provided by TST. Sub_Thread captures visual data that will be requested by Monitoring_TMO.

The reason why Sub Thread is used instead of internal thread of TMO is because the capacity is limited to 512byte in transmitting data through SvMGate. Thus, instead of TMO objects, Sub Thread is used. Future work will be conducted on the use of TMO objects by compressing images or transmission of data through RMMC.

Device_TMO get control messages from Command_TMO in ICSD and transmit the messages to equipments such as robot arms and surveillance cameras. Sensor_TMO collects sensor data and transmits the data to Sensor_Request_TMO_TMO.

The DReTS Application Framework consists of seven TMO objects and one Sub_Thread in total. In addition to the basic Application Framework, it has a merit of extending the system by adding relevant TMO objects or Sub_Thread when needed to extend additional functions.

B. Real-time Synchronization System TMO object

Figure 3 shows how the Real-time Synchronization System object is structured. The method that time-synchronization is periodically accomplished after the time
synchronization by TMO middleware (TMOSM) is as follows: When the server system time is transmitted to Time_Set_TMO at Time_Server_TMO of ICSD according to the period set, it is possible to compare and maintain the time synchronization state of the whole system while comparing the server system time and the system time of RD. As for the communication method between TMO objects, BlockingSR method is chosen out of SvMGate communication methods of TOMSM. The reason why such a method is selected is that the accuracy of data transmission is sustained as SvMGate inside TMO objects is blocked during the transmission of data.

In this way, as RD maintains the synchronized state by using the server system time of ICSD, it transmits Timestamp label information attached to each data of image data, sensor data and equipment state data to ICSD.

C. Monitoring System TMO object

Figure 4. Monitoring System TMO object

Figure 4 shows Monitoring System TMO object. The role of Monitoring System TMO object is to acquire and transmit data to ICSD functioning to collect information of image data input through webcams connected to USB.

With external Sub Thread simultaneously-processed with Pseudo_TMO Thread, image data is acquired through webcams or CCTV in RD and transmitted to ICSD via socket communication. Then, according to the period of Capture_SpM in ICSD, The Monitoring _TMO possible to receive image data and display.

D. Sensor data processing System TMO object

Figure 5 shows Sensor Data Processing System TMO object. The role of Sensor Data Processing System TMO is to acquire and transmit data from sensor. Through SvMGate provided from TMOSM, ICSD requires data transmission to RD according to the period set by Sensor_SpM. Sensor_TMO, required to transmit data will transmit result values to each SpM requiring for the transmission, by acquiring location data or various sensor data from the sensor device with Sensor_SvM implemented internally.

E. Tele-operation System TMO object

Figure 6 shows Tele-operation System TMO object. When an equipment operator gives operation commands for equipments while monitoring the site from a long distance away, Command_SpM in ICSD transmits command codes to Motion_SvM in Device_TMO of the RD via SvMGate. Then, Motion_SvM transmits the commands transmitted to the relevant equipments. The equipments to which the command codes are transmitted are operated by methods internally implemented. In addition, equipment operator confirms equipment state information and commands stored in ODS from a long distance away, it can secure the accuracy and safety while operating equipments.

V. IMPLEMENTATION AND RESULTS

A. Implementation of proposed system

In order to utilize merits of system extensibility (extension of TMO objects in RD) on the basis of the DReTS model suggested, this paper establishes a demo-system environment in the combination of Tele-operations system, able to operate three robot vehicles, Monitoring System, using wireless cameras attached to the robot vehicles, Sensor Data Processing System for the acquisition of a single Zigbee sensor data and Real-time Synchronization System, using the server system time.
Especially, in order to experiment on various equipment control environments, serial communication and a wireless LAN-based intelligent unmanned car robot are used. Without the implementation of CSD, ICSD is implemented to play the role of CSD.

The whole system is composed of ICSD (Master Node) functioning as a computer from a long distance away and RD (Worker Node) functioning as a local computer. The role of ICSD system is monitoring in real site, to transmit the server system time, to receive sensor data and image data. The role of RD system is to transmit command to the robot car physically connected, to receive the server system time to confirm the synchronization condition, acquisition of Zigbee sensor data and to acquire image data from the wireless receiver. Table1 shows the development environment for the DReTS.

### Table 1. Development environment

<table>
<thead>
<tr>
<th>System</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICSD Computer</td>
<td>Intel Core2Duo CPU E8400, 4GB RAM, Windows XP Professional SP3</td>
</tr>
<tr>
<td>RD Computer</td>
<td>Intel Core2Duo CPU 4400(2GHz), 4GB RAM, Windows XP Professional SP2</td>
</tr>
<tr>
<td>Zigbee Sensor</td>
<td>Size of 19mm – 21mm, 30m – 100m operating range, Serial 2 port(RXD1, TXD1), 9600BPS transfer rate</td>
</tr>
<tr>
<td>Intelligent Unmanned Car</td>
<td>Size of 905mm * 190mm * 11mm, 1750g, Maximum speed of 1000 mm/sec, EV2440, RS232C(19200BPS)</td>
</tr>
</tbody>
</table>

B. The DReTS service scenario

The service of DReTS embodied is provided as the following scenario:

- TMO in RD (Worker Node) begins.
- `Sub_Thread` is created to acquire image data from the wireless camera.
- The IO connection between RD and hardware device is standby.
- TMO in ICSD (Master Node) begins.
- The TCP/IP connection is established for the image data communication between RD and ICSD.
- The connection between RD and device IO is completed (Unmanned cars 1, 2 and 3 /Zigbee sensor)
- RD and ICSD TMO engine starts.
- The monitoring system is activated.
- Through the monitoring system, the remote operator can control the vehicles 1, 2 and 3.
- ICSD transmits the server system time to RD periodically.
- RD confirms the synchronization conditions by receiving the server system time.
- RD periodically transmits the data values of Zigbee acquired from ICSD.

- ICSD shows the sensor data values and server system time acquired to the remote manager.

C. Detailed structure of the DReTS

Figure 7 shows the structure of implemented the DReTS. The implemented system uses eleven TMO objects in total. ICSD is composed of six TMO objects while RD is composed of five TMO objects including the thread to capture images.

![Figure 7. The DReTS structure in detail](image-url)
synchronization. In the period of 10-second, ICSD requires for the acquisition of sensor data from the relevant SvM and ICSD outputs the sensor data transmitted on the screen. The TMO in ICSD receives image data from the image capture thread of RD through SpM, having the periodical feature of 100ms, and outputting it on the screen makes it possible for users to monitor the circumstances of sites.

D. Advantage of the Proposed Model

It is easy to implement and debug TMO nodes. Implementing and debugging of real-time controls and the timing characteristics cause pain during the development of distributed real-time applications with conventional real-time methods. But all we need to do is to use TMO communication APIs supported by the TMO tool kit.

It is easy to modify and expand the proposed TMO-based model. We often need to scale up or down the whole system in the time dimension. Many modifications could be needed with conventional real-time methods. But all we need to do is to change the scale of the real-time clock of TMO for the proposed TMO-based model.

Thus, we urge TMO-based real-time applications are suitable to systems with less strict real-time constraints such as construction equipments, space probing equipments, tsunami-detecting equipments, etc, since those equipments product relatively small amount of data in the period of SpM and are not a time-critical decision model.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we proposed an easy-to-use TMO-based real time model with less strict real-time constraints in ubiquitous environments. Using the proposed model, we designed and developed a TMO-based real-time system for real industrial applications able to be used in construction sites.

The proposed model is very promising since it provides a sound TMO-based real-time application framework, cost-effectively resolving the problems caused by conventional programming methods during the development.

On the basis of the details having been implemented so far, further researches will be conducted on how to add modules applied with synchronization methods using GPS based on the basic method of TMOSM providing I/O port transmits data acquired to ICSD again through SvMGate5. management module and synchronization methods for the support for RD and various devices. Moreover, by actually applying the system suggested to construction sites so that test beds may be established and the result of applying it to the sites, the efficiency of DRets will be verified.

In the future, by extending the function in addition to DReTS able to manage construction sites, further researches will be conducted on the application to a large number of fields where the actual circumstances of sites can be monitored from a long distance away, such as road management, bridge management and pollutant management, and the relevant equipments can be operated.

ACKNOWLEDGMENT

This research was supported by the grant (07KLSGC04) from Cutting-edge Urban Development – Korean Land Spatialization Research Project funded by Ministry of Land, Transport and Maritime affairs of Korean government. Professor Hanku Lee is the corresponding author. hlee@konkuk.ac.kr.

REFERENCES