Communication System Architecture to Manage a Containers Terminal

José L. Poza  Juan L. Posadas  José E. Simó  A. Crespo
Department of Computer Engineering
Universidad Politécnica de Valencia
Camino de Vera s/n,
46022 Valencia, SPAIN

Abstract — Currently, containers are used in the transportation of products because of the point-to-point service they offer. By using specialized cranes, operations in maritime transport and land transport can be achieved quickly. To optimize the operation costs, the whole management of the terminal is computerized proportionally with the complexity of the problem. The multi-agent systems paradigm seems suitable to solve the computational problem. The communication between agents and the corresponding data warehouse is made by means of a number of content-oriented communication channels. For supply an organized message routes to the system, an event oriented communication system has been developed. The communication channels determine the communications system architecture. This paper describes how common communications architecture for agents can facilitate the management of an industrial system. The aim is to separate the agents of the communications. The communications system used is based on the distributed blackboard architecture.

I. INTRODUCTION

Marine containers terminal is the connection between containers - maritime transport and transport by land. To make this process it is necessary to store containers temporarily in the yard, using specific machines, until they are sent to their final destination.

The yards are generally large and, due to the ecological impact of ports and their high costs, it is necessary to optimize the utilization of space and the resources implicated in the container management. This lack of flexibility is a serious drawback for system adaptation and migration to similar industrial automation systems.

Consequently, container terminal management is a very complex system; the only way it can reasonably be controlled is to develop a number of modular components that are specialized for solving a particular aspect of it [1].

The whole set of computer resources and procedures are referred to as Terminal Operating System. These systems are composed of databases, and tools intended to help human operators in the planning and control tasks. The results of off-line planning are difficult to implement due the high unpredictability of the elements to control. Commercial operative systems used in container terminals are closed. In many cases, flexibility for operative is supported by the system, but not the flexibility for the data sources and the communication system installed on the layout.

Communication between components, knowledge about system states, and the important events are a fundamental issue for solving errors and keeping the system control.

II. SYSTEM DESCRIPTION

A. System history

Set of operations in the terminal is very extensive; the alternative approaches separate some common systems [2].

Marine Side and Land Interface: These two systems focus on loading/unloading containers to/from ships or to/from land transport. Normally two or three gantry cranes (GC) are used to move containers for each ship. In [3], the process of formulation of a container load profile is detailed.

Transfer System: It transfers containers into the container storage yard. The method used in the terminal is to employ yard trucks (YT) to make the transports. Transtainers are used to move a container on the storage area of the yard. The existing approaches to this problem attempt to resolve it using operations research techniques [4].

Container Storage System: Its purpose is to allocate and control the containers in the yard. Typically, the applications for the management of container terminals divide the work into two tasks [5]: the yard configuration and the automatic container allocation. These two tasks, in the current new system, are contemplated like two different agents with two different communications channels assigned to them.
Another task is the control of the terminal access gates, identifying both the truck and the container using artificial vision techniques. In this way, an unattended gate system, which speeds up the truck admission, is achieved and global productivity of the containers terminal is increased.

Originally, the automation system was designed based on the agent’s requirements [6]. The information needed for them, was stored on an IBM AS/400 database system. In the first phase of the project, all information lies in the database. For this question, the table design determines the system structure. Based on the database to approach the design of the architecture, the system was divided based on its different tasks (figure 1).

Every agent execute their task in an independent way to the others system agents, to realize his tasks, they will have to be coordinated and to communicate its decisions by a reliable and secure communication channel. The fundamentals of the communication system, was described in detail in [7].

Nowadays, this system is being used to control the multi-agent system implemented to the container terminal of the Port of Valencia, Spain.

B. New approach

The responsibility for control, of the original system, needs a main service agent [6]. Because of, a new model of the system was designed. The new model was called GAMA: Gestión Automática Marítima (Maritime Automatic Management). In this model, the communication between agents and the corresponding data warehouse is made by means of a number of content-oriented communication channels.

For this reason, two types of channels can be distinguished in the system: event-oriented and data-oriented channels [7].

The event-oriented channel topology defines the agent structure architecture. Owing to, events transmitted into the agents, will provide enough information about which agents are involved on an operation. Moreover, communication between agents, and the knowledge about system states and the important events are a fundamental issue for solving errors and keeping the system control. Finally, the fact of have communications architecture with a message topology, provides a method for monitoring the system by watching the communication event channel and the messages prepared for the purpose [8].

C. System Operation

The relation between agents and into agents and database is implemented inherently by the communication system (figure 2). The system is divided in three differentiated layers. Table I shows the layers and the mission in the system of each one.
A graphical description of the layers can be viewer in the figure 2. The most important components of the system are the database and the communications system. In the database is stored the information about the containers and the machines. The database structure is very complex and is not the objective of the present article.

The support of the communication system is an application that connects all the agents and his messages. These messages are formatted in XML.

In the figure 2 all agents that are in the control of the terminal can be observed. Although it is very difficult to classify the agents, they can be classified in the scope in which they work.

When a container arrives at the terminal, by means of highway or by railroad, an agent registers the container data in the terminal (fig. 2-a), at the same when it leaves the terminal the agent removes the container information (fig. 2-b), but the historical information is stored in another database. This update of database causes the performance of an agent that locate it at the terminal (fig. 2-c).

The location of a container in the database implies the use of a set of machines to translate the container. The machinery agent is responsible for assign the machines to perform the movements. These agents depend on the machine parameters (fig. 2-d) and (fig. 2-f). In order to manage the machines correctly, is necessary knowing the position into the terminal. The machinery position agents are those provide this information to the database (fig 2-d) and (fig. 2-g). A similar cycle happens with the stowage plans for ships (fig. 2-h) and (fig. 2-i). The complexity of the operations does that the ship agents need much information, its work consists of prevent the set of containers with which the ship will arrive and the set of containers with which the ship will departs.

A marine door agent (fig. 2-k) must manage the location of containers in the terminal, due to the critic of the process of load and unloading of the ships; this agent needs the performance of a marine operative agents (fig. 2-l) and (fig. 2-m).

Finally, by means of the common system of communications all the agents can interact among them without interact with the database; therefore, the coherence of the information is responsibility of the agents.

III. COMMUNICATION SYSTEM ARCHITECTURE

A. Communications between agents

Communication in distributed systems presents several problems [9]. Mainly the following: synchronization and co-ordination of messages and applications, coherence in the format of the messages, fault tolerance and mark the time boundary in applications. The communication between GAMA modules is done in two levels.

These levels are defined by the message type that communication channel transport. The two types of channels that can be distinguished in the system are the event-oriented and the data-oriented channel (figure 4).

SC and gateways compound event level. Event level offers oriented services to communication and diffusion of the events produced by the terminal. A Communication Server (SC) based in the data sharing through a distributed blackboard model [10] makes this diffusion into the agents, directly (figure 3). Specific routers are the necessary bridges connecting heterogeneous systems with the SC.

Data level is based on the operating system support to the communication with the database: ODBC. This level offers database access to do queries and changes in the container yard state. ODBC offers a database directly to all agent connections (figure 3).

Event-oriented channel includes several different communication systems. For example, a serial port and a radio channel are used to communicate with the machinery agents, whereas a socket connection is used to communicate events from the database server. Using different heterogeneous systems makes it necessary to adapt the internal messages at the concrete communication system and to make use of different communication protocols. The two main components of the agent communication system are exposed next.
B. Event Communications: SC

To allow the interaction between Win32 processes of the distributed system, a communication common interface called Communication Server (SC) is used.

The SC system is based on the internal data representation through distributed blackboard architecture. The SC continually updates the data structure that represents the blackboard, showing the values of the messages sent between the processes. Through the access to the data structure of the SC, the distributed processes communicate using a common interface independently of their location.

The SC system has been developed and tested by the authors in previous works in the context of distributed systems and mobile robots [11]. The system has been subjected to extreme condition tests proving its reliability and high performance.

The SC system requires that each computer belonging to the system execute an SC instance (figure 4). These instances are in charge of fulfilling the communication tasks, updating and control data for obtaining partial copies in different computers. As a result, processes, independently of their location, can contact all the system by making accesses local SC instance.

To guaranty the consistency between the copies of the distributed blackboard, the SC instances manage the necessary communications and the data transmission between the computers establishing permanent connections with fault detection and recovery.

The system gives a bind-notification schema to associate the execution of specific code of processes with a set of events such as the updating in the system of a data in the blackboard. Thus, processes connected to the SC receive automatically the messages that they are interested in.

The configuration of the SC consists, mainly, on the specification of an object set that is used as a logic media for distributing system messages.

C. Event Communications: Gateways

There are nodes where it is impossible to execute a SC instance (for example, an embedded PC where the operating system does not have the SC requirements, UNIX systems, or AS400 systems); in these cases, the processes have to communicate with a remote SC through a specific gateway or router application. These nodes are called heterogeneous nodes.

In the normal operation of GAMA routers have the function to open, maintain and close the communication channels through sockets (TCP, UDP) and serial port (COM) between the heterogeneous nodes and the Communication Server (SC).

D. Data communication: ODBC

ODBC (Object Database Connectivity) is a Microsoft database API that allows applications to access databases over a network. ODBC supports both SQL and non-SQL databases [12].

Distributed applications have to make queries into the Database Server through ODBC channel. ODBC communications use TCP through specific drivers for AS400 from Database Server and through a specific DLL (Dynamic Linked Libraries) from GAMA modules. These specific DLL’s export a set of tools and data formats to help applications accessing terminal data in a uniform way independently of the database technology and definitions in the background. The programming tasks in the adaptation of GAMA system to a new database system are restricted to minimal changes in these DLL’s

IV. SUPPORT PROVIDED BY THE COMMUNICATIONS SYSTEM

A. Modelling the system

Every agent is connected to a set of SC Data. Every SC Data can be associated to a system and an operative of the terminal of containers (figure 5).

For each event produced, from any source, there is a route of channels and applications through the associated message have to travel. Each SC Data of the communication system can be seen as an interconnected node with others based on the route that must take the message (figure 6).
The calculation of the time for a specific message is made directly following the route of the message. Each node knows the temporary limits that must have a message related with an event. Based on this, nodes can determine the validity of the messages. Once the route of a message corresponding to an event has been determined, obtaining the time corresponding to the event is made using the equation that represents the specific route. Every type of message has its own sequence time (figure 7).

Mathematical methods using patterns of event sequences can be used to calculate the time of any sequence of messages. Another method is to accumulate the individual times of each message in the sequence.

B. Monitoring the system

Monitoring a distributed system implies generating the state log system in a concrete moment (snapshot) or the events log or a system event [13]. System state monitoring implies that each node sends information to the monitoring agent [14], which can be executed periodically or under demand. Event monitoring is a more complex task that consists of three phases: location of event detection, time of event detection and event report format [13]. The previous levels are detailed following.

Location of event detection. There are two ways of detecting events in the distributed system. One is to detect individually on the part of each node the important events, to store this information and to send it to the node that executed the analysis. This approach has several problems. The main one is the temporary overload of the nodes due to the necessity of generating the traces. The other big problem is the overload of communication channel due to the necessity of transporting tracing among nodes through the communication channel. The alternative to event monitoring in the nodes is to locate events in the same communication channel.

Time of Event Detection. In order to characterize temporarily the time consumed in the route from a message through the system, the Time Firewall method [15] is used by each application maintaining a registry of the time consumed by a message while it remains in the application.

When the message arrives, the application compares its Time Field with the corresponding internal table where temporary characteristics of the message are specified. Based on this comparison, the message will be dealt accordingly.

When the message leaves the application, the time that has remained in the application is added to the Time Field. A table to maintain the time intervals exists for every node. By means of this table, the communications state can be temporarily characterized at any moment.

This temporary characterization is very interesting because it allows knowing whether the system is tuned and works in a synchronous way any moment. In the event of arriving messages with a high value time field, it is possible to make an inspection in the route of nodes towards the origin of the bottleneck and take measures to solve it [8].

If the method used to monitor the system is the distributed node monitoring method, the analysis consists of several phases as merging traces, validation of monitoring information, database updating, combination of monitoring information, filtering of monitoring information and finally the analysis of monitoring information. As our system uses the communication channel monitoring method it is only necessary to do the filtering phase and the analysis phase.

The system allows a flexible events monitoring, because it is possible to watch out for different control levels (table II).

The system overload that monitoring introduces is minimum, because the connection of the monitoring agents is carried out in a separate machine. This implies that the rest of the agents do not perceive the effect of the monitoring node. Any agent can switch from a normal work mode to a monitoring work mode, by reading or writing the trace field in the message.

Manage a distributed industrial system is a complex question due to the difficulty inherent in the node control and changes in topology of the nodes. The communications system presented, based on the distributed blackboard model, gives the support to control and manage all nodes.

The control of the messages, transmission and reception, is responsibility of the communications system. Because isn’t necessary that the agents manage the communications, changes in the system are easy to realize (figure 8).

<table>
<thead>
<tr>
<th>Mode</th>
<th>Number of SC Data</th>
<th>Interval of events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical Events</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Snapshots</td>
<td>[1, n]</td>
<td>1</td>
</tr>
<tr>
<td>Message sequence</td>
<td>1</td>
<td>[1, n]</td>
</tr>
<tr>
<td>System evolution</td>
<td>[1, n]</td>
<td>[1, n]</td>
</tr>
</tbody>
</table>
C. System simulation.

The model of communications based on distributed blackboard, facilitates the simulation of the system. Only is necessary to simulate, the behavior of the agents by means of the injection of the information into the SC objects which the agents simulated are connected. For example, if wants to simulate the system of agents, but without using real machines, then can be injected the behaviors of the machines in the SC objects (figure 9). This causes that agents don’t appreciate the absence of the real machines, since at them gets the information in the same way that the real machines. If agents are simulated, instead of simulating the complete system, it is possible to prove the behavior of real agents interacting with simulated agents.

D. Agents adaptation: flexibility.

The adaptation of an agent to the system is immediate, due to the independence of the communications between the agents. An agent that changes his location only must be adapted to the communications system, and this adaptation is standardized for all agents. The great advantage of the standardization of agents by means of the communications system is the facility for the mobility of the agents into the system.

Functionally, the agent is connected to corresponding SC objects and the changes in his location do not imply changes in the system topology (figure 10).

In order that an agent can change his physical location, the only requirement is that the final computer has the necessary objects SC in his distributed blackboard instance.

For example, figure 10, shows an agent dedicated to the location of the containers that enter or leave, at the terminal, by terrestrial means.

E. System expansion: scalability.

System expansion implies adding new functionalities to existing subsystems. These new functionalities will be made by new agents. The new agents will need SC objects to communicate with the existing agents. If the SC objects already exist, the amplification of the system only will consist of connecting to them. Usually a system expansion implies a new operative (figure 11). In this case, the number of SC objects will be increased to give support to the new operative. Existing agents don’t need changes; his behaviors only depend of the previous SC objects.

For example, figure 11 illustrates how to integrate a new terrestrial operative. This case can be the incorporation of a new machinery of internal transport. This new operative is dedicated to the movement of containers into the terminal, for that reason does not have an SC object that belongs to input output system.

V. CONCLUSIONS

The system presented can be adapted easily to other similar industrial system without great relevant changes. A well know variation of the system topology, is not a problem due to the changes only affects to configuration of the SC Data in computer that holds a concrete agents. The GAMA software is currently working in the port of Valencia.

The core of the GAMA software is the communication system, a programming model used to develop distributed modules in a transparent and scalable way. Gateways have been developed to open the communication system to heterogeneous nodes such as currently devices installed in the plant. Time stamps, in the form of “time firewall,” are
proposed to look for “bottlenecks” and find the system limits under stressed simulation.

Basic operations for the control of an industrial system, like operations modelling, system monitoring, or simulation, can be implemented without impact in the system.

The communications system, allows to agents to move into the system without the needed of change his connections. The use of blackboard architecture simplifies the changes and the expansions of the automation system.

The terminal automation system (GAMA), based on the communications system presented in this article, is working nowadays in the Maritime Terminal of Containers “Muelle Príncipe Felipe” of Valencia (Spain).

VI. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution at this work of Eduardo Orellana from Maríntima Valenciana S.A.

VII. REFERENCES


