

# Implementation of CORBA-based data transfer between multiple machines

Yongbin Bai, Yupeng Liu, Haiming Tong, Zhen Liu

<sup>1</sup>China Satellite Maritime Tracking and Control Department, Jiangyin, 214431, China

**Key words:** Software Bus; CORBA; Data transmission

**Abstract:** In this paper, complete Virtual Radio Control System software bus, mainly multi-machine data exchange between modules of this key technology. Environmental monitoring and control for distributed systems, data exchange based on CORBA technology between multiple machine function modules.

## 1. Introduction

Software bus is the foundation platform software system, provides for the development of the upper functional modules, and integrated software framework for interoperability. Soft bus need to solve two key technologies, multi-machine data exchange between functional modules and high-speed data transmission between the single function within the module. From this data the interactive aspects of the software bus to achieve.

## 2. Overall hierarchical structure

Transmission module is designed based on CORBA event service mechanism, and to improve and extend with the characteristics of measurement and control system, and improve performance of data transmission. The system uses hierarchical design ideas, designed primarily to three levels: level agents, agent management, event channel layer, as shown in Fig. 1.

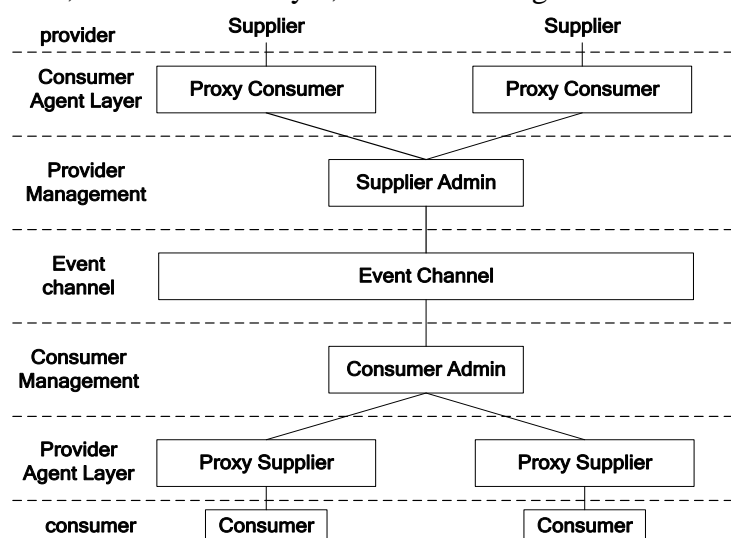


Fig.1 Hierarchical structure of the system

As shown in Fig.1, Agent layer includes agents and consumers by providing proxy; proxy management including supplier management proxy agent and consumer management.

The event service data transmission, Supplier is not directly registered on the event channel, but with the proxy layer Proxy Consumer connected. Proxy Consumer and provider in the same process space, is to simplify the event channel other consumers. Similarly, Consumer Proxy Supplier is directly connected. Visible, agents and consumers to provide a layer of soft data bus and the interactive interface, the system is the connection layer and external communications.

Supplier Admin and Consumer Admin role in the diagram is to create separate providers and consumers proxy agent. A provider manager / consumer manager can create more than one proxy provider / consumer agent.

Data generated through the algorithm module is transferred to the consumer proxy software bus, the bus and transfer agent by the provider to the consumer algorithm module need this data, thereby completing the data communication between the two modules.

### 3. Construction of a communication link

Construction of communication links including the establishment and disconnection of the communication link.

Providers and consumers for data communication, we must first establish a communication link. Process established communication link includes an acquisition event channel to establish links and link consumers end provider side.

#### a. Get event channel

This article uses CORBA Naming Service Event Channel Obtains object references. First, the service name to the event, and then naming services Naming Context interface service resolving events, create an event channel:

```
...
CosNaming::Name name (1);
name.length (1);
name[0].id = CORBA::string_dup ("EventService");
CORBA::Object_var naming_context_object=
orb->resolve_initial_references("NameService");
CORBA::Object_var ec_object = naming_context->resolve (name);
RtecEventChannelAdmin::EventChannel_var EventChannel=
RtecEventChannelAdmin::EventChannel::_narrow (ec_object.in ());
...
```

#### b. Establishing a communications link between providers.

(1) Event Channel to create a Supplier Admin:

```
RtecEventChannelAdmin:: Supplier Admin var
supplier admin = EventChannel -> for_suppliers ();
```

(2) Call provider operations manager for the end consumer to create a Proxy Consumer locally:

```
consumer_proxy = supplier_admin-> obtain_push_consumer ();
```

(3) Providers themselves connected to the Proxy Consumer, told the Event Channel Register:

```
consumer_proxy-> connect_push_supplier (supplier, get_SupplierQoS ());
```

This step has two input parameters, the first parameter indicates the provider to connect to a communication link; second parameter is the type of data provider published not qualify for this type of data communication link cannot be sent.

c. Establishing a communications link on the consumer side.

(1) Event Channel Manager to create a consumer:

```
Rtec EventChannel Admin:: Consumer Admin var
consumer_admin = EventChannel-> for_consumers ();
```

(2) Calling consumers manage the operation of the peer provider to create a local Proxy Supplier:

```
supplier_proxy = consumer_admin-> obtain_push_supplier ();
```

(3) Consumers themselves connected with the Proxy Supplier, represents registered to the same event channel:

```
supplier_proxy-> connect_push_consumer (consumer, get_ConsumerQoS);
```

This step has two input parameters, the first parameter represents the communications link to connect to the consumer; the second argument is the data type of consumer subscriptions, communications link will only meet this type of data is sent to the consumer by module.

This completes the connection data communication links between providers and consumers, as shown in Fig.2.

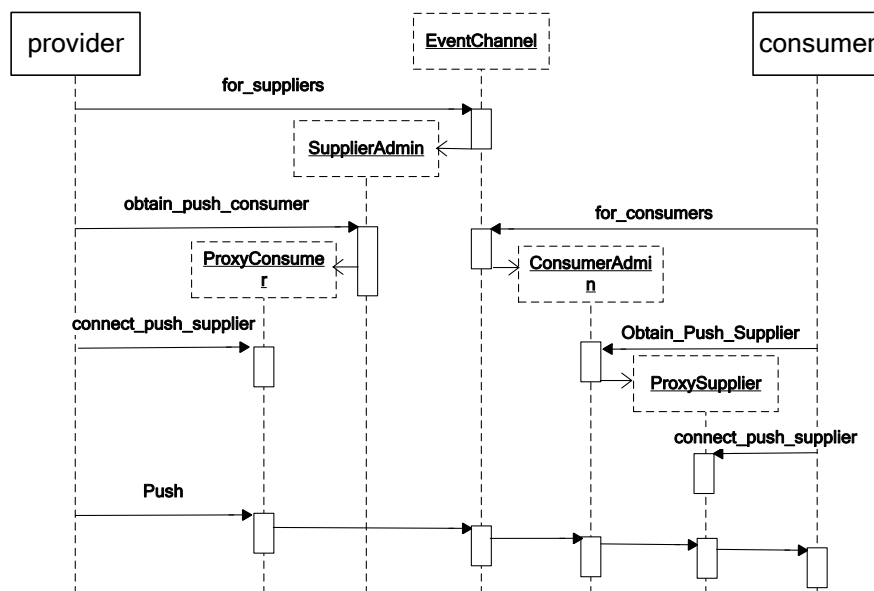


Fig.2 Communication link connection diagram

Data provider or consumer can take the initiative to disconnect the data communications link.

(1) Provider side

After the completion of the transmission data provider, consumers can call the operating agent connected thereto on:

```
// Do not send any more datas...
consumer_proxy->disconnect_push_consumer ( )
```

(2) The consumer side

Recipient of the data relative to disconnect, you can call the provider proxy operator connected thereto on:

```
// Do not receive any more datas...
supplier_proxy->disconnect_push_supplier ( )
```

#### 4. Data transmission process.

Software bus data transmission system in accordance with the "publish and subscribe" scheduling mode, the data from the module provider to subscribe communication link transmission of this data by the consumer module. Unified communications link management providers and consumers, and complete two matches by filtration mechanism to publish and subscribe to the same data source providers and consumers to link to complete the transfer of data. Definition of providers and consumers is relative, even if the data is sent after receiving consumer data to other modules for further processing, it became provider.

Provider side to achieve the data transmission requires the following steps:

a. Soft bus landing.

Soft landing bus primarily for initialization software bus, including the provider side ORB initialization, create POA adapter, and open the naming service.

b. Publish data type.

According to "publish - subscribe" scheduling mechanism, the provider-side before sending the data to published data type of the output data, the communication link according to data released by the type of data to be filtered, meet the filter criteria to be able to send data to the communication link. Published data type is designed as a four-byte string.

c. Connecting a communication link.

Communication link is a major part of the software bus, according to the event service mechanism, establish a data transfer between the provider and the consumer channel, to complete the transfer of data.

d. Produce and package data.

Data provided by module algorithm processing internally generated data to the output. Before being sent to software bus, in order to facilitate the transmission and consumer software bus terminal receives, you first need to package the data format converted into soft bus specifies a unified data structure. According to the unified data structure designed in this paper, data formats, including header and data body, described later in detail. Here is the process of packaging the data:

(1) Establishment of a data packet.

```
RtecEventComm:: Event_Set event (1); event.length (1); // define the length
```

```
RtecEventComm:: Event & e = event [0];
```

(2) Initialization header, specify the data type.

```
Event [0] .header. Type = "BPSK"
```

(3) Import the data into data volume.

```
Event [0] .data.any value <<= data
```

Provider-side data transfer program flow chart shown in Fig.3.

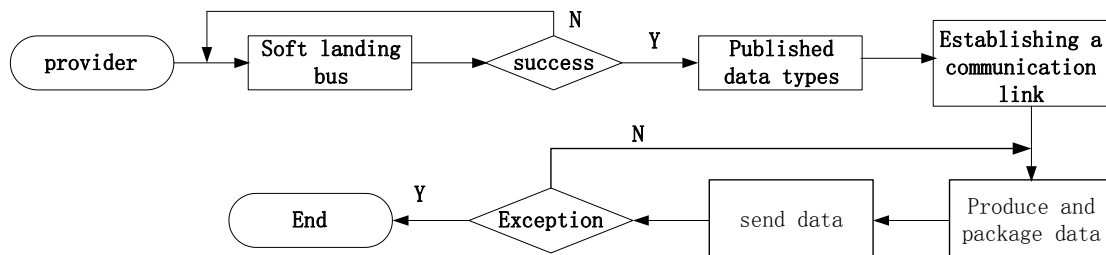


Fig.3 Provider-side program flow chart

e. Sending data.

This design task scheduling model is based on data flow-driven way to pass data, the data source is the A/D acquisition card periodically sampled. Event Service using Push delivery mode, each time the provider side to produce new data by calling the Push) interface to take the initiative to send (to the soft data bus. Soft bus to the need for this data and then the consumer side through the data channel data.

## **5. Summary**

Paper completed the virtual radio and Control System software bus is mainly based on CORBA technology to achieve multi-machine data exchange between functional modules.

## **References**

- [1] Command Ranging & Telemetry Cortex CRT-XL User's Manual [M]. 2013,12:3-12.
- [2] V. Bose, M.Ismert, M.Wellborn. Virtual radios [J]. IEEE Journal on Selected Areas in Communications, 2009, 17(4):591-602.
- [3] T. Grydeland, F. D. Lind, P. J. Erickson, J. M. Holt. Software Radar Signal Processing[J]. European Geosciences Union 2014.
- [4] J. Schamus, J. B. Y. Tsui, D. M. Lin. Real-time software GPS receiver [C]. ION GPS 2012 Proceedings, 2012: 24-27.