Underlying Communication Model of Joint Test Environment Based on Publish-Subscribe Mechanism

Zhihua Dong¹*, xiaodong Jiang¹, Heng Liu² and Binbin Hua¹

¹Baicheng Ordnance Test Center, Baicheng, Jilin, China
²Huayin Ordnance Test Center, Weinan, Shannxi, China

*Corresponding author: 576827011@qq.com

Abstract. Joint Test Inter-Connection Architecture (JTICA) contains application layer, simulation layer, and communication layer from top to bottom, which can construct a shared test space through connecting live, virtual, constructive simulation resources. Data distribution service is used as communication layer protocol, which can provide joint test with high real-time and large throughput. Heterogeneous systems and interactive models need to been mapped into equivalent entities in communication layer to set up the relationship between heterogeneous systems based on data-centric publish subscribe mechanism. Then heterogeneous systems are mapped into DomainParticipant, interactive models mapped into Topics, in communication layer. There are two map relationships between heterogeneous systems and DomainParticipants including “one-to-one” and “one-to-many”, which affect the communication efficiency of joint test. Communication delay of Prototype systems designed based on different map relationship are compared, and the result is referred to when selecting map relationship.

Keywords: joint test, Data Distribution service, simulation architecture, publish-subscribe.

1. Introduction
With the development of simulation standard, simulation systems based on different protocol co-exist. Joint test needs the interconnection and interoperation of heterogeneous systems including simulation. Data Distribution Service (DDS) was introduced into simulation to boost the communication performance recent years, especially the performance of real-time and throughput [1]. High Level Architecture (HLA) and Test and Training Enabling Architecture (TENA) are compared with DDS in Reference 2 and 3, and there are similar concepts and data distribution pattern between them. The performance of simulation system based on HLA is improved by integrating the system with DDS in reference 4. Layered Simulation Architecture is proposed in reference 5 and 6, which use DDS to interconnect heterogamous systems as the foundational communication level. But the architecture is only a concept, off-the-shelf software do not exist for joint test. The gateway between HLA and DDS is designed when developing application system, which demonstrated the interoperation feasibility of HLA and DDS legacies in some domestic universities and institutes [7, 8]. Although these studies achieve the data distribution of simulation systems based on DDS, the solutions only apply to the...
interoperation between DDS and certain architecture, and DDS is yet not the universal infrastructure for diverse heterogeneous systems [9, 10].

Learned from NCWare, SimWare and LSA, Joint Test Inter-Connection Architecture (JTICA) is proposed to be suitable for constructing joint test environments. The paper focuses on the communication layer of JTICA, to establish the map relationship between heterogeneous systems, interactive model and entities in communication layer, and find out the characteristics of communication performance.

2. JTICA architecture
JTICA can be logically divided into three layers, from bottom to top, including communication layer, simulation layer and application layer. Communication layer could provide network infrastructure of high throughput and low latency, called as "data bus". Simulation layer provide needed simulation services for diverse systems and maintain the shared test space. Applications run individually and respectively in application layer, on basic of joint test environments.

![Figure 1. JTICA hierarchy](image)

Communication layer in JTICA achieve the communication performance and data distribution service. Simulation layer realize the collaborative modeling and simulation, and shared information is described in the form of interactive models in the layer, which is called as Common Data Exchange Model (CDEM). Application layer provides universal interfaces for heterogeneous systems, by which they can access the shared space and interact with other systems.

3. JTICA communication layer model
3.1. Communication Model Based on Publish / Subscribe Mechanism
Communication layer contains communication entity and data model. Data model is distributed concept, which exists in conceptual space called as Global Data Space (GDS), and entities can publish and subscribe it according to information need. The publish/subscribe mechanism is applied in communication layer, and the publish/subscribe relationship between entities is on the basic of the systems in application layer. There is loose coupling between entities, and they are transparent each
other, called as anonymous publish/subscribe. Data model can be only published by an entity, but can be subscribed by more than one, as shown in Fig. 2.

![Publish/Subscribe Mechanism of JTICA](image)

Figure 2. Publish/Subscribe mechanism of JTICA

### 3.2. Information Exchange Mechanism in Communication Layer

As the communication infrastructure, information exchange is achieved by Data-Centric Publish-Subscribe (DCPS) mechanism of DDS. DCPS defines the communication model in the basic of data-centric publish and subscribe mechanism. The model is of platform-independent, so it is possible that entities in communication are mapped into equivalents in other protocol/platform.

There are five modules in DCPS, including infrastructure, Topic, publishing, subscription and domain, the relationship between them shown in Fig. 3.

![Relationship Between Entities in DCPS](image)

Figure 3. Relationship between entities in DCPS

There contain five types of entity in DCPS communication model including DomainParticipant (DP), Publisher, DataWriter (DW), DataReader (DR) and Data-Object. Data-Object takes the form of Topic. Publisher and DataWriter represent data sender, then Subscriber and DataReader acts as data receiver, Topic containing the content sent and received. The communication concept model of DCPS is illustrated in Fig. 4.

**Domain participant (DP) is a local process, which can be seen as the ‘container’ of other entities. A domain is a distributed concept that links all the applications (domain participants) able to communicate with each other. Data described in ‘Topic’ can be exchanged by several domain participants in the same domain. It represents a communication plane: only the entities attached to the same domain may interact.**
A Publisher is an object responsible for data distribution. It may publish data of different data types. DataWriter (DW) acts as a typed accessor to a publisher. The DataWriter is the object the application must use to communicate to a publisher the existence and value of data-objects of a given type. When data-object values have been communicated to the publisher through the appropriate data-writer, it is the publisher’s responsibility to perform the distribution. DataWriter is a strongly-typed entity, and only can write relevant data-object to publisher. Publisher is a common entity in a domain participant which can send all type of data-object to GDS.

A Subscriber is an object responsible for receiving published data and making it available to the receiving application. It may receive and dispatch data of different specified types. To access the received data, the application must use a typed DataReader (DR) attached to the subscriber. Thus, a subscription is defined by the association of a data-reader with a subscriber. This association expresses the intent of the application to subscribe to the data described by the data-reader in the context provided by the subscriber.

Data-object is described in the form of Topic, and is the basic unit of publication and subscription. Topic objects conceptually fit between publications and subscriptions. Publications must be known in such a way that subscriptions can refer to them unambiguously. A Topic usually associates a name, a data-type, and QoS related to the data itself. Topic can be programed by several languages such as IDL, XML, and UML. In addition, Topic with ‘Key’ parameter is distinguished by the value of ‘Key’, and topic objects with different ‘Key’ value represent the different instances.

4. Constructing mapping model of JTICA

Mapping model of JTICA includes two parts: heterogeneous systems in application layer mapping to the entities in communication layer which can send or receive data; interactive model between systems in simulation layer mapping to topic object. The number of entities that a system is mapped to is due to the number of other systems which interact with it.

4.1. Mapping rules from application layer to communication layer

Data exchange between systems is not achieved directly in application layer. The mapping is essentially the transformation of interaction between heterogeneous systems from application layer to communication layer. And the interactive relationship between entities in communication layer represents the one between systems in application layer. Assuming that there are n heterogeneous systems inter-connected based on JTICA, represented as $S_i (i \in [1, n])$. The mapping from application layer to communication layer is formalized as follows:
4.1.1. One to one mapping: Supposing that interaction exists between system \( S_1 \) and system \( S_x, S_y, S_z \), and interactive model is respectively \( CDEM_{ix}, CDEM_{iy}, CDEM_{iz} \), \( (x, y, z \in [2, n]) \). There does not exist intersection among the three interactive models. The four systems are accordingly mapped into entity \( S'_1, S'_x, S'_y, S'_z \) in communication layer. Publication and subscription exist between \( S'_1 \) and \( S'_x, S'_y, S'_z \) according to the relationship between \( S_1, S_x, S_y, S_z \) in application layer, as shown in Fig. 5 (a).

![Figure 5. Mapping rules from application layer to communication layer](image)

(a) one to one mapping          (b) one to many mapping

4.1.2. One to many mapping. Suppose that interaction exists between system \( S_a \) and other \( m \) systems, and between \( S_b \) and other \( n \) systems. Then \( S_a \) and \( S_b \) are respectively mapped into communication layer, which is formalized as follows:

\[
S_a \rightarrow S_{a1}, S_{a2}, \ldots, S_{am};\quad S_b \rightarrow S_{b1}, S_{b2}, \ldots, S_{bn}.
\]

The interactive model \( CDEM_{ab} \) between \( S_a \) and \( S_b \) is obtained through reasoning by the reasoning module in simulation layer. If \( CDEM_{ab} \) is mapped into \( Topic_{ab} \), and publication and subscription exist between \( S_{ap} \) and \( S_{bq} \) about \( Topic_{ab} \), then the interaction between \( S_a \) and \( S_b \) is fully mapped into the interaction between \( S_{ap} \) and \( S_{bq} \). In one-to-many mapping rule, interaction between systems in application layer can be replaced by only on pair communication entities in communication layer, which is obtain by mapping from the systems. Therefore there exist only a interaction between \( S_{a1}, S_{a2}, \ldots, S_{am} \) and \( S_{b1}, S_{b2}, \ldots, S_{bn} \), as shown in Fig. 5 (b).

4.2. Mapping rule from simulation layer to communication layer

The mapping in simulation is the interactive model mapping to data objects in communication layer. The mapping cannot change the relationship between entities according to the one between systems. In Fig. 6, \( CDEM_1 \) contains the interactive data between HLA system and TENA system, then OM1 in communication layer contain the same data between entity 1 and entity 2, which is obtained by mapping. So there is a one to one mapping relationship between interactive model and data object.
5. Demonstration and analysis

The chapter focuses on building a prototype system based on JTICA, analyzing the performance of two mapping rules. The application layer of the prototype system contains HLA systems, TENA systems, and Command and Control (C2) systems which simulate the node of radar, C2, and anti-air gun respectively. OpenDDS is applied as the communication layer.

5.1. Test Design

Ping-Pong test is employed to measure the delay between sending and receiving the same information. Supposing Topic object is the same size, and how the number of nodes in the communication layer maps from the HLA system affects time delay is tested. The test environment is shown in Tab. 1.

| Hardware Environment | Software Environment | Network Environment |
|----------------------|----------------------|---------------------|
| CPU: i3 3220          | Windows 7             | Switch 100Mbps      |
| Frequency: 3.2GHz     | OpenDDS 3.5           | NIC 100Mbps         |
| Memory: 2GB           |                      |                     |

5.1.1. Design 1: one-to-one mapping. C2 system is mapped to $DP_{C2}$, anti-air gun is mapped to $DP_{Gun}$, which are all domain participant. Several DataWriter and DataReader are constructed in $DP_{C2}$, and only one pair of DataWriter and DataReader is constructed in $DP_{Gun}$. $DP_{C2}$ creates a thread for every Topic, and the interactive relationship with $DP_{Gun}$ is set up respectively. When the number of Topic is 2, 4, 6, 16, 24, the delay is measured between $DP_{C2}$ and $DP_{Gun}$, as shown in Fig. 7.
5.1.2. Design 2: one-to-many mapping. C2 system is mapped to numbers of domain participants called $DP_{C2,i}$, which is the same number with the number of Topic. Interactions between $DP_{C2,i}$ and $DP_{Gun,i}$ are constructed. When the number of Topic is 2, 4, 8, 16, 24, the delay between $DP_{C2,i}$ and $DP_{Gun,i}$ is measured, as shown in Fig. 8.

5.2. Results and Analysis
The result of design 1 is displayed after data processing in Fig. 9. When the number of Topic increases, delay becomes larger. The time-delay caused by performance due to increasing threads can be excluded according to the CPU occupancy under different topic number.

Due to DDS rule, the type of publisher is not relevant to Topic, and DataWriter is dependent with Topic. DataWriter is in fact the subclass of Publisher, and datawriter instance need relevant publisher to publish the data. In design 1, numbers of datawriter share one common publisher. When different datawriters need to sending data synchronously, all the data are list in publisher queue, which will be sent to GDS according to QoS strategy. With the number of Topic rises, larger delay happens.
Figure 9. Delay of different number of Topic by comparison in “one to one” mapping rule

In design 2, delay is about 1000 $\mu$s, as shown in Fig. 10. Each publisher creates single datawriter, and each subscriber creates single datareader. The publish/subscribe relationship between $DP_{C2}$ and $DP_{Gun}$ is independent. Therefore, delay is not relevant to the number of Topic, and is around a constant value.

Figure 10. Delay of different number of Topic by comparison in “one to many” mapping rule

The test results of two mapping rules reflect the different affect to delay. "One to one" mode is suitable when there are few heterogeneous systems and simple interactive relationship. As the number of inter-connected systems increases more, the interaction relationship becomes more complicated, the number of topic increases accordingly. When a system interacts with more than 24 other systems, communication performance of “one to many” rule is prior to “one to one” rule. Interaction is achieved in single process in “one to one” mode, with simple development and consuming less resources, while numbers of process is created in “one to many” mode, with complicated development and consuming more resources.
6. Conclusion
The paper focuses on communication layer of JTICA which can be used for constructing joint test environment. The mapping rule from application layer and simulation layer to communication layer is proposed, supporting the construction of test space based on JTICA. Conclusions contain: (1) designing the communication model of JTICA; (2) heterogeneous systems in application layer mapping rule to entities in communication layer is proposed; (3) interactive model in simulation layer mapping rule to Topic is proposed too; (4) prototypesystem built up according to the mapping rules of “one to one” and “one to many” demonstrates the different performance of delay.

References
[1] DDS White Paper: Using DDS for scalable, high performance, real-time data sharing in next generation Modeling & Simulation systems. 2013.
[2] Joshi R, Castellote G-P. A Comparison and Mapping of Data Distribution Service and High-Level Architecture 2006.
[3] Powell E T. TENA and DDS: Some Thoughts [C].Proceedings of the Fall Simulation Interoperability Workshop, 2013.
[4] Lopez-Rodriguez J M, Martin R, Jimenez P. How to Develop True Distributed Real Time Simulations? Mixing IEEE HLA and OMG DDS standards [C].Proceedings of the Spring Simulation Interoperability Workshop, Boston, MA, 2011: 470-477.
[5] Martinez-Salio J-R, Lopez-Rodriguez J-M, Gregory D, et al. A New Approach for Converging LVC Simulation Architectures [C].Proceedings of the Spring Simulation Interoperability Workshop, San Diego, CA, 2013: 343-350.
[6] Topçu O, Oğuztüzün H. Layered simulation architecture: A practical approach [J]. Simulation Modelling Practice and Theory, 2013, (32): 1-14.
[7] Tian Shiqi. Research and implementation of distributed flight simulation system oriented real-time communication middleware [D]. University of Electronic Science and Technology, 2019.
[8] Zhang Zhihe, Shi Lusha, Zhang Bin, et al. A real-time simulation system based on DDS & HLA [J]. Electronic Design Engineering, 2017, 25 (10): 26-29.
[9] OMG. Data Distribution Service for Real-time Systems Version 1.2. www.omg.org/spec/DDS/1.2/, 2007.
[10] OMG. The Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification Version 2.1 [DB/OL]. www.omg.org/spec/DDSI/2.1, 2009.