Research on Cross-network Exchange Method of Enterprise Application Business Process Data

Xu Feng¹, Tao Sicheng²*, Li Gongliang¹, Xie Yang¹, Tao Yizheng¹

¹Institute of Computer Application, Chinese Academy of Engineering Physics, Mianyang 621000, Sichuan, China
²Swanson school of engineering, University of pittsburgh, Pittsburgh, PA 15261, USA

*Corresponding Author: Tao Sicheng; email:sit27@pitt.edu

Abstract: To solve the problem of efficient, safe and controllable transfer of enterprise-level application systems’ data across network, this paper proposes a data exchange method for application systems across physically isolated networks. Firstly, the analysis of the cross-network data exchange business model and the design of the business scenario were carried out, and two typical cross-network data exchange business models were condensed. Then, the design of the overall scheme and the detailed technical scheme of cross-network data exchange based on the one-way import device was carried out. Finally, the cross-network ferry middleware system was developed, and the experimental environment of the cross-network data exchange system based on the one-way import device was built, the personnel organization and the leave process information ferry across physically isolated networks were achieved, and the feasibility of cross-network data exchange scheme was verified.

1.Introduction
In order to improve cooperation efficiency and ensure information security at the same time, domestic and foreign government departments, military industrial groups and other units have built network infrastructures with different business positionings, and established different application systems for various types of businesses on different networks. Based on these application systems, related businesses are carried out. The Nuclear Weapon Complex (NWC) in the United States has multiple networks connecting national laboratories, factories and other related institutions, mainly including ESNet, SecureNet, Enterprise Secure Network (ESN), SIPRNet[1]. The domestic public security information network, scientific research office network of weapons and equipment research and production units and other related networks are strictly separated from the international Internet and other public information networks in accordance with the relevant national provisions [2].

With the rapid development of informatization and the increasing speed of knowledge update, the scientific research, design, production and management processes in the internal network require frequent exchange of information with external networks in different forms, which will inevitably generate a large number of input and output services [3]. At the same time, because different types of users in the organization need to carry out the same or interrelated services on different networks, the same and similar business application systems need to be deployed on different networks. The strict physical isolation between different networks cause the problem of incoherent application system business cross the network and fragmentation of business data which further resulted in the phenomenon of "information islands"[4]. Therefore, when constructing a cross-network application system, it is
necessary to design a business data exchange scheme to match it, and clarify the type and mode of data exchange of the cross-network application system. At present, a lot of work has been carried out at home and abroad on data exchange of application systems in the same network and data input and output across physically isolated networks\textsuperscript{[5]}, and some mature technology products have formed. For the research of data exchange between cross-network application systems, most of them are based on the design of software and hardware solutions and the discussion of security technology. However, for the application system business process data, especially for the research on low-level business process data exchange method based on the workflow engine\textsuperscript{[6]}, it lacks a complete and systematic solution.

Aiming at the problem of cross-network exchange of application system business process data, this paper analyzes the typical types of exchanged data, and proposes a set of modes and solutions for cross-network application system data exchange, trying to make the application system data transfer efficiently across network so as to solve the problem of business continuity of cross-network application systems.

2. Analysis of typical business data exchange models

When enterprise business process is executed across networks through application systems, the complexity of the data structure of different types of business data is inconsistent, and the frequency of data exchange is often inconsistent when different types of business are executed. According to the systematic survey and conclusion, according to the different requirements of data exchange period and integration depth of different types of businesses in the cross-network application system based on the relational database, this paper condensed two typical cross-network business models.

2.1. One-way ferry cross-network business model

The one-way ferry businesses is usually a one-way transmission of form data across the network, and the frequency of business execution is relatively low. For example, a certain account information in an application system in a network needs to be imported to another application system in another physically isolated network for aggregation. The data exchange period is usually from several weeks to several months. The business model under the scenario is a single ferry, and the one-way ferry of business data usually carries the automation of high-level business processes\textsuperscript{[7]}.

For the one-way ferry cross-network business in the application systems of related enterprises and institutions, it can be further condensed into two application modes:

1) Mode 1: Multi-network business object type. A single object businesses can be managed in a closed loop in the same network area, and data aggregation is required across the network. For this type of business, its cross-network data exchange needs are data import and aggregation to support decision-making statistics and analysis in a network. As shown in Fig.1(a), for the management of a unit's personnel, finance and other information, the structured and unstructured information generated from the application system in the external network needs be exported and imported into the corresponding application system in internal network, and summarized with the account information in the information system in internal network for further statistical analysis.

2) Mode 2: Multi-network life cycle type. An end-to-end business process involves multiple business objects. Different business objects are distributed and managed in different networks. In the life cycle of the end-to-end business process execution, business data needs to be exchanged across all networks. For this type of business, its cross-network data exchange mode is that multiple networks exchange business data back and forth in two directions to complete the closed-loop management of end-to-end business processes. For example, for a unit's management of its material procurement information, it is assumed that the unit has established three networks with different business positionings, which are temporarily called X network, Y network, and Z network. In the actual business execution process, firstly it needs to approve the purchase requisition information in the Z network, process and export the purchase requisition data to Y network and summarize it into the purchase plan, then process information in the purchase plan data and export it into the X network for bidding, the feedback process of the purchase execution information is started in the X network, across the Y network, and finally on the Z network for
material acceptance, its cross-network data transfer mode is shown in Fig.1(b).

![Figure 1. One-way Ferry Cross-network Business Data Transfer Mode](image)

2.2. Process integrated cross-network business model

Process-integrated business is a two-way cross-network transmission of process data based on workflow engine. During business execution, cross-network application system needs frequent two-way interaction. For example, during the execution of a business approval flow, because some of the process participants are on a business trip, but the process needs to be handled urgently, we can export the process execution data from application system in the internal network and publish it on the external network, and wait until the business execution in external network is completed, we it can be imported back to the internal network. The data exchange cycle requirement is usually within a few hours. This paper calls the business model in this scenario as process integration. The cross-network integration of workflow data usually carries the automated execution of low-level business processes and even the final process.

For process-integrated cross-network businesses, suppose that there are two physically isolated networks X and Y in an enterprise or institution. An application system S is deployed on both the X and Y networks, and a business workflow F is embedded in the system S, workflow F contains of 6 process links, namely Start, P1, P2, P3, P4, End. Because different process link executives need to work on different networks, by regularly and frequently exchanging the workflow execution result data on the two networks back and forth to other networks, we can make the instances of workflow F continue to execute subsequent links based on the results of the predecessor links from the front-end network in isolated X network and Y network, as shown in Fig.2.

When the instance of workflow F is executed, after the predecessor link is executed, the subsequent link can be specified to be executed on a certain network, or the link can be executed on multiple networks without specifying the execution network. Therefore, according to whether the execution of each process link is locked in the network, the cross-network execution mode of process-integration business can be divided into two categories:

1) Mode 1: Multi-network serial mode. In this mode, when the predecessor link is completed and submitted to the subsequent link, it is necessary to specify a certain network for the subsequent link to execute. In other networks where the subsequent link is not executed, the process operation will be locked. By exchanging the latest status data across the network, the execution status can be monitored in each network. As shown in Fig.2, it is assumed that the instance I of the workflow F is started in the X network, and the process initiator submits the execution of the next link of the process P1 to the X network, after cross-network data exchange occurs at the time T1 before the P1 link is not executed, the status of the current process execution can be monitored in the Y network, but because the execution network specified for the P1 link when the process is started and submitted is the X network, the process execution in the Y network will be locked and inoperable. It is assumed that after the execution of the process link P1 in the X network, the execution network specified when submitting it to the P2 link is the Y network. When data exchange occurs at the time T2, the execution of the P2 link can only be...
performed in the Y network, and the process execution in the X network will be locked, and so on... until the last link End of instance I is executed.

Multi-network serial mode

|       | Start | P1   | P2   | P3   | P4   | End   |
|-------|-------|------|------|------|------|-------|
| X network |      |      |      |      |      |       |
| Y network |      |      |      |      |      |       |

Legend
- Link actually executed in this network
- Link actually executed on other network
- Data exchange occurs at T1
- The first routing submitted to this network
- The first routing submitted to other network

Fig.2. Example Diagram of Process Integrated Multi-network Serial Mode

2) Mode 2: Multi-network parallel mode. In this mode, when the predecessor link is completed and submitted to the subsequent link, there is no need to specify the network for the subsequent link to execute. Each network obtains the latest status data through cross-network data exchange so that all networks can not only monitor the current process execution status, but also carry out execution of subsequent links. As shown in Fig.3, it is assumed that the instance I of the workflow F is started in the X network, and the process is submitted to the next link P1. At this time, P1 in the X network can be executed. After cross-network data exchange occurs at the time T1 before the P1 link is not executed, the current process execution status can be monitored in the Y network, and because the execution network of the P1 link is not specified when the process is started and submitted, the P1 link of F instance I in application system S can both be executed in the X network and Y network. It is assumed that the process link P1 is submitted to the P2 link by the executor after the execution is completed in the X network. Before the data exchange occurs, P2 can be executed only in the X network. After the data exchange occurs at T2, P2 can both be executed in the X network and the Y network, and the process executor of the link chooses to execute P2 on the Y network, and so on... until the last link End of the instance I is executed.
Multi-network parallel mode

| Start | P1 | P2 | P3 | P4 | End |
|-------|----|----|----|----|-----|

X-network

Legend

- Link actually executed in this network
- Link actually executed on other network
- Data exchange occurs at T1

Fig.3. Example Diagram of Process Integrated Multi-network Parallel Mode

In contrast, the multi-network serial mode and the multi-network parallel mode have advantages and disadvantages. The rule of the multi-network serial mode is simple, and there is no need to consider the problem of data consistency, but the disadvantage is that the efficiency of process execution is low, and the executor of the predecessor link needs to determine the network to execute the subsequent link, which has higher requirements for the process executor. The multi-network parallel mode has lower requirements for process executors. The process executors in the predecessor link do not need to consider the problem of the execution network of the subsequent links. Multi-network parallel makes the process execution efficiency higher, but related data trade-off rules must be formulated to solve the data consistency problem caused by the same link being executed multiple times in multiple networks. As shown in Fig.3, if the P1 link is executed on both X network and Y network but the processing results are different, we can choose to use the recent execution results as the final execution result of this link.

3. Design of cross-network data exchange scheme

3.1. Overall exchange scheme design

At present, there are two main methods that can be used to exchange data between application systems across physically isolation networks: disc burning[8] and gatekeeper transmission[9]. There are automated and semi-automated devices assisted by a robotic arm for disc burning currently[10]. This method has liberated human labor to a certain extent. However, on the one hand, the cost of its automated devices is high, and on the other hand, the waste of optical discs is still serious. All units still mainly adopt the full manual disc burning method. Compared with disc burning, gatekeeper transmission is a more efficient method. Gatekeepers are classified into unidirectional gatekeepers and bidirectional gatekeepers. One-way optical gate[11] uses the principle of one-way transmission of light to ensure the absolute one-way safe transmission of information.

This paper designs a data exchange system based on unidirectional optical gate device across the physically isolated networks, and its schematic diagram is shown in Fig.4. The network data sending end located is called the source network, and the network the data receiving end located is the target network. The one-way import hardware device is used to connect the source network and the target network. The one-way import device is composed of an internal end machine and an external end machine, and is physically separated by a unidirectional optical path inside the device. A cross-network ferry software middleware is deployed in the source network and the target network, which is used to receive the data packets from the application system in the source network and forward them to the one-way import device, or receive the data packets from the one-way import device in the target network.
and forward them to the application system. The application system in the source network and the target network are integrated with the cross-network ferry software middleware through the interface.

Process-integrated business requires frequent two-way transfer of business process data back and forth, so in this paper the data exchange system solution based on two-way single-lead device is used. By deploying two sets of single-direction devices with opposite data flows between the source network and the target network to support the data flow of business workflow back and forth between the two physically isolated networks, so the bidirectional exchange needs of process-integrated data is met and workflow integration is achieved.

![Cross-network Data Exchange System](image)

Fig.4 Data exchange system deployment diagram

The basic process of data exchange based on this cross-network exchange system is that the source network system A packages the exchange data and calls the integrated interface to send it to the cross-network ferry software middleware, and the source network middleware S transfers it to the exchange out file directory 1, the data can be transferred to the exchange in file directory 1 in the target network through the one-way import device 1, the cross-network ferry software middleware T in target network will grab the data packet from the exchange in file directory 1, and push the exchange data into the corresponding application system A1 specified in the target network by parsing the information in the data packet.

In order to ensure the accuracy and integrity of the transmitted data, the received data packets need to be verified on the target network. The verification method is that, before the A system calls interface of the source network middleware S for cross-network exchange, the size information of the exchange file is written into the exchange file data packet, and the target network ferry software middleware T obtain the file size information by parsing the received data packet, and at the same time measure the true size of the data packet. If the result of the measurement is equal to the result recorded in the packet, it means that no packet loss has occurred during transmission, and the received data packet is considered to be completed, otherwise middleware T will write the transmission failure information to the transmission error feedback file and send it to the exchange out file directory 2, and feed it back to the source network through the one-way import device 2, trigger break point resuming mechanism in the source network, and re-send the file package.

3.2. Application and service design

The cross-net ferry software middleware designed in this paper uses services as the basic unit for exchange and docking when cross-network data exchange. Multiple services within the same application system can participate in data exchange at the same time. In order to ensure the compliance...
of data exchange, the cross-net ferry software middleware designed in this paper provides the function of service registration for application system data exchange. Each service has a unique service code identifier during registration, and a service credential will be generated after registration. When the application system services exchange data through the cross-net ferry system, they need to submit their service credentials to the cross-net ferry middleware. Only when the service credential pass the identity authentication can the application system service exchange data through the cross-net ferry middleware.

As shown in Fig.4, applications refer to application systems such as A, B, C, etc. deployed within each network, and services refer to functional modules or functional services that exist within the application system. An application system may contain multiple services, such as A system in the source network contains services as1, as2, etc., and A1 application system in the target network contains services at1, at2, etc. Since each functional module can be identified as a service with a different code, this exchange system can meet the exchange needs of a functional module of the application system in the source network to send data to multiple target networks at the same time. For the process-integrated business, all workflows share the same service during data exchange, that is, the workflow service. The process model call is distinguished by the process model id and process instance id, and is used as the passed parameter of the service call.

3.3 One-way ferry service exchange data packet structure design
Current main data exchange technologies include ETL (Extract Transform Load)-based data exchange technology, EDI (Electronic Data Interchange)-based data exchange technology, JSON (JavaScript Object Notation)-based data exchange technology, WebService technology and XML-based data Exchange technology[12].

XML has powerful data representation capabilities, self-description capabilities, and strong extensibility. At the same time, it has achieved the separation of data and presentation[13]. XML has no fixed tags and attribute names, and users can write different tags to represent data as they need[14]. Based on these characteristics, this paper adopts XML as the medium in data exchange, so as to realize the exchange of structured and unstructured data across the network system. When data is exchanged between cross-net ferry systems, the data is transmitted in the form of compressed packets based on self-describing XML.

There are three types of files inside the one-way ferry service data package, including the original structured data, original unstructured data file, and an exchange data description file that the data objects of source service needs to package and exchange. The structured data information exchanged by the source service will be written to one or more XML files, the unstructured attachment data is directly packaged in the form of a file. The exchange data description file exchange_info.xml records the IP, port, URL, service identifier, data packet size and other information of the source service and the target service in detail, and it is used as the input file of control parameters to ensure successful data exchange. Taking personnel information one-way ferry as an example, the overall structure of the exchanged data packets is shown in Fig.5.
3.4. Structure design of process-integrated business exchange data packet

There are four types of files in the process-integrated business data package, including the structured data, the unstructured file, the exchange data description file and a workflow execution data file of the original main object associated with the process instance that the source service needs to package and exchange. The process execution data file workflow.xml contains the process instance id, the processing time of each link, the processing person, the processing opinion, the processing status and other information. When the process data is exchanged for the first time, the process definition information, such as the number of links, link name, link execution role type, predecessor or successor node, submission routing, etc. Taking the data exchange of the leave request workflow as an example, the overall structure of the data packet is shown in Fig.6.

4. System verification

Based on the above model analysis and scheme design, in this paper a cross-network ferry system based on the J2EE architecture[15] is developed, a single-guide system software and hardware experimental environment is built and a personnel management system organization service and a leave process application service are taken as examples to achieve the ferry of personnel organization information and the data exchange of leave workflow information based on multi-network serial mode across physically
isolated network, and verified the effectiveness of the exchange methods and technical solutions proposed in this paper.

5. Conclusion
This paper answered the question to a certain extent that how to ensure the continuity of the operation of cross-network applications by exchanging cross-network data safely and efficiently, and achieved the effective integration and utilization of data resources in different physically isolated networks. The scheme analyzed the cross-network business model and designs of the business scenario and the cross-network exchange businesses were divided into two categories: one-way ferry and process integration. The design of the overall scheme of cross-network data exchange, the design of application and service, and the design of the structure of the exchanged data packets based on the one-way import system were carried out. Based on the scheme, a cross-network data ferry system was developed, an experimental environment based on single-guide system was built, and the feasibility of the data exchange scheme was verified. However, that whether the two types of typical business data exchange models proposed in this paper can cover all scenarios of enterprise-level applications or not has not been strictly analyzed and demonstrated. Perhaps there are more or more complex data models for cross-network exchange needs. The business models can be further refined, and related research needs to be advanced continuely in later work.

Acknowledgement
This work was supported by the director’s fund projects of the computer application institute of CAEP under Grant Nos. “JS2019A02”and “JS2019A01”.

References
[1] Lu Hai. IT progress of the three major nuclear weapons laboratories in the United States [J]. Information and Electronic Engineering. 2009.8(in Chinese).
[2] Matrix O D. Understanding Network Security[M]. Computer Network Security,2005.
[3] Liu Dan. Research on the security technology of secret-related network information exchange [C], 2014 MIS/S&A academic exchange meeting(in Chinese).
[4] Arenas M , Pérez, Jorge, et al. Data Exchange Beyond Complete Data [J] .Journal of the ACM (JACM ) ,2013 ,60(4) :28 - 33 .
[5] Liu Meng, Chen Mingguang. Analysis of several data exchange technologies under network isolation [J], Science and Technology Consulting, 2008, 000(017) (in Chinese).
[6] Rafael F D S, Filgueira R, Pietri I, et al. A characterization of workflow management systems for extreme - scale applications[J]. Future Generation Computer Systems,2017,75: 228-238.
[7] Han Xiaoliang, Wen Chunhai, Chen Guo. Exploration and research of BPM system implementation method [J]. Information Technology and Informatization, 2020(4):75-77(in Chinese).
[8] Li Hongmin, Chen Guangping, Ling Ronghui, et al. Design and Implementation of Information Exchange Mode between Intranet and Extranet[J]. Communication Technology, 2009(4): 120-122(in Chinese).
[9] Deng Mingjun. Design and implementation of unidirectional file transmission system based on single fiber [D]. Guangzhou: South China University of Technology, 2011(in Chinese).
[10] Deng Zhiyong. Design of DVD-R disc automatic pre-write machine [J]. China High-tech Enterprises, 2012(21): 20-22(in Chinese).
[11] Liu Yongfu, Jiao Binliang, Jin Guoqing, et al. Design and implementation of network information security non-feedback one-way transmission system [J]. Computer Security, 2010(11): 38-41(in Chinese).
[12] Lu Yibo. Data exchange method applied to heterogeneous e-government platform [D]. Beijing: Beijing University of Posts and Telecommunications, 2018(in Chinese).
[13] Li Zhijian. Heterogeneous Data Exchange Technology Based on XML[J]. Journal of Leshan Normal University, 2012, v.27;No.195(012):35-37 (in Chinese).
[14] Anna K, Ching H H. Towards increasing reliability of clouds environments with RESTful web services [J]. Future Generation Computer Systems, 2018, (87): 502-513.
[15] Johnson R. J2EE development frameworks. COMPUTER. Volume:38, Issue:1, Pages:107-110, Published: JAN 2005.