Improvement of Architecture Based on Hierarchical Message Bus

Zhengwei Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 1328217689@qq.com

Kun Wang
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 644790698@qq.com

Jinhui Li
School of Computer Science and Engineering
Xi'an Technological University
Xi'an, 710021, China
E-mail: 715319152 @qq.com

***Abstract*—The traditional architecture based on the hierarchical message bus has defects such as strict component requirements and lack of support for reuse. The article introduces message-message modem and interface-message modem construction middleware to propose a new hierarchical message bus. The structure new hierarchical message bus solves the problem that the hierarchical message bus architecture requires too strict system components, and the components that fail to provide corresponding interfaces lack support for reuse. Firstly, the paper introduces the traditional hierarchical message bus architecture and its existing defects. Secondly, the article proposes new hierarchical message bus. based on the hierarchical message bus architecture. Finally, the article summarizes the architecture new hierarchical message bus proposed in this paper.**

Keywords-Hierarchical Message Bus; Middleware; Modem; New Hierarchical Message Bus

I. INTRODUCTION

One of the core questions of software architecture design is whether it is possible to use repetitive architecture patterns, that is, whether it can achieve software reuse of the architecture [1]. The structure of the traditional architecture is mostly based on the C/S or hierarchical mode. Although the C/S and hierarchical modes are easy to understand and have high efficiency, they are highly targeted and expand some large systems in specific fields. And the reuse of components is not very friendly. In reality, there is often a need for an architecture with both a stable static structure and sufficient component reuse. Based on the traditional hierarchical message bus structure, this paper further expands and improves the hierarchical message bus for different types of components, and develops intermediate components with wider application scope and better reusability to maximize software reuse. The effect is to achieve greater efficiency of component reuse.

DOI: 10.21307/ijanmc-2020-023

II. KEY TECHNOLOGY

Based on the traditional hierarchical message bus structure (HMB), this paper constructs two convenient middleware message-message modem (MMM) and interface-message modem (IMM) for conversion and adjustment of different standard types of components, so as to achieve greater efficiency of component reuse. This article developed two general middleware:

1) Message-Message Modem (MMM), a non-standard component based on the message interconnect interface, uses MMM middleware to send and receive and parse messages between the component and the message bus.

2) Interface-Message Modem (IMM), calling non-standard components of the interface, using IMM middleware to complete the conversion and transmission between the interface and the message.

III. ARCHITECTURE DESIGN

A. *Traditional hierarchical message bus architecture*

The Hierarchy Message Bus (HMB) architecture is a software architecture proposed by Yang Fuqing et al.[2]. This architecture is based on a hierarchical message bus and supports the distribution and concurrency of components.

All components pass through the message bus Communicate as shown in Figure 1.

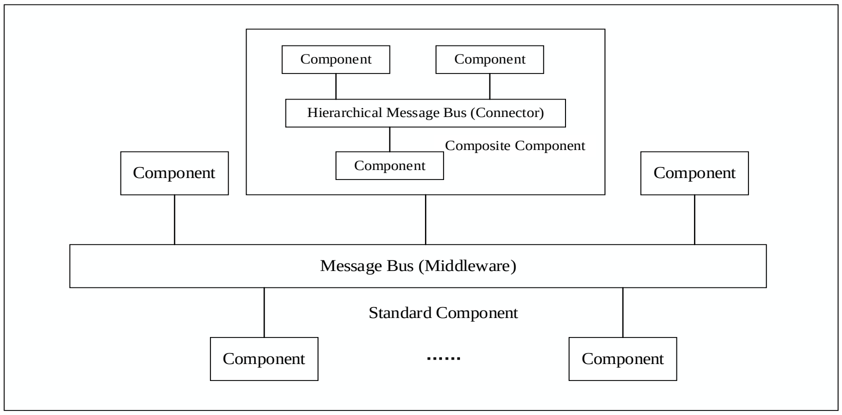

Figure 1. Hierarchical message bus architecture diagram

The connection of the system is the message bus, which is responsible for the dispatch, delivery, filtering and return of the processing results of the message [4]. The components in the system are all connected to the message bus and register the type of messages that interest them to the message bus. The component sends out the corresponding message according to the needs of the system, and the message bus is responsible for dispatching the message to all components interested in the message in the system. In the HMB architecture, the message is the only way for all components in the system to communicate. When the component receives the message from the message bus, it responds to the message according to its own state, and then returns the corresponding processing result to the target component through the message bus [5].

Since all the components in the system are connected through the message bus, it is not required that each component has the same address space or be limited to one machine, so this style can better describe the distributed concurrent system.

The system and the components that make up the system are usually more complicated, and it is difficult to obtain a complete understanding of them from one perspective, so a good software engineering method often models the system from multiple perspectives, generally including the static structure and dynamic behavior of the system And functions [6].

1) Component model:

The component model of HMB includes three parts: component interface, static structure and dynamic behavior, as shown in Figure 2 [3] .

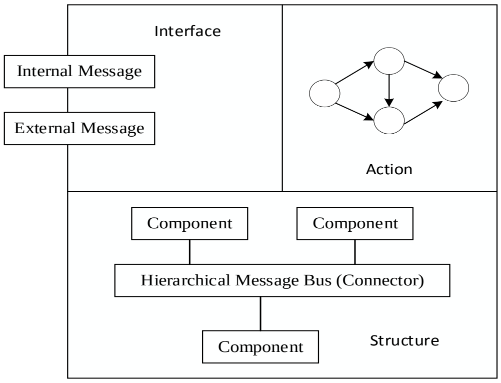

Figure 2. Component model of HMB architecture

The interface part of the component represents the set of interfaces supported by the component. Common ones are internal standard message interface, external message standard interface and external interface, and each interface defines a set of input and output messages. The behavior part of the component is represented by the finite state automaton with output. When the component receives the message, the finite state automaton responds to the received message according to the current state, resulting in a state transition. The structural part of the component represents the internal structure of the composite component. The composite component is composed of an atomic component and a local message bus. Several local components can be connected to the local message bus. The local message bus is a simple representation of the message bus. The role of the message bus it shows that it provides a unified integration mechanism for the entire system and components at all levels.

2) Component interface:

Generally speaking, the component interface represents the interaction path between the component and the external environment. In the HMB architecture, the component interface is a message-based interconnection interface, where the interface represents the set of messages sent and received by the component, and the components communicate with each other through messages, which together constitute the component interface. In the general system defined by the interconnection interface, the connection between components is usually a fixed match between the requester of the service and the service provider, while in the system defined by the HMB architecture, the component has external messages After the response, it will cause a change of state. Therefore, it can be found that, after receiving the same message, a component may have different responses at different times and in different states.

3) Message bus:

The message bus is the connection of the system. In the system, all components form a component-message response registration form by registering the types of messages of interest to the message bus. The message bus locates the corresponding message based on the type of message it receives and the information in the component-message response registration table, passes the message to the corresponding responder, and is responsible for returning the processing result. When necessary, the message bus also needs to filter and block specific messages, as shown in Figure 3.

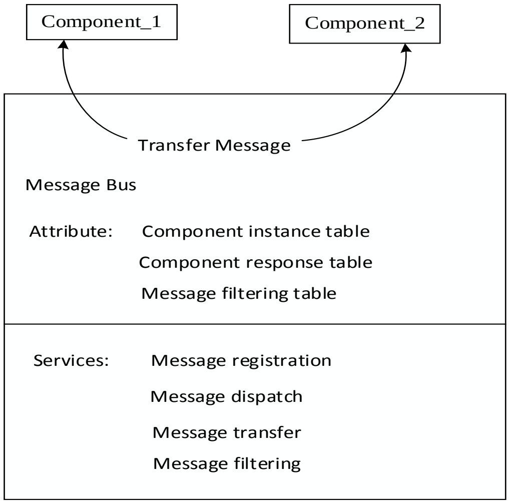

Figure 3. Message bus structure diagram

Message registration: The component registers the response message set with the message bus.

Message dispatching and delivery: Messages are passed between components through the message bus. The message bus dispatches messages to components interested in the message according to the component-message response registration form, and returns the processing results [8].

Message filtering: The message bus implements message filtering between components through message conversion or message blocking, to avoid message conflicts and message mismatch problems during component integration.

Through the above analysis, it can be found that the HMB architecture improves the degree of cohesion between the various components, while reducing the degree of coupling between the components, but there are still certain defects:

First, if there is an existing well-functioning component, but due to the different form of the interface, it will not be used in this system, and it is necessary to re-develop the same functional component that fully complies with the interface standard, which is in line with the architectural style to improve software reuse Development efficiency is contradictory.

Second, although the message filtering mechanism of the message bus has a certain message matching function, it can only process components that interact through messages, but not components that interact through interface calls, and it may be necessary to match different messages. Modifying the design of the message bus has caused some instability to the system.

Third, the architectural style of HMB has too strict requirements on the components that make up the system. Only strictly qualified components can be used, and there is a lack of reuse support for components that fail to provide corresponding interfaces.

Therefore, it is necessary to improve and expand the architecture based on the hierarchical message bus.

B. *New hierarchical message bus architecture*

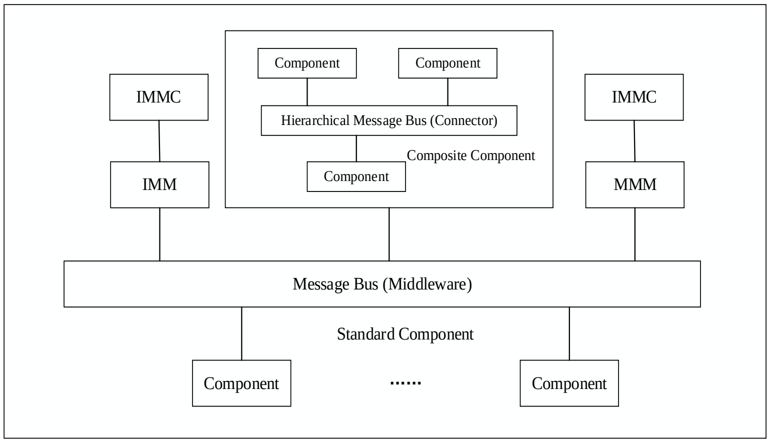

Figure 4. Schematic diagram of the NHMB architecture

This paper improves the traditional HMB architecture and becomes a new HMB architecture style (NHMB, New Hierarchical Message Bus), as shown in Figure 4 below.

Compared with the traditional HMB architecture, the NHMB architecture adds non-standard components of interface-message modem and message-message modem and message interface. Among them, the message-message modem (MMM), a non-standard component based on the message interconnection interface, uses MMM middleware to send and receive and parse messages between the component and the message bus. The interface-message modem (IMM) calls the non-standard components of the interface and uses IMM middleware to complete the conversion and transmission between the interface and the message.

Complex components in the system can be decomposed into relatively low-level subcomponents, which are connected through a local message bus [7]. Such complex components are called composite components. If the sub-components are still relatively complex, they can be further decomposed, and the decomposition continues in this way, and the entire system forms a tree-like topology. The end nodes of the tree structure are called leaf nodes. They are atomic components in the system and no longer contain subcomponents. The entire system can be used as a component to integrate into a larger system through a higher-level message bus. Therefore, the entire system and the individual components that make up the system can be characterized in a unified manner.

The improved NHMB architecture model in this paper is similar to the HMB structure, which has three parts: interface, static structure and dynamic behavior.

Interface: The interface in NHMB can be either the originally supported interface with message interaction function or an interface that only provides formal calls.

Static structure: The composite components in NHMB are also based on NHMB's architectural style, not the original HMB's architectural style.

Based on the above two changes, the architecture can greatly reduce the requirements for the specificity of components, so that existing components can be reused to a greater extent.

1) Component model:

The improved NHMB architecture in this paper is similar to the HMB architecture, which has three parts: interface, static structure and dynamic behavior.

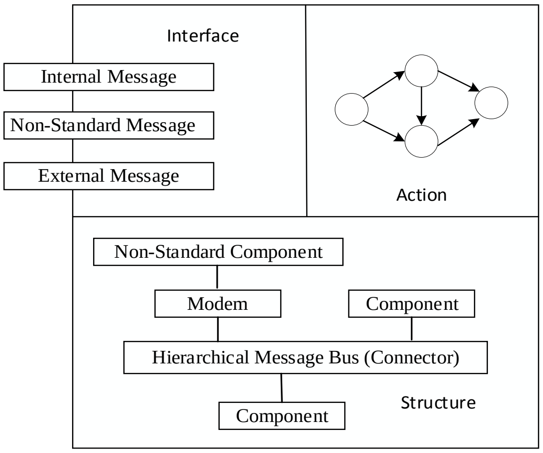

Figure 5. Component model of NHMB architecture

2) Component interface:

The NHMB architecture achieves the indirect use of such components by adding an interface-message modem or a message-message modem between the type of component and the message bus. From the overall system perspective, although a module is added to the system, it has no effect on other modules of the system. In addition, there is no need to modify the message bus. In a sense, the NHMB architecture can achieve the effect of supporting more types of components by adding corresponding middleware modules, thereby greatly improving the system development efficiency and system stability.

3) Modem middleware:

In order to support non-standard components, this article adds interface-message modem or message-message modem middleware to complete the reception and analysis of messages in the system in the NHMB architecture. So as to better adapt to the development of software systems, and greatly reduce the complexity of software system development, maintaining the stability of the system. The working principle of the modem is shown in Figure 6 below.

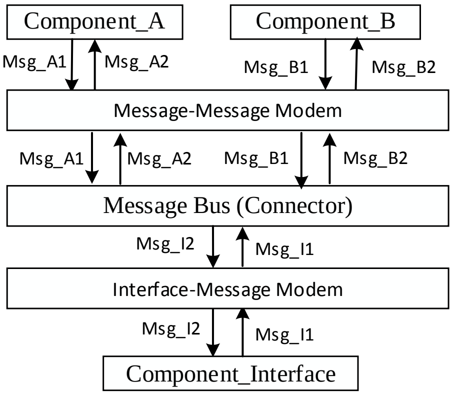

Figure 6. Message modem middleware

Component A, Component B, and Interface component send messages to the message bus together. Component A and Component B are standard components. Component Interface is an interface component. Two different types of components first send their own characteristic messages Msg_A1, Msg_B1, Msg_I1 to MMM and IMMM, then MMM and IMMM parse the message Msg_A1, Msg_B1, Msg_I1 from the component according to the rules prescribed in advance with the standard component and interface component, and then repackage the message into Msg_A2 according to the format that the message bus can handle. Msg_B2, Msg_I2, and forwarded to the corresponding message bus, so as to achieve indirect communication between component A, component B and interface component Interface to the message bus, the process of returning messages is similar to the process of sending messages.

4) Structural characteristics of NHMB system:

High robustness: The message-message modem/interface modem added in this paper has a strong resolution and modulation mechanism, and has strong fault handling and recovery capabilities for non-standard components based on the message interconnect interface.

Higher development efficiency: the system can use mature functional components, even if its interface does not meet the needs of the system, it can be easily integrated into the system by adding a simple modem.

Support the distributed operation of each functional module: since all components in the system interact through the message bus, the system can pass the messages of each module to each other without affecting the distributed concurrent operation.

More secure: The NHMB architecture designed in this paper introduces a modem. In order to increase the security of the system and prevent the communication of the message bus from being maliciously destroyed, authentication and encryption are adopted in the NHMB architecture style to ensure the security of message transmission. The introduction of modem not only reduces the requirements for components, increases the range of component selection, improves the scalability of the system, but also provides a convenient way for the system to expand the security mechanism, that is, the modem developed at this time not only contains message interface matching It also provides special functions for encryption and decryption.

Easy to expand and maintain: Each component in the system is independent of each other, so when expanding or modifying system functions, you can only operate the specific modules in it, without affecting other modules.

IV. CONCLUSION

The NHMB architecture proposed in this paper has the advantages of high robustness, higher system development efficiency, easy expansion and easy maintenance. By adding two intermediate components of message-message modem and interface-message modem on the basis of the traditional HMB architecture The operation greatly reduces the complexity of software system development, maintains the stability of the system, and solves the problem that components with good functions cannot be used because of different interfaces, and the same functional components and HMB system that meet the interface standards need to be redeveloped The structure has too strict requirements on the components constituting the system, and lacks reuse support for components that fail to provide corresponding interfaces.

REFERENCES

[1] Zhang Yousheng. The third article of software architecture series-style of software architecture [J]. Programmer, 2002(08): 45-47.

[2] Zhang Shikun, Wang Lifu, Chang Xin, Yang Fuqing. Software Architecture Description Language Based on Hierarchical Message Bus [J]. Chinese Journal of Electronics, 2001(05): 581-584.

[3] Zhang Shikun, Wang Lifu, Yang Fuqing. Software architecture style based on hierarchical message bus [J]. Chinese Science Series E: Technology Science, 2002(03):393-400.

[4] Zhang Yousheng, software architecture [M]. Version 2. Beijing: Tsinghua University Press, 2006.

[5] Zhang Shikun, Wang Lifu, Chang Xin, Yang Fuqing. Software Architecture Description Language Based on Hierarchical Message Bus [J]. Acta Electronica Sinica, 2001(05):581-584.

[6] Yang Fuqing, Mei Hong, Li Keqin.Software reuse and software component technology [J]. Journal of Electronics, 1999(02):69-76+52.

[7] Qin Guorong, Zhang Shikun. Research on Dynamic Simulation and Evolution of Software Architecture Based on Hierarchical Message Bus [J]. Computer Science, 2001(03):75-77.

[8] Wang You, Rong Mei, Zhang Guangquan. Application system model based on hierarchical message [J]. Computer Science, 2006(11): 285-288+292.