

Review Article

Trends in Architecture and Middleware of Radio Telescope Control System

Jun Li ^{1,2}, Na Wang,^{1,3} Zhiyong Liu ^{1,3}, Yining Song,^{1,2} Ning Li ^{1,2}, Lingming Xu ¹,
and Jili Wang ¹

¹Xinjiang Astronomical Observatory, Chinese Academic of Sciences, Urumqi 830011, China

²University of Chinese Academy of Sciences, Beijing 100049, China

³Key Laboratory of Radio Astronomy, Chinese Academy of Sciences, Nanjing 210008, China

Correspondence should be addressed to Zhiyong Liu; liuzhy@xao.ac.cn

Received 11 April 2021; Revised 17 May 2021; Accepted 11 June 2021; Published 29 June 2021

Academic Editor: Yu Liu

Copyright © 2021 Jun Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The control system is the central control unit of the radio telescope. It is used to monitor, control, coordinate, and manage software and hardware systems so as to satisfy the requirements of high-precision control in astronomical observation of radio telescope. The control system architecture is the foundation for the implementation of the control system, which determines the stability, scalability, and maintainability of the control system. Furthermore, the architecture design of the control system is closely geared towards the technological development of radio telescope and computer software architecture. In this article, we analyze the characteristic of the control system of a radio telescope in various steps and discuss the development of their architecture and middleware framework. System architecture and middleware framework of control system also serve as a useful reference for the design of other radio telescope control systems.

1. Introduction

The development of radio telescope drives the research of radio astronomy dramatically. Meanwhile, the progress of astronomy has put forward higher requirements for radio telescope. In the early days, radio telescopes have a simple equipment structure, a single business logic functionality of software [1, 2]. However, with the increase of scientific needs, the continuous breakthroughs in key technologies of radio telescope, the degree of automation, and the requirements of antenna accuracy are also continuously improved. Therefore, modern radio telescopes have a wide distribution and variety of hardware devices, which makes the business logic between the hardware devices complex and various, leading to an exponential increasing of control system complexity [3–5].

Since the 21st century, radio telescopes have developed rapidly in both single dish and array antennas [6, 7]. For single dish antenna, Green Bank Telescope (GBT) [8], Tianma Radio Telescope (TMRT) [9], Sardinia Radio

Telescope [10], Five-hundred-meter Aperture Spherical radio Telescope (FAST) [11], Qitai Radio Telescope (in the construction phase) [12], and other radio telescopes [13–15] with a large aperture, wide band, high sensitivity have been built. Astronomical observation introduces large quantities of auxiliary control and measuring equipment, mainly including active surface control, subreflector adjustment, laser interferometry, meteorological measurements, and electromagnetic environment monitoring. These devices are helpful for real-time tracking and high-precision pointing of astronomical observation objects by radio telescope. Furthermore, the complicated radio telescope puts forward higher requirements on the logical structure of architecture design of the control system, which greatly increases the complexity of the control system. For array antennas, large-scale array antennas such as Atacama Large Millimeter Array (ALMA) [16], LOFAR [17], Giant Metrewave Radio Telescope (GMRT) [18], Australian Square Kilometre Array Pathfinder (ASKAP) [19], and MeerKAT [20] have been built, as well as the planned Square Kilometre Array (SKA)

[21]. The multinode collaborative work of these array telescopes presents new challenges to the expansibility and cooperativity of the control system.

The control system is the core of the radio telescope. It is used to coordinate and manage the antenna, receiver, terminal, measurement sensor, and other subsystems of radio telescope. The software of these subsystems uses different technologies, such as cross platforms, multiple languages, and interface definition [22]. The design of the control system needs to comprehensively consider the runtime environment, scientific requirements of each subsystem, and incorporate the design concepts of software architecture, especially the middleware framework that accompanies the development of software architecture. In this article, we will analyze the control system of modern radio telescope and focus on the control system architecture, middleware framework, and their development trend. Section 2 analyzes the control system architecture and its development trend; Section 3 analyzes the middleware framework and its development used by the control system; Section 4 summarizes the full text.

2. Control System Architecture

Control system architecture is the general planning for the design of a radio telescope and is the foundation of control system implementation. Its design is directly associated with scientific requirements, radio telescope key technology, and computer software technology. Control systems in different periods use different software architectures; hence, these architectures are classified into three stages.

2.1. The First Stage Is before the 1990s. Owing to the relatively single scientific requirements, fewer hardware devices of radio telescope, simple device functions, and less business logic function between subsystems, the control system was built with a centralized architecture to satisfy the requirements of radio telescope. The centralized architecture was mainly applied in early radio telescopes, such as the 30 m Millimeter Radio telescope control system [1], Effelsberg control system [23], Nobeyama Radio Observatory 45 m (NRO45M) telescope control system [24], and Parkes control system [25]. This type of control system centrally processed all the business logic of antenna, receiver, and other subsystems. The control system adopted code level or library file call, which had the advantages of simple structure, short development cycle, and easy deployment. Notwithstanding, code dependence between antennas, receivers, and other subsystems was strong, resulting in a high degree of coupling and poor extensibility of the control system. When the hardware, software interface, or library file of the subsystem changes, the entire control system may need to be updated, resulting in poor maintainability and portability of the control system.

2.2. The Second Stage Is from the 1990s to the Beginning of the 21st Century. With the increase of science requirements, the growing number of hardware devices and auxiliary devices,

devices function becoming more complex, and more complicated business logic between subsystems, centralized control systems could not flexibly add, delete, or update devices or functions. Therefore, distributed architecture satisfied these requirements of radio telescope by using modular design, such as Effelsberg control system [2], NRO45M telescope control system [26], and GBT enterprise software [27]. This type of control system coordinated and managed the antenna, receiver, terminal, and other subsystems of the radio telescope. Any of the subsystems was divided into one or more modules. For example, the antenna subsystem was compartmentalized into an active surface, servo, and other modules. The development, testing, deployment, and operation of these modules were carried out independently, and a unifying communication mechanism is adopted between the modules. Above all, the design of the control system modules needed to consider all requirements of the radio telescope and the dependency relationship with other modules. In addition, the division of module boundaries and the definition of interface parameters affected the performance of the control system, making the predesign of the control system more complicated. In a word, a high-performance control system could add, delete, modify, or update functions or modules at any time so that the control system has good scalability and easy maintenance.

2.3. The Third Stage Is from the Beginning of the 21st Century to the Present. As a result of the rapid growth of scientific requirements, the growing types of hardware devices and auxiliary devices, the complex functions of devices, and more complicated business logic between subsystems, the second-stage distributed control system had poor support for multiprogramming language and cross-platform. However, the middleware framework solved these problems, which shielded complex implementation details and provided a unified interface. Therefore, a distributed architecture based on middleware framework was organized into two ways: (1) Service-Oriented Architecture (SOA) based on middleware framework [28], such as Large Millimeter Telescope (LMT) monitoring and control system [29] and TMRT distributed control software [30]; (2) the combination of SOA and event-driven architecture (EDA) based on middleware framework, such as ALMA software [31], ASKAP monitoring and control system [32], and GMRT control and monitoring system [33]. This type of control system coordinated and managed the software and hardware subsystems, such as antennas, receivers, and terminals divided any subsystem into one or more components. For example, the terminal subsystem was classified as components such as VLBI joint measurement, single antenna observation, and antenna measurement. These components were independently developed, tested, deployed, and run and were connected by a middleware framework. Middleware framework simplified complex interfaces such as operating system, network, and database into simple and unified interfaces, integrated SOA, or the combination of SOA and EDA into its design. Among them,

the control system based on SOA is mainly used for the analysis and execution of control commands. In addition, the control system based on EDA is mainly used for monitoring devices and control systems. When designing the control system, the interface parameters of each component need to consider the overall requirements of the radio telescope and the dependence on other components. At the same time, the division of component boundaries and the definition of interface parameters affect the performance of the control system and increase the complexity of the control system's early design. Compared with the control system of the second stage, the control system has lower complexity, better scalability, and higher maintainability and supports more platforms and programming languages.

In conclusion, software architecture applied to control system changes from a centralized architecture to a distributed architecture without a middleware framework and then to a distributed architecture based on middleware framework. Table 1 compares the advantages/disadvantages of the three-stage control system in terms of communication method, application requirements, business logic, and development difficulty. The first stage centralized control system has the benefits of simple development, short developing period, low cost, and easy deployment. However, this type of control system has some shortcomings, mainly including the inability to flexibly add, delete, and update functions and subsystems; it has the characteristics of weak extension, poor maintenance, and low portability; application code between subsystems has strong dependencies. The advantages of the second-stage distributed control system mainly include the following: modular design improves code reusability and development efficiency; modules can be independently developed, tested, deployed, and run; system is easy to add, delete, or modify modules or subsystems. Notwithstanding, the control system at this stage has some shortcomings. For example, the early stage of control system design is complex and cross-language and platform support is poor; calling relation between subsystems is complex and difficult to maintain; software engineers use complex underlying interfaces such as operating systems and networks to implement control system. The merits of the third stage distributed control system mainly include the following: middleware framework provides a simple, unified standard interface for control system development; the component granularity is smaller than the second-stage module granularity; control system has the characteristics of good scalability, high maintainability, and supports multiplatform cross-language. However, this type of control system has some shortcomings. For example, the division of component boundaries affects the performance of the control system; the logical relationship between components is complex, which makes it difficult to operate, test, and deploy the control system.

Because there are many kinds of radio telescope hardware devices, complicated structure, huge functions, and complex business logic, modern radio telescope control systems use a distributed architecture based on a middleware framework.

3. Application of Middleware Framework

Modern radio telescope control systems are mostly constructed using the distributed architecture based on the middleware framework in the third stage. Middleware framework is the piece of software that is located between two independent applications: an independent application and independent system. Middleware framework hides the details of the operating system, network, or database so that developers only need to pay attention to business logic [34]. Middleware framework provides a standard protocol for communication between subsystems, which is used to connect different layers (high level, low level, and device layer [35], as shown in Figure 1) in control system architecture. When designing a control system, different types of architectures use different middleware frameworks that provide mechanisms to simplify the development of the control system, such as component or service encapsulation and interaction rules. Middleware frameworks are used by the radio telescope control system (see Table 2); these middleware frameworks are divided into high-level coordination middleware framework and high- and low-level management middleware framework. The control system constructed by the high-level coordination middleware framework can coordinate the high level of Figure 1. This kind of middleware framework mainly includes CORBA [36] and ICE [37]. The control system constructed by the high-level management middleware framework can coordinate and manage the high level and low level of Figure 1. This type of middleware framework mainly includes ACS [38], the combination of CORBA (or ICE) and EPICS [39], and Tango [40].

3.1. High-Level Coordination Middleware Framework. The control system is constructed by the high-level coordination middleware framework and provides some common functions such as information management, organization, and mission planning to the high level of Figure 1. This kind of middleware framework includes CORBA and ICE, which provides a soft bus with functions such as service transparency, communication shielding, and information exchange. Middleware framework also provides a neutral language for defining interfaces, which can be compiled into different programming languages to implement operation and parameter transfer between subsystems.

CORBA is a middleware framework with a soft bus function proposed to solve the interconnection of distributed heterogeneous environments [41]. The core Object Request Broker (ORB) of CORBA is used to build a soft bus. The soft bus separates a client from a server and provides transparent network access services for a client [42]. CORBA's Interface Definition Language (IDL) is mapped to a variety of programming languages to generate static call interfaces and static IDL Stubs [43]. CORBA is used not only in early radio telescopes, such as LMT control system [44], but also in industrial equipment and scientific devices [45–48]. CORBA provides the foundation for middleware frameworks such as ACS and Tango and also provides design

TABLE 1: Comparison the characteristics of the three-stage control system.

| Name | The first stage: centralized control system | The second stage: distributed control system without middleware framework | The third stage: distributed control system based on middleware framework |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Communication method | TCP/IP, UDP | TCP/IP, UDP | Middleware (ICE, TANGO, EPICS) |
| Granularity | — | Coarse | Fine |
| Application requirements | Single scientific requirements, centralized hardware distribution, simple equipment functions, single business logic function | Many kinds of hardware equipment and auxiliary equipment, complex functions of equipment, complicated business logic | Many types of hardware and auxiliary equipment, very complicated equipment functions, and quite complicated business logic |
| Call level | Code level, library | Modularity | Component, service |
| Invocation style | Call between functions or modules | Interprocess call | Interprocess call |
| Business logic | Centralized processing of all functions | Extract core business and improve module reuse | Increase the dispatch center to manage the service, the service call is transparent, and there is no need to care about dependencies |
| Development difficulty | Low | High | Medium |
| Maintainability | Weak | Strong | Medium |
| Deployment method | Centralized deployment of all functions | Distributed deployment with independent modules | Independent distributed deployment of components and services |

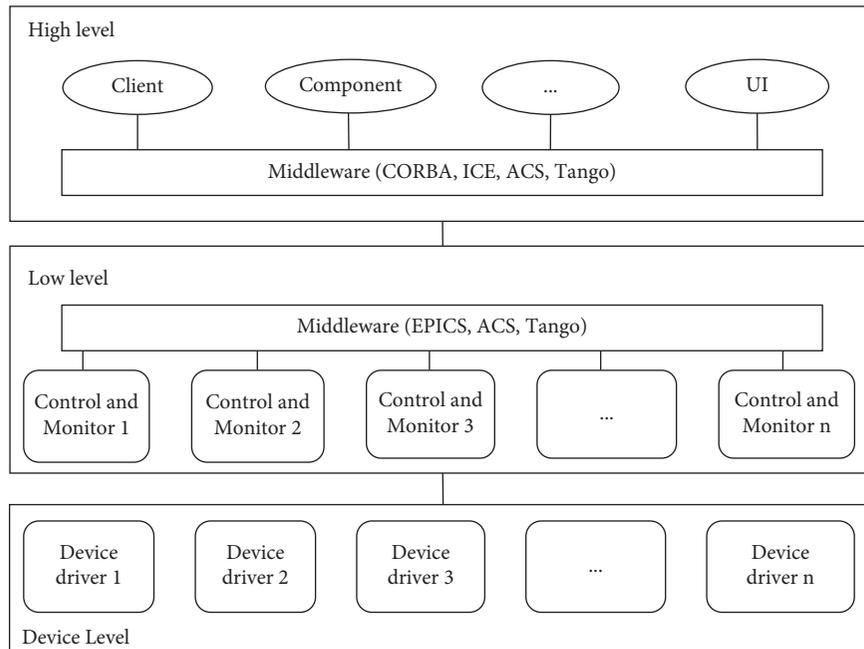


FIGURE 1: Middleware framework applied to a control system architecture of radio telescope.

ideas for ICE. However, CORBA does not provide a hardware device driver interface and has shortcomings such as complex structure, long learning cycle, high implementation cost, stopped development, and being discontinued.

ICE, absorbing the design ideas of CORBA, is a middleware framework based on remote procedure call (RPC) [49]. ICE Core not merely shields complex interfaces such as network and operating system but also provides transparent access services. Therefore, developers only need to focus on business logic [50]. Specification Language for ICE (Slice) is

a neutral language similar to CORBA IDL, which sets the contract or interface between a client and a server and provides functions including data persistence and serialization. In addition to being used in radio telescopes, such as TMRT distributed control software [51], ICE is also used in large-scale software systems, scientific devices, and industrial equipment [52, 53]. Furthermore, the control system constructed by the combination of ICE and EPICS coordinates and manages the high level and low level of Figure 1, e.g., the ASKAP monitoring and control system [54]. However, ICE does not provide a hardware device driver

TABLE 2: Middleware framework used by major radio telescope.

| Name | Diameter | Country | Middleware |
|-------|----------|-----------|---------------|
| LMT | 1 × 50 | Mexico | CORBA |
| TMRT | 1 × 65 | China | ICE |
| ALMA | 66 × 12 | Chile | ACS |
| SRT | 1 × 64 | Italy | ACS |
| FAST | 1 × 500 | China | CORBA + EPICS |
| ASKAP | 36 × 12 | Australia | ICE + EPICS |
| GMRT | 30 × 45 | India | Tango |
| SKA | — | — | Tango |

CORBA: Common Object Request Broker Architecture; ICE: Internet Communications Engine; EPICS: Experimental Physics and Industrial Control System; ACS: ALMA Control System.

interface. It is a heavyweight middleware framework, and engineers need to have a certain knowledge of the technology stack.

In summary, the control system constructed by the high-level coordination middleware framework coordinates software subsystems. This kind of middleware framework is developed from CORBA to ICE. ICE has replaced CORBA because CORBA has stopped updating and maintaining and ICE has simpler interfaces and implementation details than CORBA. In brief, the control system built with ICE has higher scalability and better maintainability.

3.2. High- and Low-Level Management Middleware Framework. The control system constructed by the high- and low-level middleware framework not only provides data management, information organization, task planning, and other functions for the high level in Figure 1 but also supplies parameter adjustment loop control, data acquisition, equipment monitoring, and fault diagnosis for the low level in Figure 1. This type of middleware framework mainly includes ACS, the combination of ICE (or CORBA) and EPICS, and Tango, which provides a soft bus with transparent access services, information transform, and other functions and at the same time, provides the functions of monitoring, controlling, and managing hardware subsystems.

ACS, custom-developed for radio telescope, is a middleware framework developed based on CORBA. It not merely integrates common issues in radio telescope but also hides the complex interface of CORBA, network, and database [55]. ACS uses CORBA to realize synchronous communication and asynchronous communication, provides services and runtime libraries for the control system, e.g., component/container services and astronomical libraries [56]. ACS is not only used in radio telescope, such as ALMA software [57] and SRT control software [58], but also in optical telescopes and physical devices [59–63]. However, ACS is seldom used in equipment and maintenance; updates have ceased after 2010.

In order to quickly develop the control system and ensure the performance of the radio telescope, ACS was replaced by two solutions: (1) the combination of ICE (or CORBA) and EPICS; (2) Tango, a single middleware framework with similar functions as ACS.

For the first method, ICE (or CORBA) is that a user sends control commands to the radio telescope, and EPICS returns the monitoring status information of the hardware device or control system to the designated location. EPICS incorporates EDA into its design. EPICS is a middleware framework originated from large-scale experimental physical devices, providing soft real-time communication functions [64]. The channel access (CA) mechanism of EPICS is based on the TCP/IP protocol, provides an application program interface for the operation interface (OPI) and input/output controller (IOC) [65]. CA is the foundation of EPICS, providing a soft bus with transparent network access service functions. IOC is the core of EPICS and provides an interface for the server application. It can control equipment through a bus or direct I/O to collect/store data information in real-time. OPI provides an interface for client application development and receives control commands sent by the upper layer and status information returned by the lower layer. EPICS is not only applied to radio telescopes, e.g., FAST control system [66, 67] and ASKAP monitoring and control system [68], but also applied to large scientific equipment such as accelerators, physical experiment devices, and optical telescopes [69–73].

For the second method, Tango is a middleware framework that originated from a large-scale physical experiment device. Tango uses CORBA and ZeroMQ to implement synchronous communication and asynchronous communication between systems, respectively [74]. Tango provides not only a simple and unified interface for devices, but also a drive interface for hardware devices, an operating environment, and development tools for the system. In addition, the soft bus provided by Tango needs to be recompiled when a new device is added, CORBA is used to send control commands, and ZeroMQ sends the status information of the hardware device to the user interface [75]. Tango provides the basis for the realization of radio telescope control systems, such as GMRT monitoring and control system [76] and SKA monitoring and control system [77]. GMRT monitoring and control system is based on the Sensor Actuator and Control Element (SACE) model and developed by Tango [78]. SKA consists of 1.3 million low-frequency arrays, 250 intermediate-frequency arrays, 2500 high-frequency arrays, with the longest baseline of 3000 kilometres [79]. Notwithstanding, SKA monitoring and control system is constructed with Tango, hierarchical design mode [80]. In addition to being used in radio telescope, Tango is also used in the development of control system for synchrotrons, lasers, and other scientific devices [81, 82].

In summary, the control system constructed by the high- and low-level management middleware framework coordinates and manages the software and hardware system. Middleware framework is developed from ACS, which is customized for radio telescope, by the combination of EPICS and ICE (or CORBA) and Tango. The combination of EPICS and ICE and Tango has gradually replaced ACS because ACS has ceased to be updated and maintained. Therefore, the control system constructed by using the combination of EPICS and ICE and Tango has good stability, high scalability, and strong maintainability.

3.3. *Analysis of Middleware Framework.* This section analyzes the application requirements of five middleware frameworks in the radio telescope control system and compares and analyzes their advantages and disadvantages.

When choosing a middleware framework suitable for a radio telescope control system, it can be considered from the development and management perspectives of software system architects, telescope managers, and software engineers. The design requirements of software system architects mainly include whether the middleware framework can satisfy the requirements of the control system. The focus of telescope managers includes the development cost and maintenance cost of the middleware framework to construct a control system. The development requirements of software engineers are which middleware framework to use to develop control system and the development cycle and difficulty of middleware framework to construct control system. Therefore, a comprehensive comparison of middleware framework characteristics is required, as shown in Table 3. The following part compares and analyzes these middleware frameworks from five aspects: “serialization interface,” “communication,” “event service,” “process management,” and “security.”

“Serialization interface” is the foundation of the realization of a cross-platform, multilanguage control system. IDL of CORBA and ACS is mapped to IDL Stub interface and IDL Skeleton interface to realize data serialization and deserialization. Compared with CORBA IDL, Tango’s serialization model better satisfies serialization/deserialization of devices, classes, and processes. ICE’s Slice draws on the design ideas of IDL, which is easier to write than CORBA’s IDL, and has the characteristics of better serialization and deserialization performance. EPICS PVDData serializes and deserializes complete data types.

“Communication” simplifies the communication method in the design and implementation of the control system. General Inter-ORB Protocol (GIOP) of CORBA, ACS is an abstract protocol that provides a set of transmission syntax and communication information format. Internet Inter-ORB Protocol (IIOP) is a specific implementation of GIOP. Tango uses ZeroMQ’s asynchronous communication to replace CORBA’s IIOP and GIOP, which can better realize asynchronous communication. ICE satisfies both synchronous communication and asynchronous communication. ICE’s RPC has richer functions and simpler interfaces than CORBA. IceStorm is an efficient publish/subscribe service for asynchronous transmission of messages. EPICS’s CA protocol is a communication protocol for transmitting information between server and client to satisfy the requirements of multiclient, multiserver message transmission.

“Event service” can improve the coordination and management capabilities of the control system. The event channel of CORBA and ACS provides event distribution service, which can satisfy the communication between subsystems. Event channel is suitable for early control systems, but it is difficult to apply to modern control systems. Tango uses ZeroMQ instead of event channel to implement event distribution. ZeroMQ uses a multicast

protocol to reduce server network bandwidth overhead compared to event channel. ICE’s IceStorm implements message distribution services, which is less coupled than CORBA’s client and server. EPICS CA provides remote access to management records and fields for IOC, which can realize search, discovery, and flow control of subsystems.

“Process management” is the basis for the effective operation of the control system. CORBA’s IOR is used to register and manage processes, while Object Request Broker (ORB) coordinates and manages message transmission between processes. Tango’s Device server provides one or more services with a smaller granularity than CORBA. On the other hand, ZeroMQ coordinates and manages message transmission and forwarding between processes. The concurrency of IceGrid, which ICE has, is stronger than CORBA, and it can better manage processes at the same time. Container/Component and ORB of ACS are used to coordinate and manage calls between processes. The caRepeater process of EPICS makes the CA client process independent of host’s IOC, and ChannelRPC is used for information transmission, coordination, and management between different processes.

“Security” is a powerful guarantee to improve the reliability of data transmission in the control system. CORBA has designed a variety of security protocols, but most of them have not been implemented, and the designed and implemented control system has low security. The security service and authorization policies provided by ACS satisfy the security of the control system. Tango uses HAProxy to realize information transmission in the network, which is more secure and has better performance than CORBA. ICE’s Glacier allows clients and servers to communicate securely through a firewall, which is more secure than CORBA. In addition to providing server security, EPICS CA gateways also provide users with secure access.

By comparing and analyzing the five middleware frameworks, it is found that Tango, ICE, and EPICS have more advantages than CORBA and ACS, which mainly include the following aspects:

- (1) They have been under continuous maintenance and update
- (2) They provide simpler serialization AND communication interfaces
- (3) They use event distribution more concisely
- (4) The process management adopted by them is more convenient AND easy to develop
- (5) The safety protection schemes adopted by them are more effective
- (6) The basic functions they provide are more complete
- (7) They have been gradually applied to radio telescopes in recent years

If the middleware framework used in the control system design only needs to coordinate and manage the software subsystems of the radio telescope, ICE can be chosen to build the control system. If the control system is used to monitor, control, and manage the hardware and software subsystems,

TABLE 3: Comparison the characteristics of five middleware frameworks.

| Name | CORBA | ACS | Tango | ICE | EPICS |
|-------------------------|-------------------------------------|------------------------------------------|-----------------------|-----------------|------------------------|
| Serialization interface | IDL, CDR | IDL or XML | Serialization model | Slice | PVData |
| Communication | GIOP, IIOP | GIOP, IIOP | GIOP, IIOP, ZeroMQ | RPC, IceStorm | CA protocol |
| Event service | Event channel | Event channel | ZeroMQ | IceStorm | CA |
| Process management | IOR, ORB | Container/Component, ORB | Device server, ZeroMQ | IceGrid | caRepeater, ChannelRPC |
| Fault diagnosis | Log service | ACS alarm system | Log service | Log service | Alarm service |
| Security | Multiple standards, not implemented | Security service, Authorization policies | HAProxy | IceSSL, Glacier | CA gateways |
| At which level | High level | High and low level | High and low level | High level | Low level |
| Maintenance | Stop | Stop | Update | Update | Update |
| Application | LMT | ALMA, SRT | GMRT, SKA | TMRT | AKSAP |

Note: GIOP: General Inter-ORB Protocol, IIOP: Internet Inter-ORB Protocol, CA: channel access, IOR: Interoperable Object Reference, RDS: Read Different than Set; CDR: Common Data Representation.

ICE can be combined with EPICS or Tango to build the control system.

4. Conclusion

This article summarizes the control system of radio telescope in different periods, analyzes and compares the software architecture and middleware framework used in the control system. Software architecture is the overall plan of the radio telescope and the basis for the realization of the control system. It develops from a centralized architecture with simple structure and short development cycle to an extensible and flexible distributed architecture without middleware framework and then to a highly extensible, portable, and maintainable distributed architecture based on middleware framework. By comparing and analyzing the merits and shortcomings of the three-stage control system, it is found that distributed architecture based on the middleware framework of the third stage is more suitable for the design and implementation of modern radio telescope control systems. Middleware framework simplifies the complex operating system, network, and other interfaces into a unified and simple standard interface, which is conducive to the design and implementation of the high-level and low-level control systems. High-level coordination middleware framework has developed from CORBA with complex interfaces and complex functions to ICE with simple interfaces and simplified functions; high- and low-level management middleware framework has developed from ACS to the combination of ICE and EPICS or Tango. After comparing the benefits and drawbacks of using middleware frameworks in modern radio telescope control systems, it is found that ICE, EPICS, and Tango are more suitable for building control systems. Therefore, in the design of the radio telescope control system architecture, it is necessary to select the middleware framework to be used in terms of software and hardware subsystems, control and monitoring software and hardware. If the control system is mainly based on service or control, system architects could choose a middleware framework with similar functions as ICE to have more

advantages. If a control system is used to monitor, control, and manage control system, a middleware framework similar to the combination function of ICE and EPICS or Tango function could be used. In addition, when choosing a suitable middleware framework to build a control system, we should consider not only the nonfunctional requirements such as extensibility and maintainability but also the functional requirements such as complete functions, stable performance, and simple interface of middleware framework.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant no. 2018YFA0404603. The research work was also partly supported by the Operation, Maintenance and Upgrading Fund for Astronomical Telescopes and Facility Instruments, budgeted from the Ministry of Finance of China (MOF) and administrated by the Chinese Academy of Sciences (CAS).

References

- [1] J. Schraml, W. Brunswig, and G. Juen, "Design and software aspects for the control system of the 30 m MRT," *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 444, pp. 122–126, 1983.
- [2] A. Jessner, "Architecture of the Effelsberg control system," *Telescope Control Systems*, vol. 2479, pp. 79–88, 1995.
- [3] N. Wang, "Xinjiang Qitai 110 m radio telescope," *Science China Physics Science China-Physics Mechanics and Astronomy*, vol. 44, no. 8, pp. 783–794, 2014, in Chinese.
- [4] P. E. Dewdney, P. J. Hall, R. T. Schilizzi, and T. J. L. W. Lazio, "The square kilometre array," *Proceedings of the IEEE*, vol. 97, no. 8, pp. 1482–1496, 2009.
- [5] A. E. Schinckel, J. D. Bunton, T. J. Cornwell et al., "The Australian SKA pathfinder," *Ground-based and Airborne Telescopes IV*, vol. 8444, no. 12, pp. 807–818, 2012.

- [6] H. J. Kärcher and J. W. M. Baars, "Ideas for future large single dish radio telescopes," *Ground-based and Airborne Telescopes V*, vol. 9145, pp. 13–23, 2014.
- [7] C. S. Wang, H. H. Li, K. Ying et al., "Active surface compensation for large radio telescope antennas," *IOSR Journal of Applied Physics*, vol. 2018, Article ID 3903412, 17 pages, 2018.
- [8] P. R. Jewell and G. Langston, "The green bank telescope: an overview," *Astrophysical Phenomena Revealed by Space VLBI*, vol. 3357, pp. 656–665, 2000.
- [9] Z. Shen, "Shanghai 65 m radio telescope," in *Proceedings of the 2011 XXXth URSI General Assembly and Scientific Symposium*, p. 1, Istanbul, Turkey, August 2011.
- [10] P. Bolli, A. Orlati, L. Stringhetti et al., "Sardinia radio telescope: general description, technical commissioning and first light," *Journal of Astronomical Instrumentation*, vol. 608, 2017.
- [11] D. Li, R. Nan, and Z. Pan, "The five-hundred-meter aperture spherical radio telescope project and its early science opportunities," *Proceedings of the International Astronomical Union*, vol. 8, no. S291, pp. 325–330, 2013.
- [12] N. Wang, "Xinjiang Qitai 110 m radio telescope," *SCIENTIA SINICA Physica, Mechanica and Astronomica*, vol. 44, no. 8, pp. 783–794, 2014, in Chinese.
- [13] F. Buffa, P. Bolli, G. Sanna et al., "An atmosphere monitoring system for the Sardinia radio telescope," *Measurement Science and Technology*, vol. 28, no. 1, 2017.
- [14] F. Fan, X. F. Jin, H. J. Wang et al., "Health monitoring system of the structure supporting the reflector of FAST," *Journal of Harbin Institute of Technology*, vol. 12, pp. 1–6, 2009, in Chinese.
- [15] Y. Jiang, J. Wang, W. Gou et al., "A position sensing device's application on TianMa radio telescope," in *Proceedings of the 2017 17th IEEE International Conference on Communication Technology (ICCT 2017)*, pp. 1904–1908, Chengdu, China, October 2017.
- [16] B. E. Glendenning, "The ALMA software system," in *Proceedings of the 2011 XXXth URSI General Assembly and Scientific Symposium*, pp. 1-2, Istanbul, Turkey, August 2011.
- [17] K. Schaaf, C. Broekema, G. Diepen et al., "The lofar central processing facility architecture," *Experimental Astronomy*, vol. 17, no. 1–3, pp. 43–58, 2004.
- [18] G. Swarup, S. Ananthkrishnan, C. R. Subrahmanya et al., "The giant metrowave radio telescope," *Large Antennas in Radio Astronomy*, vol. 60, pp. 95–105, 1996.
- [19] T. Westmeier and S. Johnston, "The Australian SKA pathfinder," *Ground-based and Airborne Telescopes IV*, vol. 8444, pp. 84442A–84512A, 2012.
- [20] L. R. Brederode, L. van den Heever, W. Esterhuysen et al., "MeerKAT: a project status report," *Ground-based and Airborne Telescopes VI*, vol. 9906, 2016.
- [21] J. Lazio, "The square kilometre array," *panoramic radio astronomy: wide-field 1-2 GHz research on galaxy evolution*, <https://pos.sissa.it/089/058/pdf>, 2009.
- [22] Z. Y. Liu, L. Jun, N. Wang et al., "The architecture design of astronomical observation and system monitoring and control software for large radio telescope," *Scientia Sinica (Physica, Mechanica & Astronomica)*, vol. 49, no. 9, p. 99509, 2019, in Chinese.
- [23] O. Hachenberg, B. H. Grahl, and R. Wielebinski, "The 100-meter radio telescope at Effelsberg," *Proceedings of the IEEE*, vol. 61, no. 9, pp. 1288–1295, 1973.
- [24] K. I. Morita, N. Nakai, M. Ohishi et al., "Telescope control system of the Nobeyama radio observatory," *Proceedings of Spie*, vol. 2479, pp. 70–78, 1995.
- [25] J. G. Ables, C. E. Jacka, D. McConnell, A. E. Schinckel, and A. J. Hunt, "The Parkes radio telescope - 1986," *Publications of the Astronomical Society of Australia*, vol. 6, no. 4, pp. 507–512, 1986.
- [26] K. I. Morita, M. Nakai, T. Takahashi et al., "COSMOS-3: the third generation telescope control software system of Nobeyama radio observatory," *Astronomical Data Analysis Software and Systems XII*, vol. 295, p. 166, 2003.
- [27] N. M. Radziwill, M. Mello, E. Sessoms et al., "An enterprise software architecture for the green bank telescope (GBT)," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5496, pp. 230–240, 2004.
- [28] C. Peter and R. Stewart, "XML schema," *Java Web Services Architecture*, vol. 118, no. 2, pp. 743–770.
- [29] K. Souccar, G. Wallace, and D. Malin, "A reusable automatically generated software system for the control of the large millimeter telescope," *Advanced Telescope and Instrumentation Control Software II*, vol. 4848, pp. 35–42, 2002.
- [30] S. L. Li, *The Design and Implementation of Distributed Control Software for Tianma Radio Telescope*, in Chinese, 2015.
- [31] G. Chiozzi, "CORBA-based common software for the ALMA project," *Advanced Telescope and Instrumentation Control Software II*, vol. 4848, pp. 43–54, 2002.
- [32] J. C. Guzman and B. Humphreys, "The Australian SKA pathfinder (ASKAP) software architecture," *Software and Cyberinfrastructure for Astronomy*, vol. 7740, pp. 467–476, 2010.
- [33] J. Kodilkar, V. Kumthekar, R. Uprade et al., "The next generation GMRT M&C system - an exploratory prototype for the SKA telescope manager," in *Proceedings of the 2019 URSI Asia-Pacific Radio Science Conference (AP-RASC)*, New Delhi, India, March 2019.
- [34] A. Dworak, P. Charrue, F. Ehm et al., "Middleware trends and market leaders 2011," in *Proceedings of the 13th International Conference on Accelerator and Large Experimental Physics Control Systems*, pp. 1334–1337, Grenoble, France, October 2011.
- [35] G. Chiozzi, K. Gillies, B. Goodrich et al., "Trends in software for large astronomy projects," in *Proceedings of the International Conference on Accelerator and Large Experimental Physics Control Systems ICALEPCS*, Knoxville, TN, USA, October 2007.
- [36] J. P. Almeida, M. V. Sinderen, D. Quartel et al., *Interaction Systems Design and the Protocol- and Middleware-Centred Paradigms in Distributed Application Development*, 2003.
- [37] Y. L. Ding, L. Z. Gu, and Y. Yang, "Research of application architecture based on distributed middleware ICE," *Journal of Computer Applications*, pp. 27-28, 2009, in Chinese.
- [38] M. Plesko, K. Zagar, M. Sekoranja et al., *ACS - The Advanced Control System*, 2008.
- [39] L. R. Dalesio, J. O. Hill, M. Kraimer et al., "The experimental physics and industrial control system architecture: past, present, and future," *Nuclear Instruments and Methods in Physics Research A*, vol. 352, pp. 179–184, 1995.
- [40] A. Senchenko, G. Fatkin, P. A. Selivanov et al., "Tango based software of control system of LIA-20," in *Proceedings of the 16th International Conference on Accelerator and Large Experimental Physics Control Systems*, Barcelona, Spain, 2018.
- [41] S. Vinoski, "CORBA: integrating diverse applications within distributed heterogeneous environments," *IEEE Communications Magazine*, vol. 35, no. 2, pp. 46–55, 1997.
- [42] J. Andrade Almeida, M. J. van Sinderen, D. Quartel et al., "Interaction systems design and the protocol- and

- middleware-centred paradigms in distributed application development,” in *Proceedings of the ECOOP 2003 Workshop on Communication Abstractions for Distributed Systems*, Darmstadt, Germany, July 2003.
- [43] E. Arulanthu, D. Schmidt, M. Kircher et al., *Applying Patterns and Components to Develop an IDL Compiler for CORBA AMI Callbacks*, 1999.
- [44] K. Souccar, G. Wallace, and D. Malin, “A standard control system for the large millimeter telescope and instruments,” *Proceedings of SPIE - The International Society for Optical Engineering*, pp. 241–249, 2004.
- [45] G. Grégory, B. Hugues, S. Michel et al., “Mapping semantics of CORBA IDL and GIOP to open core protocol for portability and interoperability of SDR waveform components,” *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 330–335, 2008.
- [46] R. Penataro, J. M. Filgueira, P. Gomez-Cambronero et al., “Application of CORBA to the GTC control system,” *Advanced Telescope and Instrumentation Control Software*, vol. 4009, pp. 152–166, 2000.
- [47] P. D. Vicente and R. Bolao, “Software tools and preliminary design of a control system for the 40 m OAN radiotelescope,” *Astronomical Data Analysis Software & Systems XIII*, vol. 314, p. 740, 2004.
- [48] N. A. Dipper, C. Blackburn, H. Lewis et al., “The use of object-oriented techniques and CORBA in astronomical instrumentation control systems,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 5496, pp. 565–573, 2004.
- [49] Y. Li, J. Zhou, L. Guo et al., “Research on distributed network communication based on ICE middleware,” in *Proceedings of the 2016 6th International Conference on Machinery, Materials, Environment, Biotechnology and Computer (MMEBC 2016)*, pp. 608–612, Tianjin, China, June 2016.
- [50] Z. H. Wu, *Zeroc Ice Authoritative Guide*, Publish House of Electronics Industry, Beijing, China, 2015, in Chinese.
- [51] J. Dong, R. B. Zhao, X. T. Zuo et al., “The research of distributed middleware technique in single-dish observation system of Shanghai 65-meter radio telescope,” *Annals of Shanghai Observatory Academia Sinica*, pp. 36–44, 2012, in Chinese.
- [52] I. Oropesa, J. A. Sánchez-Margallo, P. Sánchez-González et al., “Controlling virtual scenarios for minimally invasive surgery training using the EVA Tracking System,” in *Proceedings of the XXXIII Congreso Anual de la Sociedad Española de Ingeniería Biomédica*, Madrid, Spain, November 2015.
- [53] C. Rodríguez-Domínguez, K. Benghazi, J. L. Garrido et al., *Designing a Communication Platform for Ubiquitous Systems: The Case Study of a Mobile Forensic Workspace*, Springer, London, UK, 2013.
- [54] M. Marquarding, “Past, present and future of the ASKAP monitoring and control system,” in *Proceedings of the 15th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS’15)*, pp. 1162–1164, Melbourne, Australia, October 2015.
- [55] G. Chiozzi, B. Jeram, H. Sommer et al., *The alma common software (ACS): status and developments*, Geneva, vol. 7019, no. 6578, pp. 163–166, 2003.
- [56] C. E. Menay, G. A. Zamora, R. J. Tobar et al., “New architectures support for ALMA common software: lessons learned,” *Software and Cyberinfrastructure for Astronomy*, vol. 7740, 2010.
- [57] T. C. Shen, R. Soto, M. Mora et al., “ALMA operation support software and infrastructure,” *Proceedings of the Spie*, vol. 8451, pp. 99–104, 2012.
- [58] A. Orlati, “Status report of the SRT radiotelescope control software: the DISCOS project,” *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913, 2016.
- [59] P. de Vicente, R. Bolaño, and L. Barbas, “Development of the control system for the 40 m radiotelescope of the OAN using the alma common software,” *Astronomical Data Analysis Software and Systems XV ASP Conference Series*, vol. 351, 758 pages, 2005.
- [60] A. Caproni, K. Sigerud, and K. Zagar, “Integrating the CERN LASER alarm system with the ALMA common software,” *Spie Astronomical Telescopes + Instrumentation International Society for Optics and Photonics*, Article ID 671110, 2006.
- [61] A. Caproni and E. Schmid, “The integrated alarm system of the alma observatory,” in *Proceedings of the 16th International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS’17)*, Barcelona, Spain, October 2017.
- [62] M. A. Araya, L. Pizarro, and H. H. von Brand, “Packaging and high availability for distributed control systems,” in *Proceedings of the 16th International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS’17)*, pp. 1465–1469, Barcelona, Spain, October 2017.
- [63] V. Conforti, M. Trifoglio, F. Gianotti et al., “The DAQ system support to the AIV activities of the ASTRI camera proposed for the Cherenkov telescope array,” *Software and Cyberinfrastructure for Astronomy*, 2018.
- [64] G. R. White and M. V. Shankar, “The EPICS software framework moves from controls to physics,” in *Proceedings of the International Particle Accelerator Conference*, Melbourne, Australia, May 2019.
- [65] G. Wang, S. He, G. Gao et al., “Design and implementation of monitor system based on EPICS for EAST PF power supply control system,” *Computer Measurement and Control*, 2018.
- [66] X. H. Ying, L. C. Zhu, and W. B. Zhu, “The design of computer supervisory system for large spherical radio telescope based on fieldbus technology,” *Computer Engineering and Applications*, pp. 251–253, 2002, in Chinese.
- [67] J. Wang, J.-j. Liu, P.-y. Tang et al., “A study on generic models of control systems of large astronomical telescopes,” *Publications of the Astronomical Society of the Pacific*, vol. 125, no. 932, pp. 1265–1276, 2013.
- [68] J. C. Guzman, “Preliminary design of the Australian SKA pathfinder (ASKAP) telescope control system,” in *Proceedings of the 12th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS 2009)*, Kobe, Japan, October 2009.
- [69] G. H. Wang, Y. S. He, G. Gao et al., “Design and implementation of monitor system based on EPICS for EAST PF power supply control system,” *Computer Measurement and Control*, pp. 57–60, 2018, in Chinese.
- [70] N. Akasaka, A. Akiyama, S. Araki et al., “KEKB accelerator control system,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 499, no. 1, pp. 138–166, 2003.
- [71] C. Y. Wu, J. Chen, C. Y. Liao et al., “Control system of EPU48 in TPS,” in *Proceedings of the 5th International Particle Accelerator Conference*, Dresden, Germany, June 2014.
- [72] T. Korhonen, R. Andersson, F. Bellorini et al., “Status OF the EUROPEAN spallation source control system,” in *Proceedings of the 15th International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS 2015)*, Melbourne, Australia, October 2015.

- [73] G. White, T. Cobb, L. Dalesio et al., “The EPICS software framework moves from controls to physics,” in *Proceedings of the International Particle Accelerator Conference*, Melbourne, Australia, May 2019.
- [74] A. Go“Tz, E. Taurel, P. Verdier et al., “TANGO–can ZMQ replace CORBA?” in *Proceedings of ICALEPCS2013*, San Francisco, CA, USA, October 2013.
- [75] G. Piotr, G. Andrew, H. Vincent et al., “Towards specification of tango V10,” in *Proceedings of the Accelerator and Large Experimental Physics Control Systems*, New York, NY, USA, October 2019.
- [76] J. Kodilkar, V. Kumthekar, and R. Uprade, “Next generation GMRT monitor and control system,” in *Proceedings of the URSI Asia-Pacific Radio Science Conference (URSI AP-RASC)*, New Delhi, India, March 2019.
- [77] D. Barbosa, “A cyber infrastructure for the SKA telescope manager,” *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913, 2016.
- [78] S. R. Chaudhuri, A. L. Ahuja, S. Natarajan et al., “Model-driven development of control system software,” in *Proceedings of the International Particle Accelerator Conference*, Vancouver, Canada, May 2009.
- [79] S. Natarajan, D. Barbosa, J. Barraca et al., “SKA Telescope Manager (TM): status and architecture overview,” *Proceedings of SPIE*, vol. 9913, 2016.
- [80] M. Di Carlo, M. Dolci, R. Smareglia et al., “Monitoring and controlling the SKA telescope manager: a peculiar LMC system in the framework of the SKA LMCs,” *Software and Cyberinfrastructure for Astronomy IV*, vol. 9913, 2016.
- [81] Á. Péter, B. Sándor, J. F. Lajos et al., “Tango-kepler integration at ELI-ALPS,” in *Proceedings of the International Particle Accelerator Conference*, Virginia, VA, USA, May 2015.
- [82] P. P. Goryl, C. J. Bocchetta, L. J. Dudek et al., “Tango based control system at SOLARIS synchrotron,” in *Proceedings of IPAC2016*, Busan, Korea, May 2016.