International Journal of Internet, Broadcasting and Communication Vol.14 No.2 40-46 (2022)
http://dx.doi.org/10.7236/IJIBC.2022.14.2.40

IJIBC 22-2-7

Distributed Data Platform Collaboration Agent Design Using EMRA

Ho-Kyun Park* and Seok-Jae Moon**

** School of IT Convergence Engineering, Shinhan University, Gyeonggi, Korea*
***Professor, Department of Artificial Intelligence Institute of Information Technology, KwangWoon University, Korea*
E-mail: *hkpark@shinhan.ac.kr, msj80386@kw.ac.kr*

Abstract

Recently, as the need for data access by integrating information in a distributed cloud environment increases in enterprise-wide enterprises, interoperability for collaboration between existing legacy systems is emphasized. In order to interconnect independent legacy systems, it is necessary to overcome platform heterogeneity and semantic heterogeneity. To solve this problem, middleware was built using EMRA (Extended MetaData Registry Access) based on ISO/IEC 11179. However, the designed middleware cannot guarantee the efficiency of information utilization because it does not have an adjustment function for each node's resource status and work status. Therefore, it is necessary to manage and adjust the legacy system. In this paper, we coordinate the correct data access between the information requesting agent and the information providing agent, and integrate it by designing a cooperative agent responsible for information monitoring and task distribution of each legacy system and resource management of local nodes. to make efficient use of the available information.

Keywords: *Enterprise Cloud, Legacy System, Agent, Metadata Schema.*

1. Introduction

Currently, many institutions and companies have built and used independent legacy systems to fit their unique business characteristics [1]. As application tasks become more complex and diverse in such a distributed environment, the need to interoperate existing legacy systems through networks has increased [2]. In particular, the databases operating in these legacy systems use different DBMS [3] and have different data types, units, and formats, so these heterogeneities must be overcome [4]. In this paper, we propose a Collaborative Agent (SA) applied to a distributed data platform using EMRA*(Extended MetaData Registry Access)* [5, 6]. The cooperative agent system is responsible for providing the information extracted from the legacy system, the Information Agent (IA), and between the Request Agent (RA) who requests the information and the information-agent responsible for agent communication. Consists of a Coordinating Agent (CA). When data information is requested from the legacy system belonging to the community through the request agent,

Manuscript Received: February. 15, 2022 / Revised: February. 17, 2022 / Accepted: February. 21, 2022
Corresponding Author: msj80386@kw.ac.kr
Tel: Fax: +82-10-916-4751
Professor, Department of Artificial Intelligence Institute of Information Technology, KwangWoon University, Korea

the coordination agent determines the processing based on the local resource information extracted through the distributed platform, so that the workload of the legacy system to which the distributed data platform belongs. will adjust Therefore, it is possible to reduce the workload of each node within the community configured using EMRA, and to ensure reliability and efficiency of information requests. Chapter 2 describes related work. Chapter 3 describes the cooperative agent system design, and Chapter 4 explained the comparison of the number of filters for queries, Chapter 5 describes the conclusion.

2. Related Work

2.1 Distributed data platform structure using EMRA

The data distribution platform structure to which the cooperative agent system proposed in this paper will be applied consists of 4 layers. Each layer consists of a management part and an application part, and the management layer is again composed of a network layer, a system layer, a service layer, and an application layer. The cooperative agent system is configured in the system layer that manages system policies related to collaboration and manages agents. The Agent Manager is responsible for coordination/management between the service layer and the network layer based on the system policy belonging to the same layer. Figure 1 shows the structure of a distributed data platform to which the cooperative agent system is applied.

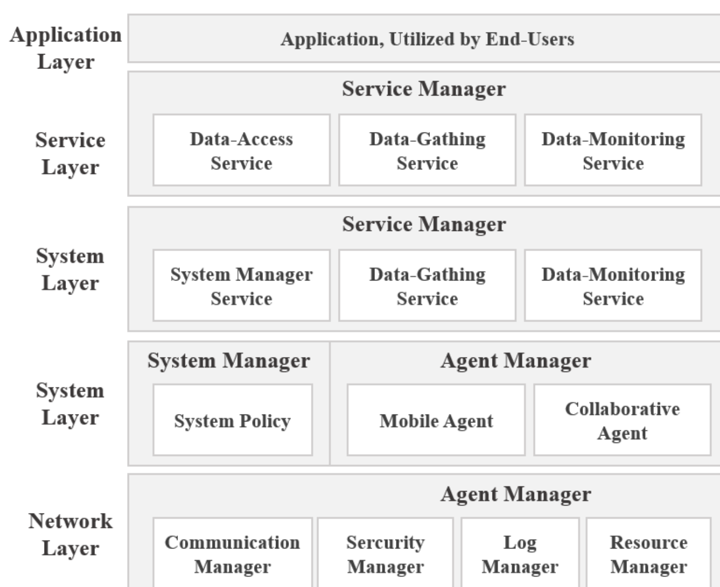

Figure 1. Distributed Data Platform Architecture

2.2 Distributed Data Platform

The cooperative agent designed in this paper is applied to the distributed data platform community environment as shown in Figure 2. And to resolve data heterogeneity between the existing legacy systems, EMRA [2] is formed. EMRA refers to an extended concept including MDR and ontology. MDR is composed by collecting schema information of databases in which legacy systems are actually used, and ontology consists of standard metadata ontology and location metadata ontology. The standard metadata ontology proposes a standard for the semantic content of metadata and the syntax of data elements to share and integrate metadata between legacy systems. provides Communication between each node uses a mobile agent technology that is platform-independent. This not only solves the problem of platform heterogeneity, but also has the advantage of reducing the burden between nodes by allowing the connection to be made only when sending an agent and when receiving a result, without the need to maintain a connection all the time.

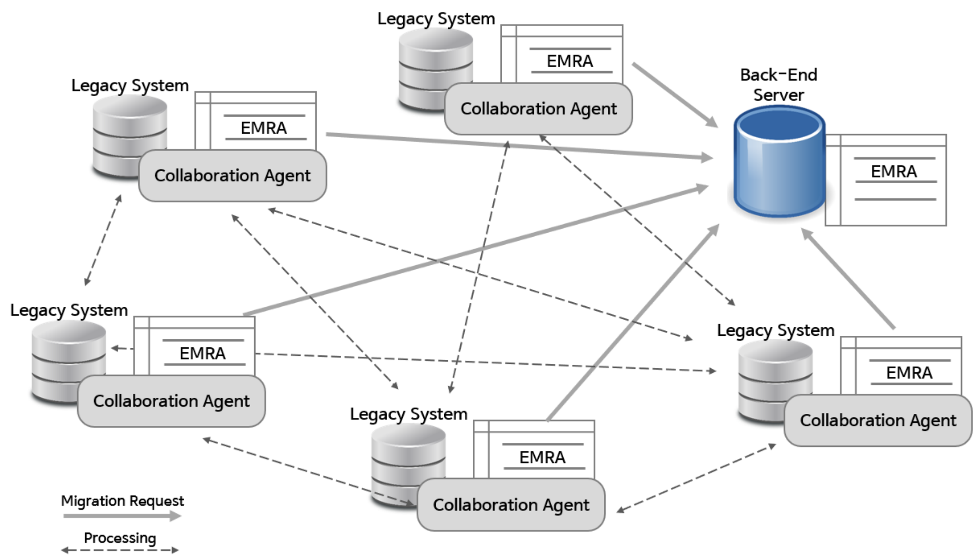

Figure 2. Distributed Data Platform Architecture

3. Cooperative agent system design

3.1 Cooperative Agent System Architecture

The cooperative agent system in this paper consists of two layers. The information agent provides log information, resource information and security information extracted from the network layer, and requested data and result information provided by the service layer. And the coordinating agent manages the communication between the information agent and the requesting agent requesting information. Execution of each agent is performed by a manager belonging to each agent hierarchy. Figure 2 is the structure of the cooperative agent.

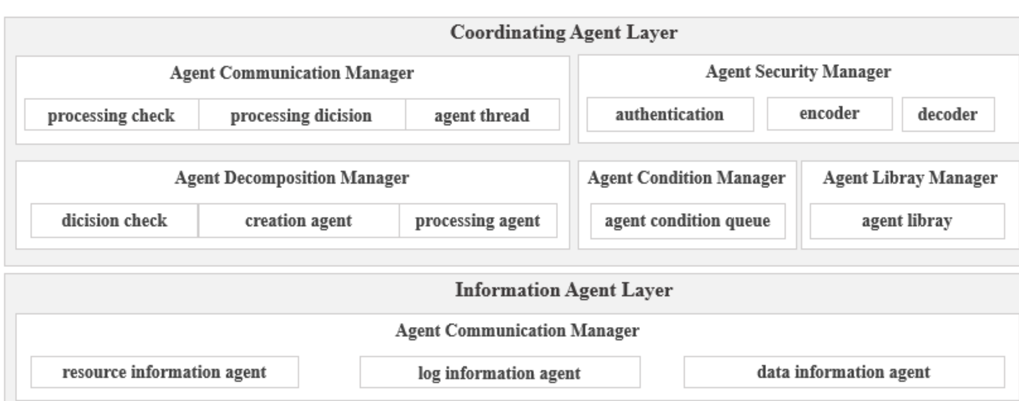

Figure 3. Structure of the cooperative agent system

- **Coordination Agent Layer.** This layer consists of 5 managers, and each manager communicates with other managers and managers of other layers.

  - Agent Communication Manager: Coordinates communication between the requesting agent and the information agent. Because communication between agents requires moving processes, states, and execution data, not simple data movement, it is necessary to implement a method of duplicating processes. There are a method of transmitting the replication process together, a method of transmitting various state data directly, and a method of requesting only necessary data after the process is moved and created. When the request agent arrives at the destination node, it must determine the process execution by determining the resource status and work status of the local node. If the work status or resource status of the local node exceeds the limit, the migration path included in the requesting agent is extracted and the priority is changed to move the requesting agent to the next node or to preserve the agent status in the agent status manager Put it in a queue to make the job wait. These policies are determined by referring to the system administrator in the system layer in Figure 2.

- Agent Security Manager: It performs authentication and encryption/decryption functions for the agent and the data or information contained in the agent. Communication between agents mainly involves process movement and exchange, so if a malicious process is included, a huge error can occur in the system of the local node. Therefore, a module for managing the authentication algorithm of request-agents coming into the local node is included. In addition, a function for encrypting and decrypting data when transmitted to the requesting node for the execution result, request result, and status result of the data requested by the request-agent is also included.

- Agent Decomposition Manager: Performs functions related to determining the type of requesting agent, creating an agent, and separating agents by service. The request-agent, determined by the agent communication manager, starts executing the process through the thread. When the agent separation manager determines the service provided by the distributed data platform by judging the type of the requesting agent that is started, it communicates and performs with the information agent suitable for the request. In addition, the agent separation manager also plays a role in creating the corresponding agent in order to transmit the request data or status information message received through the information-agent to the requesting node. At this time, the type of the generated response-agent is determined according to the service of each information-agent.

- Agent Library Manager: It performs the function of librarying the moving code. Since the request agent and response agent of the cooperative agent system proposed in this paper use the mobile agent technology, the code and process move. The size of request-agent and response agent can be reduced by libraryizing frequently used functions among moving code or processes. Also, since it is registered in the library, the execution speed can be increased.

■ **Information Agent Layer.** This agent serves to provide information and data serviced by the local node. In the information agent layer, three agents are managed by the collection manager. It provides resource information periodically collected by the resource manager of the distributed data platform, and the log agent provides log information of each agent executed in the local node and log information of the database. The data information agent serves to provide information about the services provided by the distributed data platform.

3.2 Cooperative Agent System Architecture

The execution process of the cooperative agent is shown in Figure 3. When the requesting agent arrives at the destination node, the coordinating agent performs agent authentication performed by the agent security manager. At this time, the authenticated agent decrypts the message information that is encrypted when the initial request agent is created through a decrypted. The agent whose decryption is completed notifies the agent communication manager that it has been completed and completes agent authentication. The agent communication manager requests the information agent for resource information of the local node and information about the operation process before executing the authenticated request-agent. In this case, the referenced resource information indicates the CPU occupancy rate of the local node, memory allocation information, and the number of agent threads. In addition, the information on the work execution process refers to information managed by the log manager existing in the platform and is collected by the information agent. The agent communication manager is collected by the information agent. The agent communication manager determines the performance of the requesting agent based on the resources collected by the information agent. The processing of the requesting agent after the execution is decided is as follows.

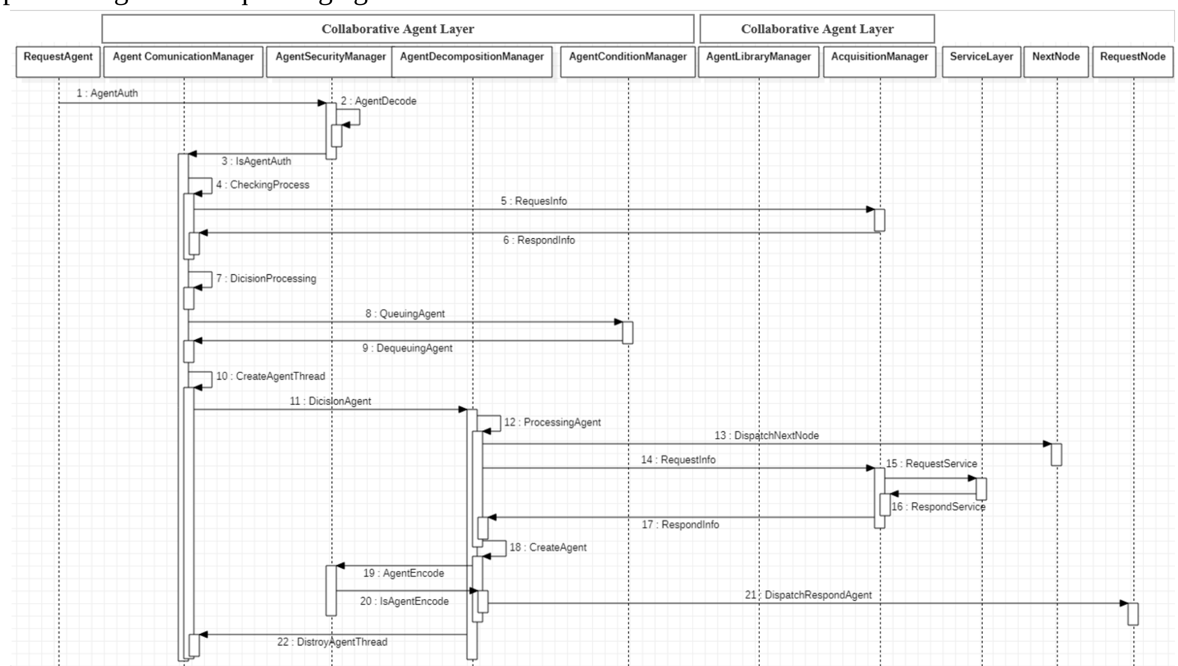

Figure 3. Cooperative Agent Manager Process

- **Process Disallowance.** If the local resource exceeds the allowed limit, the request agent in the local node is rejected. Therefore, through the coordination agent, the migration path of the requesting agent is changed so that it can be visited after starting from another node.
- **Process Standing by.** If the local resource does not exceed the permissible value, but a large number of agent threads in the local node are created and no more work can be performed, the execution of the requested agent in the local node is queued. The requesting agents that are waiting to be executed are stored in the agent state preservation queue and executed sequentially. At this time, the saved request agent does not wait for execution because it is saved, but copies it and stores it in the queue and then moves to the next node.
- **Process Permission.** If the local resource and the agent thread are within the allowable limits, the requesting agent can execute within the local node.

The agent communication manager creates an agent thread so that the requesting agent whose execution is decided can operate. At the same time that a thread is created, the agent separation manager determines the type of the requesting agent. Depending on the type of the requesting agent, it establishes a connection with the information agent that can process and starts to perform the request. After the execution is completed, the response information to the request is stored in the response agent created by the agent separation manager, encrypted by the agent security manager, and delivered to the first requesting node. After being transferred to the requesting node, it notifies the agent communication manager that the task is complete and destroys the requesting agent's thread.

4. Evaluate query filtering accuracy

In this section, we evaluate the accuracy of EMRA's query processing results through queries related to insertion, deletion, and update. Insert, delete, and update queries can be performed to determine the referential integrity of EMRA. The results were defined as evaluation factors.

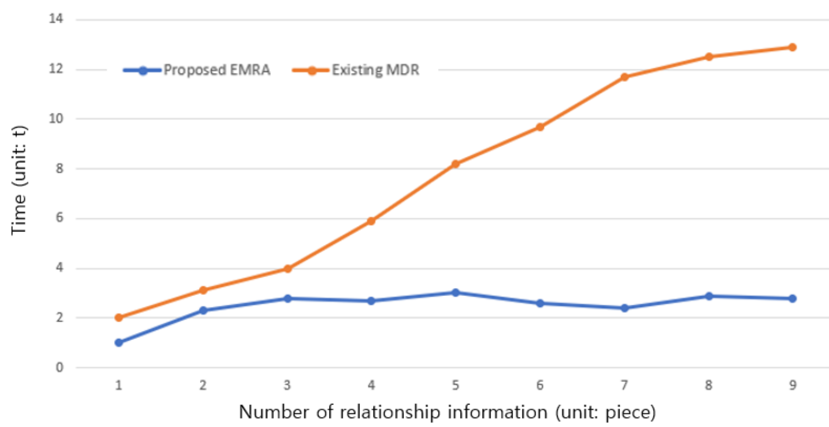

Figure 4. Time Comparsion of query modeling in MDR and EMRA

Since the EMRA system does not reflect the relationship information of the MDR, information that should be omitted from the query result is not omitted. This is a factor that lowers the reliability of the system. In Figure 12, 30 each of the queries related to insertion, deletion, and update were sampled in the system based on MDR and eMRA, and the number of queries actually executed was compared after querying the system. Each query is a query executed on a single table that reflects the relationship.

5. Conclusion

In this paper, we proposed a cooperative agent applied to a distributed data platform using EMRA. By suggesting a cooperative agent to be applied to this system, information can be utilized more efficiently than the existing system. Through the coordination agent, it is possible to reduce the load on the local node due to resource management and work distribution of nodes belonging to the community. However, it was confirmed that the response time was slower than that of the existing system through performance evaluation. The time loss due to the coordination and management of the coordination agent needs to be solved through efficient algorithm research. In addition, there is a need for research that can provide a service capable of real-time access management as well as a search service.

Acknowledgement

This paper was supported by the Shinhan University Research Grant of 2022.

References

[1] Prescott, John E., and Igor Filatotchev. "The business model phenomenon: towards theoretical relevance." Journal of Management Studies 58.2, pp.517-527, 2021. https://doi.org/10.1111/joms.12610

[2] Nilsson, Jacob, et al. "AI Concepts for System of Systems Dynamic Interoperability." Engineering applications of artificial intelligence, 2021.

[3] Kumar, Raj, et al. "Re-assessment of the depth to the base of magnetic sources (DBMS) in Australia from aeromagnetic data using the defractal method." Geophysical Journal International 225.1, pp.530-547, 2021. https://doi.org/10.1093/gji/ggaa601

[4] Hleap, Jose S., et al. "Assessment of current taxonomic assignment strategies for metabarcoding eukaryotes." Molecular Ecology Resources 21.7, pp,2190-2203, 2021. https://doi.org/10.1111/1755-0998.13407

[5] Lee, Jong-Sub, and Seok-Jae Moon. "Mobile Cloud System based on EMRA for Inbody Data." International Journal of Advanced Culture Technology 9.3, pp.327-333, 2021. https://doi.org/10.17703/IJACT.2021.9.3.327

[6] Elberskirch, Linda, et al. "Digital research data: from analysis of existing standards to a scientific foundation for a modular metadata schema in nanosafety." Particle and fibre toxicology 19.1, pp.1-19, 2022. https://doi.org/10.1186/s12989-021-00442-x