ArchiMate Customization and Architecture Repository Management Practices: for a Technology-Intensive Enterprise

Baobao Ding¹, Tong Wu*, Yingming Yang², Liang Dou² and Tiancheng Jin²

¹ CFETS Information Technology Co., Ltd, Shanghai, 200120, China
² Department of Computer Science and Technology, East China Normal University, Shanghai, 200062, China
*Corresponding author’s e-mail: wutong@chinamoney.com.cn

Abstract. Technology-intensive companies need to develop a variety of applications based on the needs of the organizations. Due to the increasing number of projects, a unified Enterprise Architecture specification is required. We propose a situation-specific Enterprise Architecture modeling method for CFETS Information Technology Co., Ltd. By customizing the ArchiMate architecture description language, developing the architectural design modeling tool and establishing the architecture repository management platform, this solution realizes the standardization and unified management of the architectural design model for the enterprise information systems. The evaluation results show that the solution is efficient and easy to use.

1. Introduction
At present, the market environment for enterprises is complex and shifting rapidly. For technology-intensive companies, there is a need to develop a variety of applications based on the requirements of the organizations. With the increasing number of projects, application complexity and circumstances of using software supply chains, outsourcing and collaborative development, the Enterprise Architecture design specifications which model the architectural designs of the various enterprise application projects are required.

China Foreign Exchange Trade System (CFETS) Information Technology Co., Ltd. is a wholly-owned subsidiary of China Foreign Exchange Trade System & National Interbank Funding Center. CFETS provides a trading platform covering interbank foreign exchange market, money market and bond market. The total transaction volume of CFETS in 2017 reached RMB 998 trillion. In order to adapt to the market demand and cope with the rapid growth of data volume, CFETS needs to develop a large number of application projects each year, including some outsourcing projects. In 2014, the delivery cycles of 178 software were less than four months. This number reached 369 in 2015 and 422 in 2016. The iterations of these application projects are very short, the quantity of the projects is increasing, and the maintenance difficulties are rapidly increasing. In order to design standard Enterprise Architecture and build more efficient enterprise information systems, CFETS applies TOGAF (The Open Group Architecture Framework) [1] in the organization, and uses ArchiMate [2], an architecture description language supported by TOGAF, for architecture modeling. However, there are many problems in practice.

Firstly, for technology-intensive companies that pay great attention on technology solutions, the interested technical details need to be described in the architectural design. ArchiMate is a relatively...
high-level architecture modeling standard [3], which can model the interactions between applications. However, it lacks the ability to describe technical details.

Secondly, the architectural design generated by each project can be treated as an enterprise repository, which needs to be stored in the Architecture Repository [4]. Currently there are no effective and unified management methods to review, maintain and reuse the architectural designs.

To solve the above problems, we propose the following solutions.

• Architecture specification customization: according to the actual needs of CFETS, the default elements and relationships of ArchiMate are streamlined and extended. Furthermore, the concept of architecture view is proposed to show a specific part of the architecture model.

• Architectural design tool implementation: in order to support architecture specification customization, redevelop the open source tool for ArchiMate - Archi.

• Construction of architecture repository management platform: based on the concept that architecture can be regarded as repository [4], an enterprise-wide architecture repository management platform is built to review, maintain and reuse architectural designs conveniently.

The rest of this paper is organized as follows. The section 2 introduces the related work. The section 3 describes the research foundation and the customization of the architecture specifications. The section 4 explains the implementation of the architectural design tool. Section 5 introduces and evaluates the architectural repository management platform.

2. Related work

Sessions [5] made an authoritative comparison and analysis of current mainstream Enterprise Architecture frameworks such as Zachman [6], FEA [7] and TOGAF, which provided an important reference for subsequent research in this field. He pointed out that TOGAF is focusing on practice and provided a practical and feasible Enterprise Architecture method with better reusability and scalability. Lankhorst et al. [8] presented ArchiMate 3.0, an enterprise modeling language that captured the complexity of architectural domains and their relations and allowed the construction of integrated Enterprise Architecture models.

In spite of the potential benefits of Enterprise Architecture, several challenges with Enterprise Architecture adoption still exist, including the inaccurate and unwieldy methods used, as well as ambiguity in terms of goals, concepts and frameworks [9]. Pittl et al. [10] show that the existing enterprise modeling approaches are inappropriate for modeling digital enterprise ecosystems comprehensively. They proposed an idea of how an extension of ArchiMate could be achieved to meet the requirements. Gill et al. [3] evaluated the applicability and integration of six modeling standards including ArchiMate, BPMN, UML, FAML, SoaML and BMM and proposed a hybrid approach for agile Enterprise Architecture modeling.

In order to derive the approaches which consider the specific requirements of organizations and industry, more researches and in-depth knowledge regarding Enterprise Architecture activity in organizations are required [11]. In this paper, we proposed the customized ArchiMate according to the requirements of CFETS. We also developed a set of tool chains, including an architectural design tool and an architectural repository management platform.

3. Architectural specification customization

3.1. ArchiMate

ArchiMate [2] is an Enterprise Architecture modeling standard for TOGAF. It is also a visual Enterprise Architecture description language that can describe the Enterprise Architecture from a high level. Fig. 1 illustrates the basic framework of ArchiMate, showing the levels and attributes of elements and relationships. The basic elements in ArchiMate can be divided into three levels: the business layer, the application layer, and the technology layer, which respectively describe different levels of abstraction in Enterprise Architecture development. The business layer mainly describes the business process. The application layer describes the software architecture of the system. The
technical layer describes the underlying technology deployment of the system. The elements can be divided into three categories based on their attributes: objects, behaviors, and subjects, where objects represent the passive elements and subjects represent the active elements in the architecture.

There can be many relationships between elements within and between layers. Relationships are divided into three categories: structured relationships, dynamic relationships, and other relationships. Relationships are used to connect different elements in the architecture. For example, the business object in the business layer is implemented by the data object in the application layer, the technical service of the technical layer serves the application component of the application layer (Serving), and the application interface of the application layer is part of the application component (Composition).

3.2. Elements and relations customization
The ArchiMate 3.0 specification includes the business layer, the application layer, and the technology layer. It involves dozens of elements and more than ten relationships. However, the technology-intensive enterprises focus on the application layer and the technology layer. We streamline and extend the original standard of ArchiMate according to requirement.

Firstly, we remove the entire business layer, leaving only the application layer and technology layer. Secondly, the elements and relationships of the application layer and the technology layer have been streamlined and extended, shown in table 1 and table 2. The basic pattern of designing an architectural model using the customized ArchiMate is shown in fig. 2. Finally, we extend the application layer and technology layer to illustrate more technical details, shown in table 3.

In addition, we propose some basic design rules to provide unified guidance on architectural designing within the company. The specific constraints include the position, the color and the allowable elements for each layer, etc. For example, the overall layout of the architecture model is vertical, and each layer uses different color.

3.3. Architectural view
Architectural view is often defined as part of the description of the Enterprise Architecture that each stakeholder focuses on, and it defines what each stakeholder can see. We summarize the requirements of CFETS and define different architectural views for different stakeholders. Table 4 lists the defined architectural views. For example, the component view only shows the component & interface layer and the reusable artifact layer for developers. The deployment view only shows the component & interface layer and the infrastructure layer for maintenance persons.

3.4. Case study
An Enterprise Architecture modeling case study using our customized ArchiMate is demonstrated here. In order to improve the user experience of the trading system, CFETS has developed the WeChat service system based on the online service mode, so that it can be the supplement to traditional trading
mode such as phone service and fax service. The architecture model of the WeChat service system is shown in fig. 3.

Table 1. Streamlined and extended elements.

| Layer                | Reserved Elements                                    | Deleted Elements                                      | Extended Elements                        |
|----------------------|------------------------------------------------------|------------------------------------------------------|------------------------------------------|
| Application layer    | Application component, Application interface, Application service | Application collaboration, Application function, Application interaction, Application process, Application event, Data object | Device, Technology collaboration, Technology interface, Path, Communication network, Technology function, Technology process, Technology interaction, Technology event, Technology object |
| Technology layer     | Node, System software, Technology service, Artifact | Device, Technology collaboration, Technology interface, Path, Communication network, Technology function, Technology process, Technology interaction, Technology event, Technology object | Linux system, Windows system, UNIX system, SQL database, NoSQL database, Web container, Message middleware, etc. |

Table 2. Streamlined relationships.

| Reserved Relationships | Deleted Relationships                           |
|------------------------|-------------------------------------------------|
| Composition, Aggregation, Assignment, Realization, Serving, Association | Access, Influence, Triggering, Flow, Specialization, Junction |

Table 3. Extensions for the application layer and the technology layer.

| Original Layer | Extended Layer | Explanation |
|----------------|----------------|-------------|
| Application layer | Application service layer | This layer shows the services provided by the software system. |
| Component & Interface layer | | This layer shows the software architecture that implements application layer services. |
| Peripheral system layer | | This layer shows the third-party systems. |
| Technology layer | Infrastructure service layer | This layer shows the services provided by the infrastructure. |
| Infrastructure layer | | This layer shows the structure and the deployment of the infrastructure. |
| Reusable artifact layer | | This layer shows the reusable artifacts, which can only be used by application components. |

Table 4. Architectural View.

| Views          | Included Layer                                      | Explanation                              |
|----------------|----------------------------------------------------|------------------------------------------|
| Deployment view| Component & Interface layer, Infrastructure layer  | Focus on the deployment of the system.   |
| Interface view | Component & Interface layer                        | Focus on the implementation of the system. |
| Component view | Component & Interface layer, Reusable              | Focus on the use of reusable              |
4. Architectural design tool

4.1. Architectural design tool redevelopment
Archimate has a native architectural design tool – Archi. Archi is an Eclipse-based and Java-based open source software for designing architectural models that conform to the Archimate specification. The current version is Archi 4.2.

We redevelop Archi to adapt to the customized Archimate specification. The original panel and menu is modified to accommodate the extended elements and the added architectural views.

Fig. 4 shows the part of customized panel in the tool. The extended elements including the operating system, Web container, NoSQL database, etc are showed in the red box.

4.2. Model check

When an architect draws the preliminary architecture model, usually there will be errors in the modeling design draft. A verification of architectural specification is needed. When an error occurs, the wrong part will be marked as red by the tool so that the architect will be alerted to correct it.

5. Architecture repository management platform

5.1. Overview

In order to implement the concept that architecture is repository, we establish an architecture repository management platform. The architects are the main producer of the architectural design, responsible for designing the architecture models and uploading them to the architecture management platform. The reviewers review the uploaded architecture models. If it does not meet the project requirements, the architecture model is returned to the architect for modifications. The project managers are responsible for the final review of the architecture model. The developers are consumers in the architecture management platform. They can log into the platform and develop systems according to the approved architecture models.
The architecture repository management platform includes three parts: architectural design tool, architecture management system, and architecture repository document library. The framework of the architecture repository management platform is shown in fig. 5.

![Architecture repository management platform framework](image)

The architecture management system is web-based. The front end of system is developed by HTML and TypeScript, and the back end of system is developed by Java.

The Architecture Repository Document Library is a system that stores all of the architectural design documents of CFETS. The library is developed by C#, and the functionality of SharePoint is packaged as Web API.

### 5.2. Architecture model context information extraction

The context of a software system comprises the knowledge that architects need to have about the environment in which a system is expected to operate [12]. However, the context information is often overlooked. This leads to software architectural design based on wrong assumptions, which may cause failure of the architectural design.

As shown in fig. 6, Bedjeti [12] summarized the various categories of context models in the software architecture. In our work, the customized ArchiMate specification covers the platform context and application context information. For example, the application context information we need to extract are the service name, function description, component category, component name, and version number, etc.

The context information are extracted during the uploading process of the architecture model, and stored in the cloud. For example, table 5 shows the extracted information of the interfaces between systems.

![Context category](image)

| Interface Name | Source System | Target System | Transmission Type | Reliability |
|----------------|---------------|---------------|-------------------|-------------|
| i7             | User Unified Authentication System (UUAS) | Wechat | DEP | no |
| i5             | RMB Trading System (RMBTS) CFETS Institution | Wechat | ETL | no |
| i6             | CFETS Institution Management System (CIM) | Wechat | ETL | yes |
| i2             | Tencent WeChat | Tencent WeChat | HTTP-REST | no |
| i4             | Tencent WeChat | Wechat | HTTP-REST | no |

### 5.3. Evaluation
The architecture repository management platform has been put into practice for a period of time. A group of main users have tested the platform, and then a satisfaction survey is conducted for evaluation.

![Figure 7. Evaluation result.](image)

The main users of the platform are categorized into developers, architects, reviewers, and project managers. Five people are randomly selected from each stakeholder category, and a questionnaire survey is conducted on them. The content of the questionnaire is prepared in advance. The score ranges from 1 to 4, e.g. for “Easy to use” question, the score 4 indicates that the platform is very easy to use. The average scores of questions are shown in fig. 7.

The results show that most stakeholders think that the platform is easy to get started, however, the platform's documentation is not sufficient enough yet. Most stakeholders believe that the architecture management platform improves the efficiency of architectural design and management as a whole compared to the previous management scheme, and the platform provides more convenient and effective functions. In general, the architecture management platform optimizes and standardizes the process of architectural design and management, which improves the efficiency of the company's architecture repository management, but it also has deficiencies such as lack of documentations, which need to be improved in the future.

## 6. Conclusions and Further Research

With the continuous advancement and development of the Internet and informatization, the enterprise's architectural design will inevitably become more important. In the future, we will continue to pay attention to the changes and developments in this field. At the same time, the deployment of Enterprise Architecture methodology in enterprises is also very important. How to guide the relevant personnel to use the architectural design method at low cost that meets the enterprise requirements is also a concern.

### Acknowledgments

This research is funded by the Science and Technology Commission of Shanghai Municipality (No. 18511103802).

### References

[1] The Open Group. (1995) The open group architecture framework(TOGAF). http://www.opengroup.org/togaf.

[2] The Open Group. (2004) The ArchiMate® Enterprise Architecture Modeling Language. https://www.opengroup.org/archimate-forum/archimate-overview.

[3] Gill, A. Q. (2015). Agile enterprise architecture modelling: Evaluating the applicability and integration of six modelling standards. Information and Software Technology, 67, 196-206.
[4] The Open Group. (2018) The TOGAF standard, version 9.2, 37.architecture repository. http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap37.html.

[5] Sessions, R. (2007). A comparison of the top four enterprise-architecture methodologies. Houston: ObjectWatch Inc.

[6] J. Zachman. (1987) Zachman framework. https://www.zachman.com.

[7] The U.S. Office of Management and Budget, Office of E-Government and IT. (1999) Federal enterprise architecture (FEA). https://obamawhitehouse.archives.gov/omb/e-gov/FEA.

[8] Lankhorst, M. (2009). Enterprise architecture at work: Modelling, communication and analysis. Springer Science & Business Media.

[9] Barn, B. S., Clark, T., & Loomes, M. (2013, February). Enterprise architecture coherence and the model driven enterprise: is simulation the answer or are we flying kites?. In Proceedings of the 6th India Software Engineering Conference (pp. 97-102). ACM.

[10] Pittl, B., & Bork, D. (2017, July). Modeling Digital Enterprise Ecosystems with ArchiMate: A Mobility Provision Case Study. In International Conference on Serviceology (pp. 178-189). Springer, Cham.

[11] Scholtz, B., Calitz, A., & Connolley, A. (2013, November). An analysis of the adoption and usage of enterprise architecture. In Enterprise Systems Conference (ES), 2013 (pp. 1-9). IEEE.

[12] Bedjeti, A., Lago, P., Lewis, G. A., De Boer, R. D., & Hilliard, R. (2017, April). Modeling Context with an Architecture Viewpoint. In Software Architecture (ICSA), 2017 IEEE International Conference on (pp. 117-120). IEEE.