System Engineering Supporting Digital Co-creation

● Tadahiro Uehara ● Akihiko Matsuo

All kinds of things are becoming digitalized, and the world of Digital Co-creation that combines these things to create new businesses is expanding. For companies to survive and grow their business in this world, they must create value by presenting their strengths to other companies in the form of application programming interfaces (APIs) and at the same time make use of other companies' strengths by way of APIs. This requires engineering technology for quick and reliable system development/operation centered on APIs. Fujitsu Laboratories calls the technical field in which this is achieved API-Publish & Compose. The Laboratories engages in research and development of system analysis technology for the purpose of making APIs of a company's particular strengths extracted from existing complex systems. We are also working on development and operation technology for quick and safe systematization of businesses using APIs offered by other companies. This paper describes the activities of Fujitsu Laboratories with regard to technologies required by both API providers and API users, and application examples.

1. Introduction

A new world where everything such as things and software are connected to the Internet is beginning. In this world, everything including work that used to be done manually until now will be digitalized, and connected and controlled by software. Therefore, the importance of application programming interfaces (APIs) as a common language is rapidly increasing.

According to a survey, the average annual domestic growth rate of the API management products market, which is a measure of API usage by companies, is predicted to be 41.1% in the period from 2015 to 2020.[1)] Another survey found that approximately one-third of responding companies generate revenue through the release of APIs, and that time to market is becoming shorter. Further, 83% of companies conducting advanced API management think that management of APIs is effective for achieving differentiation from competitors.[2)] In fact, not only advanced companies in the United States but also in Japan and elsewhere are leveraging APIs for business growth. In Japan, for example, a warehouse operator makes and releases APIs for personal storage services, and collaborates with third parties for the provision of new services through the use of these APIs.[3)]

Fujitsu Laboratories aims to realize Digital Co-creation that utilizes people, things, and ICT in a parallel, autonomous, and interactive manner to create services tailored to people. Through this, we support end users in all kinds of situations in social, economic, and everyday life activities, maximizing value and convenience for the benefit of end users.[4)]

For companies to survive and expand businesses in this world of Digital Co-creation, quick and high-quality system development and operation centered on APIs are required. However, to realize this, new engineering techniques and methods different from the traditional business system development that Fujitsu is good at are necessary. Consequently, Fujitsu Laboratories has named the technological area for realizing such development API-Publish & Compose, and conducts research and development in this area.

This paper describes the system engineering technologies required by both providers and users of APIs, including analysis technology for the provision of APIs from existing systems, as well as development and

operation technology for quick and safe systematization of businesses using APIs.

2. Two technological areas for Digital Co-creation

This section describes two strategies for business development in the Digital Co-creation world.

The first strategy is to create an ecosystem by opening up a company's strengths to the outside. The example of a US company converting its maintenance know-how for industrial machinery into a platform,[5] and the aforementioned Japanese warehouse operator, correspond to this. This strategy requires, in addition to reform of the company's business model for external disclosure, reform of the company's existing systems.

For example, consider a case where a company allows other companies to order via the Internet in a purchasing system where until now orders were input only from within the company. In this case, it is necessary to change the existing system so that other companies can access the order taking function from the Internet. Fujitsu Laboratories is engaged in R&D on technology called API-Publish to migrate existing systems so that they can be utilized in the world of Digital Co-creation. This includes converting functions into APIs in order to minimize the cost of system changes.

The second strategy is to incorporate the strengths of other companies for the creation of new value. For example, in the financial industry, the fintech market, in which venture companies provide services such as asset management and investment support that use the finance APIs of major financial institutions, is already up and running. In Japan, the revised Banking Act mandates efforts to introduce open APIs, and business creation through the use of APIs is bound to gain momentum.

However, when a company constructs a system using APIs provided by others, new problems arise. The achievement of stable system operation requires that the company manage the quality of the APIs provided by other companies, which cannot be controlled, and adapt its system according to specification changes of these APIs. Fujitsu Laboratories is engaged in research and development of technology called API-Compose that allows quick and safe construction of systems by combining APIs.[4] **Figure 1** shows the overall image of

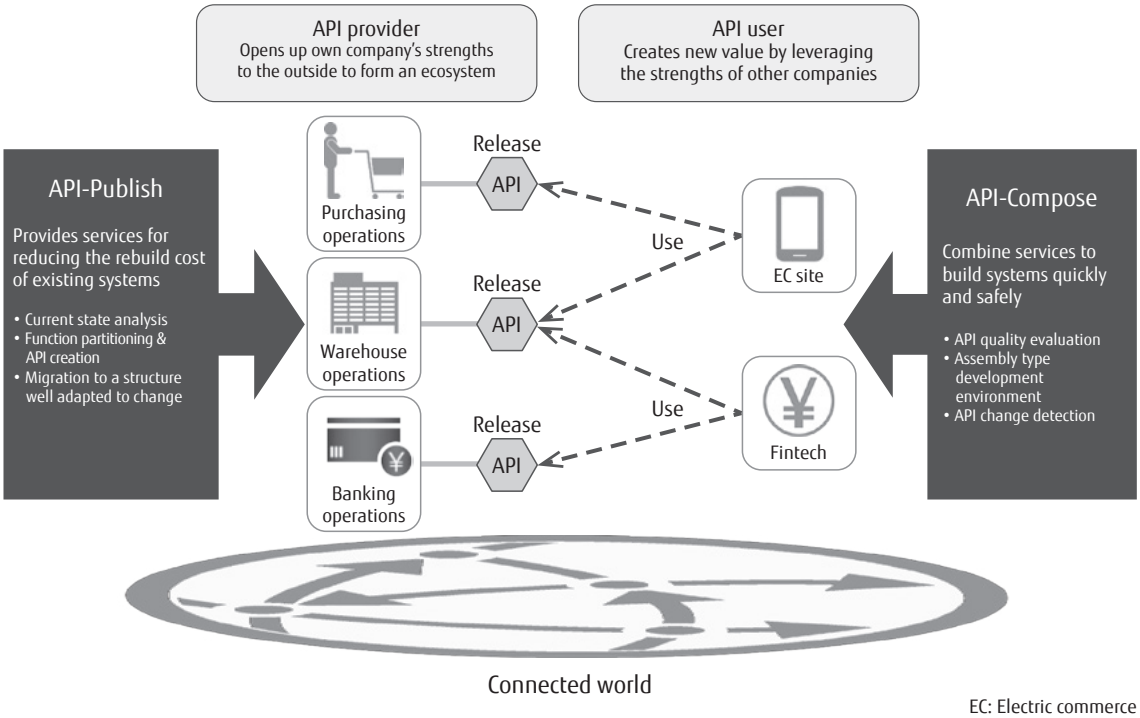


Figure 1
Overall image of API-Publish & Compose.

API-Publish & Compose.

The following sections describe API-Publish and API-Compose in detail.

3. System development technology providing APIs: API-Publish

To digitalize work that is a company's strength, it is necessary to convert business tasks such as order taking and placing into APIs so that they can be controlled by software. To achieve this, calling functions of the existing business system via APIs must be made possible. However, in the case of a new businesses that is started through the provision of APIs and the like, advance estimation of loads tends to be difficult. Therefore, an initial launch that minimizes risk and does not affect the existing established business, with gradual improvements to be made along the way according to changing needs, is desirable.

On the other hand, existing systems tend to become complicated as the result of maintenance actions over the years, and system changes are difficult to implement and often require many person-hours. To overcome this, a method to make partial or stepwise changes in a way that does not affect the entire existing system is needed.

API-Publish proposes migrating existing systems to a new system structure capable of easily accommodating changes through the following three steps, shown in **Figure 2**.

3.1 Step 1: Current state analysis

First, it is necessary to grasp the current state of the existing system and to estimate the required number of person-hours and the risks according to the purpose—for example API implementation or improvement of maintainability.[6] However, existing systems that have been in maintenance for a long time tend to have more complicated functions than when they were originally designed. As the influence of changes is often far-ranging, it can be difficult to estimate the scale of changes.

To deal with such situation, Fujitsu provides an asset analysis service that analyzes the source code of existing systems and evaluates their current state.[7] This service evaluates quality based on the scale of each system and program, the call structure, and various

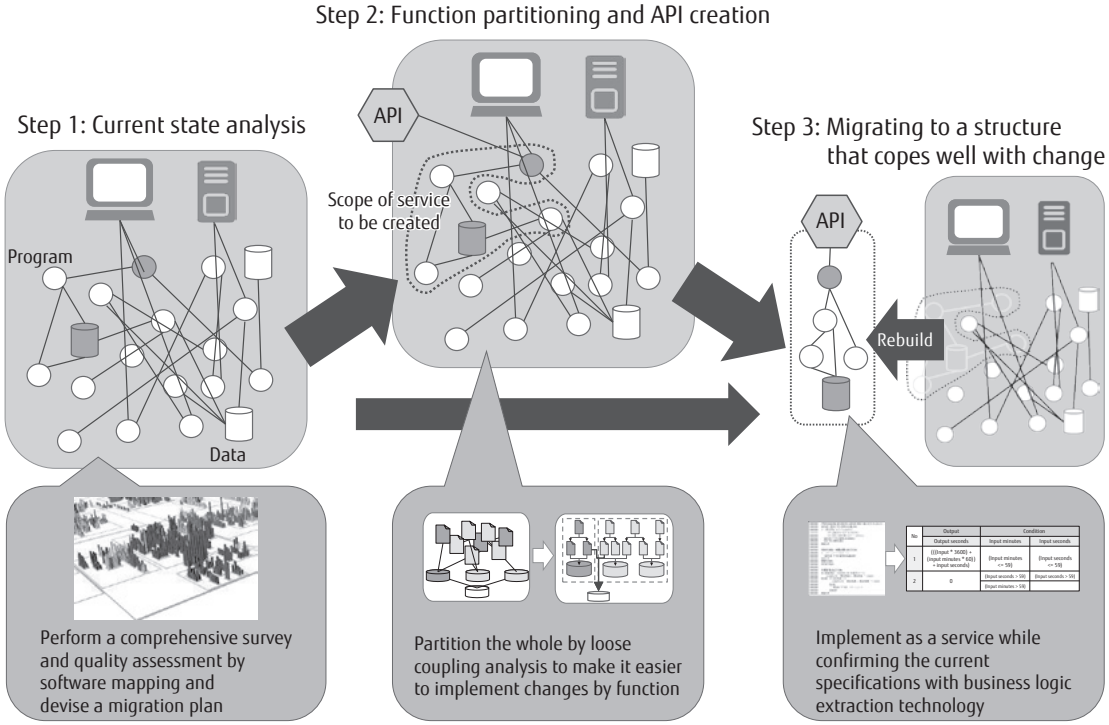


Figure 2
Steps of API-Publish.

measured metrics. It can also identify programs that are no longer used by dependency analysis that tracks call relationships.

Further, it provides also analysis and evaluation by software map technology[8),9)] that visualizes the software structure of the system in the form of a map. With this map, one can grasp the relationships among the functions in the system and any complexities, and estimate the scope of influence of any changes by examining the functions to be converted to APIs and the relations between the parts to be changed and the other parts.

3.2 Step 2: Function partitioning & API creation

Based on the results of the present situation analysis, what is needed in the next step is to make changes to the existing system according to the purpose, such as improving maintainability and adding APIs. However, inside the existing system, various functions and data directly refer to one another, and making changes somewhere will often affect the whole. For this reason, even in the case of small-scale changes, many person-hours are required for the analysis of the scope of influence and testing for operation verification, and it takes a long time until release.

Fujitsu Laboratories has developed loosely coupled analysis technology as a technique to make it easy to make changes inside such existing system.[10)] This technology analyzes the relationship between programs and data, clusters programs and data that have strong relationships, and finds the boundaries where dependencies such as program calling and reading/writing of data across clusters are least prominent.

Based on the results of this analysis, APIs are created by taking into account business needs and rearranging dependency relationships among functions. APIs can thus reduce the impact of changes on other functions and provide functions for outside users.

3.3 Step 3: Migrating to a structure well adapted to change

Offering the functions of an existing system as APIs to outside users may expose the existing system to effects such as increased load. In that case, implementing functions to be provided externally as services separately from the existing system allows API implementation without actually having to change the existing system. Further, since one can substitute only services without affecting the existing system, quick response is possible even when API changes become necessary.

This approach requires the implementation of functions equivalent to the specification of the existing system as services, and operating them in parallel with the existing system. For this reason, Fujitsu Laboratories has developed business logic extraction technology that extracts for reuse the business logic implemented in programs in a form that human beings can easily understand.[9),11),12)]

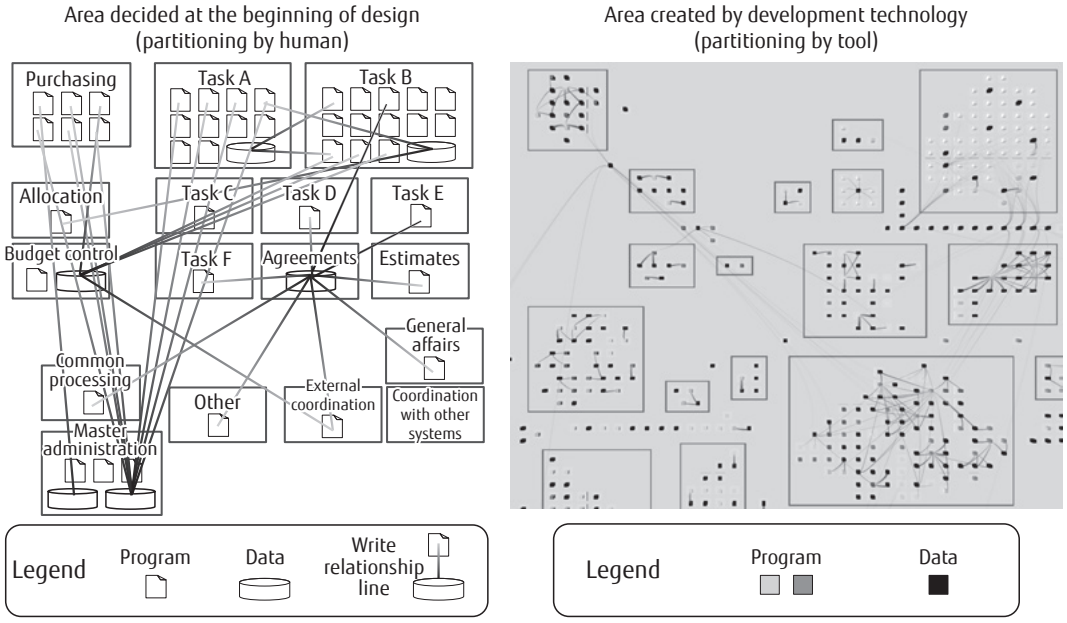
This innovation makes it easy to design services that have the same functions as existing systems by determining the range including the functions of interest from the results of the current state analysis and function partitioning, and extracting and reusing the business logic implemented within that range. Also, by analyzing similarities and differences between the specifications extracted in this way and the specifications of available external software as a service (SaaS) and in-house common services, it is easy to replace existing services.

Further, by migrating the functions of the existing system stepwise to a new system or SaaS by this approach, it is possible to finally migrate the entire system to a native cloud form.

3.4 Application examples

An example of partitioning the structure of a purchasing operations system by using loosely coupled analysis is shown in **Figure 3**. In the naming conventions of the original design, the system, which was composed of 2,151 programs, was partitioned into 71 areas. As the result of analysis with this technology, the system was partitioned into 199 areas, thereby achieving a finer arrangement with greater granularity. Despite the larger number of divisions, the total number of dependency relationships between the areas decreased to 84.4% of the number at the time of the original design, and since the data in the areas are not written from the outside, it is easy to sort the dependency relationships.

By deciding the scope of service creation in accordance with the areas obtained with this technology, independence from the outside is enhanced and the



	Partitioning by human	Partitioning by tool
Number of areas	71	
Total number of data dependencies	3,457	2,917 (84.4% of partitioning by human)
Writes to data within areas	196 (There is writing from outside the areas)	0 (There is no writing from outside the areas)

Figure 3
Example of loosely coupled analysis.

burden of creating the required APIs can be reduced.

4. System development technology using APIs: API-Compose

When constructing one's own system using APIs provided by other companies, the fact that APIs keep on changing in specifications even after system construction raises new issues. **Figure 4** shows the life cycle of the system and the new issues that exist at each step. For example, in the requirements definition & design process, issues related to the search for suitable APIs, such as "Are there APIs that meet the requirements?" and "Is this business partner reliable?" present themselves.

At Fujitsu Laboratories, among the issues shown in Figure 4, we research system development and operation technologies for examining APIs, connecting APIs, and modifying the system. Through these technologies, it is possible to realize system construction that combines APIs of other companies faster and with greater reliability. These three technologies are described below.

4.1 API quality evaluation "examining APIs"

When utilizing APIs provided by other companies, it is difficult to judge to what extent the quality of the APIs is reliable. Unless one knows the quality of APIs, it is not only necessary for users to be aware of functional and non-functional requirements that need to be taken into consideration, but it is also difficult to design the development and operation work itself. This includes for example estimating the length of time required for API learning, and estimating person-hours required for handling changes in API specifications.

In this regard, Fujitsu Laboratories is aiming for the evaluation of API quality by developing metrics that quantitatively evaluate API learning ease and specifications stability, in addition to existing evaluation metrics

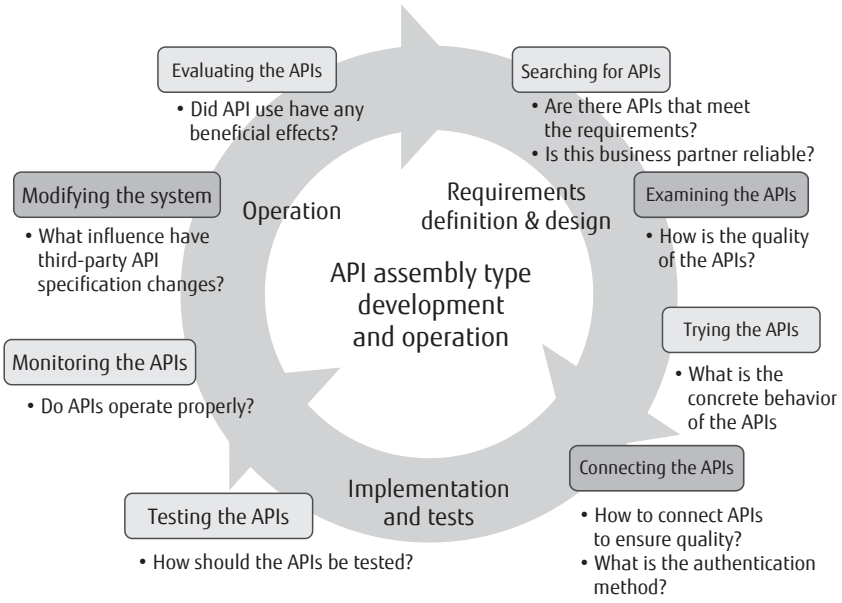


Figure 4
System life cycle and issues in API assembly type development and operation.

such as response time, availability, average recovery time, and number of users.

First, we devised sample coverage values as metrics for API learning ease. This is because it was thought that it was important that use samples be comprehensively provided to aid understanding of API specifications.

A sample coverage value is a numerical representation of the degree of correspondence between the specifications of API request and response parameters described in API documents and the samples (such as "Kyoto" for the "City" parameter). This value is the harmonic mean of the following two coverage rates. If there are few samples to aid understanding of specifications, this value is lower, and based on that, it can be estimated that time to mastery will be long.

1) Sample coverage ratio

The rate at which samples exist for the APIs whose specifications are described and their parameters.

2) Specification coverage ratio

The rate at which specification descriptions exist for the APIs in the samples and their parameters.

We also devised a compatibility maintenance value as a metric for the stability of specifications. The compatibility maintenance value is a numerical value that indicates whether the specifications maintain backward compatibility every time the API is upgraded. This value is obtained by comparing the API documents of the two versions of each API, calculating the percentage of APIs that maintain backward compatibility, and obtaining the mean value by weighting this percentage for each generation. If this value is low, it is highly likely that measures to deal with API changes will be required.

In API quality evaluation, the quality of each API is visualized by presenting the above two metrics developed originally, and four existing metrics as a radar chart on the API portal. When using APIs, this allows developers to design APIs while taking into consideration the aspect of quality.

4.2 API assembly type development environment "connecting APIs"

In Digital Co-creation, to quickly change a business according to needs, a development environment that allows quick systemization of business flows and business rules is required.

For this purpose, many flow-based development environments have been proposed. By describing functions in flow form and combining APIs, it is possible to easily assemble the business flows of the company. However, to achieve stable system operation, it is necessary to build in other elements besides business flows and rules, such as the implementation of exception

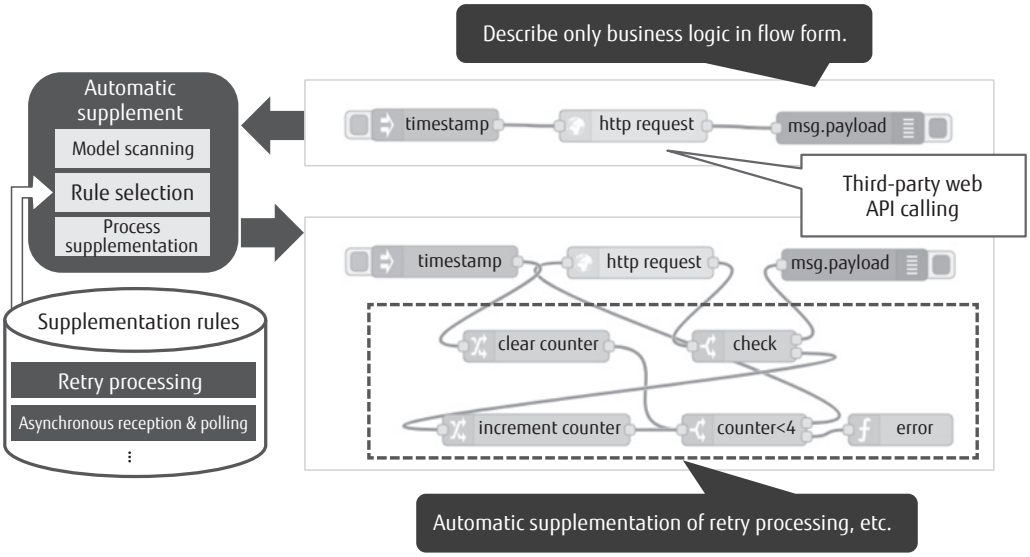


Figure 5
API assembly type application development environment.

handling. It is also necessary to implement functions necessary for improving reliability such as retry processing, timeout, and circuit breaker functions. However, implementing these functions by flow description is complicated and involves many person-hours.

To solve this problem, Fujitsu Laboratories translated these reliability enhancement functions into patterns, and developed a technology to automatically supplement the business flows described by developers.

Specifically, as a flow editor, we use Node-RED, which is a development environment provided as open source software (OSS). By converting the representation of the Node-RED flow in JSON format according to the rules for each function to be supplemented, a partial flow realizing the reliability improvement functions is embedded in the original flow. This allows developers to develop systems by concentrating on creating business flows and business rules. An example of supplementing the retry process is shown in **Figure 5**. After the developer describes the flow including invocations of web APIs, our technology automatically complements the flow to incorporate the retry process into the flow.

4.3 API change detection "modifying the system"

The specifications of APIs may change at a timing not expected by the user. When the specifications of open APIs such as Twitter APIs are changed, there are cases where applications that were able to use them until now become inoperable. However, when doing business with Digital Co-creation, it is desirable to formulate rules concerning API changes in advance between the API provider and user.

Therefore, Fujitsu Laboratories proposes to release changed API specifications in the OpenAPI Specification format[13] before upgrading APIs. This allows the user to automatically detect changes in APIs and take suitable action in advance. Creating API specifications in the OpenAPI Specification format on the API provider side also has merits in terms of development efficiency, such as automatic source code generation therefrom.

In addition, Fujitsu Laboratories has developed a technology to judge whether it is necessary to change the applications on the user side due to changes in the published APIs. Specifically, by comparing the API specifications of two different versions described in the OpenAPI Specification format, and by analyzing the corresponding end points, parameters, presence or absence of response, and so on, it is possible to determine whether backward compatibility is maintained. Notifying the developer utilizing APIs from other companies of the judgment result allows the developer to respond smoothly to changes in these APIs.

4.4 Application examples

To verify the effectiveness of the above-described development technologies, three technologies were incorporated into API gateways and API portals, and systemized as one platform. Currently, Fujitsu Laboratories is testing that system in house. Further, using APIs developed in house at Fujitsu Laboratories, we are evaluating API learning ease, specification stability metrics, and the effectiveness of change detection technology.

5. Realization of end user computing

As Digital Co-creation evolves further and everything in the world can be controlled by APIs, business and software will become even more integrated. That is, it will be assumed that business departments and executives themselves develop software using APIs.

The API assembly type development environment described in this paper is an initiative contributing to such world, but it is just a beginning. Such development environment must make it easy to develop software even if one is lacking in ICT skills, and it must have a mechanism to ensure stable operation of the whole system even in the event of problems.

Under the concept of API-Publish & Compose, Fujitsu Laboratories is working to realize new end-user computing that ensures the reliability of entire enterprise systems while making possible software development by end users of business systems such as business departments and executives.

6. Conclusion

This paper described the concept and research and development technology of API-Publish & Compose, which are API-based system development and operation technology necessary for growing businesses with Digital Co-creation.

This technology for connecting existing systems to Digital Co-creation is unique to Fujitsu, which has been working on business system development for many years, and it is a powerful weapon for providing our strengths in the form of APIs. Further, this technology is also expected to allow the realization of systems that utilize APIs easily and with the high reliability characteristic of Fujitsu.

Going forward, to meet this expectation, we will further mature API-Publish & Compose technology and conduct further R&D as we aim for a world where everyone can develop software safely using APIs.

References

1) ITR: ITR Market View: System Collaboration/Integrated Middleware Market 2016 (in Japanese). *https://www.itr.co.jp/report/marketview/M16001300.html*
2) CA Technologies: APIs: Building a Connected Business in the App Economy. *https://www.ca.com/us/rewrite/articles/api-management/apis-building-a-connected-business-in-the-app-economy.html*
3) Warehouse TERRADA: The API of the cloud storage service "minikura" by Warehouse TERRADA has received a patent. *https://www.terrada.co.jp/wp/wp-content/uploads/2016/05/20160512_minikuraAPI-Patent_Press-Release_EN1.pdf*
4) M. Kawaba et al.: Future Trends in Software Technology to Support Digital Innovation. FUJITSU Sci. Tech. J., Vol. 54, No. 3, pp. 70–76 (2018). *http://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol54-3/paper16.pdf*
5) General Electric: Industrial Data Fabric. *https://www.ge.com/digital/predix/industrial-data-fabric*
6) IPA/SEC: User Guide to Successful System Reconstruction, 2nd Edition (in Japanese). *https://www.ipa.go.jp/files/000057294.pdf*
7) Fujitsu: Application Operation and Maintenance FUJITSU Application Management and Outsourcing Asset Analysis Service (in Japanese). *http://www.fujitsu.com/jp/services/application-services/application-management-outsourcing/apm/services/assetanalyse/index.html*
8) Fujitsu Laboratories: Fujitsu Develops World's First Software Map Generating Technology to Leverage Application Portfolios. *http://www.fujitsu.com/global/about/resources/news/press-releases/2012/0229-01.html*
9) Y. Maeda et al.: An Approach to Transforming Systems for Execution on Cloud Computing Systems. FUJITSU Sci. Tech. J., Vol. 53, No. 5, pp. 71–77 (2017). *http://www.fujitsu.com/global/documents/about/resources/publications/fstj/archives/vol53-5/paper11.pdf*
10) Fujitsu Laboratories: Fujitsu Develops Technology to Partition, Visualize Complex Business Application Structures by Function. *http://www.fujitsu.com/global/about/resources/news/press-releases/2017/0518-01.html*
11) Fujitsu Laboratories: Fujitsu Develops Technology to Automatically Extract Business Specifications in Programs.

http://www.fujitsu.com/global/about/resources/news/press-releases/2016/1011-01.html
12) Y. Maeda et al.: Extraction of Business Specifications from Large-Scale Programs for Modernization of Business Systems. FOSE2016, 2016.
13) Swagger: OpenAPI Specifications. *https://swagger.io/specification/*

Tadahiro Uehara
Fujitsu Laboratories Ltd.
Mr. Uehara is currently engaged in research and development of web APIs and native cloud applications.

Akihiko Matsuo
Fujitsu Laboratories Ltd.
Mr. Matsuo is currently engaged in research and development of business system maintenance and reconstruction.