

Concurrent development by multiple developers in integrated factory automation development environment using open-source system “Git”

IWAMURA Shintaro

Along with the recent global trend, consumer needs in the market are getting more diversified, and manufacturers are required to produce new products in a short cycle. Manufacturing companies have installed new equipment and added new production patterns to existing systems to manufacture new products.

The control system of production equipment has become more complex, and the scale of control software has increased. Manufacturers have to respond to a short cycle of product release, so it is necessary to shorten the development period of the production equipment. The number of cases in which the development of the control system is shared by multiple developers is increasing. In such development, collective management of settings and control program is important, but there have been problems such as matching of duplicate changes and management of change history.

This article describes the cooperating technology that we developed for SysmacStudio, which is Omron’s integrated factory automation development environment, and Git, an open-source version control system. We confirmed that it can shorten the development period by managing changes of software developed by multiple developers and reducing the backward man-hour caused by mistakes. We also confirmed that the development period could be shortened by using the completed control software and applying this technology to the control software development (derivative development) of similar production equipment.

1. Introduction

Omron offers “SysmacStudio” as an integrated factory automation development environment, which enables the entire FA system consisting of robots and safety devices as well as sensors (input), output devices (output), and Controllers, to be developed with a single application software. The SysmacStudio makes it possible to centrally manage control software, such as settings and control programs, for devices that control production equipment. The SysmacStudio handles this centrally managed control software as project data.

As the production equipment has become intelligent and complex in recent years, the number of cases that multiple developers jointly develop the system is increasing; specialists of different devices gather and develop a production system as a team. In order to shorten development period, development of a programmable logic controllers (PLC) programs are also shared with multiple developers. In such cases, copies of the master

project data are distributed to each developer to edit and then changes are applied to the master project data at certain timing using the edit functions of the SysmacStudio.

Before applying changes, the developer needs to find the data to change, check whether other developers made any change to the same data, select a change if the same data was changed in duplicate, and check for problems on changes. It has been a problem that it takes time for the course of the tasks. In addition, since this operation is performed manually, a failure can be mixed in the control software due to a developer’s judgment and operation mistakes, which also causes a backtracking of the development.

In software development, a version control system is widely used in order to manage changes of software. The system manages change histories of source code files or document files of software, such as a change date and time, a modifier, and a content of the change. Also, the system has a mechanism to store the change history of each file. Even if one file is simultaneously edited by a plurality of people, when there is no

Contact : *IWAMURA Shintaro* shintaro.iwamura@omron.com

duplication of editing, its automatic merge function efficiently aggregates and manages file changes and automatically applies each editing to one file.

Therefore, we used this version control system to resolve issues in control software development done by plurality of people. To achieve efficient and accurate change management of SysmacStudio project data in the version control system, we selected three technical issues related to updating of project data and examined the solution.

This paper, in addition to concrete solutions for these technical issues, describes a concrete case in which the man-hours for managing changes in project data could be reduced by the integrated FA development environment in cooperation with the version control system. This technology has already been released as our product SysmacStudio Ver.1.20.0 team development option in October 2017.

2. Issues

The project data includes various file types: data saved as text such as control programs and graphic data for display devices (HMI). Data of a control function of a FA system is divided and saved in multiple files of the project data in some cases, and in other cases, to eliminate redundancy of data defined in the project data, data are mutually referenced between the files.

If the project data in such files is updated by the automatic merge function of the version control system, without using the SysmacStudio, then referencing relationship of the data can collapse. Even if a developer checks the difference of the files using the functions of the version control system, changes cannot be merged correctly because the data is unreadable. If the developer edits the file data directly, the project data might be broken.

Considering that, we examined the mechanism to manage the changes in project data and tried to combine the SysmacStudio and the version control system to prevent the abovementioned issues. To realize this mechanism, we tried to resolve the following three technical issues.

2.1 Development of client tool function

In general, developers use client tool to operate the version control system. The client tool has necessary functions to operate version control system and shows changes, change history, and difference between the versions. Since the project data includes files that developers cannot understand, it is necessary to check the changes and change history of the project data on the SysmacStudio that generates project data, without being conscious of the difference of each file. For that reason, merging the client tool functions of the version control

system to the SysmacStudio is the first technical issue.

2.2 Comparison of project data and development of merge function

A developer can compare a file being edited with the master file registered in the version control system using the client tool and register a new master file that includes information of the differences (compare and merge functions). However, this function can be used only for text files. It is not possible to register difference information in the project data including multiple file types, such as graphic files for display, to the version control system. Accordingly, realizing compare and merge functions of the project data on the SysmacStudio by linking with version control system is the second technical issue.

2.3 Assurance of consistency of project data

For project data that is mutually referenced between the files, and the consistency should be assured, it is necessary to prevent incorrect changes made by version control system and detect unauthorized changes to restore correct data from the past history in the version control system. Detecting abnormality relating to data consistency and restoring correct project data is the third issue.

3. Technology

This section explains a cooperative technology between Git¹⁾, which is one of the version control systems widely used in the field of software development, and the SysmacStudio. Git is a distributed version control system and has a feature that it can be used even on site where a network cannot be used for security reasons.

3.1 Integration of application software using window shell extension

To operate version control system on the SysmacStudio, we developed collaborative technology with the Git client tool (TortoiseGit²⁾). Specifically, the Git function is made available by calling the TortoiseGit function from the SysmacStudio through the Windows shell extension (Fig. 1). The Windows shell extension is a function to customize and expand the context menus of files and folders on the Windows explorer. We used this Windows shell extension and integrated the TortoiseGit context menu to the SysmacStudio (Fig. 2). By that, TortoiseGit functions became directly operable from the SysmacStudio.

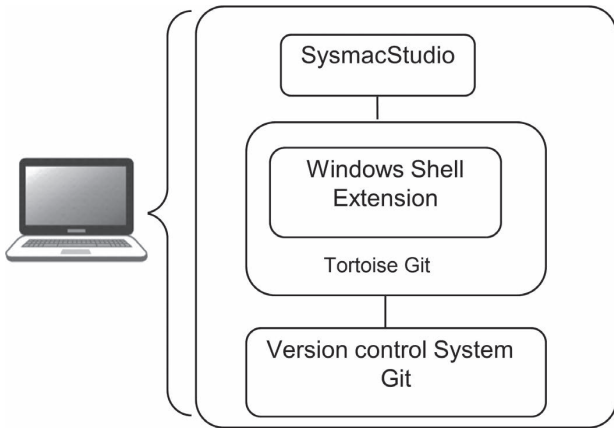


Fig. 1 Collaborative configuration with version control system

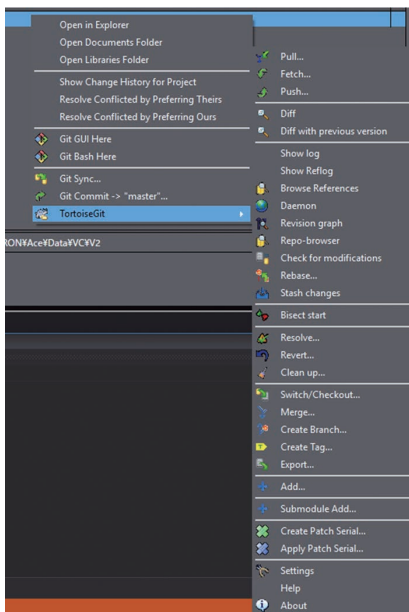


Fig. 2 TortoiseGit context menu displayed on the SysmacStudio

In addition, Windows shell extension allows us to use the latest functions of the TortoiseGit on the SysmacStudio. Since TortoiseGit and SysmacStudio are combined using the interface of the Windows shell extension, each function of application software can be strengthened independently without affecting each other. Both Git and TortoiseGit have been updated many times since Omron released the SysmacStudio that can be used with the version control system, and new functions of the TortoiseGit such as the revision graph function and the improved functions of the Git could be used immediately in the SysmacStudio. In the meantime, Omron did not need to modify the SysmacStudio to follow the evolution of TortoiseGit and Git.

3.2 Intuitive operation to show and merge the differences

We developed a function that helps users to visually compare settings and control programs of each control device that

constitutes the machine and to merge them on the SysmacStudio after confirming difference information of the project data acquired from the version control system.

This section explains the comparison function of ladder programs and screen data of the display device.

The ladder program is one of widely used program languages for the PLC to describe logic circuits that use relays. Currently, instructions for complex calculation are also prepared.

It is difficult to find changes in the ladder program that graphically displays logic circuits. Therefore, we added a function so that each element can be compared by extracting difference in each circuit element unit (Fig. 3). Circuit elements are compared to find the difference from the two directions of the input side and the output side along the connection order (power flow) of the circuit, and difference results are compared. This enables a detailed comparison of the ladder programs that have been changed on site (Fig. 4).

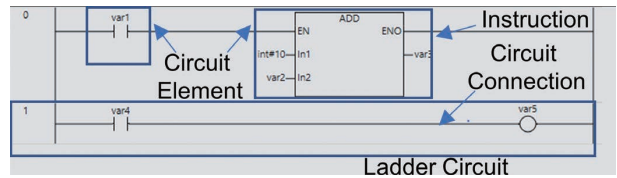


Fig. 3 Data configuration in the ladder editor

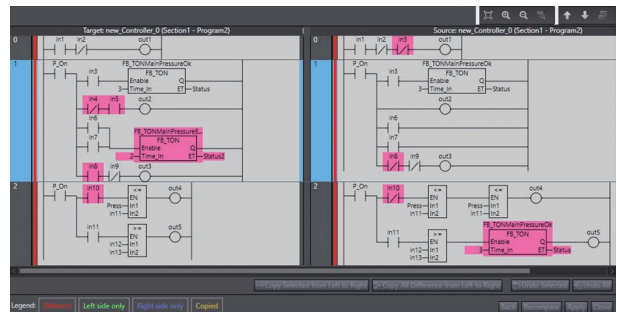


Fig. 4 Comparison window of the ladder editor

SysmacStudio has a function called PageEditor for editing screen data of the display device. The screen data to be edited on the PageEditor has multiple types of data, such as shapes, colors, and text data of the buttons to be switched depending on the status. SysmacStudio manages such data and relations of the data in hierarchical configuration. SysmacStudio organizes the difference information of the data acquired from the version control system in a hierarchical data order and shows all difference details so that they can be easily monitored on the PageEditor screens. For example, if a text string on the button in a screen is changed, PageEditor screen shows that there is a change for the button. When the screen to show detailed text data is displayed, the user can confirm the difference (Fig. 5).

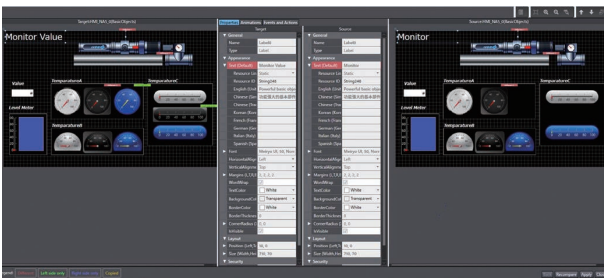


Fig. 5 Comparison window of the PageEditor

For this PageEditor comparison function, Omron is currently applying for patent (as of November 2018).

As mentioned above, we realized functions for intuitively confirming changes and collectively merging changes on the SysmacStudio.

3.3 Assured consistency between data using MD5 hashes

To detect errors relating to consistency of project data caused by the version control system, we developed a mechanism to assure consistency between the data.

In order to ensure consistency in the interdependent data, we developed a mechanism of treating the interdependent data as one data group by using MD5 hashes. MD5 hash is a hash function that generates a value of 128 bit from the original data of arbitrary length. Seeing the data generated by the MD5 hash, the user can understand whether the data has been changed. Hash value is a technology widely used in exchanging web data. There are several types of hashes including MD5, SHA-1, and SHA-2, and they are used for security purposes and ensuring the identity of files. This time, our objective is to check consistency in the contents of the files, so we adopt MD5 hashes with short calculation time and small size.

We developed the mechanism in which multiple interdependent data is grouped as one data, an MD5 hash is created for that, and the hash value is updated each time the data is changed (Fig. 6). The hash value is saved as a part of project data, and if the data with dependency relation is partially changed, inconsistency is detected in MD5 hashes. Thus, inconsistency in project data can be detected. If inconsistency is detected when reading the project data in the SysmacStudio, the version control system shows detected data and helps the user to restore correct data from the change history (Fig. 7).

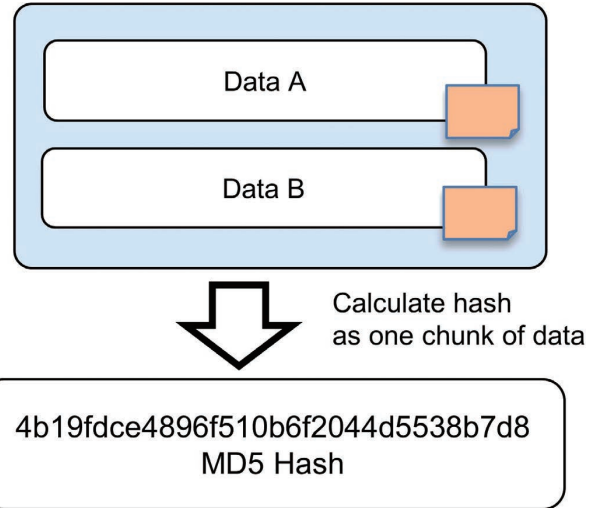


Fig. 6 Generation of MD5 hash

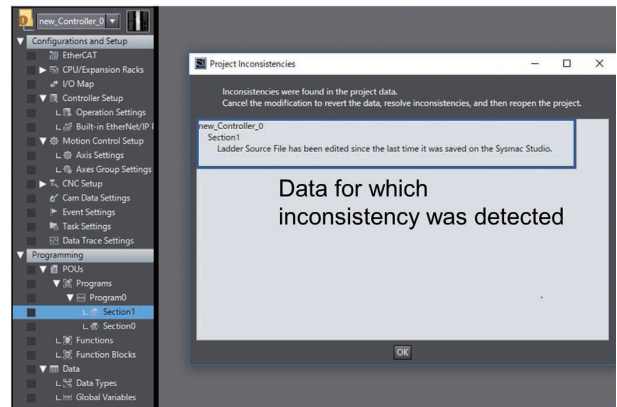


Fig. 7 UI to detect data inconsistency

4. Validation of technology

We verify that actual manufacturing equipment can be developed by a plurality of people and that the man-hours required for change management can be reduced by using the SysmacStudio in cooperation with the version control system Git. The use case for verification is defined based on the results of interviews with multiple customers that are developing production equipment by multiple engineers.

Before describing the content of the verification, we introduce the system configuration and operation of the distributed version control system Git (Fig. 8) briefly.

A distributed version control system has a storage area called a local repository in a computer of each user and a remote repository for sharing data between users. The repository stores the version control target project data and its change history information. Each user can synchronize the local repository and the remote repository at any time. When the change in the local repository is shared with other users, an operation of reflecting (pushing) the change in the remote repository is performed, and

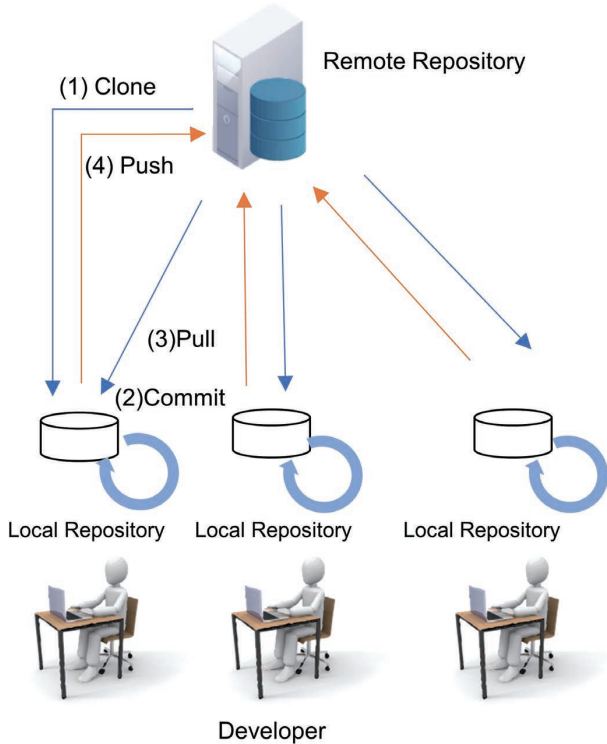


Fig. 8 Distributed version control system

to reflect the change of other users in the local repository, an operation of capturing (pulling) the change from the remote repository is performed. Users can work with local repositories and reflect them in remote repositories when changes have been accumulated to some extent.

4.1 How to manage the versions with Version Control System

There are the following three operation configurations depending on the difference in the sharing method of the remote repository. Developers can select one of them:

- Using a shared folder on a computer to share it as a remote repository
- Building a dedicated Git server to share a remote repository
- Utilizing a Git server service on the Internet to share a remote repository

We verified that all three of the abovementioned configurations can be supported by the SysmacStudio by operating an actual server. Developers can select a suitable configuration to the development of production equipment in their companies for operating the version control system.

4.2 Development of production equipment by multiple developers

This section explains the flow of developing a production equipment by two developers (developer X and Y) using the

version control system.

First, we determined the roles of the developers for editing of the project data. The SysmacStudio's version control function helps multiple developers to edit one project concurrently and to merge the change automatically by "pushing" it to the remote repository. However, if multiple developers edit the exactly same item and try to merge the change, a conflict occurs and merging fails. To prevent this conflict, the roles are shared as follows:

Developer X: Create general configuration of SysmacStudio projects and edit settings, data types, global variables, and ladder programs (ProgramX)

Developer Y: Edit ladder program (ProgramY)

Fig. 9 illustrates the relationship between the repositories and branches.

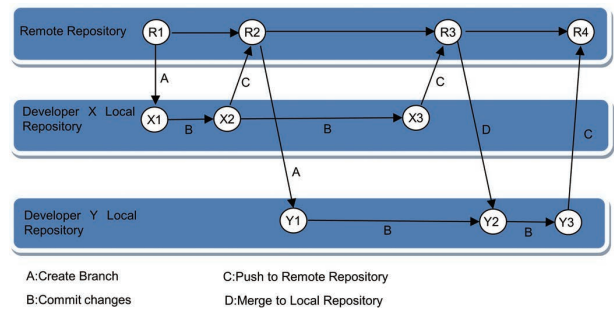


Fig. 9 Flow of development by multiple developers

Each developer creates a working branch on the local repository of their personal computers and makes some changes in the program. After editing the program, the developer "pushes" the changes in the local repository to the remote repository, and the changes are reflected. The data in the local repository of their personal computer is merged via the remote repository.

For the detailed information, refer to the SysmacStudio Startup Guide (Project Version Control Function)³⁾.

In our in-house verification, we confirmed that multiple people can develop a machine concurrently by cycling the flow in Fig. 9. In addition, as shown in Table 1, we confirmed that when the version control system is used, as compared with the conventional development in which the version control system is not used, the repeated work to manage the changes can be greatly reduced.

Table 1 Verification results

Use cases	Version Control System: Not used	Version Control System: Used
Time to find a target project. (Find the changes)	1 min	10 sec
Time to start comparing data. (Find data changed in duplication)	1 min	5 sec
Time to merge program data. (Select a change from duplicated changes)	2 min	30 sec

4.3 Development of derived production equipment by multiple developers

Next, we examined the collaborative effect with the version control system in derived development of customer's actual equipment with multiple developers. Derived development is a method used when the variation of production equipment increases; manufacturers develop a control program commonly usable for multiple similar machines with different hardware configurations.

Verified case was derived development of project data for a packaging machine by a plurality of developers.

This customer is a major packaging machine manufacturer who has strength in customizing a packaging machine in accordance with the needs of each customer. The company develops a machine by removing unnecessary functions from the basic configuration including all the packaging processes and combining functions and specifications required by the customer. Since the machine is developed to meet the customer's request, the machine configuration often becomes complicated. When a machine is developed by a plurality of people, the number of man-hours required to manage changes increases, for example, it takes time to understand who changed what in the program and to merge the project data. Even though the machine is developed by multiple people, productivity does not improve. In addition, as the scale of the machine becomes larger, it takes time to specify the cause of errors.

This time, the company adopted the SysmacStudio that has cooperative functions to use the version control system and developed production equipment.

The packaging machine like the one shown in Fig. 10 was developed under the conditions shown in Table 2 below.

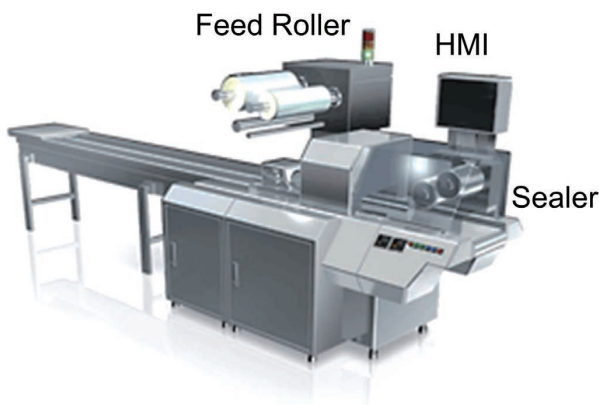


Fig. 10. Packaging machine system

Table 2 Roles of developers (5 people)

Roles of development	No. of people
Project configuration/settings in the SysmacStudio	1
Control program	2
Screen of the display device (HMI)	2

For the control system of the equipment, the following Omron devices were used.

1. NJ-series PLC CPU Unit
2. NX-series IO system
3. IS-series AC Servo System
4. NA-series Programmable Terminal (HMI)

After this development project was completed, we received the following evaluations and feedback from the packaging machine manufacturer.

Until they started to use this version control system in development, the project data was organized using the SysmacStudio export function and saved as a single file with dated file names. When a problem occurred, they compared projects using the comparison function. In this method, since the file was opened, verified, and compared each time, it took a long time to check the history of a specific correction point in time series and to specify the cause of the trouble.

After the company introduced the SysmacStudio version control system, it could shorten the time to specify error factors to 1/4 or less of the conventional version because they could search for the cause from the past history and display the comparative result in a moment. As a result, man-hours required to manage the version of the file in the development process could be reduced to half or less.

5. Conclusions

In order to develop a SysmacStudio project by multiple developers, we realized cooperative functions between Git and SysmacStudio by using the client tool TortoiseGit of the version control system Git and the Windows shell extension technology.

We also developed visualized comparing technology for various data types of the integrated FA development environment so that users can intuitively compare and merge data.

In addition, we used MD5 hashes to establish the technology to ensure consistency between data contained in SysmacStudio projects data.

We used this technology in a development project by multiple developers and confirmed its effectiveness. We confirmed that this technology is also effective not only for the development by multiple developers, but also for the derived development of the equipment.

The number of man-hours required to manage project data can be halved in the equipment development, and the introduction of SysmacStudio incorporating this technology is increasing globally.

We will continue to contribute to customer's manufacturing innovation by leveraging open-source technology evolution.

References

- 1) <https://git-scm.com/> (Retrieved 2018-11-19)
- 2) <https://tortoiseGit.org/> (Retrieved 2018-11-19)
- 3) Omron "SysmacStudio Startup Guide (Project Version Control Function), P-125

About the Authors

IWAMURA Shintaro

Software Development Dept. Controller Div.

Product Business Division H.Q.

Industrial Automation Company

Speciality: Software Engineering

The names of products in the text may be trademarks of each company.