

UMS: Software to Automate Operators' Actions

Fumihiko Yokose[†] and Akira Inoue

Abstract

We have developed a unified management support system (UMS), which is software that reduces the workload on operators and operating costs by automating operations. UMS makes it extremely easy for an operator who is not a computer expert to record and replay actions taken on systems. Since most modern systems are extremely complex, UMS can improve system operation efficiency. This article describes the features and technologies of UMS and gives usage examples.

1. Introduction

Effective operations support systems (OpSs) are essential to achieve efficient advanced operations. For example, the management of one access network requires the information management functions provided by several OpSs. Without these OpSs, we could not build and operate commercial access networks and provide high-quality network services to customers.

The key to OpS efficiency is providing support to operators because they cannot be replaced by systems because they make sophisticated judgments and perform field work. Thus, OpSs must be easy for operators to use. However, this is not always true: modern OpSs provide many sophisticated operations. The key problem is that the OpS interface is not user-friendly, and trying to work with its characteristics can be very time-consuming and expensive. An OpS interface that is unsuitable for operator interaction leads to excessively high error rates and failures.

Our solution is a unified management support system (UMS). UMS is software that automatically manipulates, or automates, the OpS interface on behalf of the operator. It improves the ease-of-use of the OpS and reduces the operator's workload. Consequently, UMS reduces operating costs.

2. Approach

2.1 Problems with OpS interface

Operations vary over time, but it is often difficult to continue using the same OpS interface. Moreover, modifying an OpS interface is not easy because it takes time and money. While it is possible to implement a new operation by using the original OpSs, the operator's workload will be much greater because the original interface was not designed to support the new operation. This leads to high operating costs.

Another problem is that databases slowly become corrupted owing to changes in the managed entities over time. The solution is to occasionally indentify the actual situation and alter the corresponding database entries. However, we cannot directly alter the databases because the data must be checked by the logic written into the OpSs, and usually only the OpSs can access and modify databases. Thus, we need a tool that eliminates the manual work needed to feed data items through the appropriate OpSs to check for consistency and perform any necessary updates.

Consider, for example, the case in which a new high-speed plan on the access network is started. The OpS that handles customer records does not to be modified, but the new plan uses new equipment, so an operator in the field should confirm the equipment with the equipment-management OpS. Thus, the operators' workload will be high.

[†] NTT Access Network Service Systems Laboratories
Yokosuka-shi, 239-0847 Japan

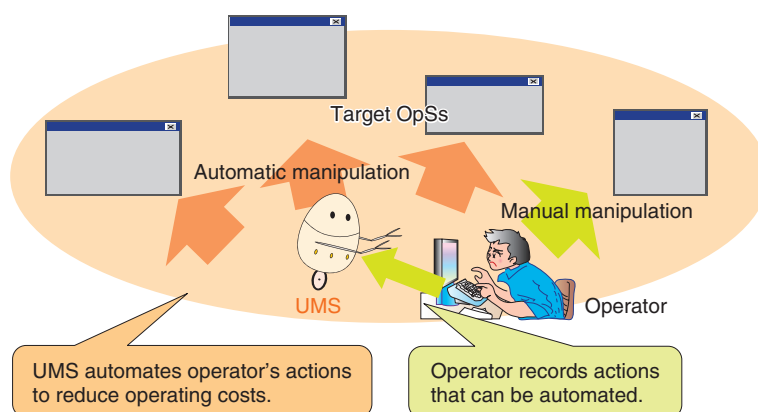


Fig. 1. Approach of UMS.

2.2 Solutions

A prime objective of UMS is to reduce operating costs. To achieve this, UMS eliminates operator actions that can be automated (Fig. 1). Thus, UMS can reduce the operator's workload and improve the efficiency of operations.

3. System overview

UMS has two particular features.

3.1 No modification of target OpS

The target OpS does not need to be modified because UMS manipulates the target OpS in the same way as the operator. Actually, OpS manipulation means manipulating the client-side application of the OpS by actions such as clicking a button and inputting a string into a textbox on a personal computer or other terminal. UMS can record and replay these actions without any OpS modification. Thus, UMS can automate most operations at low cost and in a short time.

This approach is similar to the macro function well known to software programmers. However, creating a macro function would require alterations to the OpS. Moreover, a macro function would not be able to access or control other OpSs. UMS automates operations that involve multiple OpSs without any of them being modified.

3.2 Automation by field operators

UMS is designed to support field operators since they are the intended users of OpSs. While field operators are specialists in operations, they are not necessarily computer specialists, and most operators

have difficulty in creating advanced programs. To overcome this problem, UMS provides a visual editor and automatic recording via operator example. As a result, it can be used by any operator, even computer novices.

4. Usage examples

Two usage examples are shown in Fig. 2 and described below.

4.1 Automation of repeated and stylized operations

UMS can automate repetitive and stylized operations. An example of a repeated and stylized operation is database correction. In this operation, operators input a lot of correct equipment data into an OpS at a time. The operation proceeds as a specific action flow consisting of, in this example, initiating a search, entering the search target, selecting a search result (target equipment), opening a window containing information about the target equipment, updating the target equipment's data items in the window, and closing the windows.

UMS lets the operator easily record the actions needed for one cycle, and it then executes the actions iteratively. Therefore, UMS can easily automate repeated and stylized operations.

4.2 Prevention of data mismatch

UMS is effective at preventing data mismatches arising from human error. When the operator must enter data items into several OpSs, he or she must accurately complete all the entries by hand. The operations involved are troublesome and errors such

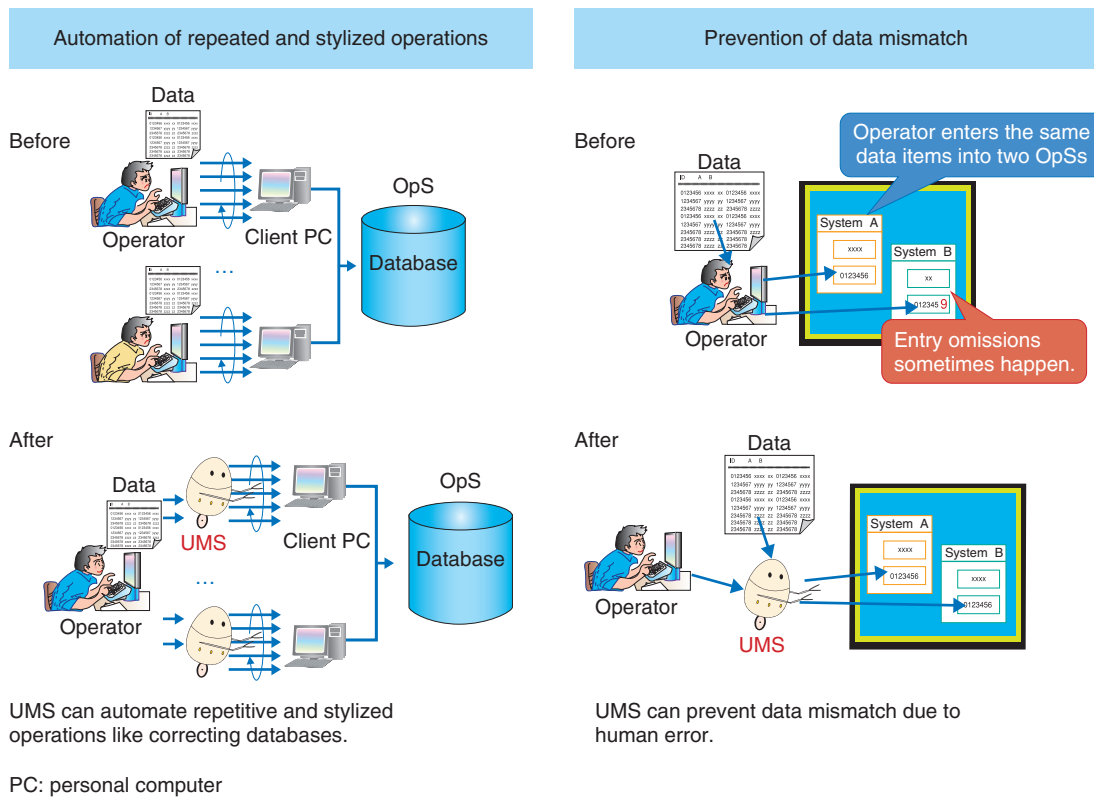


Fig. 2. Usage examples of UMS.

as entry omission and erroneous data entry may occur. As a result, the operations are likely to produce data mismatches.

With UMS, the operator can automate the operations needed for populating multiple OpSs with the appropriate data items, while, at the same time, automatically checking the correctness of data items against the logic of each OpS. Therefore, UMS can prevent data mismatches caused by human error.

5. Technologies

UMS has four key modules: a scenario auto-generation module, a visual editor module, a module for handling data at runtime, and a plug-ins module (Fig. 3). This section describes the technologies used in these modules.

5.1 Scenario auto-generation

A scenario is information that specifies actions and the interface manipulation flow for automation. The operator can automatically generate a scenario by performing desired actions manually. UMS detects

the operator's OpS manipulation events and infers his/her actions. It adds the actions to a scenario according to the manipulation flow.

5.2 Visual editor

UMS uses flowcharts for intuitive viewing and editing of scenarios (Fig. 4). An operator can understand a scenario as a visual presentation (flowchart) and can edit the scenario in an intuitive manner without coding.

5.3 Data handling at runtime

The same operation will use different data in each cycle, so UMS provides a data management function in addition to the scenario management function. When UMS reruns a scenario, it identifies the variables in the scenario and accesses the appropriate data list to discover the values of the variables to be entered, record by record (Fig. 5).

After recording the scenario, UMS creates an empty Excel or CSV (comma-separated values) file (the data list) whose column headers reflect the variables entered by the operator and those returned from

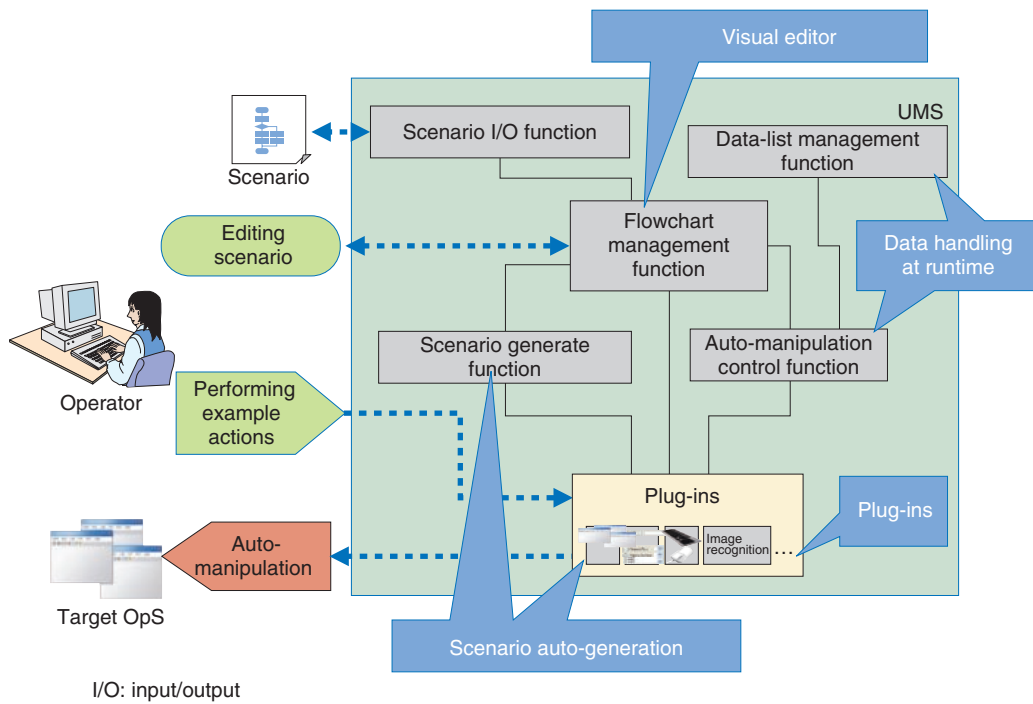


Fig. 3. UMS technologies.

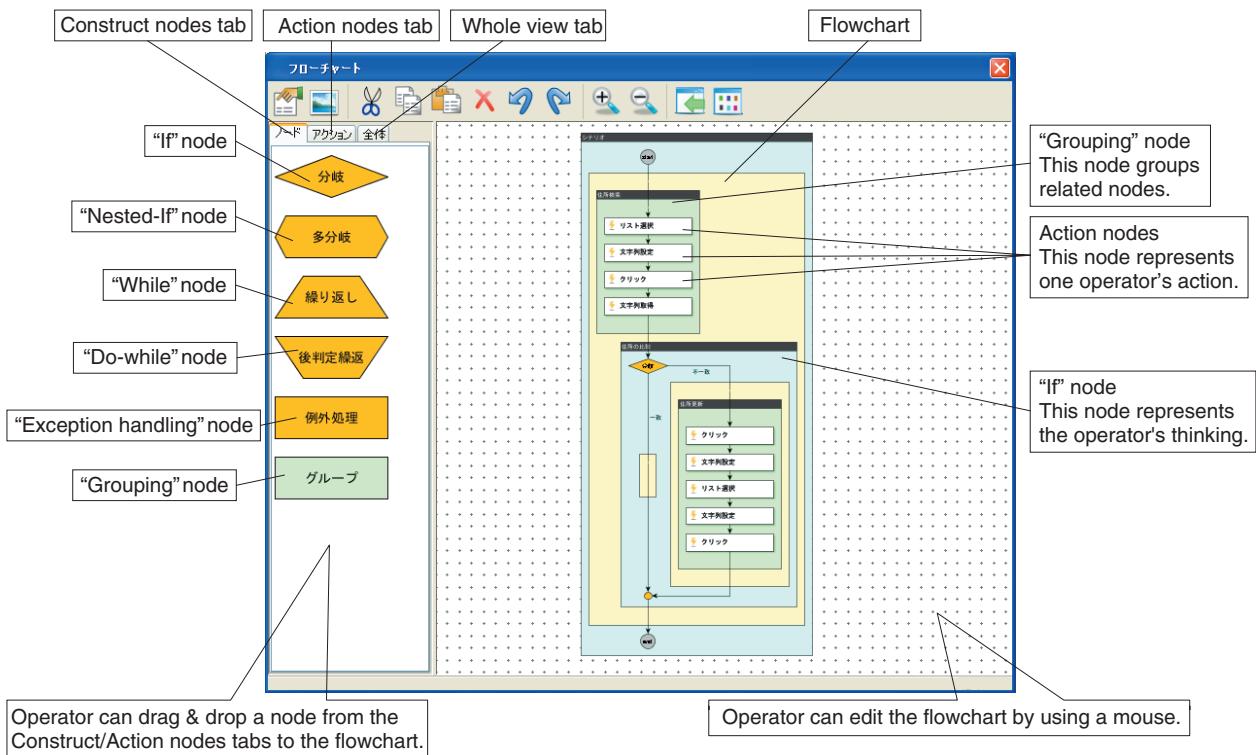


Fig. 4. Visual editor of UMS.

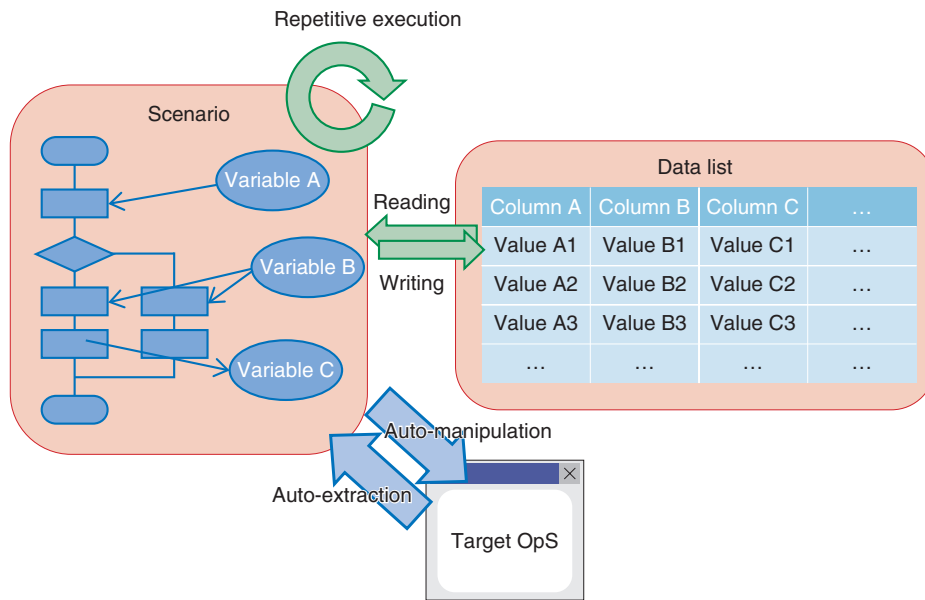


Fig. 5. Data handling at runtime.

Table. 1. Comparison with similar software.

	UMS	UWSC ^{*1}	Test automation software ^{*2}
Features	UMS is mainly for field operators and enables the operator to automate an operation intuitively.	UWSC has general versatility because it is based on replaying mouse and keyboard actions. It has good flexibility due to its high-level script functionality.	Software specialized for developing software. It has many test automation functions.
Required skill	Operator can make a scenario intuitively without coding by using scenario auto-generation, the visual editor, and other features.	Operator should create/edit text-based scripts. This requires considerable coding skill.	A visual editor is available, but it is aimed at software developers.
Data handling	Operator can input/output data to/from operations easily by using a CSV or Excel file (data format is fixed).	Operator should write scripts to input and output data.	Operator should write scripts to input and output data.

*1: <http://www.uwsc.info/>

*2: Micro Focus TestPartner, HP WinRunner/QuickTest Professional, IBM Rational Functional Tester Plus, etc.

the front OpS. The operator can also create data lists manually. Moreover, the operator can also use data lists that were extracted from an OpS. While executing a scenario, UMS can copy data output by an OpS and enter it into the appropriate data list record for later reuse.

5.4 Plug-ins

Considering future scalability, we designed UMS's architecture so that its functions can be easily extended. UMS contains program code that depends on the

operating system, browser, and so on because it must detect the operator actions and replay them. Accordingly, we separated this code from the main UMS program as plug-ins. In the future, when we adapt UMS to other environments, such as a new operating system or new browser, we will only have replace the plug-ins. Moreover, we can add new functions to UMS by means of new plug-ins.

5.5 System requirements

The current version of UMS runs on Microsoft

Windows and can automate actions related to typical graphical-user-interface-based applications and Microsoft Internet Explorer. In addition, low-level UMS functions include mouse and keyboard control. These functions can automate applications on other platforms (e.g., Java and Flash) and applications running in a server-based computing environment (e.g., Citrix XenApp).

6. Comparison with other similar software

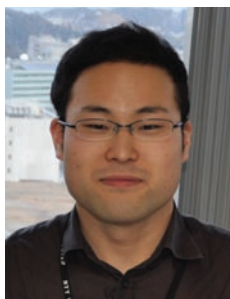
A lot of software has been created to automate users' (operators') manipulation actions. UMS is compared with alternatives in **Table 1**. UWSC and test automation software offer high functionality, but require the user to have advanced computer skills. On the other hand, UMS provides a visual editor and other functions that eliminate the need for advanced computer skills. Since its main purpose is to automate

operations, UMS makes it easier to input/output data than the alternatives. Some other software has more flexibility in terms of data input/output, but requires the operator to write scripts.

7. Conclusion

Our unified management support system, UMS, can automate and link many OpSs without modification; an operator can automate operations by himself or herself. UMS is especially effective at automating repetitive and stylized operations. As a result, it reduces the operators' workloads and operating costs.

We are now utilizing UMS in the field and gathering results. We intend to discover and eliminate its weaknesses. In the future, we will try to further reduce operating costs.



Fumihiko Yokose

Engineer, First Promotion Project, NTT Access Network Service Systems Laboratories.

He received the M.E. degree in electrical engineering from the University of Electro-Communications, Tokyo, in 2007. He joined NTT Access Network Service Systems Laboratories in 2007. He has been engaged in R&D of access network OpSs. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).



Akira Inoue

Senior Research Engineer, First Promotion Project, NTT Access Network Service Systems Laboratories.

He received the M.E. degree in mechanical engineering from Osaka University in 1994. He joined NTT in 1994 and is currently engaged in the development of access network OpSs. He is a member of IEICE.