

# UMS: Software for Automating Operators' Actions to Rapidly Improve Operational Efficiency at Low Cost

*Hiroyuki Adachi, Kentaro Hotta, Fumihiko Yokose, Takahiro Toyoda, and Akira Inoue*

## Abstract

Operational efficiency is a universal theme required in every business domain. In this article, we introduce our product called UMS (unified management support system), which enables anyone to easily automate terminal operations in office work and to thus improve operational efficiency.

*Keywords: UMS, automation, operational efficiency*

## 1. Introduction

Companies use various operation support systems (OSSs) to improve operational efficiency and ultimately reduce costs. The OSSs need to be modified in accordance with changes in operational flow caused by, for example, the launch of a new service. However, such modification typically involves very high costs and considerable time, so it is therefore difficult to modify OSSs in a timely manner. In some cases, companies abandon the idea of modifying OSSs at all. In such cases, operators have to add operations that the OSSs cannot handle, resulting in complicated, lengthy, or repetitive operations. This leads to heavy workloads and lengthy operation time. Moreover, it frequently leads to human errors.

To address this issue, we have developed the unified management support system (UMS) [1], which is software that automates users' terminal operations.

## 2. Features of UMS

UMS has three main features (Fig. 1), which are described in the following subsections.

### 2.1 Automation of operations

UMS is able to automate complicated user terminal operations on various applications (APs) that run on the environment indicated in Table 1. UMS provides various functions for automation, as shown in Fig. 2. A typical function is an operation to the graphical user interface (GUI)<sup>\*1</sup> components of an AP, such as clicking on a button or inputting a string into a text-box. In addition, UMS provides functions that adapt to various behaviors of an AP such as waiting for a window to appear, image recognition on the window, conditional branches, iteration, cooperation with external commands, and interactive processing. Furthermore, UMS can acquire a string from a textbox or a list box on an AP and store it as data.

Since UMS identifies target AP windows and GUI components, it can fully operate even with GUI components arranged outside of the screen if the AP is running on Internet Explorer or comprises Windows standard components.

<sup>\*1</sup> GUI: A type of user interface that allows users to interact with electronic devices such as a mouse through graphical icons and visual indicators.

## Feature 1: Automation of operations

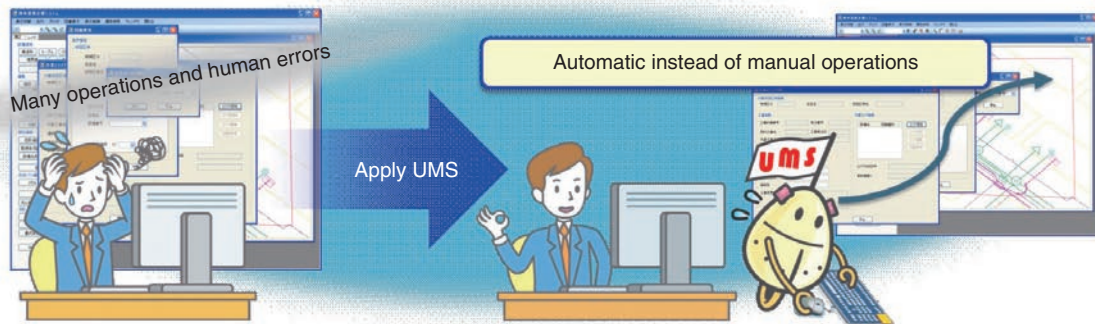
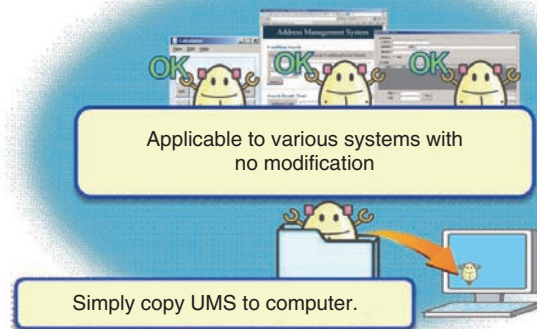
Feature 2: Automatic scenario generation;  
Visual editorFeature 3: No system modification  
or installation required

Fig. 1. Main features of UMS.

Table 1. System requirements.

Requirement	Class	Description
Hardware requirements	CPU	Intel Pentium 4, 2.5 GHz or greater
	HDD	Over 500 MB
	Memory	Over 1 GB (over 1.5 GB in Windows 7)
Software requirements	OS	Windows XP SP2/SP3 (32 bits) Windows 7 (32 bits)
	Others	Internet Explorer 6–9 if UMS is applied to web browser. .NET Framework Ver. 2.0 or greater if UMS is applied to application on .Net Framework.

CPU: central processing unit

HDD: hard disk drive

OS: operating system

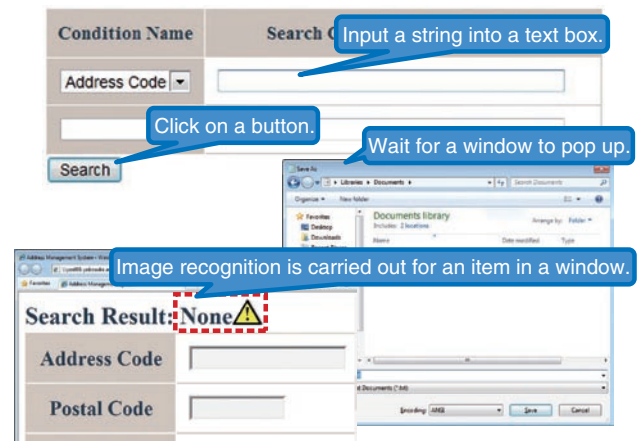


Fig. 2. Applicable operation examples.

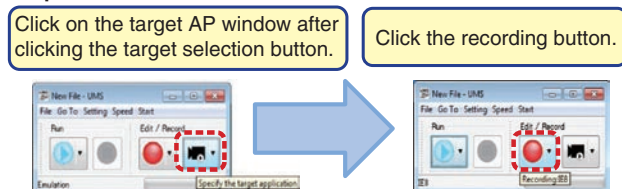
Table 2. Comparison with other similar software.

	UMS	UWSC <sup>*1</sup>	Test automation software <sup>*2</sup>
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 create a scenario intuitively without coding by using automatic scenario generation, the visual editor, and other features.	Operator must 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 easily input/output data to/from operations by using a CSV or Excel file.	Operator writes scripts to input and output data.	Operator writes scripts to input and output data.

CSV: comma-separated values

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

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

**Preparation**

This procedure is necessary when the target window changes.

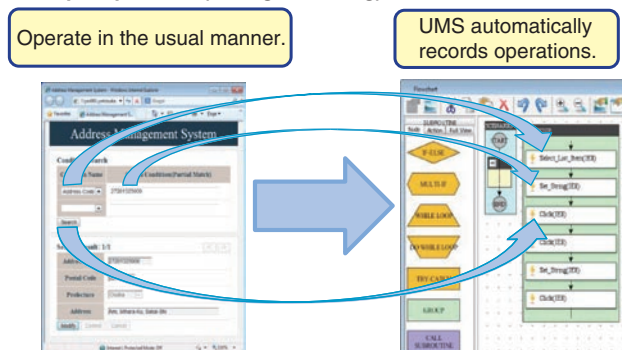
**Example operation (during recording)**

Fig. 3. Automatic scenario generation procedure via operator example.

**2.2 Automatic scenario generation and visual editor technology**

There are a number of software programs to automate user operations; a few are listed in **Table 2**. However, they are often difficult to use because considerable programming skill is required to create the *scenario*, or automation program. UMS solves this problem by providing automatic recording via operator examples and a visual editor on a GUI. Thus, any

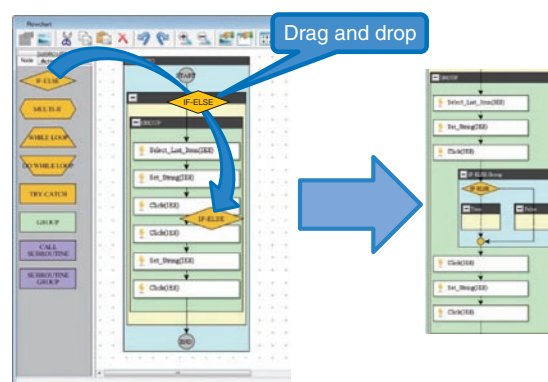


Fig. 4. Creating a branch flow.

user, even a personal computer (PC) novice, can use UMS. The procedure for creating a scenario via an operator example is shown in **Fig. 3**. If the user manipulates the target AP in the usual manner, UMS automatically records the operation on a flowchart as a node whose granularity is easy to understand, for example, *Click* or *Set String*. A user can freely edit the scenario merely by manipulating the mouse to copy and paste, delete, or rearrange nodes.

Furthermore, if an operation diverges because of various conditions, the user can describe the branch process merely by manipulating the mouse to drag and drop the IF-ELSE node from the left-hand node palette to the flowchart (**Fig. 4**).

**2.3 No modification of target OSS required**

The target OSS does not need to be modified because UMS operates the target AP in the same way as a user operating it manually. Thus, UMS can be

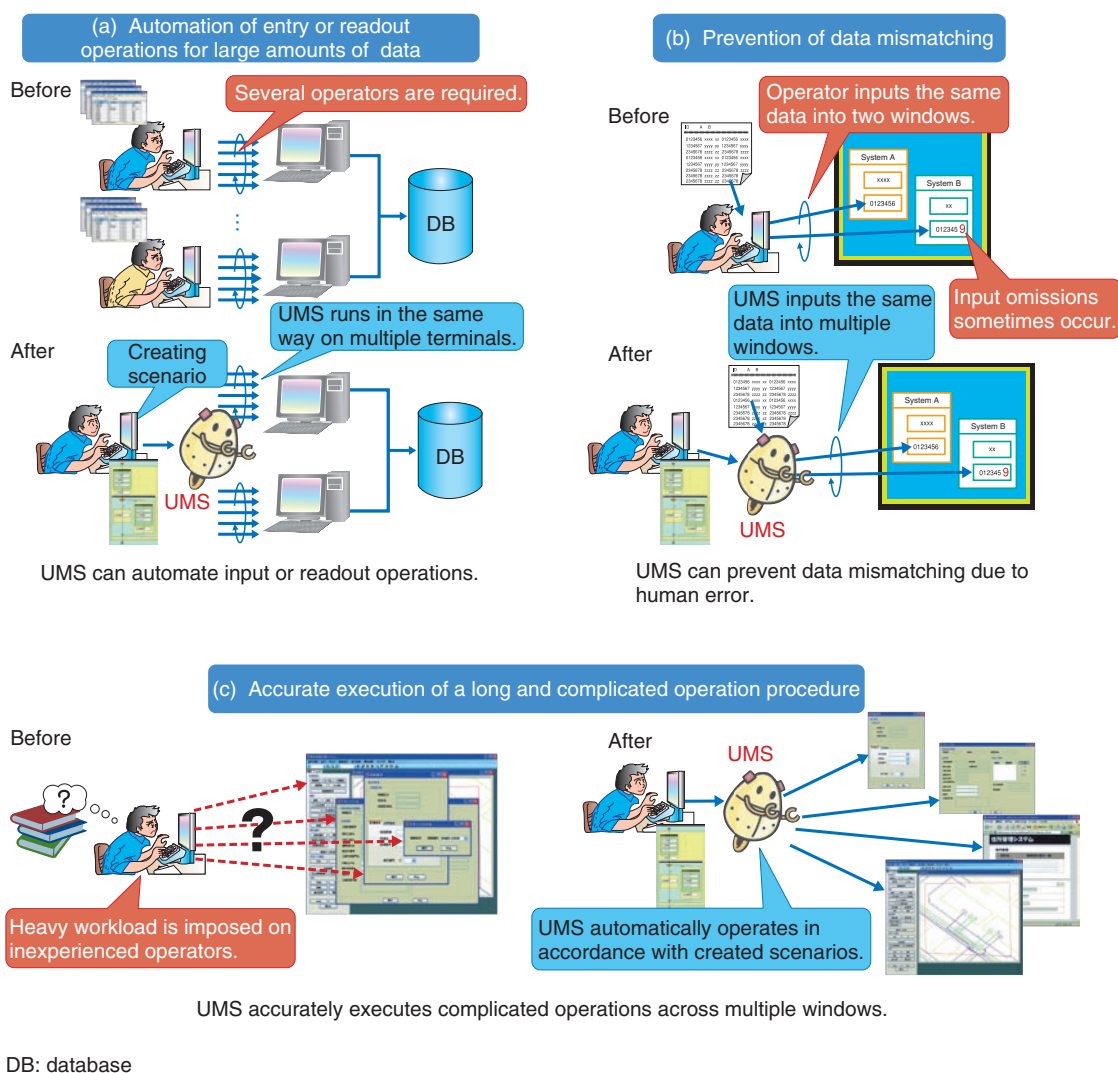


Fig. 5. UMS application examples.

applied to various systems at low cost and in a short time. Additionally, since UMS can be used merely by copying a few associated files to the PC, it can minimize the negative effects on the terminal environment.

### 3. Application examples

UMS works most powerfully in the case of repetitive entry operations involving data prepared for an AP. Typical application examples are shown in Fig. 5.

The first example involves entry or readout operations that are performed for a large amount of data and that require many operators. UMS can dramatically reduce the operators' workload by automatically

processing the data (Fig. 5(a)).

The second example involves entry operations for inputting the same data into several OSSs, which might occur when OSSs cannot share data with each other. In this case, the operations are likely to yield data mismatches due to entry errors. However, UMS can prevent data mismatches caused by human error by accurately performing entry operations to the OSSs (Fig. 5(b)).

The third example is of a time-consuming and complex operation procedure. As operation procedures become longer and more complicated, inexperienced users require more operation time and make more mistakes. However, UMS solves this problem by enabling even long and complicated operations to be



performed smoothly (Fig. 5(c)).

In these three cases, the more data that are prepared, the more effectively UMS works.

#### 4. New features of UMS

Currently, UMS is being applied to various operations in different fields within the NTT Group. A few functional problems have been revealed as the system's usage has become more widespread.

We have been working to expand UMS's application range, and in October 2013, we developed the latest version of the system (Ver. 3). This new version is equipped with new functions that can overcome these problems.

The major functions are described in the following subsections.

##### 4.1 Automatic prevention of malfunctions caused by user operations

###### (1) Background

If the user unintentionally inputs information to the target AP through mouse or keyboard operation while UMS is operating automatically, malfunctions such as input errors or inadvertent deletions may occur.

###### (2) New functions

- If unintentional mouse or keyboard operations are detected, this function suspends the scenario and displays a message to the user.
- Automatic confirmation of input strings' correctness is provided.

##### 4.2 Response to unexpected errors

###### (1) Background

Because UMS works in accordance with a created scenario, the user should take every possible behavior of the target AP into consideration when creating the scenario. However, since it is quite difficult to foresee every possible behavior and error, the scenario is suspended with an error message if UMS encounters an unexpected event on the AP during automatic operation.

The problem, therefore, is responding to unexpected events.

###### (2) New function

- The scenario continues to be executed even when an unexpected error occurs.

\*2 VBScript (Microsoft Visual Basic Scripting Edition): A scripting language that runs on Microsoft Windows. It can automate external applications such as Microsoft Office.

##### 4.3 Expansion of applicable scope

###### (1) Background

UMS provides fundamental functions required for automatically creating scenarios. Users can utilize combinations of these functions to create scenarios. However, automation is difficult with some operations merely by using these fundamental functions. Certain operations are needed in specified cases, including handling complicated files, processing data formats, processing data strings, and operating with highly functional APs such as Microsoft Office and its macro programs.

###### (2) New functions

- VBScript<sup>\*2</sup> can be used to freely define new UMS functions.
- A library is provided that makes it possible to reuse a group of nodes created on other scenarios.

#### 5. Summary and future plans

We have developed UMS, which enables anyone to easily automate terminal operations. UMS is a versatile system and is now being applied to various operations in NTT EAST, NTT WEST, NTT Communications, and NTT DOCOMO [2]–[5]. In December 2013, we transferred the UMS technology to NTT Advanced Technology Corporation, and they intend to further develop and commercialize it.

#### References

- [1] F. Yokose and A. Inoue, "UMS: Software to Automate Operators' Actions," NTT Technical Review, Vol. 9, No. 8, 2011.  
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201108ra4.html>
- [2] S. Numahata, H. Yamazaki, and K. Sasaki, "Implementing UMS to Improve Efficiency of Optical Fiber Path Testing," NTT Technical Review, Vol. 12, No. 2, 2014.  
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201402fa2.html>
- [3] K. Nakayama, "Expanded Use of UMS in the Tokai Area," NTT Technical Review, Vol. 12, No. 2, 2014.  
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201402fa3.html>
- [4] K. Uchiyama, K. Senba, and Y. Tsunashima, "Improving Efficiency of Company-wide Operational Work by Applying UMS," NTT Technical Review, Vol. 12, No. 2, 2014.  
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201402fa4.html>
- [5] T. Daido, K. Ichihashi, and T. Nonobe, "A UMS-based Default-setting Tool for Smartphones and Tablets," NTT Technical Review, Vol. 12, No. 2, 2014.  
<https://www.ntt-review.jp/archive/ntttechnical.php?contents=ntr201402fa5.html>

**Hiroyuki Adachi**

Fourth Promotion Project, NTT Access Network Service Systems Laboratories.

He received the M.S. degree in physics from Osaka City University in 2011. He joined NTT WEST in 2011 and worked in the Nagoya Branch from 2011 to 2013. He moved to NTT Access Network Service Systems Laboratories in 2013. He is currently engaged in R&D of operation support systems of access networks.

**Takahiro Toyoda**

Section Chief, Business Promotion, Kyoto Office, Kansai Branch, NTT FIELDTECHNO.

He received the M.E. degree in information engineering from Fukui University in 1995. He joined NTT in 1995 and worked in NTT Access Network Service Systems Laboratories from 2010 to 2013. He moved to NTT FIELDTECHNO in July 2013.

**Kentaro Hotta**

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

He received the M.E. degree in systems engineering from Hiroshima University in 1995. He joined NTT in 1995 and has been engaged in the development of access network OpSs. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE).

**Akira Inoue**

Senior Research Engineer, Supervisor, Fourth 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 developing operation support systems for access networks. He received the Academic Encouragement Award from IEICE in 2002. He is a senior member of IEICE.

**Fumihiro Yokose**

Engineer, Fourth 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 operation support systems of access networks. He is a member of IEICE.