

# USER APPLICATION UNIT PROCEDURES IN Exapilot

SATO Keiji \*1 MATSUNO Takuya \*1

---

*Rapid progress in Information Technology has fueled the shift toward open architecture systems and networking in the manufacturing industry. However, even though IT has enabled data integration between systems, for surviving the global competition, it is essential to process such data strategically through closer collaboration between systems and to reform role-sharing between humans and systems. Therefore, we have newly developed and added VBScript Unit Procedures and user application unit procedures to “Exapilot” operation efficiency improvement package, thereby enabling users to tailor the icons of procedures to their needs. This upgrade facilitates interlinking of Exapilot with other systems, and then turning Exapilot into the core software in collaborative operation packages. This paper introduces a predictive control system as an example of collaboration using Exa Series solution-based packages.*

---

## INTRODUCTION

Remarkable progress in information technology has led to open architecture systems and networking in the manufacturing industry. In such industry, remote monitoring systems based on public communication lines or the Internet, and or collaboration of control, information, and corporate systems based on standardized interfaces such as OLE for Process Control (OPC) are now widely used via networks. Despite such data integration between systems, however, closer collaboration and processing of data for strategic use, and changes in role-sharing between humans and systems are needed in order to survive the global competition.

This paper introduces an approach for closer collaboration of systems using the Exapilot operation efficiency improvement package.

## OPERATION SUPPORT COLLABORATION

Our customers use various systems in their actual manufacturing plants, including:

- Process control systems such as distributed control systems (DCSs) and programmable logic controllers (PLCs)
- Database systems containing process data
- Production management systems
- SCADA systems for supervising operation and production information
- Simulation systems for implementing safer and more stable production
- AI systems such as fuzzy and neural computing systems
- Office automation systems such as e-mail and groupware

These systems were developed to meet respective user needs, and have since become an essential part to keep day-to-day production activities. However, users need to increase operation and production efficiency, at lower cost and with shorter lead times in the age of surviving the global competition. One approach is for our system integrators to link and coordinate existing independent systems and configure the systems enabling the production based on speed up and more accurate decisions to be made. We call this type of efficient production with close

---

\*1 Industrial Automation Systems Business Division

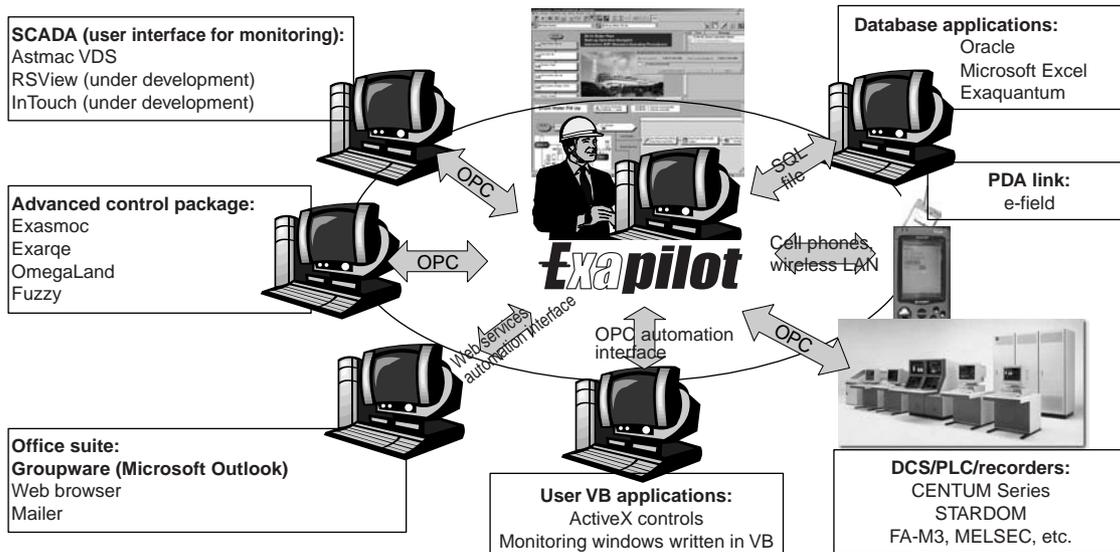


Figure 1 Concept of Operation Support Collaboration

collaboration among existing systems “operation support collaboration.”

Figure 1 shows the concept of this collaboration. Exapilot is a software package designed to automate the work of humans (operators). With functions for collaborating with other systems added, Exapilot becomes core software for operations.

## EXAPILOT OPERATION EFFICIENCY IMPROVEMENT PACKAGE

### Function Overview

Exapilot is a software package that enables plant operator to computerize their knowledge of plant operations by themselves. To do this, they only put the icons of unit procedures corresponding to particular actions, such as plant operations including data setting in a control system, plant anomaly monitoring, and manual operation request to the site, to draw a flowchart as shown in Figure 2.

For example, the following are routine tasks of the operator:

- (1) Every day, the production order for the current day is sent from headquarters via the production management system.
- (2) The manufacturing division, based on experience, determines settings in the control system to manufacture the ordered amounts to meet the ordered lead times.
- (3) The operator makes settings in the control system and manipulates the field devices.
- (4) The operator monitors manufacturing statuses.
- (5) The operator checks the manufacturing amounts from the control system.
- (6) The operator inputs the manufacturing amounts in the production management system.

Exapilot makes it easy to interface with the control system, but the VBScript unit procedures newly added in Exapilot R3.10

make it even easier: the entire six-step procedure shown above can be performed by Exapilot, and thereby leaving the operator free to focus on more important work instead.

### VBScript Unit Procedures

Visual Basic (VB) is a widely-used programming language, and VBScript (VBS) is a compact and easier-to-use version of Visual Basic. VBScript unit procedures enable users to add custom functions to Exapilot by themselves.

Examples of collaboration by VBScript are introduced below.

- (1) Collaboration with Windows operating system
  - Opening, copying, deleting, and finding files
  - Reading and writing text files
- (2) Collaboration with Microsoft Office components
  - Reading and writing data from/to Excel files
  - Interlinking with calculation functions in Excel or VBA

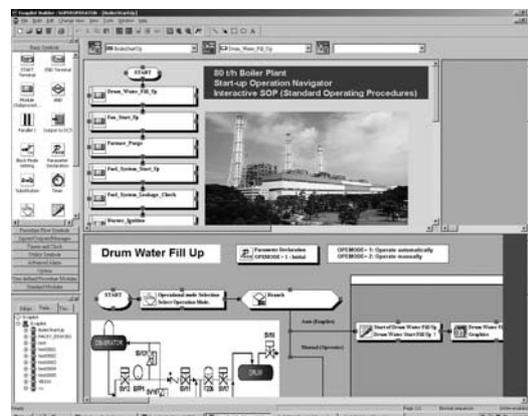


Figure 2 Defining an Operation Flow in Exapilot

```

Sub lag ()
  ARG03 = ARG04*exp (-1/ARG02)
          + (1 - exp(-1/ARG02))*ARG01
  ARG04 = ARG03
End Sub
'DATA_IN   : ARG01
'TIME      : ARG02
'DATA_OUT  : ARG03
'DATA_PRE  : ARG04

```

**Figure 3** VBScript of First Order Lag

- Reading and writing data from/to Access databases
  - Sending mail, and writing schedules and jobs to Outlook
- (3) Collaboration with MATLAB

Collaboration with MATLAB, which provides programming tools for data analysis, modeling, simulation, and so on, allows operation based on more sophisticated control theory. Using the MATLAB automation server capabilities as the interface with MATLAB, the following actions in MATLAB can be performed from Exapilot:

- Carrying out MATLAB commands
- Reading matrices
- Writing matrices

#### User Application Unit Procedures

Nevertheless, VBScript is still a programming language and is not always intuitive for those directly involved in manufacturing. Exapilot therefore allows users to easily add functions that they have developed as VBScripts, as custom unit procedures which are referred to as “user application unit procedures”.

An example of creating a user application unit procedure having a first order lag, which is often used in control calculations, is shown below.

```

[revision]
  formatrev=0
[file]
  filename=File.txt
[function1]
  name=lag
  dispname=first order lag
  arg1=DATA_IN
  io1=I
  type1=R8
  rem1=input value
  arg2=time
  io2=O
  type2=R8
  rem2=time constant
  arg3=DATA_OUT
  io3=O
  type3=R8
  rem3=output value
  arg4=DATA_PRE
  io4=IO
  type4=R8
  rem4=previous value

```

**Figure 4** Information File for Unit Procedure of First Order Lag



**Figure 5** Dialog Box for Defining Unit Procedure of First Order Lag

The transfer function of a first order lag can be expressed as:

$$G(s) = k/(1 + Ts)$$

Transforming this expression for digital control systems yields:

$$\text{DATA\_OUT} = \text{DATA\_PRE} * \exp(-1/\text{TIME}) + (1 - \exp(-1/\text{TIME})) * \text{DATA\_IN}$$

where

DATA\_IN = input value

TIME = time constant

DATA\_OUT = output value

DATA\_PRE = preceding output value

To create this unit procedure of a first order lag, the user only has to create a script file and unit procedure information file and to put them in a designated folder.

- (1) Script file

Code the desired calculation in VBScript. (Figure 3 shows the script for a first order lag.)

- (2) Unit procedure information file

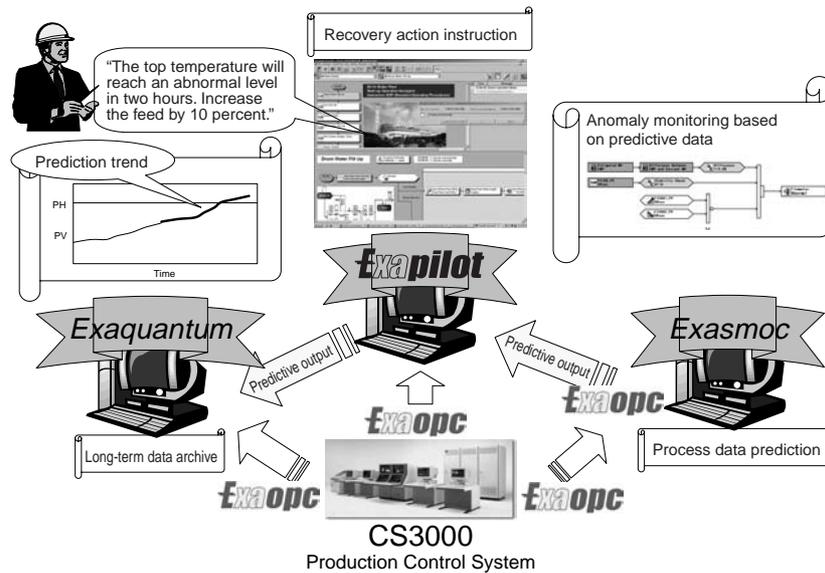
Make settings in the User Application dialog box (Figure 5) to set the type and name of the function, arguments and their set values, and comments. (Figure 4 shows the contents of the unit procedure information file for a first order lag.)

- (3) User Application dialog box

A dialog box will be automatically created based on files (1) and (2) above, as shown in Figure 5. MSScript.ocx is used as the VBScript execution engine and is designed to run in a different process from that for execution of Exapilot main tasks so as to minimize the impact of any bugs in user applications on Exapilot main tasks.

## PREDICTIVE CONTROL SYSTEM

As an example of operation support collaboration, a predictive control system comprised of Exa Series solution-based packages is described (see Figure 6). This system predicts future values of specific process data using a multivariable model,



**Figure 6** Predictive Control System

performs anomaly monitoring on the data thus predicted, and gives instructions to the operator to avoid future abnormalities. This ensures steady, safer plant operation and hence more efficient production. The Exa Series solution-based packages of this system have the following roles:

- (1) Exaopc OPC interface package acquires process data from the CS 3000 via OPC interfaces.
- (2) Exasmoc multivariable model-based predictive control package usually performs process control (modulation of setpoints and manipulated variables) based on the predicted data, but performs only prediction of changes in values of process data in this system.
- (3) Exapilot operation efficiency improvement package develops and executes process anomaly monitoring logic and gives instructions for proactive recovery actions. A major advantage is that process anomaly monitoring is based on the predicted future data.
- (4) Exaquantum plant information management system displays real-time and historical data graphically or in the form of trend graphs. A major advantage is that by collaborating with Exasmoc, predicted future data can be displayed.

## CONCLUSION

We have been developing software that assists collaboration for the benefit of users as well as software developers in terms of quicker development with fewer resources.

VBScript is not new, but as with Web services on the Internet, can be used to improve collaborative functionalities. We will continue to incorporate the latest technologies into Exapilot to extend such collaboration. ◆

## REFERENCES

- (1) Niina Nobuhiro, et al., "Unusual Operation Support Package Exapilot," Yokogawa Technical Report, Vol. 41, No. 4, 1997, pp. 17-22 (in Japanese)
- (2) Kobayashi Yasunori, "Exapilot, Assistant Package for Improvement of Operational Efficiency," Yokogawa Technical Report, Vol. 43, No. 3, 1999, pp. 17-20 (in Japanese)
- (3) Takahashi Kimikazu, "Advanced DCS Usage and Improvement in Production Efficiency with Operation Support," Keiso, Vol. 43, No. 11, 2000, pp. 42-47 (in Japanese)
- (4) Niina Nobuhiro, et al., "Operational Efficiency Improvement Package Exapilot Lite," Yokogawa Technical Report, Vol. 45, No. 1, 2001, pp. 59-62 (in Japanese)

\* Exapilot, Exaopc, Exaquantum, Exasmoc, CENTUM, and Astmac are registered trademarks of Yokogawa Electric Corporation. Other product and company names appearing in this paper are trademarks or registered trademarks of their respective holders.

