FIELDBUS SYSTEM ENGINEERING

TOMITA Shoji [*1]   *MORI Hiroshi* [*1]

Emerging fieldbus technology is revolutionizing most of the control technologies in process industries, with this engineering revolution capturing a great deal of user interest. Traditionally, control-loop architecture was designed and incorporated inside distributed control systems (DCS). Now however, this architecture can be incorporated into the whole system including the field devices, because the control functions are distributed throughout the field. Engineering a fieldbus system will not only involve designing control loops but also developing and maintaining fieldbus devices used at different places by a variety of persons with different backgrounds.

This paper discusses the fieldbus engineering system that evolved from the search for a new method of fieldbus engineering, describes the latest features and merits of the fieldbus engineering tools developed for this system, and gives a practical example of how to engineer a fieldbus system using the CENTUM CS 1000 or CS 3000 system.

INTRODUCTION

The Fieldbus Foundation, which was formed in 1994, set out a list of fieldbus specifications under the name FOUNDATION™ Fieldbus in 1998. Since these specifications were established, fieldbus-compatible products have continued to be successively released and subjected to field tests. The fieldbus system we developed was designed with the aim of achieving "Integration with CENTUM CS." This was one of the first fieldbus systems.

Emerging fieldbus technology is revolutionizing most of the control technologies in process industries. One area that will see change is the engineering of different instrumentation. Traditionally, control-loop design was employed inside DCSs. Now however, it's possible to distribute the control functions throughout the field. This raises the following issues and/or tasks:

- Designing of control schemes (i.e., function blocks) to be built inside fieldbus devices
- Designing of fieldbus systems with control loops that incorporate some function blocks in each fieldbus device, and communication links between different devices
- Startup, operation, and maintenance of the fieldbus system

The scope of fieldbus engineering is not only limited to control-loop design but also covers the development to maintenance of fieldbus devices, with different activities performed at different sites by users of different background. Thus to ensure that the engineering of a fieldbus system can be carried out across multiple sites, it is necessary to develop new engineering technologies and a system that enables the exchange of information among those sites. Further, engineering activities need to be performed independently while still allowing the parties to collaborate with each other.

This paper discusses the engineering of a fieldbus system first, then describes the core engineering tool used to achieve such engineering, and finishes with an practical example of fieldbus system engineering using the engineering tool in a CENTUM CS 1000 or CS 3000 system.

FIELDBUS ENGINEERING

How Is It Different from Traditional DCS Engineering?

Probably the most notable difference is in the cabling work because of the change from 4- to 20-mA analog-signal wiring to two-way multi-drop communication. However, even bigger differences in engineering have emerged from assigning the control functions to various field devices scattered throughout the

[*1] Industrial Automation Systems Business Division

plant, and allowing each of these devices to perform distributed control independently. Such differences relate to:

- Mutually dependent fieldbus device selections and control-loop design
- Design of communication and scheduling information
- Setting of engineering data in fieldbus devices

The selection of field devices and design of control loops, which until now have been performed completely independently, become interrelated. Control loops are designed to realize the required control schemes using the capabilities and resources of the available devices. Thus, the selected devices are those that can help achieve these control schemes and the required control performance. Device selection and control-loop configuration is therefore a process of trial and error. Other unconventional activities include setting up fieldbus communication, devising a communication schedule and block execution schedule, and downloading engineering data into fieldbus devices via communication.

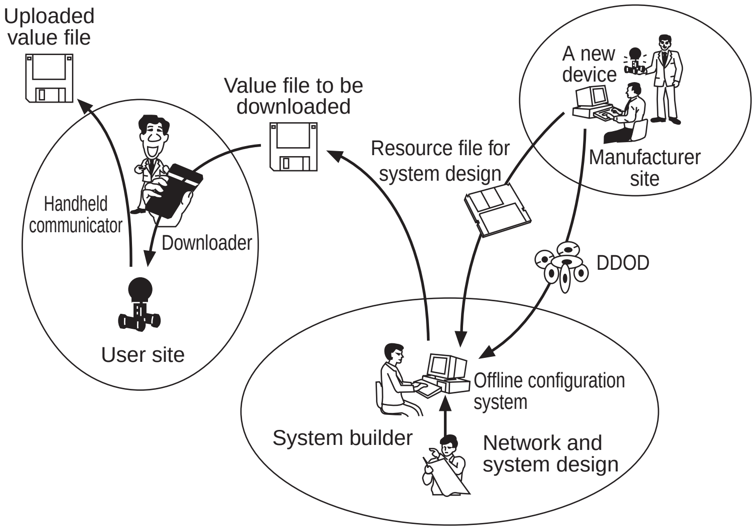

Uploaded value file
Value file to be downloaded
A new device
Resource file for system design
Manufacturer site
Handheld communicator
Downloader
DDOD
User site
Offline configuration system
System builder
Network and system design

Figure 1 Fieldbus Engineering Activities

Systematic View of Engineering

Now, let's view the fieldbus system engineering from a system engineering point-of-view. Figure 1 shows how the various activities of fieldbus system engineering can be spread over three sites.[1]

1. At the manufacturer site, the device vendor determines the function of the device and the parameters to be communicated over the fieldbus.
2. At the system builder site, the fieldbus system designer determines the required control function, and then selects the devices, control-loop architecture and parameter values to be used to realize this control function.
3. The user is the person who actually operates the fieldbus system. They perform engineering when the fieldbus is started up, adjusted for changing plant conditions or being maintained.

These activities are often done independently at each site; hence new technologies need to be developed to allow engineering at each site to be done concurrently. For instance, even if a device is not available to a system builder, a system designer should be able to define the structure of the fieldbus and control loops using the devices. Alternatively, even if the fieldbus design has not yet been completed, a user should be able to mount devices and check their operation using suitable parameter values.

Offline configuration (or offline engineering) refers to engineering work carried out by a system designer when there are no actual devices; it forms the essence of fieldbus system engineering. A system designer should be able to carry out system design freely, which means he/she must be able to flexibly and freely define device configurations and control loop architectures while offline. The ability to this is extremely important from a system engineering point-of-view.

Two kinds of information must be transferred to enable offline configuration.

1. Information about the resources of each device, which must come from the device supplier
2. Values specific to each device, which is initially obtained from the system designer and later from the user

An external interface needs to be defined to allow this kind of information to be exchanged between sites. This enables engineering to be done at individual sites concurrently. The most suitable technique to provide this external interface is to use a file for transmitting information. Data can be easily transferred to other sites or machines (including computers) using an external storage medium such as a floppy disk. The specifications of this external interface are defined by the Fieldbus Foundation in the article, FOUNDATION™ Fieldbus Common File Format.[2] It defines the file containing the first set of information (above) as a resource file, and the file containing the second set of information as a value file. Using these files as external interfaces is good way to establish mutual connections across multiple sites, thus allowing fieldbus system engineering to be structured systematically.

Is Fieldbus System Engineering Difficult?

The engineering of a system using fieldbus devices does increase in volume when compared with that of a system using conventional field devices due to the increased scope. Thus, the question that must be asked is "What should be done to make engineering easier?" The key to this issue is whether an engineering tool that requires simple data entry operations to accomplish engineering can be provided, and how simple those data entry operations are designed to be. Thus, we shall now discuss the functions required in an engineering tool.

The system designer must perform many challenging tasks, namely:

- Reference the resource files and DDOD files[*2]
- Define the devices to be used (entering the device tag names and node addresses)
- Assign the device tag names
- Assign the node addresses
- Configure control loops
- Determine and set the communication parameters
- Determine and set the communication schedule and block execution schedule
- Determine and set the parameter values of blocks
- Determine and set the parameter values of link objects[*3]
- Adjust the parameter values

Hence, the tool must include functions that:

1. Facilitate entry of engineering data (device configurations, control loop configurations, and parameter values) while offline, as it is essential to be able to start engineering prior to procurement of devices and installation work;
2. Import resource files and DDOD files; and
3. Download the determined parameter values into each device via fieldbus communication.

Another critical function is integration with a DCS.[(3)] Presently, a DCS is employed in most plants to control the processes. It cannot be expected that all control functions will be replaced by a fieldbus system, rather it is more likely that a fieldbus system will be used for some parts of plant control only or applied in steps, and will be used together with a DCS to achieve control. Integration of a fieldbus system and DCS requires a function for supplying engineering data of the fieldbus system to the DCS and a system that allows the control functions of the DCS and those of each fieldbus device to be combined to configure control loops.

FIELD BUS ENGINEERING TOOL

The engineering tool[*4] developed provides the following functions (shown in Figure 2):

1. Builder: Generates data compliant to Foundation™ Fieldbus and which is to be set in the fieldbus devices.
2. Network startup function: Downloads data to fieldbus devices.
3. Maintenance function: Uploads parameter values of blocks (values of tuning parameters, etc.) from fieldbus devices.
4. As the master device in the link, it can generate and download set values for host interface devices.
5. It provides an external interface that imports resource files and imports and exports value files in a format that is compliant with the Foundation™ Fieldbus Common File Format.
6. Host-file set generator: Generates engineering data about the fieldbus system for host applications.

Items 1, 2, and 5 satisfy the basic tool requirements mentioned above. Item 6 realizes the function aforementioned for integration with a DCS that provides the engineering data of a fieldbus system to the DCS for the purpose of integration with a DCS.

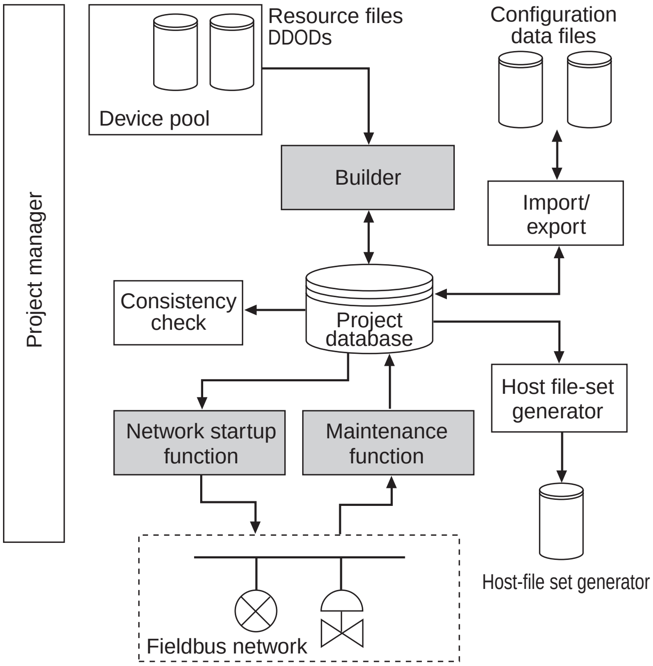


Figure 2 Functional Configuration of Engineering Tool

The CENTUM CS, CENTUM CS 1000, and CENTUM CS 3000 system is prepared with an I/O module (ACF11) referred to as the host interface device for connecting a fieldbus system to a DCS. The ACF11 acts as a fieldbus device having the link master function (see Figure 4). In line with the new concept of "software wiring terminals" for each fieldbus device's function blocks, the tool allows the control functions of the DCS and fieldbus devices to be connected in order to configure a control loop. Using the FBAP configurator[*5] (the middle window in Figure 3), the user can view a function block in each fieldbus device as if it is a function block inside the DCS and connect it to other function blocks. Software wiring terminals are interfaces outside each fieldbus device that are used to connect to host applications (such as a DCS) and blocks in other fieldbus devices in other fieldbus segments. For each fieldbus device, software wiring terminals

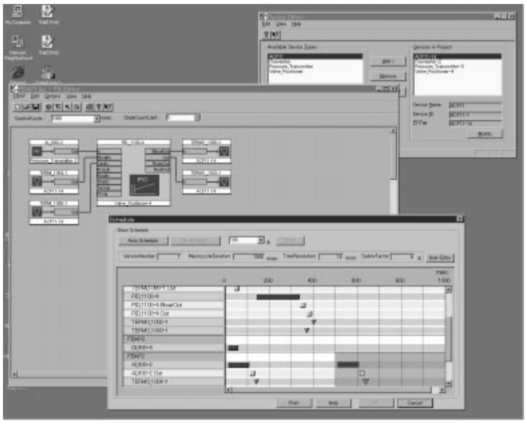


Figure 3 Builder Windows

simply refer to the terminals; they do not have anything to do with its control. However, if the software terminal information is given to the connected DCS as one of the host files (refer to item 6 in the functions listed above), the user can connect the data obtained from the terminals of each fieldbus device's control functions in order to control functions inside the DCS and configure cascade or advanced control.

The advantages of this fieldbus-engineering tool include:

(1) Automatic generation of parameter values for communication and scheduling: Parameter values for communication and scheduling are automatically generated based on the FBAP configuration, thus allowing the user to set the optimum values without needing to know the complicated Fieldbus communication specifications.

(2) Feasibility check by simulation using the scheduler: A scheduler generates a schedule to perform communication and run function blocks. The generated schedule is graphically displayed as a Gantt chart as shown in the front window in Figure 3. In this window, the user can visually check whether control of all loops can be accomplished within each control period and modify the schedule parameter values, FBAP configuration, or the schedule itself. Thus, users can run the schedule repeatedly while making changes in order to try and find the most satisfactory schedule settings.

(3) Ease of replacing device: When replacing a device with exactly the same model, upload the data from the device (refer to item 3 in the functions listed above) to equalize the saved data. Then download the data to a new device in which the tag and address have already been set up to complete the replacement. When replacing a device with the same model but of a different version or a similar device from a different device vendor however, engineering usually needs to be done again based on the resource file for that device since the capability and available resources differ. In this case, first export the value files for the entire project, and then replace the resource file for the replaced device. Next, import all the value files of the project that was exported, and then reconfigure the project. Lastly, download the data to the replaced device. This completes the replacement. The system does not require major modification to continue operation with a new device.

AN EXAMPLE OF FIELDBUS SYSTEM ENGINEERING

The following introduces an example of fieldbus system engineering from a CENTUM CS 1000 or CS 3000 system. Figure 4 shows the system configuration of a CENTUM CS 1000 or CS 3000 connected to a fieldbus system. The CENTUM CS 1000/3000 engineering functions and the fieldbus engineering tool run on a personal computer connected to the VLnet (for the CENTUM CS 1000) or Vnet (for the CENTUM CS 3000). The H1 fieldbus is connected to an ACF11 installed in an I/O nest. The DCS and fieldbus system can be engineered concurrently as follows:

(1) Engineering of fieldbus system
Engineering of the fieldbus system is performed in three steps: registration of devices, control loop configuration, and configuration data generation. The resource file for devices intended for use should be imported beforehand. In the first step, using the device editor (the rear window in Figure 3), register the devices to be used and set the tag names of the devices. Then, using the FB editor, configure control loops. Lastly, generate the configuration data.

(2) Engineering of DCS
Register the AFC11 via the System View (the rear window in Figure 5) in the same way as you do for other I/O modules. In the property sheet of the AFC11, specify the directory in which the fieldbus information resides. In the window of the IOM definition builder (I/O module definition builder, the front window in Figure 5) that opens when the AFC11's icon is double-clicked, define blocks tag names, parameter names, directions, and data types, then load the I/O module configuration data. During this loading process, the builder program reads the fieldbus information and sets the necessary information in the ACF11 file in the project database.

(3) Download to Fieldbus
Using the fieldbus system engineering tool, download the configuration data to the fieldbus devices. This sets up the data in the fieldbus devices.

(4) Operation and monitoring via HIS
The status of each fieldbus device can be monitored via an instrument faceplate displayed on an HIS screen, as shown in Figure 6.

As above, implementing the three-step actions using the fieldbus system engineering tool and defining the ACF11 configuration will complete the engineering of a fieldbus system.

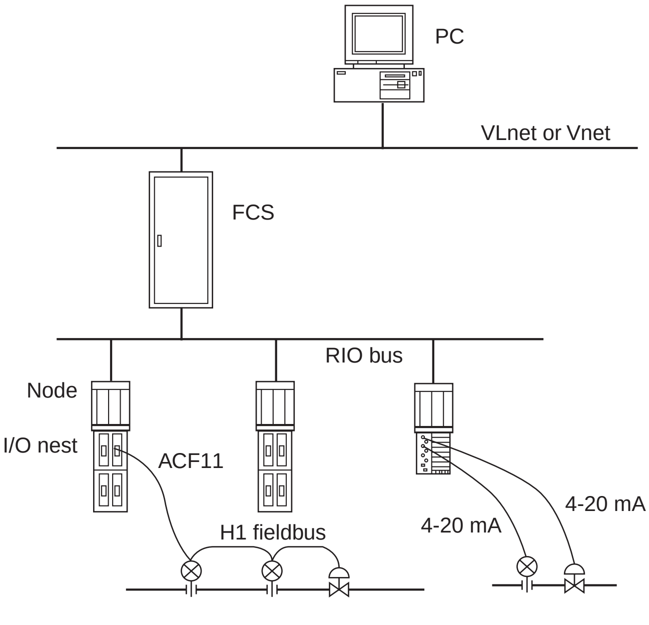
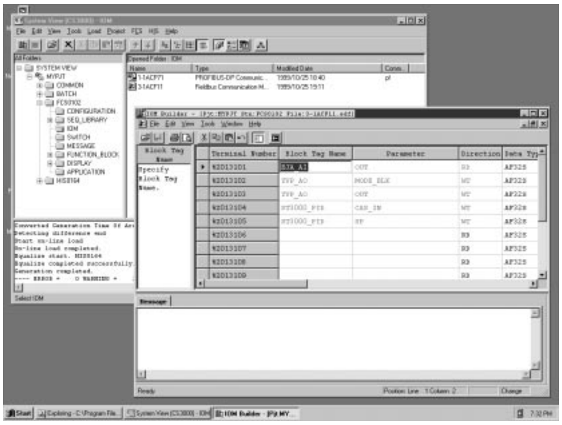


Figure 4 CENTUM CS 1000/3000 System Configuration



Name	Type	Modified Date	Conn...
1-1ACF71	PROFIBUS-DP Communic...	1999/10/25 18:40	pf
3-1ACF11	Fieldbus Communication M...	1999/10/25 19:11	

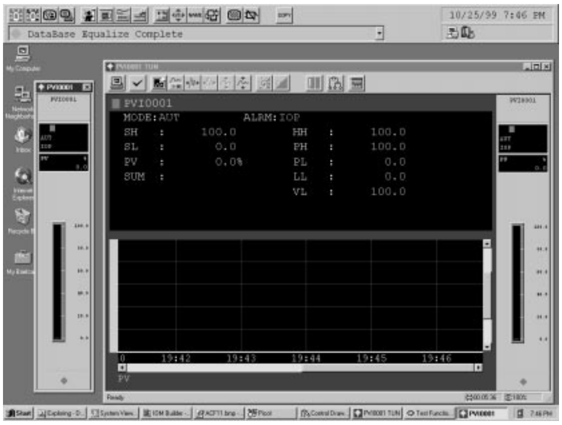
```
Converted Generation Time Of Ac
Detecting difference end
Start on-line load
On-line load completed.
Equalize start. HIS0164
Equalize completed successfully.
Generation completed.
---- ERROR =    0 WARNING =
```

Select IOM
IOM Builder - [Pjt:MYPJT Stn:FCS0102 File:3-1ACF11.edf]
File Edit View Tools Window Help
Block Tag Name
Specify Block Tag Name.

	Terminal Number	Block Tag Name	Parameter	Direction	Data Typ
▸	%Z013101	DIK_AI	OUT	RD	AF32S
	%Z013102	TYP_AO	MODE_BLK	WT	AF32S
	%Z013103	TYP_AO	OUT	WT	AF32R
	%Z013104	PT1000_PIB	CAS_IN	WT	AF32R
	%Z013105	PT1000_PIB	SP	WT	AF32R
	%Z013106			RD	AF32S
	%Z013107			RD	AF32S
	%Z013108			RD	AF32S
	%Z013109			RD	AF32S

Message
Ready
Position: Line 1 Column 2
Change
Start
Exploring - C:\Program File...
System View (CS3000) - IOM
IOM Builder - [Pjt:MY...
7:32 PM

Figure 5 Defining an ACF11 Configuration



```
DataBase Equalize Complete
```

```
PV10001
 MODE:AUT           ALRM:IOP
 SH   :      100.0        HH   :      100.0
 SL   :        0.0        PH   :      100.0
 PV   :        0.0%       PL   :        0.0
 SUM  :                   LL   :        0.0
                          VL   :      100.0
```

Figure 6 Monitoring and Operation via HIS

TO MAKE ENGINEERING MORE FLEXIBLE AND FREE

Flexibility and freedom in engineering is one of the main objectives of offline configuration. To further facilitate engineering, a function that enables a system designer to configure control loops before determining the devices to be used is needed. For this, we are currently studying the development of the universal block. A block model is a new concept that can be assigned to the corresponding device after determining the devices to be used. The universal block enables the devices to be selected and control loops to be configured concurrently, and makes it unnecessary to consider the sequence of engineering procedures. This is an evolutionary concept in plant engineering.

Other things we are presently considering for development include: an algorithm that automatically finds the optimum parameter values to achieve the full performance of each device; a function for displaying user guidance; and a incremental database download function that downloads only the modified data or only the differences from the last data save.

CONCLUSION

Fieldbus system engineering was discussed from a system engineering point-of-view, and we introduced an engineering tool that was developed for system engineering. Use of files in a common file format as an external interface was useful not merely when systematizing the engineering of a fieldbus system but also when developing the tool. By abiding by the common file format will allow us to use a variety of tools from various device manufacturers at the same time. The interoperability of not only fieldbus devices but also tools becomes apparent.

Fieldbus technology has not yet reached maturity. We believe that the only way to spread and development of the fieldbus is to achieve a steady, constant effort by acquiring more field results to pursue true interoperability. ◆

REFERENCES

(1) Tomita S. et. al., “Standardized file exchange eases fieldbus interoperability design,” InTech, vol. 44, no. 3, pp. 46-49, 1997

(2) FF-103 FOUNDATION^{TM} Fieldbus Common File Format

(3) Mori H. et. al., “Fieldbus System Integrated into DCS,” ISA/96 Advances in Instrumentation and Control, vol. 51, 1996

*2 Acronym of Device Description Object Dictionary; a binary file that is supplied from the device vendor and contains description information about the use of blocks in a device.

*3 A link object is one of the objects comprising function block application processes (FBAPs); it contains information about connections between blocks.

*4 This engineering tool can cover engineering for only one segment of H1 fieldbus and cannot be used to engineer devices on a segment connected via a bridge.

*5 A function that allows the user to connect function blocks in fieldbus devices to each other to configure control loops; the word FBAP is an acronym of Function Block Application Process, the generic name of a control function built up with function blocks.