NIPPON STEEL TECHNICAL REPORT No. 89 January 2004

UDC 681.3:699.1

Application of Open System to Process Automation in Iron-and-steel making

Nobuo SUMIDA*[1] Toshiaki UENO*[1]

Abstract

Nippon Steel Corporation led the steel industry in adopting general-purpose personal computers and operating systems as process computer systems for automatic optimal control and developing the middleware NS SEMI SYSTEM in the activity of open system solution. The enhancement of middleware function has made the extension of adopting process field. Nowadays, the renewals of process computers projects mostly have adopted the middleware and great cost reduction has been achieved. The open system application technology including PLC and DCS fields is accumulating and the middleware and support tools originally developed as well are shared with all works. The middleware has been adopted not only in the steel industry including overseas but also in other industries. To keep up with various needs quickly, Nippon Steel Corporation is promoting open system solution activities and development on the latest open system technology.

1. Introduction

In computer systems for steelworks, a large-scale hierarchical system is configured for an extremely large process and facility, to enable production of a diverse range of product types at high quality (see **Fig. 1**). On the other hand, there is a strong need to build diverse systems which reduce cost and exploit IT technology. So instead of using conventional dedicated process computers, which are highly functional but expensive, more and more facilities are using general-purpose computers with open specifications. This report discusses actual cases of using open system solutions, primarily in the fields of process computers and electrical/instrumentation equipment.

2. The Shift to Open Process Computers (C)

2.1 Background of the shift to open systems

Steel process control has the following features:

(1) The stringency of control precision requirements (on the μm order), even though extremely heavy objects are controlled
(2) The reliability to withstand 24-hour continuous operation in a plant where operations never stop
(3) Massive information processing due to the large scale of the facility
(4) The stringency of the requirement for high response, seen in rolling equipment

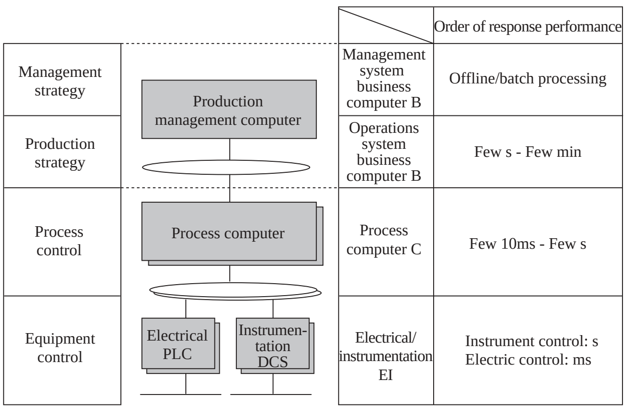

Fig. 1 Division of function between computers at steelworks of Nippon Steel

*[1] Environment and Process Technology Center, Technical Development Bureau

(5) The difficulty of stable operations and "building-in," due to the complexity of the manufacturing process.

Construction of process computer systems satisfying these stringent requirements has been ongoing since the 1960s, and at present, approximately 700 systems are operating throughout Nippon Steel. Within this process, we will focus here on the history leading up to total self-production of process computer software, and adoption of open systems.

(1) Purchased both hardware and software from computer manufacturers (1960s)

(2) Began self-production of software, starting with control models to serve as the core (1970s)

The control models and operation screens of process computers contribute to quality competitiveness in steel products, and are a form of operations know-how for product manufacturing. Therefore, this type of technical capital was regarded as crucial for ensuring corporate competitiveness.

(3) Achieved 100% self-production of software (1980s)

The primary needs in this period were: to reduce cost and improve speed of software maintenance in routine operations and equipment improvement, and to reduce system introduction costs when installing new lines for new products. In the process of improving the self-production rate, two issues arose due to human factors in software design and production: (1) Improvement of software quality, and (2) Improvement of engineering productivity. Therefore, by about the middle of the 1980s, efforts were being made to standardize engineering and software structures, and a system for 100% self-production of software was built.

(4) Adoption of open systems (since 1996)

Based on software design and development technology from the shift to self-production, Nippon Steel defined and developed middleware for building open process control systems (the NS SEMI SYSTEM*1)), and today, open process control systems based on this middleware are used throughout our facilities (except in some processes like hot rolling).

2.2 Middleware for open system control: "NS SEMI SYSTEM*1)"

Middleware for open control is general-purpose basic software which allows dedicated process computers for control to be replaced with commercially available work stations and personal computers (PC). Nippon Steel developed the self-produced "NS SEMI SYSTEM*1)" (see **Figs. 2** and **3**) as a form of such middleware, and this middleware is being widely used in open process control systems. This section describes the functions and features of the system, its track record, cases of actual use, and future development.

2.2.1 Functions and features

The functions and features or this middleware are as indicated below (**Fig. 4**).

(1) Improvement of reliability and processing performance to enable use of open operating systems (OS) for core system process control

The functions necessary for self-development of standard application software (AP software) for steel process control have been available since the era of software self-production using dedicated process computers (prior to the adoption of open systems), and the system is equipped with these functions. The system also has general-purpose function sets suitable for development of highly functional AP software with high reliability and processing performance, and a variety of support capabilities to enable efficient execution of software, OS and middleware setting, and problem isolation for AP software. This wealth of capabilities makes it possible to build open systems for highly reliable core system control using general-purpose PCs and workstations, for everything from small-scale to large-scale systems.

(2) Compatible with almost all of the main open OS

The system is compatible with almost all the main open OS, including WindowsNT®*2), Windows®2000*2), Linux®*3) and various types of UNIX®*4), and this enables selection of the optimal open server for each system. The middleware is built by analyzing the instructions of each OS (including past versions) from the standpoint of compatibility and function/performance, and limiting the system only to universal OS instructions (Win32API, UNIX® system calls). This ensures high reliability, minimizes the effects of specification differences due to OS upgrades, and makes it easier to keep up with the latest OS.

(3) Ensures portability of AP software

High portability is achieved by formulating and standardizing a universal function interface with AP software suited to the distinctive features of the C/C++ language (like function types and structures). Templates are provided for AP software developers, so even ordinary programmers with no specialized knowledge of the instructions of each OS can efficiently develop highly reliable software. This also enables quality optimization, like that of a uniform source program with good maintainability.

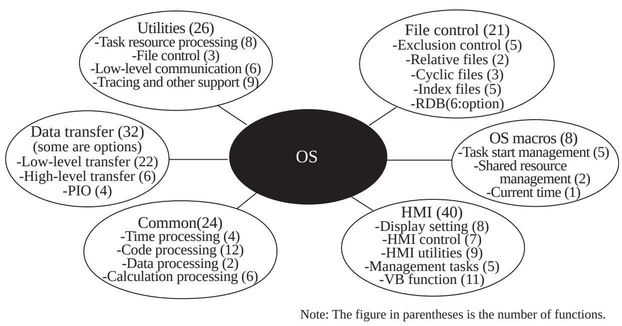

Fig. 2 NS SEMI SYSTEM*1) functions

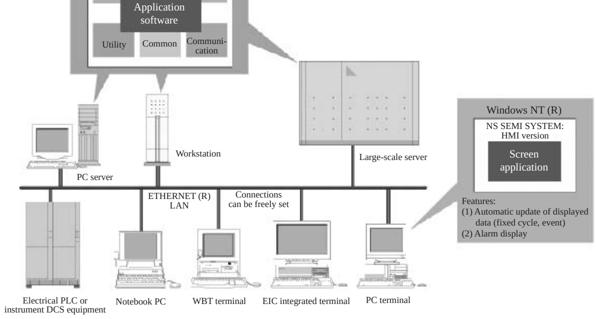

Fig. 3 System configuration using NS SEMI SYSTEM*1)

*1) "NS SEMI SYSTEM" is a registered trademark in Japan of Nippon Steel Corporation.

*2) Windows, Microsoft Office, Excel, Word and Visio are registered trademarks or trademarks of the Microsoft Corporation (U.S.) in the U.S. and other countries.

*3) Linux is a registered trademark of Linus Torvalds.

*4) UNIX is a registered trademark in the U.S. and other countries, licensed exclusively by the X/Open Company Limited.

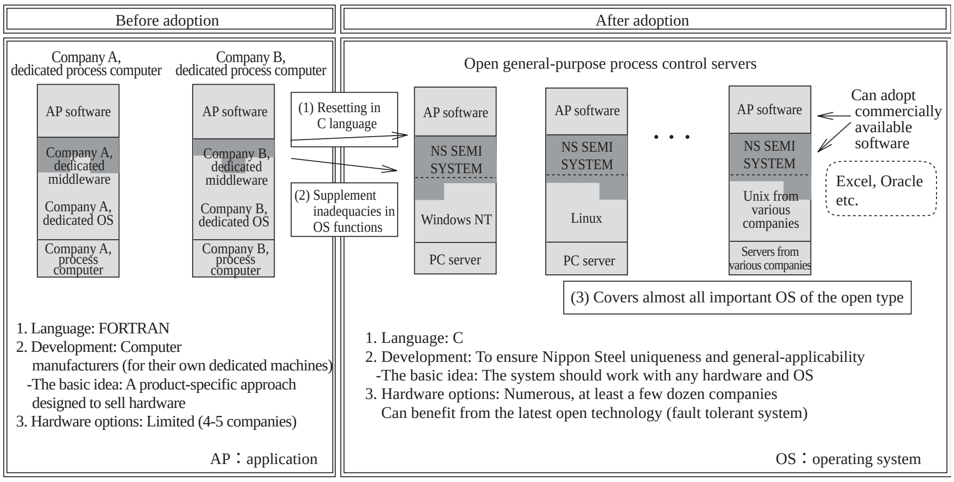

Fig. 4 NS SEMI SYSTEM[*1] features

(4) Achieves a human interface (HMI) with outstanding real-time performance

The system has a function for automatically updating display data at a fixed cycle, or in response to events, and a wealth of graphs and other display setting components needed for process control. It also has functions for central monitoring of alarms which occur over multiple servers, and can be used with web browsers.

(5) Equipped with a wealth of interface menus for linking with other systems

In the area of communication middleware and process I/O middleware, the system has an extensive track record of linking with a diverse variety of electrical sequencers/PLCs, instrument DCS and other computers, using methods like TCP/IP, BSC and no-protocol, so it is easy to connect with various systems and devices. For TCP/IP communication, conventions have been formulated to enable various types of error detection and recovery for automatic control, and TCP/IP use basically always follows these conventions. When necessary, it is also possible to simplify interface testing by providing the communication function part of the middleware on the other system side. The system also has middleware conforming to various open communication conventions like OPC (OLE for Process Control).

2.2.2 Track record and actual cases of use

Full-scale development of the "NS SEMI SYSTEM[*1]" began in 1995, and continuous durability testing was conducted over a long period. In 1997, a WindowsNT® PC server was used for the first time for steel process core control, as a billet continuous casting process computer at the Kimitsu Works[1]. This system achieved a significant cost reduction of 80% for software and 20-30% for software (compared with figures for the previous year at Nippon Steel), and has continued to operate smoothly up to the present time. When process computers were upgraded at the time of the No. 3 blast furnace renewal at the Kimitsu Works (in 2001), servers equipped with Linux® were used for the first time, in a similar fashion, as large-scale process computers[2].

In fields other than steel process control (both inside and outside of Nippon Steel), the system is being used as middleware for building MES (Manufacturing Execution Systems) for integrated control, management and linkage with high-level information systems at production sites in the chemical field, and other manufacturing industries (including some applications overseas). The system has now been used in a cumulative total of 140 systems (approximately 300 or more servers) (**Fig. 5**).

The next section takes a look at some actual cases employing the new open communication standard OPC in recent, large-scale open process control systems.

Previously, manufacturer-specific packages with built-in OPC interfaces have been sporadically used mainly for small-scale transmission/reception of data between electrical equipment, instrumentation and data collection PCs, but there are differences between manufacturers in the scripting languages for AP software development, and there was inadequate support for diverse needs. So, we adopted the OPC interface throughout for data transmission/reception between electrical devices, instrumentation and process computers which comprise the large-scale process control system. As

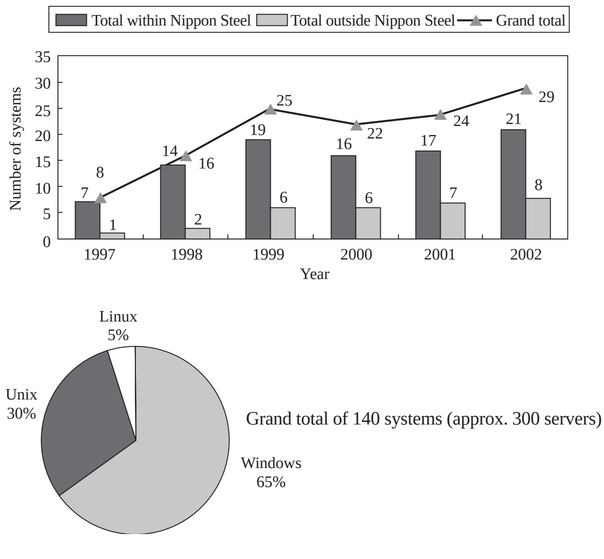

Fig. 5 Track record of NS SEMI SYSTEM[*1] use

shown in **Fig. 6**, a general-purpose client middleware OPC program was developed to facilitate AP software development in the C language, and this made it possible to build flexible systems not dependent on packages made by manufacturers. OPC communication capabilities with the Linux® system were developed, and this enabled use in multi-vendor configurations with OS other than Windows®.

2.2.3 Future outlook

The executable modules of AP software and the NS SEMI SYSTEM*1 in an open process computer system (which operated under WindowsNT®3.51 from 6 years ago) were copied, as is, to a Windows®2000 server (**Fig. 7**), and operation was validated by long-term running. The results confirmed that operation was normal. This is because the middleware was constructed by limiting the NS SEMI SYSTEM*1) to universal OS instructions (Win32API), and Win32API compatibility has been ensured within the scope of those OS instructions. The systems is configured to use standard conventions of the TCP/IP-based NS SEMI SYSTEM*1) in the interface with the outside, and to employ a TCP/IP converter with TCP/IP for BSC transfer. Therefore, in future open system server upgrades, it will be possible to achieve upgrading to the latest server model in an extremely short time and at low cost. Even if OS compatibility problems arise, they can be handled efficiently with minor, partial corrections within the NS SEMI SYSTEM*1), and we have obtained conclusive evidence that changes in AP software are not necessary.

The system is compatible with almost all main OS of the open

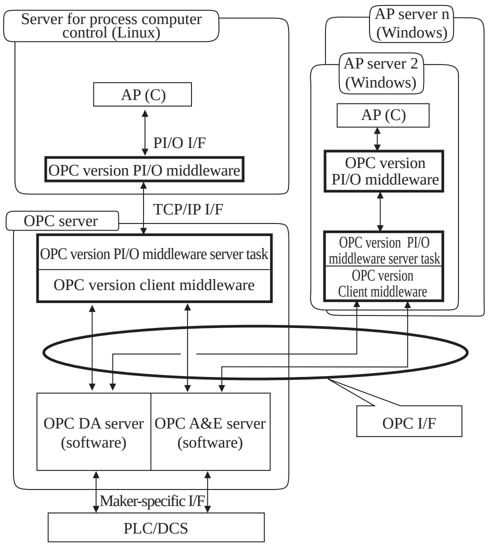

Fig. 6 NS SEMI SYSTEM*1) case history -OPC-

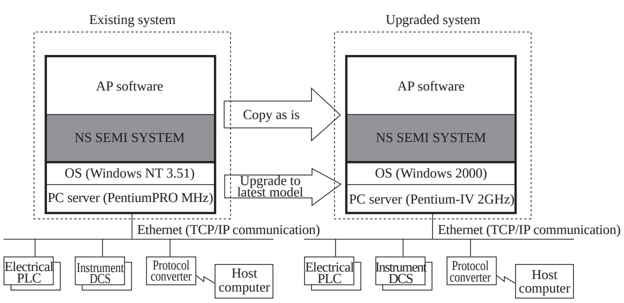

Fig. 7 NS SEMI SYSTEM*1) simple upgrade validation

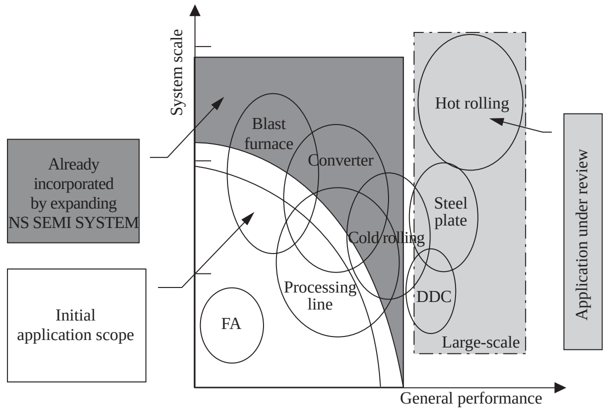

Fig. 8 Expansion of processes using NS SEMI SYSTEM*1)

type, so our policy for the future is to use the NS SEMI SYSTEM*1) in all upgrades and new installations. Future issues regarding the scope of application in the current steel process (**Fig. 8**) are as follows:

(1) In expanding application to high-speed processes like hot rolling, steel plate and cold rolling, it will be necessary to realize even higher reliability and real-time control, and examples of our efforts include: integration into open real-time Linux, and development of high-speed data sharing and communication functions between distributed servers.

(2) In order to handle obsolescence upgrades of the many types of existing manufacturer process computers, we are expanding the menu of middleware so that existing manufacturer middleware software can be replaced with the NS SEMI SYSTEM*1).

(3) In order to continue to keep up with the latest OS versions and software development environments, we have established a specialized team system (both within and outside of Nippon Steel) to ensure maintenance and long-term support of the NS SEMI SYSTEM*1), and enable response to the latest system technology applications. In the future, we will further expand and automate features like general-purpose durability test programs which are internal tools.

2.3 Process computer obsolescence upgrades

Most of the process computers within Nippon Steel will have to be continually upgraded in the future as suppliers stop providing maintenance. From the standpoint of ensuring quality in the midst of extreme labor-saving and automation of operations know-how, this must be done because system downtime leads to operation stoppages. For these upgrades, we are developing automatic conversion systems for porting currently operating AP software to open systems at minimal cost, and support systems for improving software development productivity when redeveloping AP software anew under the NS SEMI SYSTEM*1).

2.3.1 Development of conversion system for porting existing AP software to open systems

This section describes an actual case where a manufacturer process computer from more than 10 years ago was efficiently upgraded to an open system while the existing software was left as is. This was done via automatic conversion, and by expanding the NS SEMI SYSTEM*1) (see **Fig. 9**).

In porting existing process computer AP software to a new system, it was previously difficult to write software for accurately converting a source program developed using existing FORTRAN middleware specifications to C language source using other

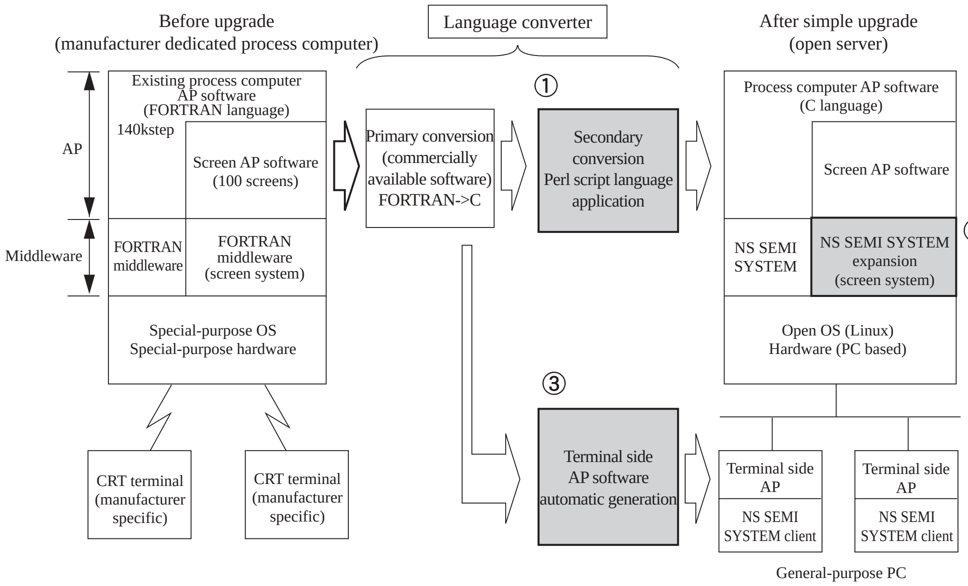

Fig. 9 Development of converter for process computer AP software upgrade

middleware specifications. This is because the differences in middleware specifications are fixed, so in the conversion software itself, it is only possible to implement simple conversions using a procedural processing language like C, and a large of amount of manual work remained.

In reusing existing process computer screen software, the existing CRT terminals are manufacturer specific, so new client software had to be written on the terminal side. Therefore, to convert from the FORTRAN language to the C language, we used commercially available software, expanded the NS SEMI SYSTEM*1), formulated specifications with good conversion efficiency, and used the Perl open source script language for conversion software. Perl is a text processing language with powerful pattern matching capabilities, and furthermore, it is open source software which can be distributed over the internet and obtained for free, just like Linux®. Complex conversion rules can be formulated by spiraling up though a process of trial and error, and it was possible to easily implement these in rule form, so the process was completed with the minimum necessary amount of manual work.

It was also possible to develop tools for automatically generating terminal side client AP software by expanding the NS SEMI SYSTEM*1) for screens, and to use all of the converted screens.

Finally, we established an upgrade system for automatically converting existing process computer AP software to the open system, with an accuracy of more than almost 99%, and thereby achieved a productivity improvement of 2~3 times (compared with the previous productivity at Nippon Steel), even when comparing with productivity in conventional simple upgrades.

In the future, this conversion development system will be adapted for all of the main manufacturers. Also, for languages other than FORTRAN, we have already developed language conversion tools for converting (for example) from industrial BASIC to the C language by combining the accumulated Perl components. For existing AP software which uses high-speed process I/O, the system is being further expanded and developed, by making general-purpose high-speed process I/O for the open system an additional menu of the NS SEMI SYSTEM*1).

2.3.2 Reduction of the amount of new AP software development, and development of productivity enhancement tools

There are cases where existing process computer AP software directly uses the OS instructions of special-purpose process computers, or where development has been done in a special language, or where maintainability declines because existing AP software has been added to or changed due to changes in the form of operations or in equipment over many years, and these changes are not always reflected in documentation. In obsolescence upgrading of process computers, it may be necessary to analyze and recreate AP software anew while improving visibility. However, in order to reduce development cost and time, we are expanding and developing existing software productivity enhancement tools and AP packages to deal with process computer obsolescence upgrades, and combining these with the wealth of commercially available software for supporting open development (**Fig. 10**).

(1) Application packages

We are developing packages comprised of common function sets and processing tasks for reducing the amount of AP creation for tracking and data collection, and subsystems with maximal specifications for some processes.

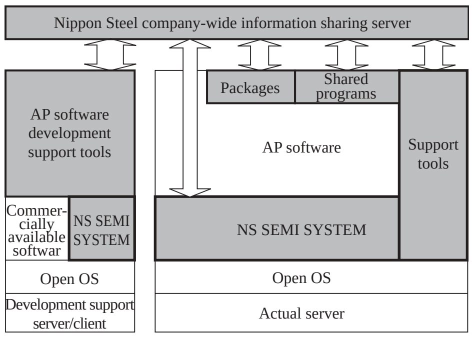

Fig. 10 NS SEMI SYSTEM*1) development support tools

(2) Development support tools

The system is equipped with a production support tool for partial automatic generation of AP source from design documents in Excel format, and a coding agreement checker tool for interactively checking quality of AP source programs using Nippon Steel in-house standards. Existing system analysis tools for static analysis of existing AP source programs have also been integrated into menus using Perl components.

A screen document development support tool automatically generates screen AP source programs in accordance with the client/server system and variations in the3-layer screen AP structure. It also supports features like multiple monitors and large screens at EIC integrated terminals in large-scale processes (**Fig. 11**). To stay in step with the Internet era, the system enables as-is use of graphical process computer screens, and the use of web browsers at on-site terminal. Installation and environment setting work are unnecessary for various types of client software on the terminal side, and this greatly improves maintainability (i.e. replacement when a terminal PC is malfunctioning, or expansion).

There are also security setting functions, and these enable staff to perform real-time monitoring, with screen images exactly the same as online screens, from OA PCs in offices remote from the production site. This feature is powerful for understanding and analyzing the operations situation.

(3) Test support tools

In addition to a combination/general test simulator, and test scenario creation support tools, the system captures IP packets necessary for the network, transfers them to a parallel running server in real-time, and these can be used with the parallel running tool for later playback. Two protocol converters are inserted between existing BSC lines, and these are also used for existing message analysis and test data creation with the parallel running tool.

(4) Nippon Steel company-wide sharing system

The above application packages and tools are centrally managed with a company-wide shared server. This enables access to AP development software and documents at each steelworks. In this way, we are working to accumulate shared AP function sets and packages, and achieve sharing of software development and engineering know-how.

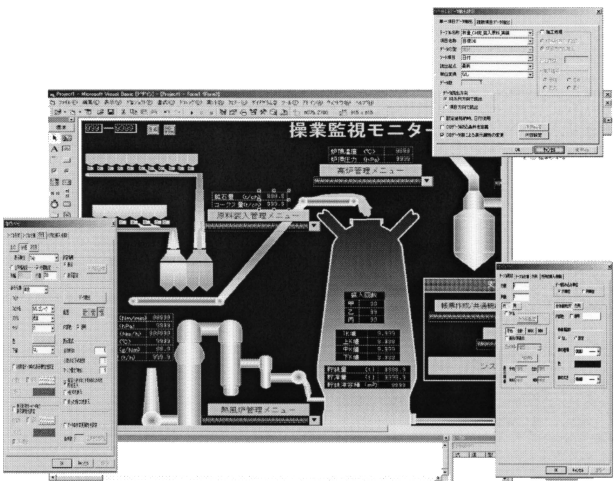

Fig. 11 Process computer screen development support tool

3. Open Conversion of Electrical/Instrumentation (EI) Software

3.1 Issues in EI open conversion

Looking back on changes in system structure (including electrical/instrumentation facilities), we were dependent on manufacturer-specific technology until around 1990.

(1) Two-level system where C and EI are connected via I/O (up to about 1975)

(2) Three-level system with network connection between C and EEI (DDC conversion) (1975 to 1985)

(3) HMI integrated system for EIC (1985 to 1990)

From about 1990 to the present, conversion to open systems has progressed to the level indicated in the example in **Fig. 12**. Open conversion of EI is still in progress, and a conceptual diagram of the system configuration after final open conversion of EIC is shown in **Fig. 13**.

As issues in the open conversion of EI, there are the issues of open conversion of EIC communication and EI networks, and open conversion of PLC and DCS, but this section will discuss development efforts aimed at standardizing the EI software needed in the process of realizing the latter (i.e. open conversion of PLC and DCS).

3.2 Standardization of electrical/instrumentation software production

3.2.1 Overview

In software production for electrical PLCs and instrumentation DCS, standardization has been delayed due to factors like: (1) The existence of various formats in documents for software production

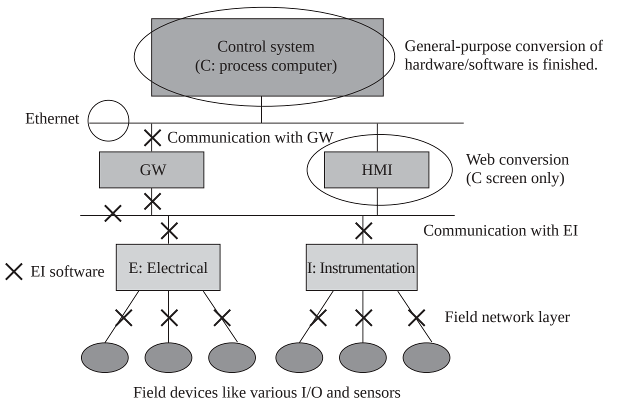

Fig. 12 Issues for EIC open conversion (example of current blast furnace system configuration diagram)

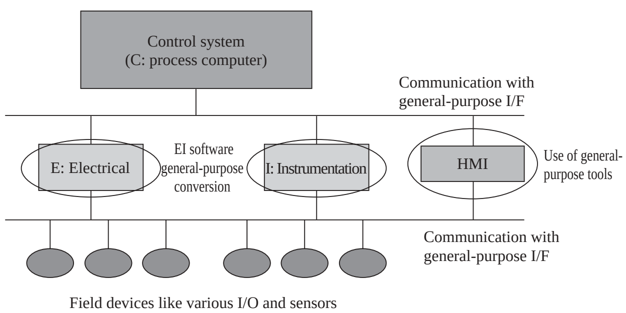

Fig. 13 System configuration after EIC open conversion

Table 1 Issues and basic policy for production of instrumentation DCS software

Previous software production procedure	Document (function specifications) => Software (DCS)
Previous issues	1) Document format was not standardized, and improvement of reusability was difficult. 2) High work load because all document preparation was manual work.
Basic policy	1) A software production tool with the following functions is needed: (1) Equipped with software components and document formats standardized within Nippon Steel. (2) Equipped with input support functions for document preparation. (3) Equipped (within document) with function for importing software creation related information to DCS.

(function specifications, software processing specifications etc.), and (2) Differences in programming languages between hardware manufacturers. This has interfered with improvement of document and software reusability, and the associated improvement of software quality. The following specifically describes issues and basic policy for software production in the fields of electrical equipment and instrumentation.

3.2.2 Issues and basic policy for instrumentation DCS software production (See **Table 1**)

3.2.3 Issues and basic policy for electrical PLC software production

For electrical equipment, the programming language in most cases is ladder language, so the following issues are involved, in addition to the issues with instrumentation[3]:

(1) Programs based on ladder language are logical combinations, and these are difficult to handle together with software created as documents. (2) Multiple pieces of software can come into being from the same document, so it is hard for anyone other than the software author to read that software, and subsequent maintainability is degraded. (3) Software module function is brittle, so it is difficult to reuse software.

In order to solve the above problems, improve software readability, and achieve greater modularity, new types of PLCs are being disseminated in Japan. In addition to ladder language, these employ the IEC61131-3 standard language (JIS B 3503) which encompasses SFC (Sequential Function Chart) language, FBD (Function Block Diagram) language and ST (Structured Text) language.

So, in addition to the basic policy in Table 1, the adopted basic policy for electrical equipment is to use the IEC standard language as the software language, except at plants where the ladder language is sufficient for software readability (i.e. small-scale plants).

3.3 Overview of electrical/instrumentation software production support tools

Electrical/Instrumentation software production support tools (hereafter referred to as “CASE”) were developed in accordance with the basic policy in Section 3.2 above. The composition of CASE for instrumentation and electrical software are shown in **Figs. 14** and **15**, respectively.

The main functions of CASE are as follows.

3.3.1 Standardization of software component symbols and document formats (Example: **Fig. 16**)

(1) Customization of things like software components, in conformity with JIS symbol components and the IEC61131-3 language, using general-purpose software (Microsoft Office: Excel, Word, Visio)

(2) Creation of coding rules to enable simplification of coding from documents to software

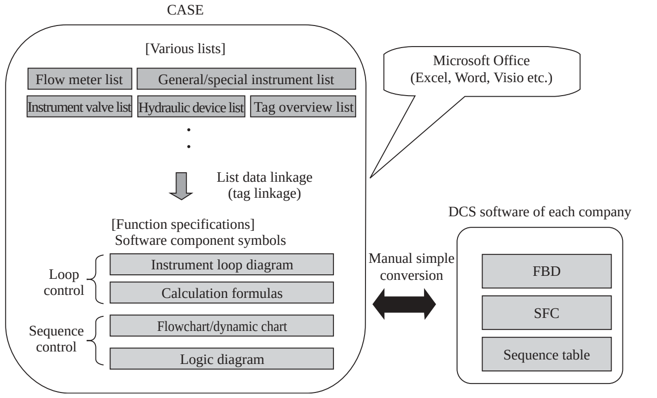

Fig. 14 Overview of CASE for instrumentation

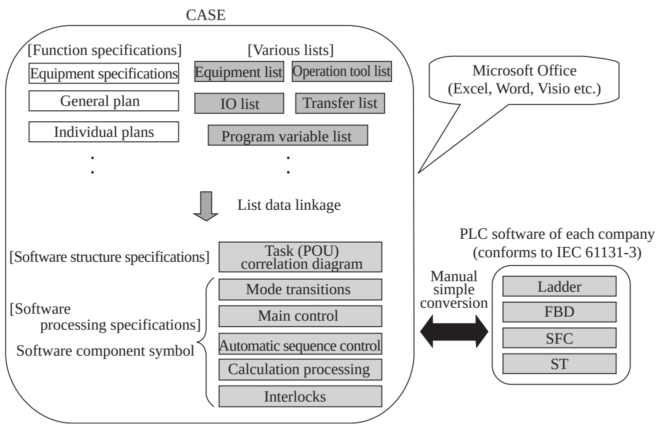

Fig. 15 Overview of case for electrical

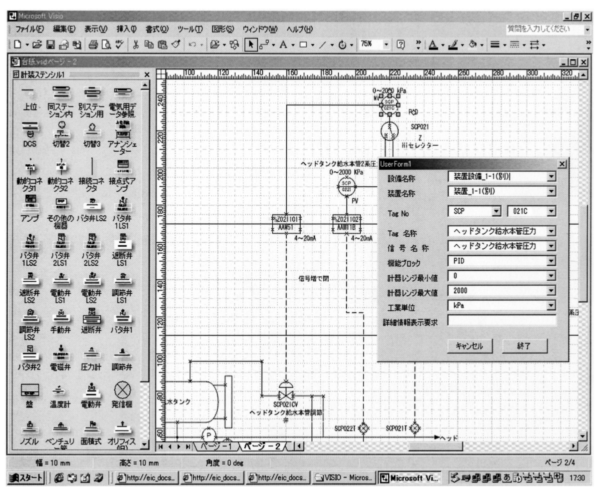

Fig. 16 Example of CASE document (1)

3.3.2 Data linkage of various lists and documents using a data base (Example: Tag linkage etc., Fig. 16)

3.3.3 Input support functions for improving efficiency of variable name input and software component input (Example: **Fig. 17**)

In the future, the plan calls for accumulating documents with these CASE tools, thereby improving reusability of documents for electrical/instrumentation control equipment within Nippon Steel, and studying the construction of an import system for reducing the manual coding load from CASE documents to software.

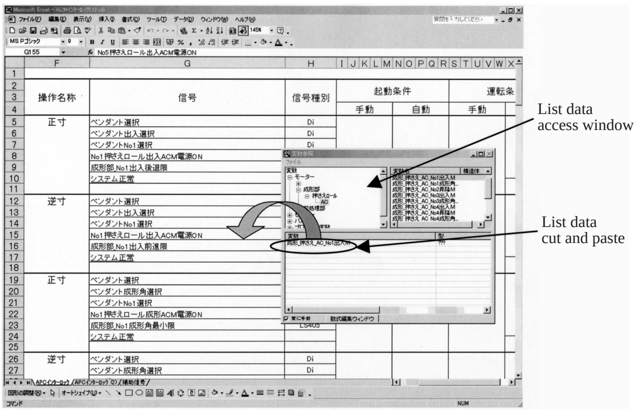

Fig. 17 Example of CASE document (2) (input support function)

4. Conclusions

Based on the core of the NS SEMI SYSTEM[*1)] middleware for open process control, we have expanded functionality, systematically developed support tools for development and expanded application in actual equipment. The system is being gradually applied to process computer obsolescence upgrades and will be used in future development of electrical and instrumentation software. Also being considered are: applications at steel companies other than Nippon Steel, and applications in other manufacturing industries. We will further expand our application development and solution activities for the latest open systems, in order to meet the diversifying needs of the ubiquitous era, and at the same time, we will continue our long-term support system for the middleware and tools which are the foundation of the Nippon Steel company-wide process control system, while keeping an eye on universal technology.

References

1) Kawahara et al.: Application of Rightsizing and Open System to Steel Processes. Shinnittetsu Giho. (363), 37 (1997)
2) Shimoi et al.: Application of Open Technology to Blast Furnace Process Control. 143rd Spring Symposium, Iron and Steel Institute of Japan (ISIJ) 2002
3) Sekiguchi et al.: Technology Trends in Sequence Control. Technical Report, Institute of Electrical Engineers of Japan (IEEJ) (781), 22 etc., (May 2000)