

# 統合エンジニアリングツール

Integrated Engineering Tool

岩瀬 修  
IWABUCHI Osamu

垂石 肇  
TARUIISHI Hajime

野島 章  
NOJIMA Akira

生産システムの高度化・多様化に伴う、エンジニアリングの複雑化および負荷の増大に対し、アプリケーションプログラムのソフトウェア生産性と信頼性の向上が求められている。このような背景を踏まえ、PLC (Programmable Logic Controller) の世界標準言語であるIEC61131-3に準拠し、コントローラ規模や制御種別に依存しない、さらに上流設計から設備稼働後のメンテナンスに至るライフサイクルを一貫してサポートする統合エンジニアリングツールを商品化した。

The costs of engineering are increasing in line with the advancement and diversification of production systems. Against this trend, it is necessary to improve the productivity and reliability of application programming.

In response to this situation, we have developed an integrated engineering tool based on the IEC61131-3 standard. This tool is independent of the size of controller and classification of control, and provides support throughout the life cycle from upstream design to maintenance after the equipment enters operation.

This paper describes the functions of the integrated engineering tool.

## 1 まえがき

生産システムの高度化、多様化が要求され、これに呼応して制御用コンポーネントは、機能、性能面で大きく進化してきた。その反面、生産システムを具現化するためのアプリケーションプログラムは複雑化し、生産システムによっては、ループ制御とシーケンス制御など従来別々であった領域の制御が混在して扱われる機会が多くなっている。生産システムの構築からメンテナンスまでのエンジニアリングが、ユーザーにとって過大な負荷となってきた。

そこで当社は、ループ制御、シーケンス制御、コンピュータ制御をシステムの規模および用途に応じて柔軟に対応できるコントローラを提供する。さらに、それら制御用コンポーネントを統一的にサポートし、かつ単独のプログラミングツール(ローダ)ではなく、システム全体をカバーするとともに、システム構築からメンテナンスに至るエンジニアリングのライフサイクルのサポートができ、また、部品化、カプセル化に代表されるソフトウェアの高生産性と高品質を重視したオブジェクト指向型のエンジニアリングツールを提供する。

## 2 エンジニアリングツールの支援環境

統合エンジニアリングツールの開発にあたり、機種および規模の統合、オープンとスタンダード、プロダクティビティ、ライフサイクルの基本コンセプトを主眼に置いた(図1)。以下、エンジニアリングツールの支援環境について述べる。

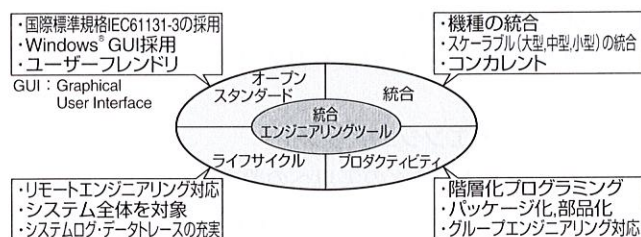


図1. エンジニアリングツールの基本コンセプト 中央の四つを主要開発コンセプトとした。  
Concept of engineering tool

### 2.1 ユーザーフレンドリな支援環境

ユーザーフレンドリなエンジニアリングツールとするため、汎用パソコン(PC)を採用し、基本ソフトウェア(OS)にPCのDFS(De Facto Standard)であるWindowsNT®(注1)、Windows®(注2)98を採用した。

画面操作はユーザーインタフェースとしてマウスオペレーション、キーボード(キーボードだけでも可能)、ショートカットキーオペレーションと様々な要求にこたえた。

操作作法はWindows®のスタイルガイドに則したものとされているため、直感的にエンジニアリングツールを操作することができる。

### 2.2 シングル・マルチユーザー対応を実現した支援環境

エンジニアリングツールの構成では、システムの規模、プログラム数などによりスタンドアローン、またはクライアント/サ

(注1)、(注2) WindowsNT、Windowsは、Microsoft社の商標。

サーバ(サーバマシン1台とクライアントマシンを複数台で構成:以下、C/Sと略記)の形態を取ることができる。このため、複数のクライアントマシンを配置しプログラムのモニタおよび編集が可能となる。このことは、設計時のグループワークに対応するとともに、コンカレントなプログラム開発を可能としている。

### 2.3 統一されたエンジニアリングツール

統合コントローラVシリーズは、システム規模によってスケラブルに構成できるコントローラであり、制御機能別に大型(3モデル)、小型(2モデル)、超小型(1モデル)の3タイプのCPUモジュールが準備されている。また、制御機能別として、シーケンス制御モジュール(S3, S2)、ループ制御モジュール(L3, L2, L1)、コンピュータモジュール(C3, C2)の計7種類のCPUモジュールが準備されている。

これらのCPUモジュールとネットワークモジュールを含む、システム全体を対象に、システム構築、プログラミングおよびモニタリング、各種パラメータの設定、RAS(Reliability, Availability and Serviceability)情報の収集・監視などのすべてをサポート範囲としているため、エンジニアリング作業の統一が図られる。

また、システム全体の監視、複数モジュールのプログラムの実行状態の監視を集中して同時に行えることが可能となりシステム状況把握、あるいはプロセスの異常状態の対応を早めることができる。

### 2.4 リモートエンジニアリング

リモートエンジニアリングの実現への要求は、ますます強くなっている。それは、工場の無人化、広域分散化、24時間連続稼働など、エンジニアをそれぞれの設備(工場)に合わせて配置することができないため、あるいは、エンジニアリングの遠隔集中管理の要求に起因する。

このエンジニアリングツールは、基本的にDCOM(Distributed Component Object Model)をベースとしたインターネットエクスプローラ(IE)によるオートダウンロード機能を利用したC/Sシステムであることから、ISDNなどの秘匿回線を使用し公衆回線網上にクライアントマシンを接続することにより、遠隔地からのエンジニアリングが可能である。

## 3 高生産性、高品質を目指したプログラミング支援

PLCの世界標準言語であるIEC61131-3に準拠したプログラミング言語の採用により、次のメリットがある。

- (1) 機種に依存しないプログラムの移植性・流用性
- (2) 階層化プログラミングによるソフトウェアの構造化
- (3) カプセル化、部品化による高生産性・高品質
- (4) 変数宣言の厳密な規約による誤ったプログラムの排除
- (5) 変数の配列・構造体利用によるプログラム間インタフェースの容易性

さらにこのエンジニアリングツールでは、下記を可能としたことでエンジニアリングコストの低減を図っている。

- (6) グローバル変数の階層化によるシステムレベルのプログラム記述性
- (7) LD, SFC, FBDの3言語の混在表記による制御論理の記述性、理解容易性

### 3.1 IEC61131-3言語

IEC61131-3では、グラフィカル言語として、LD(Ladder Diagram:シーケンス処理型言語)、FBD(Function Block Diagram:データ処理型言語)、SFC(Sequential Flowchart:状態遷移処理型言語)と、テキスト言語として、ST(Structured Text:構造化テキスト言語)、IL(Instruction List:リスト言語)の計5言語が規定されている。

これらの言語で記述されたプログラム、FB(Function Block)の単位をPOU(プログラム構成単位)と総称し、それぞれのPOUは、自由に他のPOUを使用して構成することが可能である。

例えば、複雑なプログラムを作る場合は、最初に機能の大部分を記述している大きなFBに分解し、次いで、より詳細な機能を記述した小さなブロックに分解し、最終的には命令語レベルのFBを作成することにより、階層的なプログラミング設計が可能となり、プログラムの構造化ができ、可読性、保守性がより高まる。

### 3.2 高生産性と高品質を実現したオブジェクト指向型エンジニアリングツール

プログラミング支援において、検証されたFBを再利用する場合、そのプログラムの内部構造を意識せずに、そのFBが提供する機能と、それが使用する外部パラメータを理解するだけで利用可能となるユーザー定義FBが準備されている。これはプログラムのパッケージ化であり、カプセル化や内部構造の隠蔽(いんぺい)化により実現されている。カプセル化は、従来のレジスタNo.などの物理アドレスを意識したプログラミングから、変数名によるプログラミングを基本とし、ローカル変数とグローバル変数が区別して扱われることにより実現させた。また、データにおいても変数の配列化や構造体化により、同様にカプセル化が可能となっている。

このように、オブジェクト指向型エンジニアリングツールとなっているので、再利用性が向上しアプリケーションプログラムの高生産性と高品質が期待できる。

### 3.3 データのロケーションを意識させないプログラミング

IEC61131-3におけるグローバル変数はコントローラ(IEC61131-3で言うリソース)内のプログラム間のデータインタフェースとして規定しているが、統合コントローラVシリーズでは、これをシステムレベルまで3階層の定義を可能とした。さらに進めて、グローバル変数の3階層とは、①コントローラ内のプログラム間で使用するコントローラグローバル、②ステーション内のマルチホスト対応として、ステーション内コントローラ

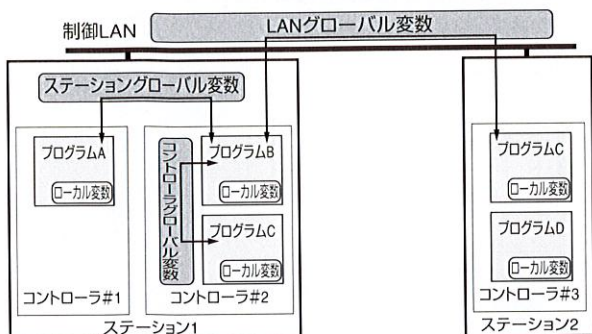


図2. 変数の階層化概念 グローバル変数の3階層の概念を示す。  
Concept of three-layered variables

間で使用するステーショングローバル、③制御用LANで接続されたステーション間で使用するLANグローバル、となっている。この3階層のグローバル変数は、それぞれの階層でデータのロケーションを意識することなく、相互に参照が可能となり、コントローラ間のデータの結び付けがきわめて容易になった。

このように、前項のローカル変数のプログラム内のリロケート化とグローバル変数のシステム共有化により、それぞれのプログラムの独立性を保つとともに、データインタフェース処理手続きを容易かつ合理的なものとしている。これら変数の概念を図2に示す。

### 3.4 言語に左右されないプログラムのモジュール化

それぞれの言語は、それぞれのプログラムシート(ワークシートとも言う)に記述されるのが一般的であるが、統合コントローラVシリーズはグラフィカル3言語のLD, FBD, SFCを混在表記できるエンジニアツールを実現した。

各言語は表1に示す特長をもっており、3言語を自由に混

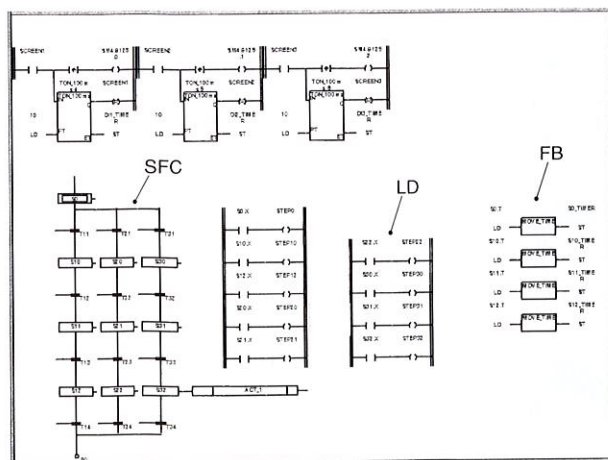


図3. グラフィカル3言語の混在表記例 仕様記述性、プログラムの可読性が向上する。  
Mixed description of three types of programming language

表1. グラフィカル3言語の特長

Summary of three types of programming language

制御演算	言語	LD	FBD	SFC
論理演算		◎	○	×
数値演算		×	◎	×
文字処理		×	○	×
順序処理		○	○	◎

◎:最適, ○:可能, ×:不可能

在できることで、仕様記述性が著しく向上する。また、各言語を同一編集平面に表記できるので、実行状態のモニタ時に呼び合いをたどってみる必要がない。一度に全体を俯瞰(ふかん)することができるので、プログラムの可読性が向上する。3言語混在表記の例を図3に示す。

## 4 エンジニアリングのライフサイクルサポート

統合エンジニアリングツールはシステム構築から設備稼働後のメンテナンスまでのすべてのライフサイクルをサポートしている。システム構築とメンテナンスのフェーズで使用する主な機能について述べる。今までのプログラミングツール(ローダ)と大きく異なる点を表2に示す。

表2. ローダとエンジニアリングツールの比較

Comparison of program loader and engineering tool

項目	プログラムローダ(従来)	統合エンジニアリングツール
エンジニアリングの対象	コントローラ(1:1)	システム全体
支援環境	スタンドアローン	C/Sシステム リモートエンジニアリング グループエンジニアリング
システム構築	システム構築	ネットワーク、ステーション、 コントローラ、I/O構成
言語	シーケンス	ラダー、SFC
	計装	計装専用言語
データ管理(変数)	レジスタNo.	変数(ローカル、グローバル) 配列、構造体
プログラミング	メイン、サブルーチン	断層化、モジュール化、 カプセル化、3言語混在表記

### 4.1 システム構築(システムコンフィグレーション)

統合エンジニアリングツールのシステム構築において、ハードウェアのシステム構築とプロセス信号のデータベース構築が一括して行える。

**4.1.1 ハードウェアのシステム構築** ハードウェアのシステム構築には、ステーションノードの構成、ネットワーク系統、ステーションのコントローラモジュール構成、I/O構成を登録する。その例を、図4に示す。

**4.1.2 プロセス信号のデータベースの構築** 対象となるシステムのプロセス信号として、I/O変数、グローバル変数があり、それらの登録を行う。

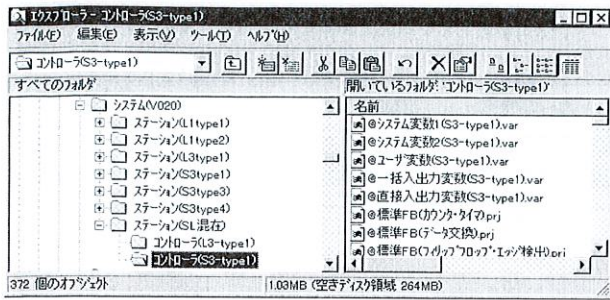


図4. システム構築画面 システムの構築を行う画面を示す。  
System configuration display

## 4.2 メンテナンス

**4.2.1 変数のクロスリファレンス** すべての変数に対して、プログラム上での使用箇所をデータベースで管理していることから、ある変数の参照しているプログラムをモニタしていた場合に、その変数に対して書込み(コイル出力など)しているプログラムに自動展開が可能であり、その範囲はコントローラ間、ステーション間を問わず展開が可能である。

主な機能は次のとおりである。

- (1) システム全体の変数使用先一覧表示
- (2) 一覧リストから使用先へのプログラム表示展開
- (3) 直近で使用しているプログラムへの展開

**4.2.2 オンラインプログラム変更** システムの稼働中にコントローラが、運転状態でのプログラム変更はもちろん、接点、コイルのフォース<sup>(注3)</sup>も使用できる。これは、プログラム部分変更機能として、プログラムの翻訳(コンパイル)をすることなく、直接コントローラのプログラムを変更する機能であり、保全時に迅速な対応が取れる。

**4.2.3 プロセス監視画面上でのプログラム表示** エンジニアリングツールはC/Sシステムであるため、ヒューマンインタフェース(HI)マシン(PC)をツールのクライアントとするだけで、監視用画面から指定のプログラムが表示可能となる。プロセスオートメーションの場面において、インタロック条件を監視画面として作成・表示していることが多いが、インタロックシーケンスを表示することによってその煩雑な作業が削減される。また、SFCアクティブステップ状態の表示においても同様の効果が発揮される。監視画面でのプログラムモニタ例を図5に示す。

### 4.2.4 参照関係図によるプログラム変更時の影響表示

統合コントローラVシリーズのプログラムは、プログラム、FB、ファンクションの組合せから成る。あるプログラム(FB、ファンクション)は、他のプログラム(FB)の中で使われている

(注3) プログラムの実行状態にかかわらず、状態を保持すること。

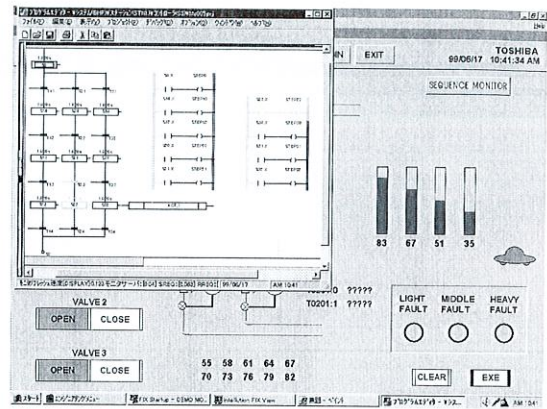


図5. 監視画面でのプログラムモニタ 監視画面にプログラムのオンラインモニタを表示。  
Program monitor display on monitoring screen

場合があり、修正変更したときには、それを使用している機能に影響を及ぼす可能性があるため、プログラム、FB、ファンクションの相関関係を右向グラフにまとめ、相関を表示し、影響把握ができる。

## 5 あとがき

統合コントローラ Vシリーズのエンジニアリングの概要について述べた。当社の様々な分野でのシステムインテグレーション技術を踏まえ、トータルエンジニアリングコストの大幅な低減をコンセプトに、統合エンジニアリングツールを開発した。

今後さらに、使いやすさと、使い勝手の良いエンジニアリングツールの充実に邁進(まいしん)する所存である。

## 文献

- (1) IEC1131-3ハンドブック . PLCopen JAPAN発行. 1998-5・15.



岩淵 修 IWABUCHI Osamu

情報・社会システム社 府中情報・社会システム工場 計測制御機器部主査。エンジニアリングツールの企画・開発に従事。電気学会会員。

Fuchu Operations-Information and Industrial Systems & Services



垂石 肇 TARUISHI Hajime

情報・社会システム社 府中情報・社会システム工場 計測制御機器部主査。エンジニアリングツールの開発・設計に従事。計測自動制御学会会員。

Fuchu Operations-Information and Industrial Systems & Services



野島 章 NOJIMA Akira

情報・社会システム社 府中情報・社会システム工場 産業制御・計装システム部主査。鉄鋼システムの開発・設計に従事。電気学会会員。

Fuchu Operations-Information and Industrial Systems & Services