

制御ロジックの図形言語化による ソフトウェア生産性・信頼性の向上

Improvement of Productivity and Responsibility
by Visual Programming of Control Logic

技術本部 西崎純一*1 桑原青子*2
石井圭*3 桑原耕治*4

製品搭載用制御ソフトウェアでは、開発期間の短縮と同時に、より厳しい信頼性向上が要求されている。この要求にこたえるために、機械装置への適用を目的として、可搬性に優れたノートパソコンで動作する図形言語システムを開発した。このシステムは、制御ロジックを図形で表現するため、機械設計者にも容易に理解できるとともに、ロジック再利用のためのライブラリー機能を有する。さらに、C言語のソースコードを生成するのでポータビリティに優れ、機械制御向けの高速制御演算も可能という特徴を持つ。本システムをクレーン自動運転制御装置に適用し、従来と同等な30msの制御演算周期を実現するとともに、検証作業時間の30%短縮を確認した。

On-board machine control software requires a short development time and high reliability. We have developed portable visual language programming for mechanical plant control systems that runs on laptop PCs. The control logic is comprehensive for mechanical designers and the control logic is represented by visual logic elements. The system's library functionality enhances software reusability. The system's high-speed real-time operation is suited to mechanical plant control and its excellent portability is realized by C source generation. Applying the system to building a crane control system for example, we achieved a 30 ms control interval and reduced software verification time by 30%.

1. ま え が き

製品搭載用制御ソフトウェアでは、開発期間の短縮と同時に、より厳しい信頼性向上が要求されている。この要求にこたえるためには、ソフトウェアの可視化・ライブラリー化が必須となる。さらに、機械用制御装置の場合には、実行の高速性とソフト開発支援装置の可搬性が要求される。

このたび、機械装置への適用を目的として、下記の特徴を有する図形言語システムを開発した。

- (1) 制御ロジックを図形言語で図面化するため、機械設計者にも容易に理解できる。さらに、制御プログラムの動作状態が、図面上で視覚的に確認でき、ロジックチェックが簡単に行える。
- (2) 制御ロジックの階層化が可能で、ロジック再利用のためのライブラリー機能を有する。
- (3) SFC (Sequential Function Chart) の選択実行機能を活用しているため、高速制御が可能である。
- (4) ノートパソコン上で動作が可能である。
- (5) C言語のソースコードを生成するのでポータビリティに優れる。

本報では、図形言語システムの構成、特徴及び図形言語による制御プログラムの開発手順についてまとめるとともに、実機制御装置への適用結果について報告する。

2. 図形言語システムの構成

このたび開発した機械制御向け図形言語システムのシステム構成を図1に示す。

本システムは、制御ロジックを図形言語を使用して図面化し、その図面情報から自動的に制御プログラムを作成するもので、作成した図面上で制御ロジックの実行状態を視覚的に確認しながら、

ロジックを検証する機能を有している。制御プログラムの作成は図形言語システムパソコンのみで可能であり、制御コンピュータはプログラムデバッグ段階でのみ必要となる。

図形言語システムのソフトウェアは、次の4機能から構成される。図2に機能ブロック図を示す。

- (1) 制御プログラム作成機能
図形言語を用いて、制御プログラム(C言語)を生成する機能
- (2) 制御プログラム検証・調整支援機能
生成した制御プログラムの実行状況を監視、制御する機能
- (3) マクロライブラリー管理機能
作成した制御ロジックをライブラリー化して、再利用できる形で、保存・管理する機能
- (4) 制御装置用実行プログラム生成機能
図形言語システムパソコンで作成した実行用プログラムを制御コンピュータにダウンロードする機能

3. 特 徴

図形言語システムの主な特徴を以下に示す。

- (1) 機械制御に適した図形表現
ロジックの図面化により、複雑な機械の動きを図面化でき、以下の特徴がある。
 - ① 各図形要素の演算順序を工夫することにより、制御周期単位での処理時間遅れを防止し、機械制御の高速性を確保している。
 - ② SFCで記述する動作・状態遷移のシーケンスとFBD (Functional Block Diagram) で記述する各動作の制御ロジックを同じシート上に書くことができる(図3参照)。
 - ③ FBD要素としてC言語で作成した関数を使用できるため、既存のC言語モジュールの活用が容易である。

*1 広島研究所制御システム研究室長

*3 高砂研究所制御システム研究室

*2 広島研究所制御システム研究室

*4 エレクトロニクス技術部長

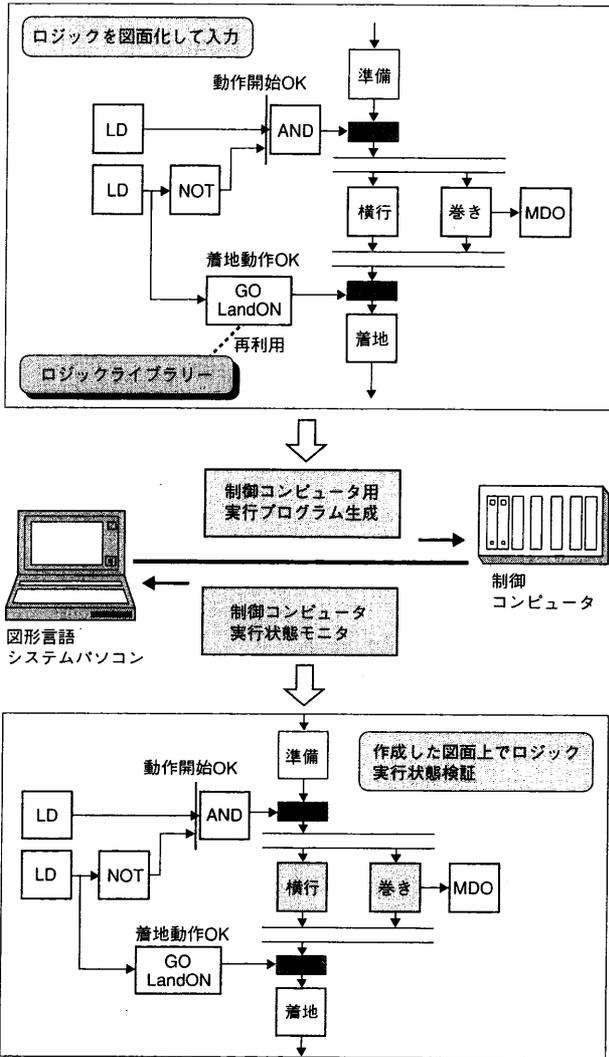


図1 図形言語システム全体構成 図形言語システム全体の構成を示す。
Configuration of visual language system

(2) 図面からC言語プログラムを自動生成

制御プログラムは、ロジック図面からC言語プログラムを生成し、そのC言語プログラムから対象の制御コンピュータのハードウェアやOSに合った機械語プログラムコードに自動的に変換して作成する。したがって、ハードウェアやOSに依存した部分を改造するだけで、別機種のマイコンやパソコンを制御コンピュータとすることができ、拡張性に富んでいる。

(3) マルチタスク構成が可能

優先度の異なる複数のタスクで制御システムを構成することができる。これは、クレーン自動制御システムのように、高速な自動運転制御処理と同時に、ホストコンピュータとの情報授受処理等の複数の並行処理を必要とする多機能システムには必須の条件である。

(4) 制御コンピュータでの実行状況をビジュアルに監視

制御ロジックの制御進行実行状態を、作成した図面上での色替え表示により、視覚的に確認できる。

(5) 制御ロジックの強制変更が可能

制御ロジック実行中に、状態値を書換えてテストしたい状態にすることができ、実際の機械を動かさなくても制御ロジックのテストができる。

(6) ライブラリー管理機能の充実

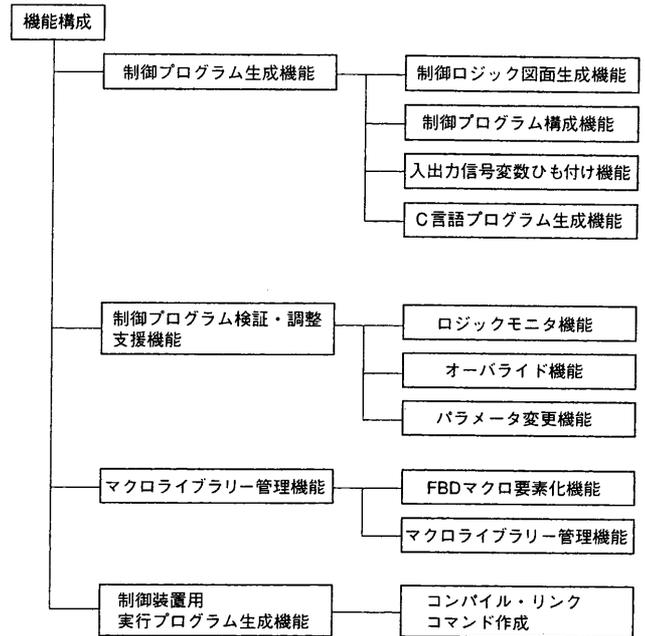


図2 図形言語システムソフトウェア機能ブロック図 図形言語システムは、4個の大きな機能から構築される。
Software function block diagram of visual language system

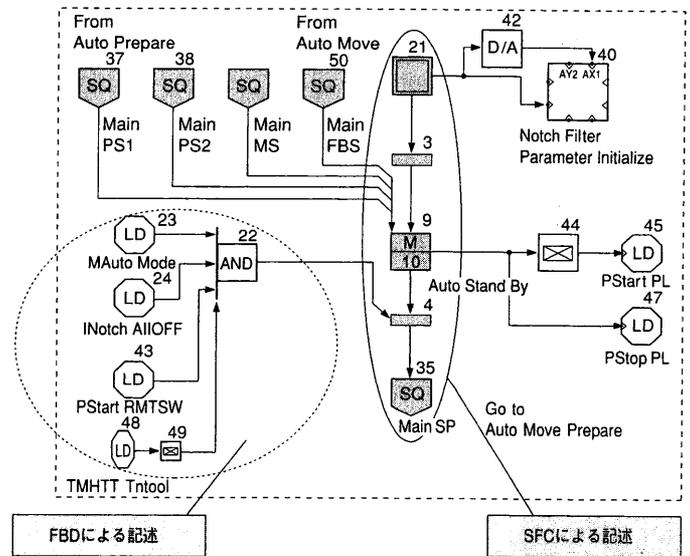


図3 SFC, FBD 混合記述図面例 SFCとFBDの図形要素の混合記述が可能を示す。
Example of mixed description by SFC and FBD

FBDによるロジック記述部分をマクロ要素（モジュール）として定義し、ライブラリー化して登録管理する機能を充実させた。マクロ要素化の利点として次の点がある。

使用例を図4に示す。

- ① 制御ロジックを構造化して階層構造で記述でき、図面を分かりやすくできる。
- ② ロジックを機能単位でライブラリー化し、再利用することで、ロジック設計者の判断に依存するロジック部分が少なく、設計者でなくてもソフトウェアの保守が容易にできる。

4. 制御プログラムの開発手順

本図形言語システムを用いて制御プログラムを開発する手順を以下に示す。

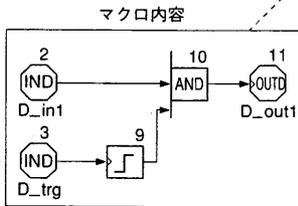
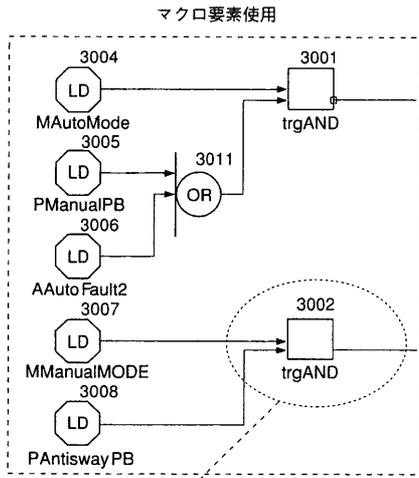


図4 マクロ要素使用例 マクロ要素により、ロジックの階層化が図れることを示す。
Example of macro element

4.1 制御プログラムの作成

制御ロジック図面の作成からC言語の制御プログラムを作成する。図5に、ロジック図作成からC言語プログラム生成までの処理フローを示す。

(1) 制御ロジック図面の作成

実行する制御ロジックを、ロジックシート上に図面化する。制御ロジックをシート図面にするための図面エディタ部と、作成した図面のロジックチェック部があり、この機能により、シート単位で矛盾のないロジックを作成することができる。

ロジック図は、図面上に図形要素を配置・接続して作成していく。本システムで利用できる図形要素を表1に示す。図形要素で使用するパラメータ値や対応する信号変数名も同時に設定・変更できる。

作成した図面は、シート単位でロジックチェックされる。ロジックエラーがある場合、図面上のどの要素のロジックに不具合があるか、その詳細内容を表示して、ロジックの図面化作業が効率良く行えるようにサポートしている。

(2) 入出力変数のひも付け

ロジック図面上で使用する変数は、内部変数と呼ばれ、一元管理されている。一方、外部機器からの入出力信号は、外部変数と呼ばれ、一般に内部変数とは異なる。入出力変数ひも付け機能を用いて、内部変数と外部変数の対応付けを行う。

(3) 制御プログラム構成の定義

制御プログラムをどんな機能タスクで構成するか、タスクやサブタスクの起動条件(定周期起動やイベント起動)、タスクの優先順位等を設定する。この定義情報に基づいて、制御コンピュータ上で動作するタスク単位のC言語プログラムが生成される。

(4) C言語プログラムの生成

前記(1), (2), (3)で入力した情報を基に、C言語の制御プロ

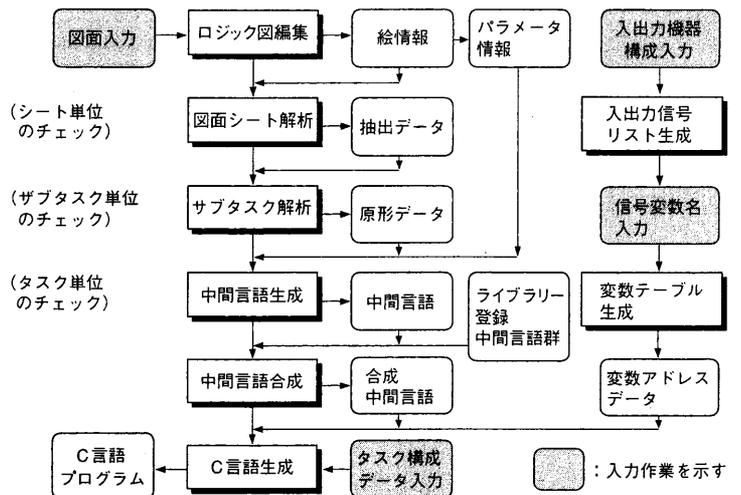


図5 制御プログラム生成フロー 図面入力からC言語生成までの処理フローを示す。
Process flow diagram which generates C-language program

表1 図形要素の種類

Classified table of visual language elements

種類		個数	
FBD	入出力要素	アナログ	5
		デジタル	5
	機能要素	—	31
	数式演算要素	—	5
SFC	—	15	

グラムに変換する。

作成したロジック図面は、シート単位でロジックチェックしてエラーなしを確認した後、サブタスクを構成する各シート間のロジック接続関係をチェックして、サブタスク単位の制御ロジックとして矛盾はないかがチェックされる。

サブタスク単位でのロジックエラーがなければ、最後にタスク単位で、使用変数の定義・参照状態をチェックして、制御プログラムとして変換可能か否か判定する。

サブタスク単位、タスク単位のチェックでエラーがあった場合、図面シート間のどの要素の接続に不具合があるか等、その詳細内容を表示して、ロジックの図面化作業が効率良く行えるようにサポートしている。

制御プログラムとして変換可能であれば、まずロジック図面情報をロジック中間言語情報に変換し、それからANSI-C規格のC言語プログラムに変換する。

(5) 制御実行プログラムの作成

前記(4)で作成したC言語プログラムをコンパイルリンクし、ロードモジュールを生成した後、制御コンピュータにダウンロードする。

4.2 制御ロジック検証

制御ロジックの検証作業は、ロジックモニタ機能・調整支援機能をフルに活用しながら実施する。

(1) ロジックモニタ機能

制御コンピュータ上での制御ロジック実行状態を、作成したロジック図面と同じ画面上に表示する。

ロジック図面上に、下記情報①②③を刻々更新表示する。

① SFCのステップ状態(活性・非活性状態)色替え表示



図6 テストクレーン 図形言語システムの性能検証用に使用したテストクレーンを示す。
Test crane used for performance test

表2 図形言語システムの評価
Evaluation of visual language system

	従来	図形言語システム
プログラム作成時間	1	1
実行プログラムサイズ	1	1.13
制御演算周期	30 ms	30 ms
検証時間	1	0.7

- ② デジタル現在値のON, OFF状態の色替え表示
- ③ アナログ現在値の表示

この機能により、制御ロジックの状態遷移と状態値を視覚的に確認でき、ロジックの検証を容易にする。

またロジック遷移があるステップで停止した場合、次に遷移するための条件のうち、どの条件が不足しているかが、すぐ判断でき、さらに図面上でその条件入力線をさかのぼって追うことにより、条件不成立の元の原因を究明することが簡単にできる。

(2) オーバライド機能

ロジックモニタ画面上で、デジタル値、アナログ値を強制的に変更する機能である。この機能を使用することで、制御コンピュータ側に入出力機器を接続しない状態で、制御ロジックの部分検証が可能となる。特に、機械制御装置の場合は、実際の機械駆動系を接続する前に、非常停止や異常処理等の制御ロジックを検証しておくことが不可欠であり、この機能はそうした事前検証に非常に有効である。

また、遷移条件が満足されずに、シーケンス遷移があるステップで停止した場合にも、不足している遷移条件のデジタル信号を、強制的にONにオーバライドすることで、次ステップへ遷移させることができ、次ステップ以降のロジックの確認を継続することが可能となる。

オーバライドしているデータはロジック図面上に強調表示されており、容易に識別される。

(3) パラメータ変更機能

ロジックモニタ時に、パラメータ設定値を変更する機能であ

る。ロジック図面作成時に図形要素ごとに設定するパラメータ値（例えばタイマの遅延時間）は、実際の制御ロジック動作時に、実状に合わせて設定変更したい場合がある。このような場合、制御ロジックを実行させながら、ロジックモニタ画面上で調整したいパラメータの設定値を変更して、動作確認することにより、適切なパラメータを決めることができる。

5. 適用例と評価

当社広島製作所の試験設備として製作、設置されているクレーンの自動運転制御装置に、図形言語システムを適用した。クレーンの外観を図6に示す。本自動運転制御装置は、従来の手動運転機能のほかに、移動経路自動生成機能や、吊り荷の振れ止め及び位置決め自動制御機能を有しており、プログラムサイズは、約1 MBytesである。この制御装置を対象に、プログラム作成時間、実行プログラムサイズ、実効速度、検証作業時間に関して、従来のC言語で開発した制御装置との比較を実施した（表2参照）。

(1) プログラム作成時間

初めての適用にもかかわらず、従来とほぼ同等の時間で作成できた。今後ロジック図の標準図面化、オペレータの習熟により、大幅な工数低減が期待できる。

(2) 実行プログラムサイズ

図形言語システムでは、図面化したロジックから定型的に自動生成するため、従来、人が工夫して直接コーディングしたものに比べて、C言語プログラムの実行サイズが、大きくなることが懸念された。しかしながら、図形言語からC言語への変換部の工夫により、プログラムサイズの増加を13%までに抑えることができた。

(3) 実行速度

本システムの特徴の一つに、SFC言語の採用がある。SFC言語の利用により、制御ロジックを選択実行させる記述が可能となり、実行速度が速くなっている。このため、図形言語化することによって懸念された実行時間の遅れは発生せず、従来と同じ30 msの制御周期が確保できた。

(4) プログラム検証時間

本システムでは、デバッグ作業が、従来のメーカー提供のデバッグツールによるものから独自のロジックモニタ機能を利用した視覚的な作業に置換わっている。さらに、制御ロジックの強制変更機能やパラメータ変更機能を有効に活用することにより、制御プログラムの検証時間を約30%短縮できた。

6. まとめ

SFC, FBDの混合記述が可能で、ロジック再利用のためのライブラリー機能を持ち、ノートパソコン上で動作する機械制御向けの図形言語システムを開発した。

クレーン自動運転制御装置への適用の結果、図形言語システムがプログラムの実行速度、プログラムサイズの点で機械制御に適用できることが確認できた。さらに、図面上で制御実行状態を確認できることから、プログラム作成時間、検証時間を大幅に短縮できた。

今後、適用したクレーン自動運転制御装置での使用に伴う改善に取組み、本図形言語システムの完成度を高めるとともに、社内各種機械制御装置への適用拡大を図っていく予定である。