

# PLCのプログラミングにおけるXML利用技術

機械制御技術部 生産システムチーム 水 矢 亨

PLCプログラミングにおいては、テキスト言語とともにグラフィック言語が多用されているが、プログラミング言語の規格ではソースコードの互換性が確保されていない。これに対して、ソースコードの共通フォーマットを定める動きがあるが、そこではXML技術が利用されている。こうした動向を紹介するとともに、XMLによる共通フォーマットを利用したPLCソフトウェアの開発ツールの開発についても報告する。

キーワード：PLC, XML, グラフィック言語, ソースコード

## 1 はじめに

シーケンス制御は、生産システムなどの制御系において、幅広く用いられている技術である。シーケンス制御用のコントローラの主流はPLC (Programmable Logic Controller) となっており、シーケンス制御システムを構築するには、PLC上で動作するソフトウェアを開発する必要がある。制御システムの高度化やネットワーク化により、システム構築におけるソフトウェア開発の占める割合が増大しており、その効率化は重要な課題となっている。

ソフトウェアの開発効率を向上させる手法として、一般的には、モジュール化やオブジェクト指向などの手法が知られている。PLC用のソフトウェア開発においては、そうした技術が十分に活用されているとは言い難い。その原因として、グラフィック言語(ラダー図など)を利用したプログラムのソースコードにおける互換性の問題が挙げられる。PLC上のプログラミングでは、テキスト言語だけでなくグラフィック言語が多用されており、この互換性を確保することは非常に重要である。この互換性確保については、XMLの利用が有望視されている。本報告では、PLC上のソフトウェア開発におけるXML技術の利用について検討する。

## 2 グラフィック言語のソースコード

PLC用のプログラミング言語に関する国際規格として、IEC61131-3<sup>1)</sup>がある。IEC61131-3では、テキスト言語に加えて、ラダー図(LD)とファンクションブロック図(FBD)のグラフィック言語について規定されている。グラフィック言語では、接点やコイルなどの言語要素に対して相互関係を結線によって表現する。図1は、LDのプログラムの例である。

IEC 61131-3では、グラフィック言語に対して、プログラムの表示についての仕様が定められている。それに従って、プログラムを表示するには、次のような情報が必要となる。

- ・ 表示用の座標系に関する情報,
- ・ 言語要素の配置情報(位置や大きさ),
- ・ 結線の情報(始点と終点, およびルーティング).

IEC61131-3の場合、どの情報がソースコードに記述されるのかという点も含め、ソースコードのフォーマットについては何も規定されていない。これが、グラフィック言語によるプログラムのソースコードにおいて互換性が確保されていない主な原因となっている。

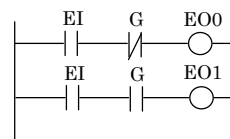


図1 ラダー図の例

## 3 XMLによるソースコードの共通化

最近では、ソフトウェアの再利用性や可読性の向上が強く求められるようになっており、それに伴って、開発ツールなどに依存しないソースコードの共通フォーマットの必要性も強く認識されてきている。

共通フォーマットを定めるには、開発ツールの設定情報が共通化されていることを前提としない限り、前章で示したような情報はソースコードに格納しておく必要がある。さらに、情報の有効活用のためには、一般性のある形式により構造化・階層化されたデータとして格納しておくべきである。構造化されたデータの記述・保存のための技術としては、XMLが広く用いられている。XMLでは、タグや属性を定義することができ、テキストファイルに構造化データを保存することができる。さらに、スキーマを利用することで、構文の指定や検証を行なうこともできるようになっている。

PLCの分野でも、プログラムのソースコードをXML形式にして活用しようとする試みが行なわれている。その一つに、PLCopen(<http://www.plcopen.org/>)によるPLCopen-XML<sup>2)</sup>があり、2005年4月に正式版であるVersion 1.0が

リリースされている。日本国内では、PLCopen Japanにより、PLCopen-XMLによる互換性などを検証するための活動が行われている<sup>3,4)</sup>。

## 4 XMLとPLCでのソフトウェア開発環境

### 4.1 エディタの共通化と開発ツールの統合化

ソースコードの共通フォーマットが流通していない現状では、ソースコードを作成・編集するには、ソースコード形式毎に対応ツールを用いる必要がある。通常、開発環境は、エディタ、各種ライブラリ、コンパイラや実行ファイルの配備ツールなどから構成される。ソースコードの共通フォーマットが利用可能になれば、このうちのエディタは共通化できるようになる。エディタを共通化して固定できることは、開発作業の効率において大きな効果があるものと考えられる。

しかし、エディタを共通化・固定しても、他の開発用ツール群(以下、SDKと呼ぶことにする)との組み合わせによる一連の作業が煩雑では、その効果が十分に得られない可能性があり、ツール間の連携を容易にすることも重要である。例えば、図2に示すように、各開発ツールを、統合開発環境(IDE)用のプラットフォームに対するプラグインとして連携させることが考えられる。開発ツールをプラグイン構造で連携させることができれば、機種毎にSDKを切替えて利用するだけでなく、テストツールやシミュレーションツールなどの補助ツールとの連携も容易になることが期待される。

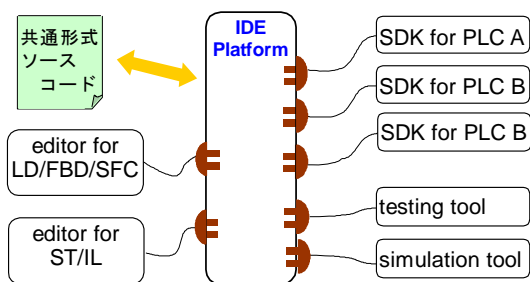


図2 プラグインベースのPLCプログラミングツール

### 4.2 エディタの試作

本研究では、IDEのプラットフォームとして提供されているEclipse<sup>5)</sup>を利用し、そのプラグインとしてPLCopen-XMLのエディタの試作開発を行なった。Eclipseは、JavaのIDEとして注目されているが、プラグインベースでIDEを構築できるプラットフォームとして開発されている。図3のような構成となっており、図2で示したような開発ツールを実現できるプラットフォームとなっている。

図4は、Eclipseプラグインとして開発したエディタの実行画面の例である。LDおよびFBDについて、グラフィッ

ク言語部分の表示、GUI操作による言語要素の配置・接続などによる作成・編集などの機能を有している。また、スキーマを利用した検証機能も利用可能となっている。

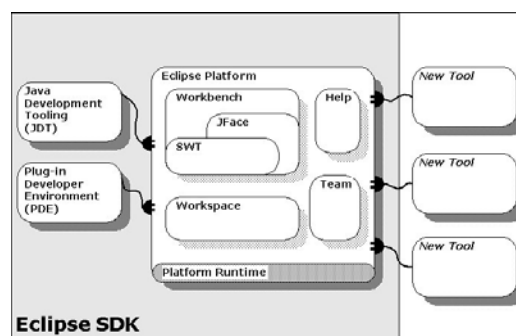


図3 Eclipseの構成<sup>6)</sup>

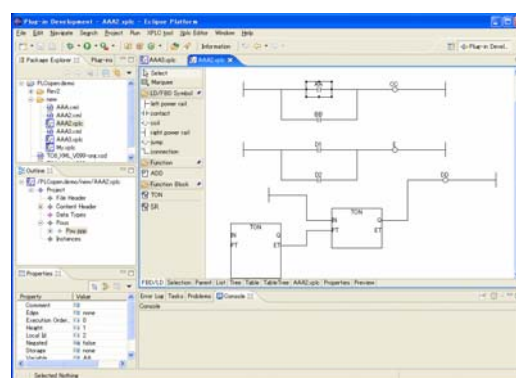


図4 Eclipse上のLD/FBDエディタ

## 5 まとめ

PLCプログラミングにおけるXML利用の動向について、ソースコードの互換性の問題とあわせて紹介した。また、XMLによる共通フォーマットを利用する開発ツールの試作開発について報告した。

## 文献

- 1) International Electrotechnical Commission : IEC 61131-3 (Ed. 2.0); "Programmable controllers – Part 3: Programming languages"(2002).
- 2) PLCopen : "XML Formats for IEC 61131-3", Technical Paper, Version 1.0 – Official Release, PLCopen Technical Committee TC6 (2005).
- 3) 宮澤・垂石: 「PLCにおけるXMLとインタオペラビリティ」, システム・情報部門学術講演会2004 講演論文集, pp.229-234, 計測自動制御学会(2004).
- 4) 垂石: 「PLCプログラミングの標準化の動向」, 計測技術, Vol. 33, No. 9, 1-5(2005).
- 5) <http://www.eclipse.org/>
- 6) <http://help.eclipse.org/help30/topic/org.eclipse.platform.doc.isv/guide/arch.htm>