

B\_033

# 制御系モデリング言語 CSML を用いた制御系設計 Control System Design with a Modeling Language CSML of Control System

古賀 雅伸†  
Masanobu Koga

竹井 佑介†  
Yusuke Takei

## 1. 背景

制御系の開発プロセスは図1に示すように、制御対象のモデリング、制御器の設計、シミュレーション、実装という手順を踏む。制御対象や制御器を記述したデータや、シミュレーションで使用されるデータのフォーマットは、使用するツール毎に異なる。そのため、開発途中で使用するデータに変更が生じた場合、関連するデータのフォーマットを全て修正する必要があり、制御系設計者への負担は非常に大きなものとなる。このため、制御系の開発において統一的に扱えるデータ形式が求められている。さらに、インターネットの普及により、データを遠隔地と交換する機会が増え、ネットワーク等の環境に依存しないデータフォーマットであることが望まれる。

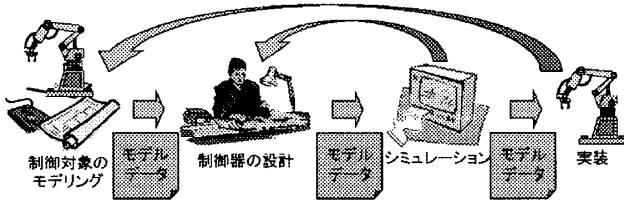


図1: 制御系開発のプロセス

本研究グループでは、昨年までに制御系の汎用モデルデータ作成のためのモデリング言語 CSML(Control System Modeling Language)[1]を開発した。現在も改良が重ねられ、より複雑なシステムの表現も可能になっている。

また、CSMLを用いて定義したデータを、各々の設計ツールで利用可能なデータ形式に変換するためのツールを開発した。CSMLを変換し、シミュレーションプログラム、および制御系ドキュメント等の自動生成を行うシステムを開発した。このシステムを用いることにより、MATIX[2]やMATLAB[3]の知識がなくても自動生成されたファイルを用いて即時シミュレーションを行うことが可能となる。

## 2. CSML

CSMLはブロック線図の構造、システムの状態等の制御系に関するメタデータを構造化して記述するXMLベースのフォーマットRSF(Rich System Format)という位置付けである。XML[4]ベースであることから、OS、ネットワークに依存せず、拡張性・再利用性に非常に優れたものとなっている。

### 2.1 CSMLの構造

CSMLはブロック線図のブロックのリストと各々のパラメータ、ブロック同士の結合関係・信号の流れを表現

している。図2は、ステップ応答の信号に積分器をかけた単純なシステムの表現している。2行目の<system>要素にシステムについての情報を記述している。属性nameは設計者が適当な名前を付けることができ、他のモデルと名前が競合しない限り、別のCSMLで再利用することも可能である。3~10行目までは、システムを表現するブロックのリストを記述してある。<block>要素に各々のブロックがとりうるパラメータを記述し、属性nameで定義された名前はシミュレーションを行う際に変換プロセッサが必要とするJavaクラスファイルの名前と対応している。11~16行目は、各々のブロックの接続関係を記述している。<connector>要素の属性fromとtoに記述されたブロックが、それぞれ出力元と入力先を示している。

このように、CSMLは各々のブロックの情報とブロックがとりうるパラメータ、接続関係を記述しているもので、詳細な数学的モデルはJava言語で定義されている。このCSMLのJava言語のマッピングにはJAXB 2.0[5]を使用している。また、本研究では再利用性、豊富なデータ定義という観点からXML Schema[6]を用いてCSMLを定義した。

```

1:<csml>
2:  <system name="システム名">
3:   <blockList>
4:     <block name="step">
5:       <param name="gain">5</param>
6:       <param name="offset">0</param>
7:     </block>
8:     <block name="integrator" />
9:     <block name="output" />
10:   </blockList>
11:   <connectList>
12:     <connector from="step"
13:               to="integrator" />
14:     <connector from="integrator"
15:               to="output" />
16:   </connectList>
17: </system>
18: <document>
19:   .....
20: </document>
21:</csml>

```

図2: 非線形動的システムのCSML記述例

### 2.2 CSML生成GUI

本システムで提供されるGUIを図3に示す。本システムでは、簡単なマウス操作でブロック線図を作成し、システムの状態および数式モデルの設定することが可能である。このGUIは、

†九州工業大学 情報工学部, KIT

- ブロック線図の作成・編集
- データのセーブ・ロード
- システムの情報の変更

等の機能がある。作成したデータは、本システムに組み込まれている変換プロセッサを用いることで別のデータフォーマットへと変換することが可能である。

本 GUI を用いてブロック線図を作成した後は、我々の研究グループで開発された数値計算ライブラリ NFC(Numerical Foundation Classes)[7] を用いることでシミュレーションが可能となる。

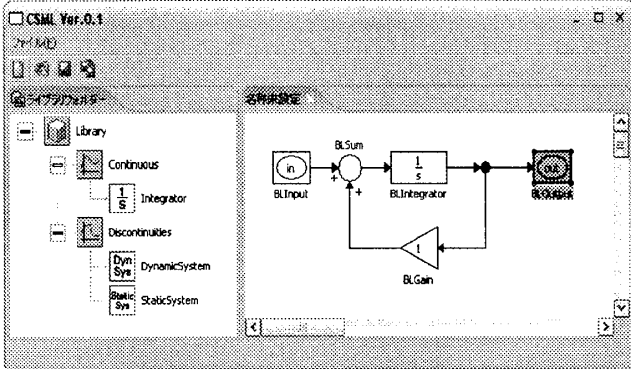


図 3: CSML 生成 GUI 操作画面

### 3. CSML を用いた制御系設計

#### 3.1 シミュレーションプログラム自動生成

モデリングした制御対象をどの様に制御するかという仕様から制御器を設計するが、制御器を設計する際 MaTX や MATLAB といった数値計算言語を使用。これらの数値計算言語の制御器設計やシミュレーションに用いるデータに CSML から変換することが可能である。

図 4 にシミュレーションプログラム自動生成をする処理の流れを示す。CSMLParser は CSML を読み込み、自動生成に必要な情報を取得し、シミュレーションプログラムを生成する。この際、XML Schema を用いて妥当性の検証を行う。XSLT[8] は本システムでは補助的なものとして使用される。生成されたシミュレーションプログラムを、各々の環境で使用することでシミュレーションを行うことが可能となる。

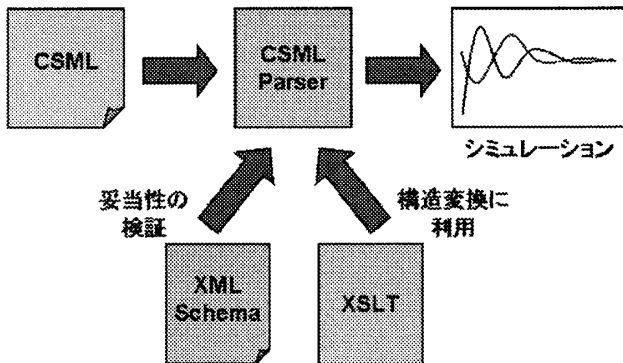


図 4: 自動生成の流れ

### 4. まとめ

本研究では、制御系の統一データフォーマットとして CSML を開発した。また、CSML を他言語のデータフォーマットへ変換・自動生成するシステムを開発した。

現在は比較的簡単な制御システムしか対応していないが、今後は複数のブロックからなる制御システムにも対応する予定である。また、既存のプログラムコードや PDF 等の参考文献からデータを抽出し、CSML を自動生成することが可能となる予定である。現在、XHTML[9] と SVG[10] による Web 上での視覚化を可能とするシステムを開発中であるが、図 5 に示す様に他の有用なツールとの関係を進める予定である。

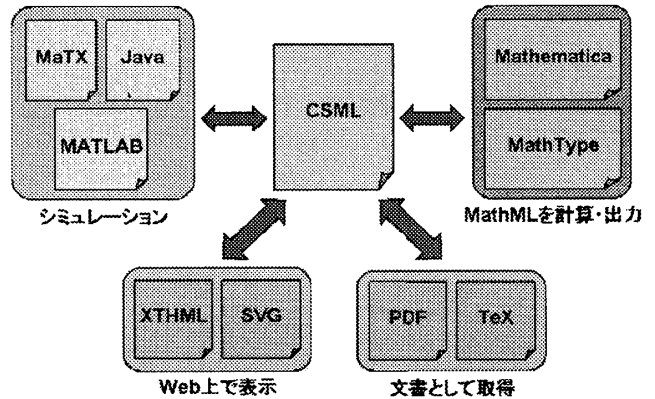


図 5: CSML との関係図

### 参考文献

- [1] 竹井佑介, 古賀雅伸. 制御系の汎用モデルデータ作成のためのモデリング言語 CSML の開発 第 4 回情報科学技術フォーラム pp. 149-150, Sep, 2005
- [2] 古賀雅伸. MaTX による数値計算. 東京電機大学出版局, Feb 2000.
- [3] The MathWorks. *MATLAB*. <http://www.mathworks.com>.
- [4] W3C. *XML* <http://www.w3.org/XML/>.
- [5] Sun Microsystems. *JAXB 2.0* <http://java.sun.com/webservices/jaxb/index.jsp>
- [6] Eric van der Vlist. *XML Schema*. O'REILLY ISBN4-87311-120X, Mar. 2003.
- [7] 松木 毅. Java 数値計算パッケージの開発と制御系設計への応用. 2002 年度卒業論文, Feb 2003.
- [8] W3C. *XSLT* <http://www.w3.org/Style/XSL>
- [9] W3C. *XHTML*. <http://www.w3.org/Markup/>.
- [10] W3C. *SVG* <http://www.w3.org/Graphics/SVG>.