

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

Development of Modeling Language CSML for General-Purpose Model Data Description of Control Systems

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。

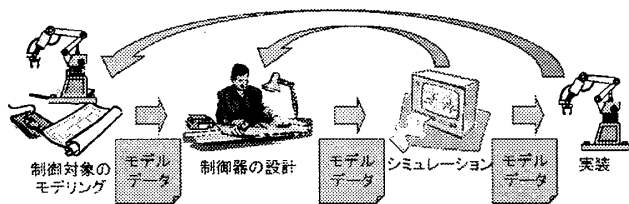


図1: 制御系開発のプロセス

本研究では、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)を開発した。CSMLはXML[1]をベースとしているので、拡張性・再利用性の点において非常に優れている。

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、MATX[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[1]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。

[†]九州工業大学 情報工学部, KIT

2.1 CSML の構造

CSMLは制御システムを基本的なシステムの結合として表現する。ほとんどの連続時間システムと、一部の非線形離散時間システムに対応しており、残りのシステムは現在開発中である。CSMLには各々のシステムに、制御システムとしての情報として、直達項の有無、内部状態の更新への入力の実用性等を持たせてある。また、システムの状態を表す状態方程式と出力を得るための出力方程式を定義してある。ただし、線形システムについては係数行列のみを記述し、非線形システムについては方程式全体を記述する。静的システムについては出力方程式のみを記述し、動的システムについては状態・出力方程式の両方を記述する。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[4]を用いてCSMLを定義した。

2.2 MathML による数式表現

CSMLは、制御システムの数式モデルを表現するためにMathML(Mathematical Markup Language)[5]を採用した。MathMLには、数式のレイアウトを定義するPresentation Markup形式と、数式の意味を定義するContent Markup形式がある。本研究ではCSMLにそれら二つの表現形式を混在させたMixed Markup形式を採用した。そのため、1つのCSMLデータを印刷や数値計算等の多くの目的に利用できる。

2.3 CSML 生成 GUI

本システムで提供されるGUIを図2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、

- 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パー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。

また、我々の研究グループで開発されているシミュレーションツールJamox[6]を用いることで、ブロック線図を作成して即時シミュレーションが可能となっている。

3. 制御系開発プロセスにおけるCSML

3.1 シミュレーションプログラム自動生成

現在、MATX, MATLAB, Javaのシミュレーションプログラムが生成可能である。図3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。

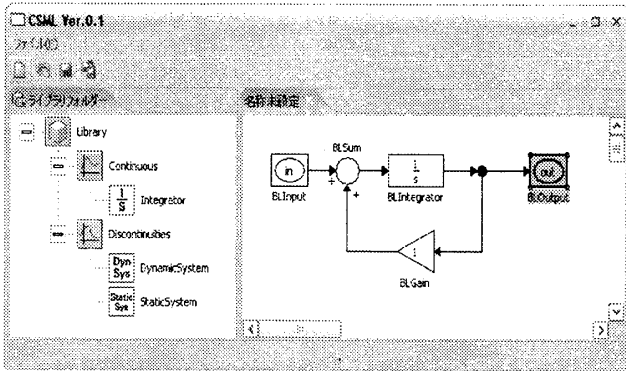


図 2: CSML 生成 GUI 操作画面

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。このツールは、MathML の Content Markup 形式を使用する。この際、XML Schema を用いて妥当性の検証を行う。XSLT 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。

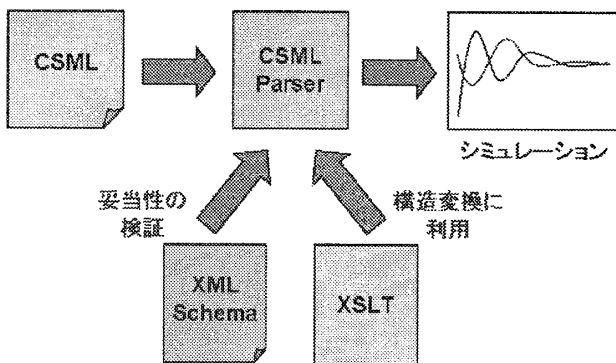


図 3: 自動生成の流れ

3.2 CSML の視覚化

CSML は XML で記述されている為、情報量が多くなると可読性が悪くなる。そこで本研究では視覚化するために CSML で記述された制御モデルを XHTML[7] に変換するツールを開発した。このツールは、MathML の Presentation Markup 形式を利用する。図 4 に CSML から生成された XHTML をブラウザで表示した様子を示す。人間にとって非常に見やすい形式で表示されている。

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 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[7]

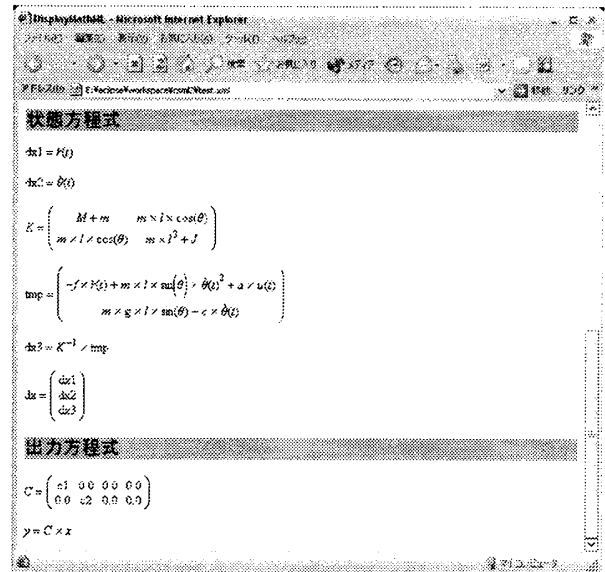


図 4: XHTML での表示

と SVG[8] 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。

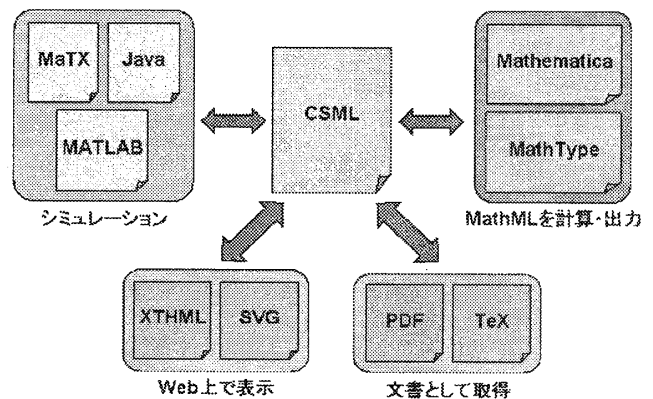


図 5: CSML との関係図

参考文献

- [1] XML. <http://www.w3.org/XML/>.
- [2] 古賀雅伸. MATLAB による数値計算. 東京電機大学出版局, Feb 2000.
- [3] MATLAB. <http://www.mathworks.com>.
- [4] Eric van der Vlist. XML Schema. O'REILLY ISBN4-87311-120X, Mar. 2003.
- [5] MathML. <http://www.w3.org/Math/>.
- [6] 古賀雅伸, 筒井勇介. ブロック線図を用いたダイナミックシステムのモデル化・シミュレーションツール. 第3回情報科学技術フォーラム講演論文集, pp. 611-612, 2004.
- [7] W3C. XHTML. <http://www.w3.org/MarkUp/>.
- [8] SVG. <http://www.w3.org/Graphics/SVG>.