4J-05　　XMLを用いた組込みソフトウェアの動的仕様記述言語EMLの提案

岡本 鉄兵　　小泉 寿男
東京電機大学大学院理工学研究科

1 はじめに

近年，各種電子・電気機器の制御用組込みソフトウェア(Embedded Software)の需要は増大傾向にある．さらにマイクロプロセッサの高性能化に伴い，組込みソフトウェア自体の巨大化および複雑化も進んでいる．このことは，組込みソフトウェアの開発生産性の低下を招き，オブジェクト指向を導入したソフトウェアの再利用による効率化を目指す試みも行われている[1].

本稿では，部品化されたソフトウェアをどのように組合せて必要とする組込みソフトウェアを構成するのか表記する方法として，動的仕様記述言語EML(Embedded Markup Language)を提案する．また，EMLを用いた組込みソフトウェアのトップダウン型設計方式の一例についても述べる[2].

EMLは，XML(eXtensible Markup Language)に準拠している[3]．動的仕様の記述フォーマットにXMLを用いたことで，ネットワーク上での設計情報のやり取りを容易にし，かつ各種開発ツール間で汎用のデータフォーマットの実現をも可能にする．このようにEMLは，Webベースでの組込みソフトウェア開発環境の構築を情報共有の側面から支援することを目標とする．

2 前提条件

EMLで必要とする組込みソフトウェアの動的な仕様を記述するには，予め適応分野毎の交換可能な機能別ソフトウェア部品（コンポーネント）がライブラリ化されているものとする．これは，EMLがそれらコンポーネントをどのように組合せて必要とする組込みソフトウェアを構成しているのか記述する言語であるからである．また，ライブラリ化するコンポーネントの選択には，過去の開発事例から得られたノウハウを反映することで，最適なコンポーネントが選択されているものとする．

A Proposal of EML based on XML for Dynamic Specification Description of Embedded Softwares
Teppei OKAMOTO, Hisao KOIZUMI
Graduate School of System Engineering,
Tokyo Denki University
okamoto@itlab.k.dendai.ac.jp, koizumi@k.dendai.ac.jp

3 EMLの仕様

EMLで記述される組込みソフトウェアの動的な仕様は，以下の4つの要素で構成する．

(1) 状態(state)
　ある時点でのシステムの内部状態
(2) 事象(event)
　システム外部からの刺激（入力信号）
(3) アクション(action)
　特定状態下で，特定事象が発生したとき
　実行される機能または処理
(4) 遷移(transition)
　ある状態から別の状態への移動

これら4つの要素を用いて，『いつ（事象）』，『どこで（状態）』，『何をする（アクション）』と『どうなるか（遷移）』を記述する．

3.1 構造

EMLの構造を**図1**に示す．EMLは，ヘッダ部で適用マイクロプロセッサの指定と便宜的なポート指定を行い，ボディ部で動的仕様の詳細を記述する．

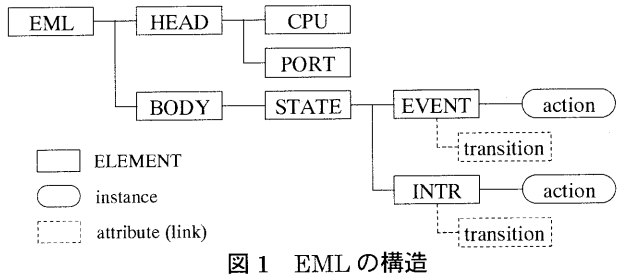

図1　EMLの構造

3.2 記述

3.2.1 事象およびアクション

EMLによる動的な仕様記述の最小単位は，アクション記述である．ある特定事象が発生したとき，どのようなアクションを起こすのか記述する．アクションがEMLにおけるインスタンスとなる．

事象およびアクションの記述

表記　<EVENT　port="1">action A</EVENT>

意味　ポート1で指定された事象が発生するとアクションAを実行する

3.2.2 状態

ある特定状態下に起こり得る全ての事象は，ひとまとめにして記述する．また，状態の階層構造については，タグの入れ子構造で表現する．

状態の記述

表記　<STATE　name="状態X">
　　<EVENT　port="1">action A</EVENT>
　　<EVENT　port="2">action B</EVENT>
　　<EVENT　port="3">action C</EVENT>
　</STATE>

意味　状態Xのとき，起こり得る事象は3パターン

状態の階層構造

表記　<STATE　name="状態X">
　　… 状態Xで起こり得るの事象群 …
　　<STATE　name="状態Y">
　　… 状態Yで起こり得る事象群 …
　　</STATE>
　</STATE>

意味　状態Xの下位状態として状態Yが存在する

3.2.3　遷移

EML における状態遷移記述は，ハイパーリンクを用いる．同一文書内リンクで記述する他，文書間リンクを用いることで，必要とする組込みソフトウェアを機能や構造などの視点で分離させた記述が可能となる．このことは，EML で記述された設計情報を再利用するのに適している．

4　EML の記述例

二階調式照明の動作をEMLで記述した例を示す．

```
<EML version="1.0">
<HEAD>
  <CPU name="Z80"/>
  <PORT id="0">電源オン</PORT>
  <PORT id="1">電源オフ</PORT>
  <PORT id="2">切替スイッチ</PORT>
</HEAD>
<BODY>
  <START type="exclusive">
    <STATE name="消灯">
      <EVENT port="0" xml:link="simple" href="#点灯" >点灯</EVENT>
    </STATE>
    <STATE name="点灯" type="exclusive">
      <STATE name="明るい" type="exclusive">
        <EVENT port="1" xml:link="simple" href="#消灯" >消灯</EVENT>
        <EVENT port="2" xml:link="simple" href="#暗い" >豆電球点灯</EVENT>
      </STATE>
      <STATE name="暗い" type="exclusive">
        <EVENT port="1" xml:link="simple" href="#消灯" >消灯</EVENT>
        <EVENT port="2" xml:link="simple" href="#明るい" >蛍光灯点灯</EVENT>
      </STATE>
    </STATE>
  </START>
</BODY>
</EML>
```

図 2　EML の記述例

5　EML を用いた開発プロセス

我々の考える開発プロセスの例を図 3 に示す．開発者は，EML に対応させた各種 CASE ツールや EML エディタを用いてコンポーネントを適用しながら設計を進める．設計情報は EML 文書として管理され，過去に作成した EML 文書の再利用による設計も可能である．

EML 記述の妥当性は，EML パーサによって検証され，設計情報の閲覧は EML ブラウザによって行う．設計上の誤りが発見された場合は，上流の設計工程にフィードバックして対処する．また，EML ブラウザにて設計上の誤りは無いと判断された設計情報から仕様書を作成することもできる．

各種マイクロプロセッサ対応の EML コンパイラでは，EML 記述された設計情報をコンパイルし，設計工程で適用したコンポーネントを実装させて必要とする組込みソフトウェアを生成する．

このように，EML で設計情報を管理することで，実装ソースプログラム生成や仕様書作成など複数の用途に再利用可能となる．また，EML は XML に準拠していることから，Web ベースの分散開発環境におけるデータ受け渡しや，各種 CASE ツール間での情報共有化の実現に適しているだろう．

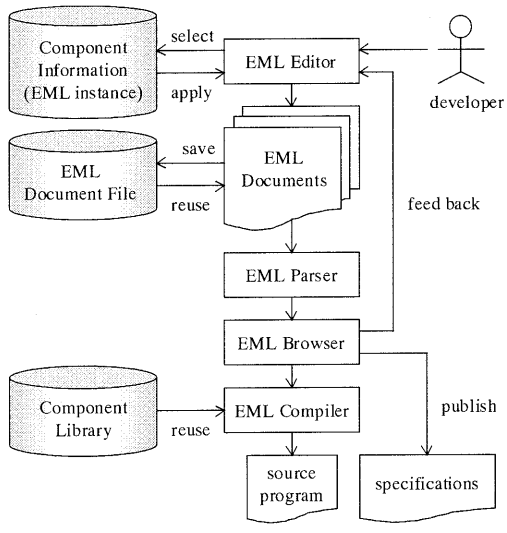

図 3　EML を用いたトップダウン型設計方式の例

6　おわりに

本稿では，組込みソフトウェア開発において動的仕様を記述する XML 準拠の設計言語 EML を提案した．また EML を用いたトップダウン型設計方式の一例についても述べた．今後の課題は，本システムのプロトタイプによる実装を目指す．

参考文献

[1] 川口，岸，門田，“組込みシステムの設計手法－オブジェクト指向を中心にして－”，情報処理，Vol.38，No.10，pp.879-885，1997

[2] 岡本，清尾，小泉，“制御用組み込みソフトウェアのトップダウン型設計方式”，情報処理学会，第 59 回全国大会，4ZC-7，1999

[3] W3C，“eXtensible Markup Language (XML)”，http://www.w3.org/XML/