

Executable UMLの生みの親に聴く xUMLの最新事情

話し手：スティーブ・J・メラー

聞き手：長瀬 嘉秀

日時：2006年5月1日

場所：御殿山メンターグラフィックス本社

INTERVIEW



Executable UML (xUML) の生みの親であるスティーブ・J・メラー (Stephen J. Mellor) 氏に、xUML の最新事情についてインタビューした。インタビューは同氏の著書の翻訳なども行ったテクノロジックアートの長瀬嘉秀氏。

1. モデルから実行可能なコードを生成できる xUML

長瀬：まず、UML と xUML (Executable UML, 図1) の関係について教えてください。

メラー：UML にはいくつかの方向があります。実装のために構造を正確に表現したいという人もいれば、構造的な問題を表現したいという人もいます。こういった異なるアプローチがあるので、UML の範囲は広いものとなっています。そのため、いろいろな製品も出てきています。簡単に言えば、UML の実行可能なサブセットが xUML です。xUML は UML の標準規格に含まれています。

長瀬：今、すでに xUML は UML の仕様の一部ということですね。xUML を使えば、100% 実行可能なコードが生成できるのでしょうか。

メラー：答えはイエスです。xUML を使えばモデルから 100% 実行可能なコードが生成できます。どんな OS 上でも動きますよ。OS なしのシステムでも、チップ上で動かすことが可能です。

長瀬：ハードウェアとして回路にもなりうるというのですが、ハードウェアとしての実装は、具体的に言うとうどのような形になりますか。

メラー：モデル・コンパイラを、ハードウェアを合成できる言語に変換します。たとえば VHDL などです。もちろん、タイミングなど記述できないものもあります。この部分については、我々もいろいろと取り組んでいるところです。

長瀬：ところで、習得するためのコツなどはありますか。

メラー：いちばん効果的な方法は、学ぶのではなく、使うことです。

長瀬：xUML ですが、どのような分野で利用されているのでしょうか。

メラー：大きなところから小さなところまで、いろいろな場面で使われています。小さなもので言うと、血液中の化学物質の濃度を測定して薬剤を送り、正しく送られたか確認するという、薬剤投与機器で使われているのを見たことがあります。その一方で、大規模な通信システムでも使われています。私の専

門はずっと組み込み分野なので、コピー機やファックスなど、いろいろなプロジェクトで使われていることは知っています。

逆に、UML を利用しない分野もあります。たとえば、xUML を使ってコンパイラを作るようなことはしません。コンパイラを作るには、特別な言語が必要になりますからね。

それから、たとえば、大型トラックのように非常に大きなものには xUML を使用しません。油圧がどのくらいになるとエンジンの温度はどのくらいになるとか、そういう項目まですべてリストアップする必要があるからです。

xUML には厳密に定義された形式があるので、ある種の数学であるとも言えます。xUML の背景にある主要な考え方は MDA (Model Driven Architecture, 図2) の背景にある考え方と同じです。それは、ある問題について、モデルをビルドして実装まで落とし込めるという考え方です。

話を xUML に戻すと、xUML は強力なステート・ベースのシステムに適しています。それから、たくさんの並列処理が必要なシステムにも適しています。

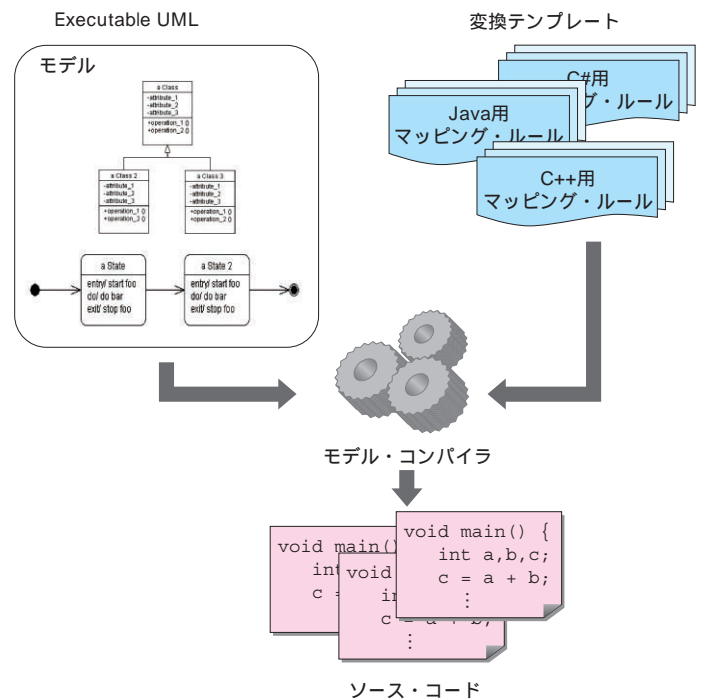


図1 Executable UML の概要

Executable UML とは、「サブセット化した UML」と「アクション言語によって定義されたプラットフォーム(実装技術)に依存しないモデル」に、あらかじめ定義しておいたマッピング・ルールを適用して実行可能なソース・コードを生成する開発手法のこと

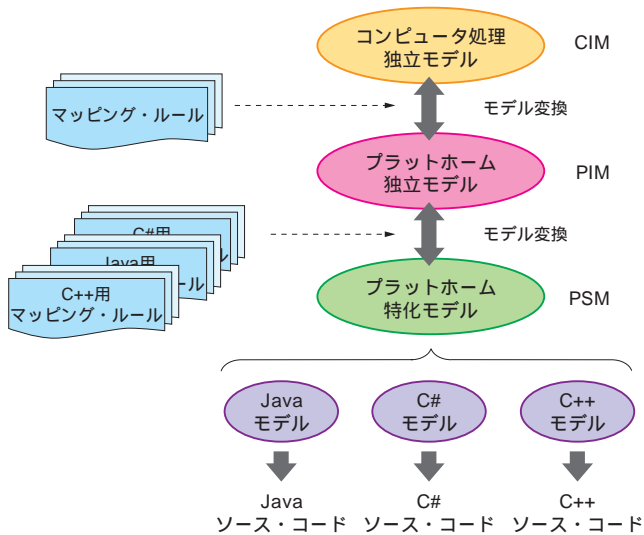


図2 MDA(Model Driven Architecture)の処理の流れ

MDA(Model Driven Architecture)とは、OMGによって提案されたモデルを中心とした開発手法のことである。CIM(Computation Independent Model), PIM(Platform Independent Model), PSM(Platform Specific Model)の視点に分けてモデルを開発することで、ビジネス、コンピュータ処理方式、プラットフォーム(実装技術)の仕様変更に対応できるようになる。なお、これらのモデルは、変換定義言語によってマッピング・ルールを定義することで、相互変換が可能になる

2. UMLの組み込み向け拡張

長瀬：話はUMLに移りますが、UML1.xは組み込み分野で使用するには不十分だったという意見もあります(図3)。それを受けてUML2.0ではダイアグラムが増えましたが、組み込み分野での使用に十分耐えるようになったのでしょうか。

メラー：組み込みシステムに関わる部分でいうと、UML2.0で新しく追加されたダイアグラムはタイミング図です。これが組み込みシステム開発に有効であることはまちがいありません。

それから、UML2.0の仕様は読んでわかりやすいものになっています。仕様も厳しくなっています。



xUMLの生みの親であるスティーブ・J・メラー氏

UML1.xへの要求

- MDAの実現
- 新しい領域のモデリングのサポート(組み込み、SOAなど)

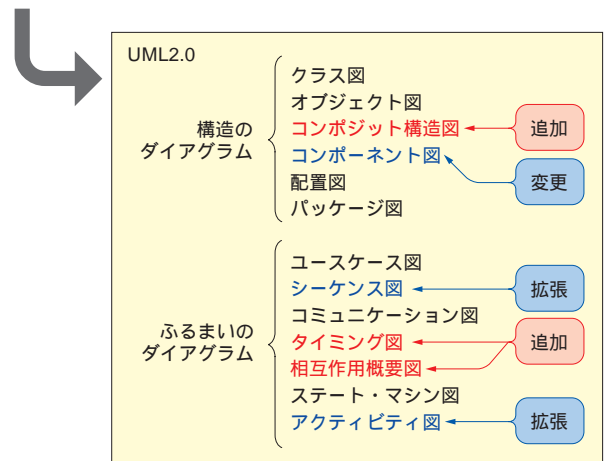


図3 UML1.xとUML2.0の違い

UML2.0ではさまざまな拡張や変更が行われたが、その内容は大きくMDAの実現のためのコア・レベルの見直しと、新しい領域のモデリングをサポートするための表現力の向上に分けられる

長瀬：UMLプロファイルにはいくつか種類がありますが、組み込み関係でUMLプロファイルは使われているのでしょうか。

メラー：使われていますよ。興味深い例もあります。一つ目は日本から始まったSoC(System on Chip)用のプロファイルです。二つ目は、組み込みシステムに特化したプロファイルで、MARTE(Modeling and Analysis for Real Time Embedded system)と呼ばれています。これは現行の1.0x Real Time Embedded Profileに置き換わるものとされています。現在のバージョンでは分散しているので、これを一つにまとめることを目標にしています。SoCは前回の会議で確定しましたが、MARTEは進行中です。ただ、私は個人的にはプロファイルについては懸念を抱いている部分もあります。

3. MDAとxUMLの関係

長瀬：先ほどちょっと出ましたが、MDAとxUMLの関係についてお聞かせください。

メラー：MDAは一般にMDD(Model Driven Development)とも言われています。MDAおよびMDDの考え方の技術的なコアは、xUMLと同じです。言語表記を定めて詳細に定義し、正しいソフトウェアへと落とし込む、それだけです。MDAは形もさまざま、使用する人もさまざまです。

現在、MDAには二つの流れがあります。一つは、完全なモデルをビルドし、100%コードにトランスレートするという考え方です。もう一つはモデルを使用し、モデルを自動的にトランスフォームして、それらにコードを追加するというものです。これはまさに従来型の手法とxUMLの手法の違いです。

しかし、私は後者がMDAで推進されているとは思いません。