モデル駆動開発に向けたマルチメタモデリング

細合 晋太郎†

近年，IoT における組込みシステムとクラウドサービスのように，一つの技術領域のみならず，複数の技術領域をまたぐコンピュータシステムが増加している．このようなシステムの設計においては，複数の領域のメタモデルを統合的に設計することが重要となる．本稿では，複数のメタモデルを用いたモデル駆動開発について述べる．

Multi Meta Modeling for Model Driven Development

Shintaro Hosoai†

Recently, multi domain system like IoT system is increasing. These system needs multi technology area and connecting. In these system designing, Integrated multi metamodeling is important. This paper, we propose multi metamodeling and using for model driven development.

1. はじめに

近年，IoT や CPS など複数の技術領域を横断するコンピュータシステムが増加している．このようなシステムの設計では，特定の技術領域を捉えたメタモデル（ドメインモデル）を定義し，モデル間の矛盾のない設計を行うことが望ましい．またメタモデルに沿ったモデルであればモデル駆動技術[1]を用いてモデルのバリデーションやコード生成に活用することができる．

しかしながらこのように複数のメタモデルを用いた設計では，個々のメタモデル内は整理されるがメタモデル間やメタモデル・モデル間の依存関係や変換関係を俯瞰的に扱うことが難しい．

このようなメタモデル・モデル間の関係を表す方法として，Bezivin ら[2]は Megamodel を提案している．またその実装環境として，AMMA（ATLAS Model Management Architecture）[3]を提供している．例として MDA における PIM（Platform Independent Model）と PSM（Platform specific Model）の間の関係や，UML 以外のモデルとの関係の定義について取り上げている．

また Vignaga ら[4]もまたこのようなメタモデル間やモデル間の管理に GMM（Global Model Management）を提案している．

Megamodel や GMM ではメタモデルに沿う情報を対象としているが，開発に関係する情報はすべてメタモデルとして定義されているとは限らない．仕様書やデータシートなどの入力情報や，ソースコードや設定ファイルなどの出力情報も明確に定義して取り扱うことが望ましい．

本稿ではソースコードや仕様書などのメタモデルに沿わない情報も加えた複合的なメタドメインモデルについて提案する．

†九州大学
Kyushu University

2. 複合メタドメインモデルの定義と表記

複合メタドメインモデルでは，メタモデルやモデル，それらの間の関係に加え，その他の形式のファイルの入出力についても扱う．

図 1 に複合メタドメインモデルのメタモデルと表記法を示す．図内中心はメタモデル，橙色破線で囲んだ部分がそのメタモデル要素に対する表記法を示す．メタモデルは大きく要素（Node）と線（Edge）に分けられ，要素はメタモデル（Metamodel），モデル（Model），テキスト定義（MetaText），テキスト（Text）からなる．また線は，関連・参照（Reference），継承（Generalization），モデル変換（Transform），コード生成（Generation），構文解析（Parsing）となる．線は基本的に短方向で指定し，コードとモデルの双方向の変換が可能な場合はそれぞれの線を記述する．モデル変換，コード生成，構文解析は属性として，実際の処理を行うモジュールへの参照を持つ．

3. 複合メタドメインモデルの利用

複合メタドメインモデルは，メタモデルの俯瞰とモデル変換のプロセス定義の二種類の用途を想定している．メタモデルの俯瞰では，メタモデル（ドメインモデ

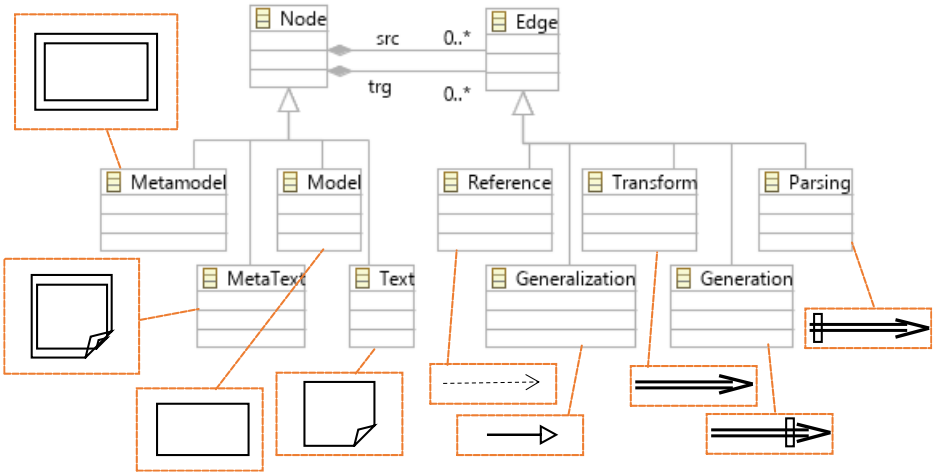

図 1 複合メタドメインモデルのメタモデルと表記法

ル）間の関係と具体的にどのようなモデルが利用されるかを記述することで，システムの全体アーキテクチャを検討する．

プロセスの記述では，メタモデル間の変換の流れを記述し，仕様書などの入力情報から，出力情報であるコードまでの変換の流れを記述する．

プロセス記述はモデル変換の定義フローである MWE（Model Workflow Engine）[5]へ変換し，MDD のモデル変換に用いる．モデル変換やコード生成，構文解析はそれぞれ MWE のモジュールに関連付けており，具体的な変換処理はモジュール内で行う．

現時点ではモデル変換やコード生成は Xtend 言語[6]での定義を想定し，構文解析は Xtext[6]による DSL 定義や POI[7]などの Office ファイル操作ライブラリを利用することを想定している．

複合メタドメインモデル上でメタモデルやメタモデルを用いたプロセスを定義することにより，システムの複合的なモデルを俯瞰できるとともに，開発プロセス自体も開発資産として運用，再利用することが期待できる．

4. おわりに

本稿では，複数のドメインモデルや仕様書，ソースコードなど開発に関する入出力情報も含めた複合的なメタドメインモデルについて提案し，その利用方法について述べた．ワークショップでは，本モデルの他の設計方法との関連や設計プロセスについて議論を行いたい．

参考文献

[1] Schmidt, Douglas C. "Model-Driven Engineering." IEEE Computer 39, no. 2 (2006): 25--31.

[2] Bézivin, Jean, Jouault, Frédéric and Valduriez, Patrick. "On the Need for Megamodels." Paper presented at the meeting of the Proceedings of the OOPSLA/GPCE: Best Practices for Model-Driven Software Development workshop, 19th Annual ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, 2004.

[3] Bezivin, J, Jouault, J, Touzet, D. "An Introduction to the ATLAS Model Management Architecture" Research Report, LINA (05-01)

[4] Vignaga, Andrés, Jouault, Frédéric, Bastarrica, María Cecilia and Brunelière, Hugo. "Typing artifacts in megamodeling." Software and System Modeling 12, no. 1 (2013): 105-119.

[5] The Eclipse Foundation, “Modeling Workflow Engine”,http://wiki.eclipse.org/Modeling_Workflow_Engine_(MWE) (accessed 2016-01-10)

[6] itemis, “Xtext”, http://www.eclipse.org/Xtext/ (accessed 2016-01-10)

[7] The Apache Software Foundation, “The Apache POI Project”, https://poi.apache.org/ (accessed 2016-01-10)