ソフトウェアシステムのプロトタイピング環境の構築

1R－3

安間 その美　高橋 大輔　上田 賀一

茨城大学 工学部 情報工学科

1 はじめに

システムが大規模化・複雑化するほど要求者と開発者との思考の隔たりが大きくなり，要求の正確な伝達が困難になる．そこでシステムの実行可能なモデル(プロトタイプ)を早期に作成し，その振舞いを確認することで要求を明確にするプロトタイピングの必要性が高まっており，プロトタイピングを支援するCASEツールも存在する．しかしそれらのほとんどは状態遷移図などシステムのある特定の側面のプロトタイピングを支援するように特化されている [1]．様々なドメインを統合的に扱える環境としてSmalltalkが挙げられるが [2]，Smalltalkはプログラミング環境をも包含しており，一般的なドメインユーザに扱いやすい環境とはいえない．

本研究ではプロトタイピングを容易にするために，システムを構成するコンポーネントをドメイン毎に予め用意し，それらを用いてシステムのプロトタイピングを視覚的に行なうことを考える．つまりコンポーネント開発者とプロトタイプ開発者を分離することで，ドメインユーザがプロトタイピングを行なうことを可能とする．さらに様々なドメインのプロトタイピングを支援するために，プロトタイピングツールをコンポーネントに関する定義から生成することも考える．これらを実現するための基盤機構を述べる．

2 基盤機構の基本構造

本基盤機構ではプロトタイピングにおけるオブジェクト指向アプローチの有効性から，オブジェクト指向を取り入れており，全ての要素をオブジェクトとして扱う．

また，システム開発時に本基盤機構ではコンポーネント開発者とプロトタイプ開発者を単一の環境で支援するために，表現系をメタメタモデル，メタモデル，モデルと階層化した [3]．メタメタモデルは我々が提供するが，メタモデルはコンポーネント開発者(以下，メタモデル開発者)が，モデルはプロトタイプ開発者(以下，モデル開発者)が作成する．

2.1 メタメタモデル

メタメタモデルはERモデルにフィールドの概念を追加したものである．メタメタモデルのコンポーネントは以下の3つである†．

ENTITY
　システムを構成する実体オブジェクト

RELATIONSHIP
　ENTITYやFIELD間の関連オブジェクト

FIELD
　ENTITYやRELATIONSHIPが入る場オブジェクト

これらのコンポーネントの関係を以下に示す．

- ENTITYとFILEDは継承可能である
- RELATIONSHIPは2項関連である
- FIELDは集約オブジェクトとして用いることが可能である
- FIELDを場とみなした場合FIELD同士の重なりを許す

またコンポーネントはオブジェクトなので，それぞれ属性やメソッドを持つ．視覚化に関して，ENTITYとFIELDはノードで，RELATIONSHIPはアークで表示するため，その表示に関する情報も属性として持つ．また視覚化に関するメソッドもメタメタモデルコンポーネントに持たせることにより，メタモデル作成者は視覚化に関する記述を軽減される．

2.2 メタモデル

メタモデルはモデルを構成するコンポーネントとその間の関連をメタメタモデルに基づいて記述される．また各コンポーネントが持つ属性やメソッドについても定義する．

メタモデル自身もFieldとみなせるので，FIELDを用いて定義される．そのためメタモデルであるField

Design and Implementation of Prototyping Environment for Software System

Sonomi Anma, Daisuke Takahashi, Yoshikazu Ueda
Department of Computer and Information Sciences, Faculty of Engineering, Ibaraki University

† メタメタモデルのコンポーネントは全て大文字で，メタモデルのコンポーネントは頭文字のみ大文字で，モデルのコンポーネントは全て小文字で表す

を含む Field を定義することによって, メタモデル内にメタモデルを含む階層的なメタモデルを作成することもできる.

2.3 モデル

モデルはシステムのプロトタイプそのものであり, メタモデルに基づいて記述される. モデルコンポーネントの振舞いはメタモデルで規定されているため, モデル開発者は新たに振舞いを記述する必要がない. またメタモデルを変更すれば, 様々なドメインのモデルを構築することが可能となる. そのためドメインユーザでも簡単にモデル作成を行うことができる.

3 プロトタイピング環境

以上を踏まえてメタモデル, モデルを視覚的に構築, テストするプロトタイピング環境を構築した. 構築にあたり実行時に動的な振舞いを柔軟に記述できるよう, C++等のコンパイラ言語ではなくインタプリタ言語である incrTcl 上で構築した. 各階層のコンポーネントは incrTcl のオブジェクトとして実装されている. しかしコンポーネントの具象化にクラス-インスタンスを用いているため, メタメタモデルからメタモデル, およびメタモデルからモデルへのスムーズな移行が難しくなっている. これについてはメタ階層を持つ新たな言語の開発を研究室内で進めており, 言語の完成を待っている状態である.

3.1 メタモデルの作成

メタモデル作成ツールは我々が予め提供するメタメタモデルのインスタンスとして生成される. メタモデル作成者はコンポーネントの属性, 例えば Entity であれば表示用のアイコン, メッセージを受信した際のアクション, その他固有の属性を, Relationship であれば線種, 矢印の種類を設定する.

また, モデル作成時のコンポーネント間の関連 (多重度) の定義も行なう.

3.2 モデルの作成・実行

モデル作成・実行ツールはメタモデル作成者が作成したメタモデルのインスタンスとして生成される. モデル作成者はモデルコンポーネントを視覚的に配置し, 属性に値を代入することでモデルを生成することができる. また, コンポーネント配置時にメタモデルで記述される構造の制約がチェックされる.

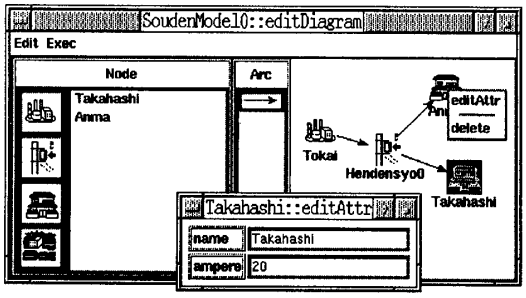
生成したモデルはツール上で動作を視覚的に確認することができる. この際, 実行時の属性に対する制約がチェックされる.

4 適用例

プロトタイピング環境上で, 電力配送のシミュレータを作成した. まず電力配送のシミュレートに必要なメタモデルを作成する. メタモデルのコンポーネントは以下のものである.

発電所	電力の供給元
変電所	電力を下流の施設に分配する
家	電力を消費する
電線	各施設を接続し, 電力を伝達する
都市	内部に各施設を持ち, 外部から入って来る電力を内部の施設に供給する

発電所, 変電所, 家は Entity, 電線は Relationship, 都市は Field である. 作成したメタモデルより電力配送シミュレート用のモデル作成ツールが生成される. このモデル作成ツールを用いて実際にモデル作成している画面を以下に示す. また作成したモデルを用いてシミュレーションを行なうことも可能である.



電力配送モデル

5 おわりに

本研究ではメタ階層に基づくプロトタイピング環境を設計・構築した. 今後は今回構築した環境上で実用例を作成するとともに, メタ階層を有するインタプリタ言語上で同様の環境を構築し, その有用性について検証していくつもりである.

参考文献

[1] フィル・サリー著, 本位田真一監訳 : “オブジェクト指向モデリング”, 日経 BP 出版センター (1995)

[2] 所真理雄, 松岡聡, 垂水浩幸 : “オブジェクト指向コンピューティング”, 岩波書店 (1993)

[3] 上田賀一, 安間その美, 高橋大輔 : “メタ階層に基づくモデルベースソフトウェア開発基盤の提案”, ソフトウェア工学の基礎 II, pp.11-20, 近代科学社 (1995)