メッセージフローに基づくエンドユーザ主導型アプリケーション開発技法*

石樽 久嗣　中所 武司†
明治大学大学院理工学研究科基礎理工学専攻情報科学系‡
{hisashi, chusho}@cs.meiji.ac.jp

1 はじめに

インターネットやイントラネットに接続されたパソコンの普及とともに，業務の専門家が自ら情報システムを構築する必要性が高まっている．たとえば，基幹業務システムのように投資に対する効果が明確なものは，従来のように情報処理の専門家が開発すればよいが，小さな部門あるいは個人の業務を対象とするものや頻繁に機能変更が発生するものには従来の開発方法はなじまない．このような動向に対応して，我々は，エンドユーザが自分に必要なアプリケーションを自ら開発し，自ら保守するための開発環境 **M-base** を実現した [1].

本稿では，その技術的特徴である動的モデルからの静的モデルの自動生成，３種類の図式表現を併用したシミュレーション技法，およびルールを用いたメッセージフロー制御方式について述べる．

2 モデリング&シミュレーション

2.1 モデリング手順

従来のオブジェクト指向分析設計技法 [2] のように最初にクラス間関連図を作成する方法は業務の専門家には難しいので，インスタンスベースのモデル構築を基本とした．すなわち，開発の初期段階で業務の動的振る舞いを明確にした後，必要に応じて静的構造を定義するという手順をとる．

2.2 動的モデルの定義

動的モデルの構築はユースケース [3] 単位で行う．一つのシステムを複数のユースケースに分割することにより，モデリングの複雑さを回避できる．また，ユースケースは利用者の視点によるシステムの使用例であり，業務の専門家にとって理解しやすいものである．

ドメインモデルにおける１つの業務を擬人化したオブジェクトに割り当てることにより，メタファベースのモデル化を行う．動的モデルの構築は，ビジュアルツールを用いて行う．具体的には，図 1 に示すように，オブジェクトをアイコンとして配置し，メッセージのやりとりを矢印で表現する．さらに各メッセージにおける属性（メッセージ名・パラメータ）を定義する．図 1 は，会議開催システムの動的モデルの例である．会議開催の指示が出されると，事務局は会議室および備品の予約を行うと共に，会議のメンバに会議開催通知を出し，出欠を管理するという業務を表している．

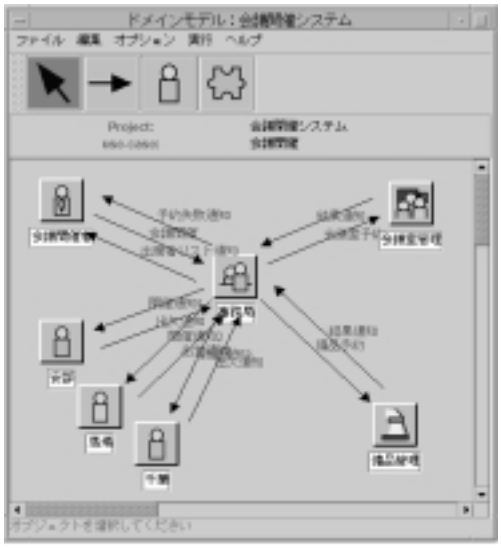

図 1: 会議開催システムのドメインモデル

既存のビジュアルツールでも，オブジェクト間を結んでアプリケーションを構築していくものがある．しかし，それらはデータベースから読み込んだデータをグラフ表示するといった典型的な処理には向いているが，多種多様な業務をモデル化の段階から支援してアプリケーションを構築するようにはなっていない．

2.3 オブジェクト内部の詳細化

次に必要に応じて，個々のオブジェクトに固有のプロパティ，メソッドを定義する．しかしプログラミング知識の十分でない業務の専門家にとって，動的側面

*Enduser-Initiative Application Development Based on Message Flows
†Hisashi ISHIGURE and Takeshi CHUSHO
‡Computer Science Cource, Major in Sciences, Graduate School of Science and Technology, Meiji University.

から静的側面への視点の切り替えには大きなギャップがある．そこで **M-base** では，先に定義した動的モデルから静的モデルを導き出すことを可能にした．具体的には，メッセージ属性で定義したパラメータをプロパティに，メッセージ名をメソッドにそれぞれ割り当てることにより，静的構造のスケルトンを生成する．その後，生成されたメソッドに対して業務の用語を用いて処理内容を記述し，必要に応じてスクリプティングへと移っていく．この機能の実現により動的モデルから静的モデルへの変換をシームレスに行うことができ，エンドユーザの負担を軽減できる．

2.4 シミュレーション

シミュレーションは，作成した業務フローの動作確認のために行う．シミュレーションもユースケース単位で行う．業務フローの各メッセージをステップ実行することにより，エンドユーザによるシミュレーションを容易にした．

シミュレーションは次の３つの図式表現を用いて行う．

- コラボレーション図（図 1）
- シーケンス図（図 2 左）
- プロパティシート（図 2 右）

コラボレーション図は，オブジェクトどうしの協調関係を示すものである．ユーザが作成したモデル上で動作の確認が行えるため，直感的に理解しやすい．また，並列処理を表現するのにも優れている．

シーケンス図は，オブジェクト間通信を時系列で表示するものである．システムが動作を開始すると，起動されたメソッドの処理内容を業務の用語で表示するため，エンドユーザにも判りやすい．

プロパティシートは，実行中のオブジェクトプロパティの値を表示または変更するためのツールである．実際の値を確認できることで，より詳細な検証を行うことができる．

このような３種類のシミュレーション方法の併用により，エンドユーザ自身が自ら作成したモデルの検証を行える．

3 メッセージフロー制御方式

アプリケーションの実行時において，あるメソッドを実行した後，次にどのオブジェクトへメッセージを送信すべきかを動的に決定しなければならない場合がある．そこで **M-base** では，柔軟性や記述容易性を考慮してルールを用いてメッセージフローを制御する．

ルールはオブジェクトの各メソッドに対応させて記述する．ルールは条件部と行動部から構成される．条

図 2: シーケンス図（左側）とプロパティシート（右側）

件部はプロパティの値による論理演算の組合せにより表現し，行動部は次に実行するメッセージ名を指定する．例えば，事務局の会議室予約結果受理メソッドにおけるルールは次のように記述できる．

1. 会議室予約結果=” 失敗” ならば 予約失敗通知
2. 会議室予約結果=” 成功” && 備品の有無=” 必要” ならば 備品予約
3. 会議室予約結果=” 成功” && 備品の有無=” 不必要” ならば 会議開催通知

既存のツールにおけるコンポーネントの連携としては，スクリプト言語を用いる方法や２つのコンポーネントのイベントとメソッド間を結ぶ方法などがある．前者は複雑な処理を記述できるが習得が難しい．後者は分かりやすいが機能が限定されるなどの欠点をもつ．

ルールは，構造が単純でありかつ柔軟に対応できるため，エンドユーザによる多種多様な業務のモデリングに有効な手段であるといえる．

4 おわりに

本稿では，エンドユーザによるアプリケーション開発技法について述べた．今後は，様々なアプリケーションに適用して実用性を検証する．

参考文献

[1] 中所武司, 小西裕治, 松本光由: メッセージフローに基づく分散協調型アプリケーション開発技法, 情報処理学会論文誌, Vol. 40, No. 5, pp. 2387–2396 (1999).

[2] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W.: *Object-Oriented Modeling and Design*, Prentice-Hall (1991).

[3] Jacobson, I., et al.: *Object-Oriented Software Engineering*, Addison-Wesley (1992).