

エンドユーザによるソフトウェア合成のためのモデル化支援インタフェース*

古川 慈之 徳永 仁史 手嶋 吉法 澤田 浩之 松木 則夫
産業技術総合研究所 デジタルものづくり研究センター

1 はじめに

1.1 背景と目的

我々は、エンドユーザが自ら必要なソフトウェアを作成すること(エンドユーザ開発 [1])を目標として、ソフトウェア開発基盤 MZ Platform[2][3]の研究開発を行ってきた。MZ Platformの開発によって、ソースコードを記述せずにソフトウェアを構築することは可能となったが、ソフトウェアに対する要求の抽出やそれに基づく設計は支援の範囲外であった。現在は、本来のエンドユーザ開発の実現に近づくために、エンドユーザが処理のプロセスをモデル化することで自動的にソフトウェアを合成する研究を進めている [4]。本報告は、エンドユーザが入力するモデル記述とその入力インタフェースに求められる機能について述べる。

1.2 関連技術

ソフトウェアの構築や保守の効率化への貢献が期待される技術の一つとしてモデル駆動開発(MDD)が挙げられる。ソースコードを書く代わりにモデルを記述することでソフトウェアを開発する手法である。近年はMDDと表現される対象範囲が広がっているが、元々はソースコードと一対一に対応するモデルを記述することを指す。エンドユーザ開発に求められるように少ない入力という観点では、ドメイン特定モデリング(DSM)に対する期待が大きい。モデルを基本とした開発という意味ではMDDの一種であるが、ドメインを特定することでモデルの共通部分を大きくして少ない修正で個別のソフトウェアを開発するための技術である。これらは一般的にはソフトウェア開発者を対象とした技術である。一方、ユーザが業務プロセスを記述してソフトウェアを動作させる製品としてワークフロー管理システムがあり、そのためのプロセス記述法の一つとしてBPMNがある。こちらはエンドユーザ側に寄った技術とも言えるが、対象はタスクの配分や進捗管理が主であり、ユーザのタスクそのものとは切り離されている場合が多い。

*An Interface for End-User Modeling and Software Generator
Yoshiyuki Furukawa†, Hitoshi Tokunaga†, Yoshinori Teshima†,
Hiroyuki Sawada† and Norio Matsuki†

† National Institute of Advanced Industrial Science and Technology (AIST), DMRC, 1-2-1 Namiki, Tsukuba, Ibaraki 305-8564

2 モデル記述によるソフトウェア自動合成

2.1 モデル記述方式

本研究で用いるモデル記述方式の概要を図1に示す。処理のプロセスは開始ノードから終了ノードまでの一連のタスクで示し、まとめてタスクフローと呼ぶ。各ノードは有向エッジで接続され、制御ノードとして分岐が存在する。一連のタスクはグループ化して階層タスクとして定義することもできる。この場合、階層化された一連のタスクは詳細フローと呼び、タスクフローと同様に記述する。タスクフローを構成する要素に関するクラス図を図2に示す。

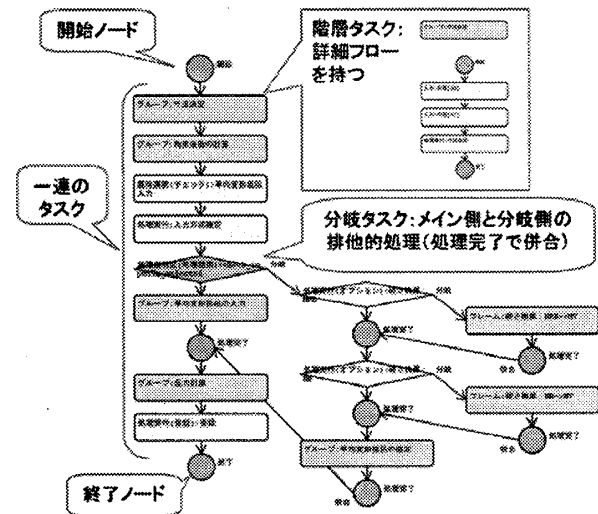


図 1. モデル記述方式の概要

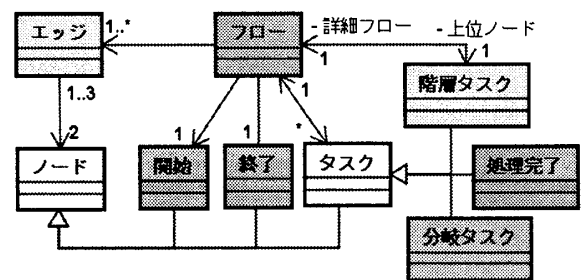


図 2. モデル記述要素のクラス図

2.2 ソフトウェア合成と MZ Platform

記述されたタスクフローからソフトウェアを合成する機能および、合成されるソフトウェアは全て MZ Platform を用いて実現している。MZ Platform の説明は既報 [2][3] に譲るが、特徴を簡単に述べると、コンポーネントの接続でソフトウェアを構築するためのフレームワークであり、その接続が全て統一したオブジェクトを用いて実現されている。そのため、ソースコードを書かずにソフトウェアを構築することが可能であり、動作している状態で動的に機能の追加や修正が可能である。この特徴により、エンドユーザが記述したタスクに対応したコンポーネントの接続を動的に作成し、実行することが可能である。記述されたプロセスに対応するコンポーネントの生成を MZ Platform 上で定義された変換器によって保証し、かつコンポーネントの接続を自動的に行うことによって MZ Platform 上のアプリケーションとして合成する [4]。

ソフトウェアの自動合成を実現するために、本研究では対象とするエンドユーザを個人または少人数によるグループまでとし、ソフトウェアに要求される機能をそのエンドユーザが責任を有する閉じたプロセスを実行することと定義する。このようなプロセスの例としては、設計や意思決定などの知的作業を念頭に置いており、合成されるソフトウェアは設計支援や意思決定支援を実現するツールとなる。つまり、ソフトウェアによる支援が有益な高度に知的な作業でありながら、パッケージソフトウェアでは機能が不足するような業務である。当然ながら、実際の業務プロセスは閉じたプロセスだけではなく他者とのインタラクションのもとに成り立つものが多いが、本研究ではモデル化の時点でインタラクションを排除することで対象外とする [4]。

2.3 実行可能なモデルの維持

本研究では、プロセスのモデル化はタスクフローの記述によって行い、個々のタスクはコンポーネントに対応させている。そのため、タスクフローの記述が正しければ、それに対応した何らかのソフトウェアが常に合成可能な状態が維持される。このような環境を提供することで、エンドユーザはソフトウェアの動作を確かめながらプロセスのモデルを修正することができる。エンドユーザはソフトウェアに対する自らの要求を常に明確に認識しているわけではないため、このような方式はエンドユーザ開発に対して有効である。

上記のように正しいタスクフロー記述を維持するために、ユーザインタフェース側での制約が必要となる。図 3 にその一例を示す。フロー形式の図を編集できる図形描画ツールは数多く存在するが、丸や四角などの図形を個別に書いてそれらを矢印で接続するという操作は途中で正しくないフロー記述の存在を許すことになる。本研究で構築した環境ではタスクや分岐の挿入時に必要なタスクや有向エッジをまとめて挿入し、常に正しいタスクフローの記述を維持する。

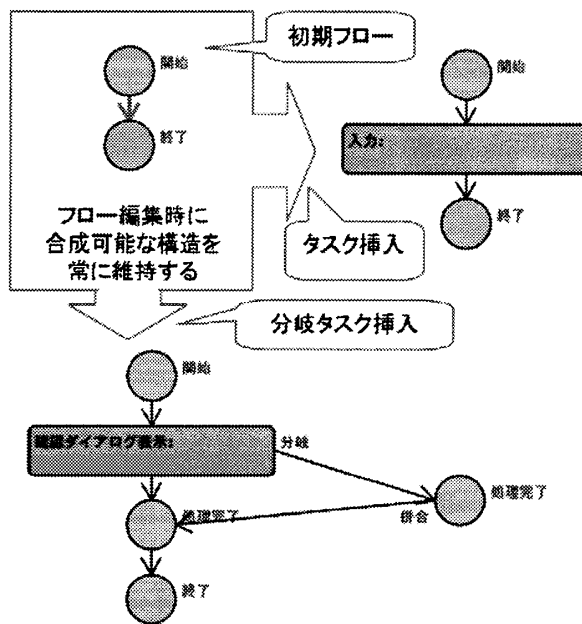


図 3. インタフェースによる操作の一例

3 まとめ

本研究は、エンドユーザ開発の実現を目標として、処理のモデル化に基づくソフトウェア合成と、そのために必要なモデル記述方式およびインタフェースについて述べた。今後もエンドユーザ開発に貢献する技術について研究を継続する予定である。

謝 辞

本研究は、NEDOプロジェクト「ものづくり・IT 融合化推進技術の研究開発」および、「中小企業基盤技術継承支援事業」の一部として行われた。図中に示したタスクフローは産総研の篠崎吉太郎氏および江端幹夫氏の協力によるものである。ここに感謝の意を表す。

参 考 文 献

- [1] Sutcliffe, A., Mehandjiev, N., "End-User Development", *Communications of The ACM*, 47(9), pp.31-32, 2004.
- [2] Sawada, H., Matsuki, N., Tokunaga, H., Furukawa, Y., "A Manufacturing Software Development and Operation Framework "MZ Platform" and its applications in Industry", In *Proc of Advanced Engineering Design*, 2004.
- [3] 古川慈之, 澤田浩之, 富澤拓志, 松木則夫, "MZ Platform: イベント駆動型コンポーネント指向開発環境を用いたエンドユーザ開発への試み", *情報処理学会第 68 回全国大会講演論文集 (1)*, pp.199-200, 2006.
- [4] 古川慈之, 徳永仁史, 澤田浩之, 松木則夫, "エンドユーザによるプロセスのモデル化に基づくソフトウェア合成", *情報処理学会第 69 回全国大会講演論文集 (1)*, pp.177-178, 2007.