

エンドユーザ開発へ向けた ソフトウェア自動合成ツール

澤田 浩之^{†1} 徳永 仁史^{†1} 古川 慈之^{†1}
手嶋 吉法^{†1} 松木 則夫^{†1}

UML による Java ソースコードの生成など、近年、ソフトウェアを自動合成する技術や製品の研究開発が進められている。これらはソフトウェア開発の効率化を図る上で有用ではあるものの、専門のソフトウェア技術者による使用を前提としたものであり、情報技術 (IT) の専門家ではないエンドユーザが利用することはきわめて困難である。我々は、企業の IT 化を推進するためにはエンドユーザ開発が有効との認識から、高度な IT 知識を必要とせずに利用可能なソフトウェア自動合成ツールの研究開発を進めている。このツールは、企業の業務における情報の流れを記述した業務フローモデルに基づいてアプリケーションを生成するものであり、生産管理システム等の業務管理系ソフトウェアの作成に適している。

An Application Software Generator toward End User Development

HIROYUKI SAWADA,^{†1} HITOSHI TOKUNAGA,^{†1}
YOSHIYUKI FURUKAWA,^{†1} YOSHINORI TESHIMA^{†1}
and NORIO MATSUKI^{†1}

Recently, research and development on techniques and tools that automatically generate software, such as Java source code generation by UML, has been conducted. Though they are useful for professional software engineers in improving efficiency of software development, it is quite difficult for end-users without sufficient knowledge about information technology (IT) to make use of them. We consider that End-User Development is effective in introducing IT systems into companies, and have been developing a software generator which can be used without professional IT knowledge. This tool generates a software application based on the *work flow model* that describes the flow of information among working units in a company. This tool is suitable for building up work control software applications including a production control system.

1. はじめに

製造業等における企業にとって、業務改革を推進し競争力の維持向上を図るためには、IT 化への取り組みが不可欠であると広く認識されている。しかし、特に中小企業では、IT システムの開発や導入、運用のための負担が非常に大きく、IT 化を進められないというケースが多く見られる。このような問題を解決するためには、エンドユーザである企業の従業員が高度な IT 知識を必要とせずに、自ら IT システムを構築・運用できるような環境を整えることが有効であると考えられる。

このような観点から、我々はエンドユーザ開発¹⁾の実現を目標としてソフトウェア開発実行ツール MZ Platform²⁾の研究開発を行い、2004 年より一般への公開を行っている^{3),4)}。MZ Platform は、プログラムを書く代わりにコンポーネントと呼ばれるソフトウェアの部品を組み合わせることによって IT システムを作り上げるツールである。バーコードや QR コードの作成・印刷、データベースとの連携など、専門家でないとは通常は開発が難しい機能が標準のコンポーネントとして用意されているので、従来よりも容易に IT システムの開発を行うことができる。

MZ Platform は、すでに製造業数社において実運用されるなど一定の成果を挙げている。その一方、IT システムの設計や、必要なコンポーネントの選択および組み合わせ方は依然としてユーザに委ねられているため、プログラム作成に先立つこれらの作業へのサポートが求められている。

本研究報告では、IT システムの設計からプログラム作成までをサポートするソフトウェア自動合成ツールについて説明し、そのプロトタイプシステムを紹介する。まず次節では、企業における IT 化の手順について説明した後、その中で本ツールの位置付けを述べる。次に第 3 節において、自動合成ツールの構成および機能について述べ、第 4 節では適用事例を示す。

2. 企業における IT 化の手順と自動合成ツールの位置付け

図 1 に示すように、通常、企業における IT 化は、業務分析、アプリケーション設計、プ

^{†1} 産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

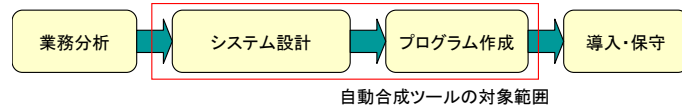


図 1 IT 化の手順
Fig. 1 Process of Introducing IT Systems

プログラム作成，導入・保守の手順で進められる。

業務分析 業務における問題点の特定と改善案の提示を行う作業である。対象となる業務を俯瞰的に捉え，その内容と手順を整理することが必要となる。対象業務に関する豊富な知識と深い理解が必須であり，相応のキャリアを持った従業員の関与が不可欠である。
システム設計 業務分析の結果として得られた改善案を，コンピュータ上で動作するプログラムとして実現するためのモデルを構成する作業である。具体的には，フローチャートの作成や UML⁵⁾ を利用したクラス図の作成などがこれに当たる。

プログラム作成 システム設計の結果をもとに，プログラム言語を用いてコンピュータ上で動作するプログラムを作成する作業である。この作業に用いられる既存ソフトウェアとしては，Eclipse⁶⁾ や Visual Basic⁷⁾ などの開発ツールがよく知られている。MZ Platform もまた，プログラム言語を直接操作するものではないが，この種のソフトウェアとして分類される。このほか，前述の UML で記述されたモデルからプログラムのソースコードを生成するソフトウェアなども市販されている。

導入・保守 作成されたプログラムを業務展開し，業務形態の変更や現場の作業実態に合わせたプログラムの修正ならびにトラブルへの対処を行う作業である。IT 化を進める上で見落とされることが少なくないが，これらの負担を予め正確に把握しておくことは重要である。

図 2 は，業務分析からプログラム作成までの手順の例を示したものである。社内の業務を分析することにより，情報やデータの流を含めた業務フローが明確化される。次に，業務フローに基づいて IT 化の対象を特定し，システム設計が行われる。図 2 では，生産計画から加工・組立に至る過程が IT 化の対象となっている。そして，システム設計の結果に基づいてプログラムが作成される。

本研究報告で述べる自動合成ツールは，これらのうち，システム設計とプログラム作成を対象とする。すなわち，自動合成ツールは，業務フローモデルの記述機能とそのモデルに基づいたプログラム作成機能を持つ。

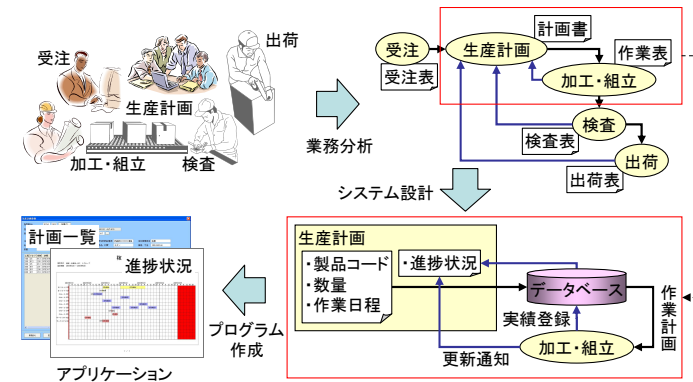


図 2 IT 化の例
Fig. 2 Example of Introducing IT Systems

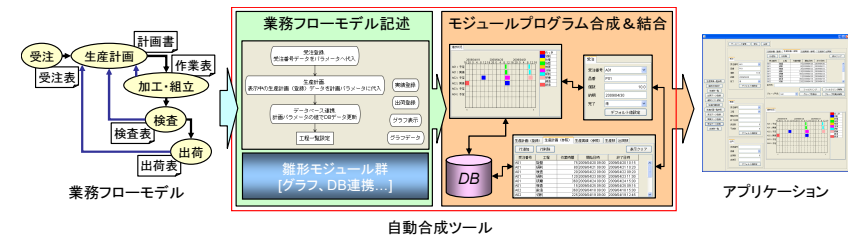


図 3 自動合成ツール機能概要
Fig. 3 Outline of the Function of the Software Generator

3. 自動合成ツールの構成および機能

3.1 機能概要

自動合成ツールの機能概要を図 3 に示す。ユーザは，業務分析の結果として得られた業務フローモデルを自動合成ツール上で記述する。自動合成ツールは，プログラムの基本構成要素となる雛形モジュールを持っており，業務フローモデルで指定された各モジュールの仕様に基いて，雛形モジュールからモジュールプログラムを合成する。そののち，合成されたモジュールプログラムを結合してアプリケーションを生成する。

自動合成ツールは MZ Platform のアプリケーションとして実装されており，生成される

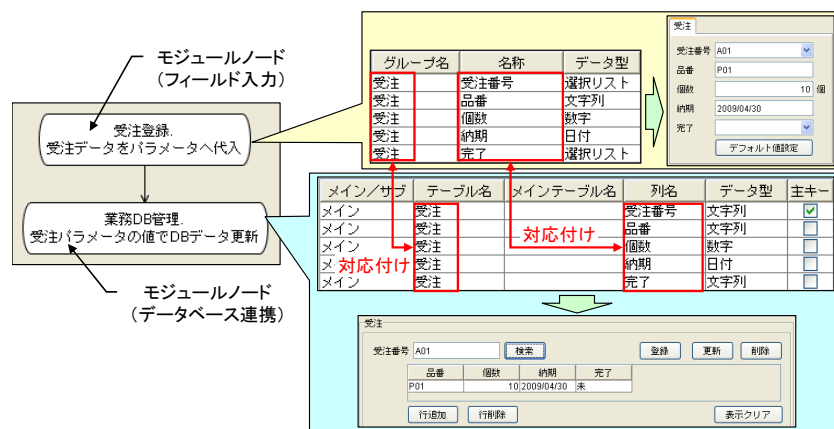


図 4 業務フローモデルとモジュール合成
Fig. 4 Process Flow Model and Module Formation

ソフトウェアもまた、MZ Platform のアプリケーションである。MZ Platform の概要については、付録 A.1 節を参照されたい。

3.2 業務フローモデル

業務フローモデルは、実行する処理の手順をフロー図として表現するものである。フロー図の各ノードは処理の実行単位を示しており、IT システムの構成要素であるモジュールプログラムに対応している。モジュールプログラムは、自動合成ツールが保持している雛形モジュールをベースとして合成される。現在、グラフ表示やデータベース連携、フィールド入力など 10 種類の雛形モジュールが用意されている。図 4 に業務フローモデルとモジュール合成の例を示す。

図 4 は、受注情報を入力してデータベースへ登録する、受注登録処理フローを表している。「受注登録」、「業務 DB 管理」の各ノードはモジュールプログラムに対応しており、それぞれ「フィールド入力」および「データベース連携」雛形モジュールから合成される。

モジュールプログラムの合成情報は、テーブルデータとして与えられる。各雛形モジュールは自己合成機能を備えており、与えられたテーブルデータに基づいて自分自身をモジュールプログラムとして合成する。図 4 には、「フィールド入力」と「データベース連携」の合成情報テーブルを例示してある。ここでは受注情報として、「受注番号」、「品番」、「個数」、「納期」、「完了」の 5 つのデータを入力するように仕様が与えられているものとしている。

「受注登録」の合成情報テーブルには以下の情報が記述される。

- 各項目のデータ型
- 全項目を「受注」というグループにまとめること

これらに基づいて「受注登録」の画面とともに、入力されたデータの取得や表示設定を行うための機能（メソッド）が合成される。

一方、「業務 DB 管理」の合成情報テーブルには、データベース内に作成するテーブル情報が記述されており、この情報に基づいてデータベースの管理と連携を行うためのインタフェースが合成される。このインタフェースは外部のデータベースへの接続機能を持つ。これを利用することで、指定したデータベーステーブルの作成や、データの登録、検索、削除、更新を行うことができる。

「受注登録」画面からの入力データと、「業務 DB 管理」によってデータベースへ登録されるデータとの対応付けは以下のようにして行われる。「受注登録」で指定されたグループ名「受注」は「業務 DB 管理」で指定されたテーブル名「受注」に対応し、これによって登録先のデータベーステーブルが対応付けられる。そして、「受注登録」で指定された各項目名は「業務 DB 管理」で指定された「受注」テーブルの各列名に対応付けられ、それぞれのフィールドに入力されたデータは、対応する列に登録される。

このように、本自動合成ツールでは、モジュールプログラム間のデータのやり取りは、すべてパラメータ名による対応付けに基づいて行われる。

3.3 プロトタイプシステム

プロトタイプシステムの概観を図 5 に示す。まず、ユーザはの業務フローモデル作成画面でノードの配置と処理フローの記述を行う。ここで配置するノードはあらかじめ用意された雛形モジュールと対応付けられており、ユーザは必要な雛形モジュールを一覧から選択する。これらの雛形モジュールはすべて MZ Platform のコンポーネントを組み合わせて作成されているため、MZ Platform を利用して雛形モジュールを新規に追加していくことも可能である。

各雛形モジュールに与える合成情報テーブルは、のモジュール合成情報編集画面で作成する。ここでは、業務フローモデル作成画面で選択したノードの合成情報テーブルの編集を行うことができる。一度作成された各モジュールの合成情報および関連するパラメータは、それぞれおよびの画面で参照可能である。

各モジュールで合成された機能（メソッド）はのモジュール機能選択画面に一覧表示される。処理フローの作成に際し、ユーザは各モジュールノードで起動する機能をここから選

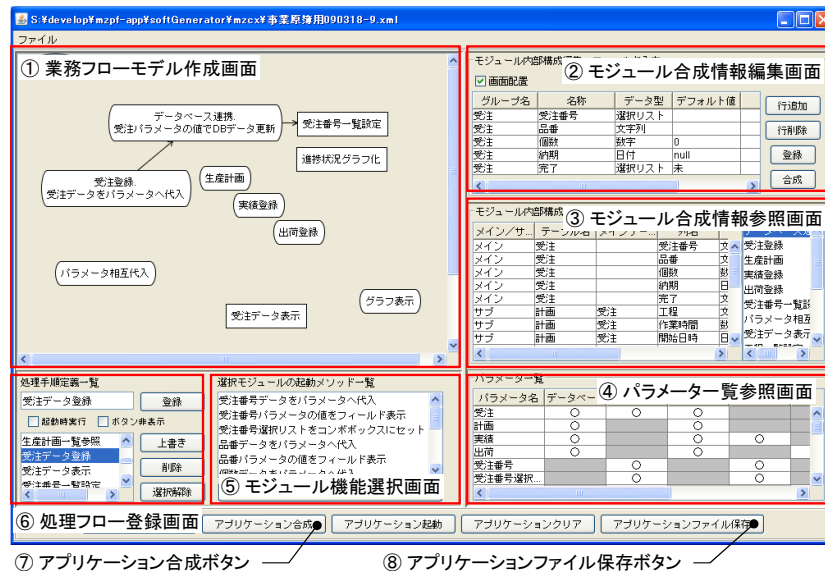


図 5 プロトタイプシステム概観

Fig. 5 Overview of the Prototype System

択する。例えば、前節で例に挙げた受注登録処理フローの作成では、「受注登録」で起動する機能として「受注データをパラメータへ代入」をここから選択することになる。

定義された処理フローの登録は、の処理フロー登録画面で行う。処理フローは複数の登録が可能であり、登録済みの処理フローをリストから選択することで表示切り替えが行われる。また、登録された処理フローを別の処理フローから呼び出すこともできる。

このようにして記述された業務フローモデルは XML 形式でファイル保存され、再利用が可能である。

アプリケーションは、のアプリケーション生成ボタンをクリックすることにより生成され、のアプリケーションファイル保存ボタンのクリックにより別ファイルとして保存される。生成されたアプリケーションの画面レイアウトの調整は、別途 MZ Platform を用いて行う。

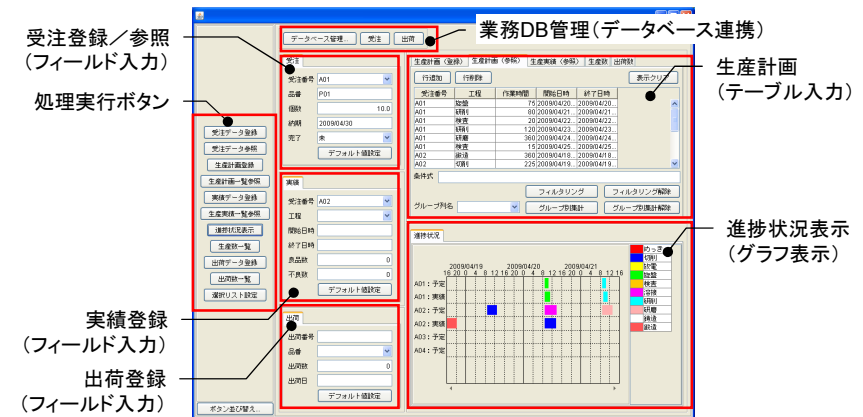


図 6 生成された生産管理アプリケーション
Fig. 6 Generated Production Control System

4. 適用事例

適用事例として、受注登録から生産計画登録、実績収集、進捗管理、出荷登録までを行う生産管理アプリケーションの生成例を図 6 に示す。作成に要した時間は、データベースの設計と構築も含めて半日程度である。このアプリケーションは、処理フローに基づいて生成された以下の機能を持つ。機能ごとに処理実行ボタンが生成され、それらをクリックすることで対応する処理フローが実行される。

- (1) 受注データ登録
受注登録 / 参照画面で入力されたデータをデータベースへ登録する。
- (2) 受注データ参照
受注登録 / 参照画面に、プルダウンリストで選択された受注番号の受注データを表示する。
- (3) 生産計画登録
生産計画画面内の生産計画 (登録) テーブルに入力されたデータを、データベースへ登録する。
- (4) 生産計画一覧参照
データベースに登録されている生産計画データの一覧を、生産計画画面内の生産計画

(参照) テーブルに表示する .

- (5) 実績データ登録
実績登録画面に入力されたデータをデータベースへ登録する .
- (6) 生産実績一覧参照
データベースに登録されている生産実績データの一覧を、生産計画画面内の生産実績(参照) テーブルに表示する .
- (7) 進捗状況表示
データベースに登録されている計画データと実績データを収集し、各工程の作業予定と実績の比較を進捗状況表示画面にガントチャートとして表示する .
- (8) 生産数一覧
受注番号ごとの生産数を、生産計画画面内の生産数テーブルに表示する .
- (9) 出荷データ登録
出荷登録画面に入力されてデータをデータベースに登録する .
- (10) 出荷数一覧
出荷番号ごとの出荷数を、生産計画画面内の出荷数テーブルに表示する .
- (11) 選択リスト設定
画面上の各プルダウンリストのデータを更新する .

また、このアプリケーションを構成するモジュールプログラムは以下の通りである . 括弧内は、ベースとなった雛形モジュールを示す .

- (1) 業務 DB 管理 (データベース連携)
受注、計画、実績、出荷の各データのデータベースへの登録や検索を行う . このアプリケーションでは、接続先データベースとして MySQL⁸⁾ を用いている .
- (2) 受注登録 / 参照 (フィールド入力)
受注データの登録および参照用のインタフェースを提供する .
- (3) 生産計画 (テーブル入力)
工程スケジュールなどの生産計画の登録と参照、また、実際に行われた作業の開始 / 終了日時や生産数、出荷数などの実績データ参照のためのインタフェースを提供する . この他、各データの集計機能を提供する .
- (4) 実績登録 (フィールド入力)
作業実績データとして、開始 / 終了日時や良品数 / 不良品数の登録用インタフェースを提供する .

- (5) 出荷登録 (フィールド入力)
出荷日や出荷数など、出荷データの登録用インタフェースを提供する .
- (6) 進捗状況表示 (グラフ表示)
各工程の進捗状況を表示するためのガントチャートを提供する .

5. おわりに

企業の IT システム開発では、情報の流れを正確に捉えることが本質となる . それは、例えば、受注時に付与された受注番号に対してどのようなデータが関連付けられ、さらにそれらが製造ロット番号を始めとする生産データとどのように関係し、そして出荷ロットとどのように対応するのかを明らかにするなどの作業である . 実際、ある企業では社内の生産管理システム開発に当たり、運用しているすべての帳票を 1 枚の模造紙に貼り、部署間での情報の流れの整理を実施し、その結果に基づいて開発計画の立案を行っている .

従来のシステム開発では、この情報の流れに基づいてシステム設計を行い、システム全体を構成する各モジュールプログラムの開発を行うことが必要である . また、一度開発したシステムを修正する際、場合によっては基本となる情報の流れから見直すことが求められることも少なくなく、そのための作業負担は非常に大きい .

これに対して本自動合成ツールは、システム開発の基本となる情報の流れを各構成モジュールで扱うパラメータ相互の対応付けとして記述し、それに基づいてアプリケーションを自動的に生成する . そのため、システム設計やプログラム作成の負担は大きく軽減される . 特に、IT の非専門家には扱いが困難なデータベースの設計や構築を自動化したことの効果は大きい . また、業務における情報の流れもすべて業務フローモデルの形で保存、参照、編集が可能であるため、開発したシステムの修正にも容易に対応できる .

今後、本自動合成ツールは、MZ Platform の付属アプリケーションの 1 つとして、一般に公開していく予定である .

謝辞 本研究開発は、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 . ご助言、ご協力いただいた産総研デジタルものづくり研究センターの皆様へ感謝の意を表する .

参 考 文 献

- 1) Sutcliffe, A. and Mehandjiev, N.: End-User Development, *Communications of the ACM*, Vol.47, No.9, pp.31-32 (2004).

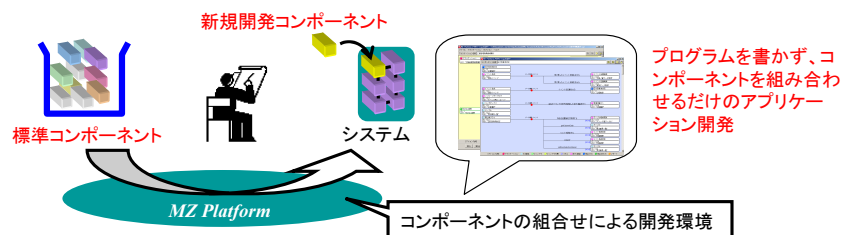


図 7 MZ Platform 概要
Fig. 7 Outline of MZ Platform

を 60%以上削減できることが例証されている。

MZ Platform は、現在、産総研コンソーシアム “MZ プラットフォーム研究会⁴⁾” を通じて、年会費 1000 円にて一般に配布されている。

- 2) Sawada, H., Matsuki, N., Tokunaga, H. and Furukawa, Y.: A Manufacturing Software Development and Operation Framework "MZ Platform" and its applications in Industry, *Proceedings of Advanced Engineering Design* (2004).
- 3) 澤田浩之：設計製造支援システム開発実行ツール:MZ Platform, 産総研 Today, Vol.9, No. 2 (2009). http://www.aist.go.jp/aist_j/aistinfo/aist_today/vol109_02/special/p14.html.
- 4) MZ プラットフォーム研究会: http://unit.aist.go.jp/dmrc/mzpf/mz_top.html.
- 5) Object Management Group – UML: <http://www.uml.org/>.
- 6) Eclipse Foundation: <http://www.eclipse.org/>.
- 7) VisualBasic デベロッパーセンター: <http://msdn.microsoft.com/ja-jp/vbasic/>.
- 8) MySQL: <http://www.mysql.com/>.

付 録

A.1 MZ Platform 概要

MZ Platform は、コンポーネントベースのアプリケーション開発実行環境であり、高度な IT 知識を必要とせずにプログラムの作成や修正を行うことができるツールである。図 7 にその概要を示す。

プログラムを作成する場合、ユーザはメニューから必要なコンポーネントを選択する。コンポーネントとは、特定の機能のみを実現する小規模プログラムを指す。そして、処理の流れをチャートとして記述する。その際、プログラムを記述する必要はなく、起動するメソッド（関数）等はすべてメニューからの選択によって指定できる。MZ Platform には、煩雑なプログラム記述を行う必要がないためにタイプミスのようなケアレスミスを排除できる、また、コンポーネントを活用することによって非専門家には扱いが難しい高度な機能も容易に取り込むことができるなどの利点があり、従来の開発環境と比較してプログラム作成工数