社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ＳＥ－160 (15)
2008－EMB－ 9 (15)
2008／6／20

業種特化型設計開発環境構築基盤の提案

村田大二郎　　団野博文　　三部良太

株式会社日立製作所　システム開発研究所

システムインテグレータの競争優位のためには，ビジネス分野に特化した設計開発能力の向上が重要である．設計開発能力を向上するためには，分野特有の情報を扱い易い形式で編集可能なツールを用いると効率的であるが，必要なツールを全て揃えるには時間が掛かる．本論文では，短期に分野向けの設計開発環境を構築するとともに，構築された環境を用い開発時間を短縮する方法を提案する．また，手法を実現する業種特化型設計開発環境基盤のプロトタイプについても説明する．

A proposal of a environment construction platform for business domain specific design and development

Daijiro Murata　　Hirofumi Danno　　Ryota Mibe

Systems Development Laboratory, Hitachi, Ltd.

An ability of conducting effective design and development that is tailored to the given specific business domain is a competitive advantage of system integrators. To improve their ability, each system integrator needs tools to edit information, which is specific to the domain with easily treatable format. However, supplying all tools for all information formats is time consuming. In this article, we propose a method to quickly develop design and development environments suitable for each domain and therefore achieving shortened development time with the environment. We also present a prototype tool for the proposed development environment construction method.

1. はじめに

近年，ソフトウェア及びソフトウェア開発に関する情報の公開は増大の一途を辿っている．情報システムのプラットフォームを販売するベンダ各社は，自社製品のマニュアルや無料の製品などの各種の情報を公開している．また，各種の団体が，共通規格と共通規格に基づいたソフトウェアとを提供している．

従って，プラットフォーム製品を用いてシステムを実装するシステムインテグレータ各社にとり，公開情報に基づく開発技術は依然重要であるものの，競合他社との差を出し難い物となりつつある．

一方，特定のビジネス分野に特化した設計開発能力は，守秘義務のため情報が公開され難く，かつ顧客の問題を解決する上で重要である．

このため，分野に特化した設計開発能力の向上が，システムインテグレータが競争優位を確保する上で，より一層重要となる．

分野に特化した設計をするに当たっては，分野特有の情報に注意する必要がある．分野に関する法律や，業界における慣習や，顧客の慣習などが，図面形式の異なる要因として挙げられる．

一般に，情報作成の効率化のためには，情報を扱い易い形式で編集するためのツールを使用する．しかし，図面形式の異なる要因が変化した場合においては，必要なツールの準備が間に合わない場合がある．

結果としてシステムインテグレータは，部署ごとに別々のツールを使ったり，十分なツールなしに設計したりすることとなる．ツールが不十分であると，情報の作成に時間が掛かる．あるいは異なるツールへの情報の入力し直しや図面形式の変更といった非効率な作業を繰り返すこととなる．結果，プロジェクトの効率化が図られずコストが増大する．

そこで，短期に分野向けの設計開発環境を構築するとともに，構築された環境を用い設計及び開発の時間を短縮する方法を提案する．また，方法の基盤となる業種特化型設計開発環境構築基盤を提案する．

2. 分野に特化した設計開発における課題

2.1 課題の全体像

以下，設計開発環境の構築と，環境を用いる設計開発とにおける特徴及び課題と解決策を説明する．

図 1 に特徴と課題と解決策との関係を示す．図 1 に示される特徴と課題と解決策との詳細は 2.2 節と 2.3 節において説明する．

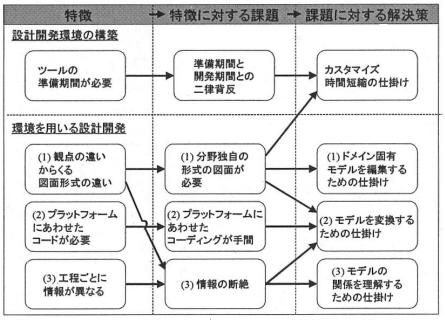


図 1 課題の全体像

2.2 設計開発環境の構築における課題

2.2.1 設計開発環境の構築における特徴

工程が開始される前に，ツールが揃っている必要があるため，ツールの準備期間が必要である．

一方，ツールの機能が複雑化するとツールの準備期間はより長くなる．

2.2.2 設計開発環境の構築における課題

2.2.1 節に示す特徴のために，設計開発環境の用意に当たっては，準備期間と開発期間との二律背反が発生する．

すなわち，ツールがない場合は開発プロジェクトの作業時間が長くなる．一方，ツールの準備に時間を掛けた場合も，開発プロジェクト全体の作業時間は長くなる．

2.2.3 課題に対する解決策

2.2.2 節に示した課題を解決するためには，ツール準備時間短縮の仕掛けが必要である．

設計開発環境の準備になるべく時間を掛けず，かつ設計開発環境を作業者のニーズにあわせる必要がある．

2.3.2 節に示すように，分野独自の形式の図面を作成するため，図面にあわせ複数のツールが必要となる．

2.3 環境を用いる設計開発における課題

2.3.1 情報システムの設計開発の特徴

情報システムの開発時には，顧客の業務を改善すべく，多段階の工程を経て，様々な役割の関係者がシステムを開発する．従って，情報システム開発プロジェクトは下記(1)～(3)の特徴を有する．

(1) 観点の違いからくる図面形式の違い

各工程に属する関係者の立場ごとに，作成する情報及び情報をチェックする観点が異なる．観点が違うため，立場ごとに異なる図面形式を使用する．

(2) プラットフォームにあわせたコードが必要

システムの実行基盤となるプラットフォームにあわせてコードを作成する必要がある．

(3) 工程ごとに情報が異なる

システム開発では情報を各工程で順番に詳細化する．よって，工程ごとに作成する情報及び参照する情報は異なる．

2.3.2 環境を用いる設計開発の課題

2.3.1 節に示した特徴のために，情報システムの設計開発は下記(1)～(3)の課題を有する．

(1) 分野独自の形式の図面が必要

工程や立場や状況に応じ，関係者がチェックする観点が異なる．このため，UML[1]や Business Process Modeling Notation (BPMN)[2]等の標準規格の図面だけでは，チェックに必要な事項全てを表すことはできないため不便である．

また，標準規格の図面を理解するためには規格を知っておく必要がある．複雑な標準である場合は，関係者の理解および図面の作成に時間が掛かる．

よって，関係者にとって理解し易い形式の図面を定義する必要がある．また，関係者が短時間で形式を理解し，図面を作成できねばならない．

(2) プラットフォームにあわせたコーディングが手間

組織がプラットフォームを用いるためには，プラットフォームの理解が必要である．

特に，分野特有の要件を満たすために，分野向けの特殊なフレームワークを使用する場合がある．フレームワークに向けた特殊なコードを作成するためには，設計者は特殊なフレームワークを理解した後に開発せねばならず，手間である．

(3) 情報の断絶

工程ごとに情報の形式が異なるため，プロジェクトの作業者にとり，自分の属さない工程の情報は理解し難くなる．また，同じ工程においても，関係者の観点が違うため，作成する図面の内容に関係者によってばらつきが出る事がある．

結果，関係者の間で必要な情報が伝わらない，あるいは誤って伝わるという問題が発生する．

また，工程をまたがる作業に工数が掛かる．工程をまたがる作業には，情報を変更した場合の影響箇所の把握や，影響のある箇所への変更の反映が挙げられる．

2.3.3 課題に対する解決策

2.3.2 節に示した課題を解決するためには，次の(1)～(3)が必要である．

(1) ドメイン固有モデルを編集するための仕掛け

関係者に理解し易い図面を作成するには，分野で必要な情報を簡潔に表したドメイン固有モデル(DSM: Domain Specific Model)[3]を作成する方法がある．

DSM を効率的に編集するためには，編集のためのツールが必要である．

(2) モデルを変換するための仕掛け

以下の理由により，モデルを変換するための仕掛けが必要である．

プラットフォーム固有のコードを効率的に作成するには，モデルの情報を変換しプラットフォームのコードを出力するツールが必要である．

また，関係者間の情報の断絶を防ぐためには，工程内でモデルの描き方を一致させ，工程間でモデルを変換する仕掛けが必要である．

(3) モデルの関係を理解するための仕掛け

工程間でのモデルの関係を理解するためには，モデルの形式の違いを吸収し，各モデルの要素の関係を出力する仕掛けが必要である．

3. 業種特化型設計開発環境構築基盤

3.1 業種特化型設計開発環境構築基盤の概要

2.2.3 節及び 2.3.3 節で示した解決策を現実化すべく，我々は業種特化型設計開発環境構築基盤を開発した．

業種特化型設計開発環境構築基盤の使用方法を図 2 に示す．

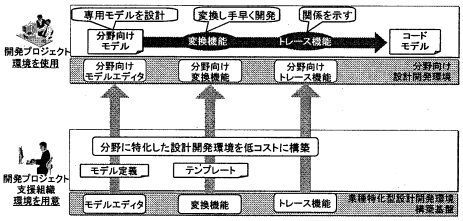


図 2 業種特化型設計開発環境構築基盤の使用方法

開発組織の生産技術部門などの開発プロジェクトを支援する組織は，3.2 節以降に述べるモデル定義やテンプレートなどを手軽に設定し，分野に特化した設計開発環境を短期に構築する．

開発プロジェクトは，支援組織から渡された分野向け設計開発環境を用い，下記(1)～(3)の機能を使用することで，システムを効率よく開発する．

(1) モデルエディタ

DSM を編集するためのエディタを用意する．

モデルエディタは，分野固有の図面の作成効率を向上することを目的とする．

モデルエディタは，様々な図面向けに簡単にエディタをカスタマイズし開発環境の準備時間を短縮するためのモデルの定義機能を有する．

(2) 変換機能

パターン変換機能とコード生成機能の二つを用意する．

パターン変換機能は，パターンとして定義された情報を元に，モデルの構成要素を追加する．既存の知識を利用して図面を手早く詳細化するとともに，図面の描き方のばらつきをなくすことを目的とする．

コード生成機能は，モデルの情報を利用しテキストファイルを生成する．情報システムのソースコード作成の手間を減らすことを目的とする．また，モデルフ

ファイルも Extensible Markup Language (XML)ファイルであるため，コード生成機能を用いモデルを生成することができると考えられる．

(3) トレース機能

トレース機能は，モデルの要素間の関連情報を可視化する．工程間の情報流通を効率化し，また影響把握や変更反映の助けとすることを目的とする．

2008 年 5 月現在，上記(2)の変換機能を利用してトレース機能を開発中である．

図 3 に 2.3.3 節で示した解決策と，上記の機能との関係を示す．

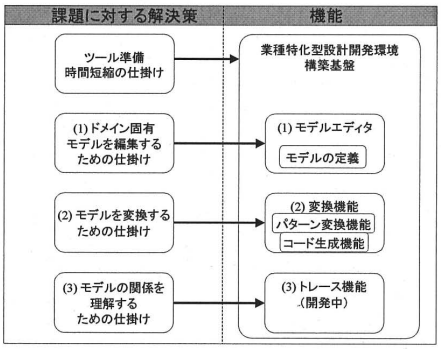


図 3 機能と解決策との関係

2008 年 4 月までに，機能(3)を除く機能(1)と機能(2)を開発した．実装した機能と，ツールの内部構成の詳細を 3.2 節から 3.4 節において説明する．

3.2 モデルエディタ

3.2.1 モデルエディタの使用方法

モデルエディタの使用方法を図 4 に示す．

モデルエディタにおいて特定の DSM を編集可能とするには，DSM の定義を作成しモデルエディタに読み込ませる必要がある．以下，DSM の定義をモデル定義と呼ぶ．モデル定義の詳細は 3.2.2 節に示す．

モデル定義を編集すると，図 4(1)に示されるように，エディタ左側の領域であるパレットにモデルの構成要素が登録される．

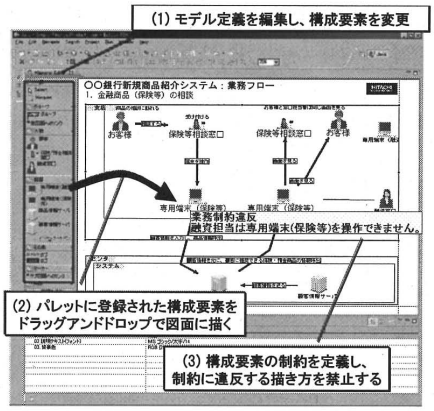


図 4 モデルエディタの使用方法

図 4(2)にあるように，パレットに登録された構成要素をエディタ領域にドラッグアンドドロップすると，図面にモデルの構成要素が追加される．

モデルエディタに定義したモデル構成要素間の制約に反し構成要素を関係付けると，制約に違反してはならない旨のメッセージが表示される．メッセージの例を図 4 (3)に示す．

制約を含むプロジェクトの設計標準をモデルエディタに定義し，開発組織内の設計者に展開することで，開発組織で決めた設計標準をプロジェクトの設計者に守らせることができる．

3.2.2 モデルエディタの内部構成

モデルエディタの内部構成を図 5 に示す．

モデルエディタ本体は，統合開発環境 Eclipse[4]のプラグインとして提供する．モデルエディタは，全てのモデルの種類に対して同一である．

モデル定義は，画像ファイル群と XML ファイル群から構成される．

画像ファイル群は，モデルエディタのパレットとエディタ領域とに表示するモデルの構成要素の画像から成る．

XML ファイル群は，モデルの構成要素と構成要素の関係との描画規則と，モデル及びモデルの構成要素の属性情報と，モデルの構成要素間の制約情報とを有する．制約情報は，モデルの構成要素間の関係に対し，関係の許可ないし拒否を示す情報である．

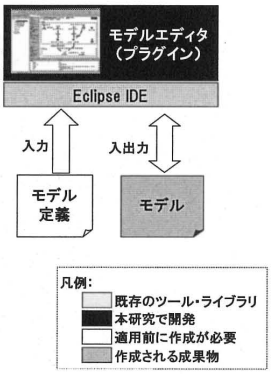


図 5 モデルエディタの内部構成

XML ファイルと画像とを変更することにより，様々な種類の図面を編集できる様にエディタをカスタマイズできる．

モデルファイルは，モデル全体の属性情報と，モデルの構成要素の情報からなる一つの XML ファイルである．

3.3 パターン変換機能

3.3.1 パターン変換機能の使用方法

パターン変換機能の使用方法を図 6 に示す．

パターン機能を起動すると，使用可能なパターンの一覧が表示される．

ユーザはパターンごとに表示されるパターン概要を読み，一覧の中からパターンを選択する．

続いてパターン内の要素に対応する要素を選択しボタンをクリックすると，改変されたモデルが表示される．

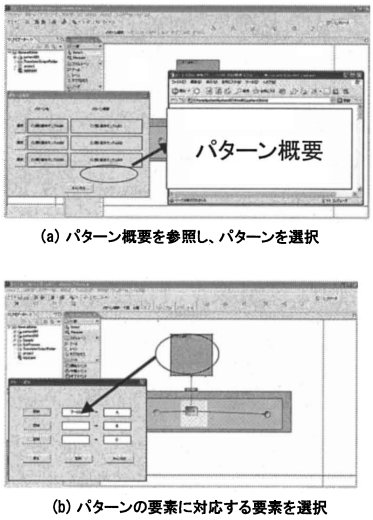
3.3.2 パターン変換機能の内部構成

パターン変換機能の内部構成を図 7 に示す．

パターンを定義する際はパターンのテンプレートを作成する．テンプレートは入力パターン及び出力パターンの２つのモデルファイルと，ダイアログ入力テンプレート及び関連定義と，パターン概要とからなる．

入力パターンは入力となるモデルの形を，出力パターンは出力となるモデルの形を表すモデルファイルである．

関連定義ファイルは入力の要素と出力の要素の関係を定義する．



(a) パターン概要を参照し、パターンを選択

(b) パターンの要素に対応する要素を選択

図 6 パターン変換機能の使用方法

ダイアログ入力テンプレートは，要素の内でユーザによる選択が必要な要素を定義する．

パターン概要は，ユーザにパターンの内容を示す情報を記した HyperText Markup Language (HTML) ファイルである．

文字列の変換など複雑な処理を行う際は，スクリプト言語を用いてユーザ定義関数を作成する．このため，パターン変換エンジンはスクリプト言語のエンジンを利用する．

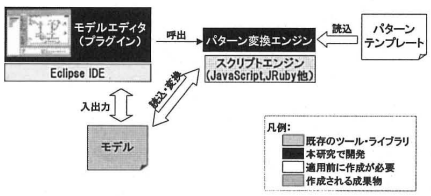


図 7 パターン変換機能の内部構成

3.4 コード生成機能

3.4.1 コード生成機能の使用方法

モデルエディタ内のボタンをクリックすると，コードが生成される．

3.4.2 コード生成機能の内部構成

コード生成機能の内部構成を図 8 に示す．

コード生成機能は，テンプレートエンジン Apache Velocity[5]及び，Velocity 上に構築される変換エンジン Apache Anakia[6]を使用する．

コードを出力するロジックを Apache Anakia の変換テンプレートに記述する．コード生成機能は，モデルの情報をテンプレートに入力し，出力するコードの情報を作成した後に，コードを生成する．

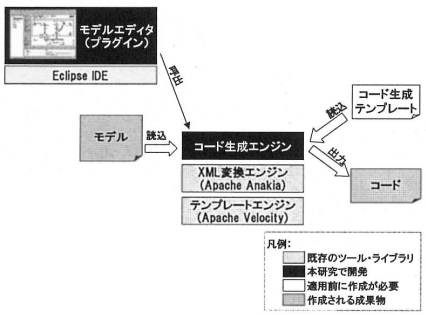


図 8 コード生成機能のアーキテクチャ

4. 業種特化型設計開発環境構築基盤の適用例

4.1 適用対象の設計開発プロセス

Service Oriented Architecture (SOA)[7]を用いるシステムの設計開発プロセスに対し業種特化型設計開発環境構築基盤を適用する方法を考案した．

SOA は，サービス連携によりシステムを構築するシステムアーキテクチャである．一般的に，サービスの実装には Web サービスを，サービスを連携するビジネスプロセスの実装には図 9 に示すような XML ファイルである Business Process Execution Language (BPEL) [8]を用いる．今回は，サービスの開発は除き BPEL の設計のみを考案した．

```
<process name="…" …>
  <partnerLinks>
    <partnerLink name="…" partnerLinkType="…" myRole="…" />
    <partnerLink name="…" partnerLinkType="…" partnerRole="…" />
  </partnerLinks>
  <variables>
    <variable name="…" messageType="…" />
  </variables>
  <sequence>
    <invoke name="…" …/>
    <receive name="…" …/>
  </sequence>
</process>
```

図 9 BPEL ファイルの例

BPEL はビジネスプロセスを表す言語であるが，実行言語であるため詳細に記す必要があり，かつ顧客の業務部門には理解し難い．よって，単純な情報からより複雑な情報に順次詳細化する設計開発プロセスを採用することとした．

以下，(1)～(3)に BPEL の設計開発プロセスの各工程を示す．

(1) 業務フロー図の作成

システムの開発に当たっては顧客である組織の業務を明確化する必要がある．

明確化に当たっては，組織の業務部門に所属する人に業務内容をヒアリングする必要がある．ヒアリングのために，顧客の業務部門とプロジェクトの設計者とのやり取りに用いるために，業務の流れを表す図である業務フロー図を描く．業務部門にとって理解しやすくするため，図 10 に示すように業務フロー図にはなるべく分かり易いアイコンを用いる．

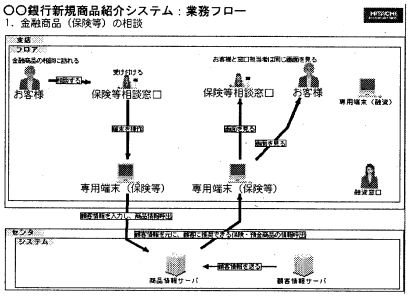


図 10 業務フロー図の例

また対称顧客ごとに顧客組織内で使用してきた業務フロー図の形式が異なる場合は，適宜異なる形式で業務フロー図を描く．

(2) BPMN 図の作成と詳細化

業務フローのままでは，システム上で実行するための必要な情報がない．そこで，業務フロー作成後，顧客のシステム部門と開発組織の設計者との間で段階的に業務フローを詳細化していく．

この際，BPMN 仕様を用いて形式的にプロセスを描き直す．以下，BPMN 仕様を用いた図を BPMN 図と呼ぶ．

(3) BPEL ファイルの作成

BPMN 図を詳細化し，システム上で実行するに十分な情報を追加した後に，実行言語である BPEL に変換する．

4.2 適用内容

4.1節に示したSOA構築プロセスをサポートするために，業種特化型設計開発環境構築基盤の機能を用いる方法を考案した．

SOA システム構築プロセスにおける業種特化型設計開発環境構築基盤の機能の利用方法と目的を，下記(1)～(4)及び図 11 に示す．

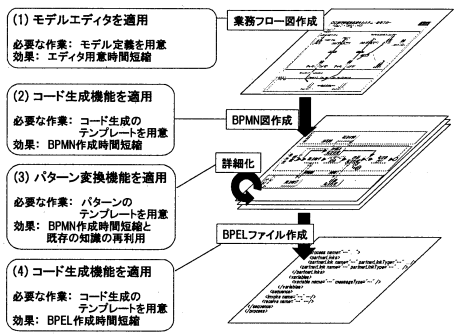


図 11 開発プロセスとアプローチの対応

(1) 業務フロー図の作成

エディタを用意するため，業務フロー図のモデル定義を作成した．作成には，基盤のインストールを含め一日弱掛かった．

設計者は，業務フロー図のモデル定義を読み込んだモデルエディタを用いて業務フロー図を作成することができる．

(2) 業務フロー図から BPMN 図へ変換

業務フローから BPMN 図への変換にはコード生成機能を用いる．設計者はコード生成機能を用いることにより，BPMN 作成に掛かる時間を短縮することができる．業務フローから BPMN へのコード生成のテンプレートは2008年5月現在開発中である．

生成した BPMN は，BPMN のモデル定義を読み込ませたモデルエディタ上で編集する．このため，BPMN のモデル定義も作成した．

ただし，作成したモデル定義は，BPMN の仕様の内，一部のみを実装している．

(3) BPMN ファイルの段階的詳細化

パターン変換機能を用い，BPMN ファイルを段階的に詳細化していく．

パターンを使用することにより，設計者による BPMN の作成時間を短縮すること，及び既存の知識を再利用することを目指す．

BPMN 図を詳細化するパターンについては，試験的に一つのみを作成した．

(4) BPEL ファイル生成

BPMN 図からコード生成機能を用い BPEL ファイルを生成する．

コード生成機能を利用することにより，設計者による BPEL 作成の時間を短縮することを目指す．

4.3 今後の課題

今後の課題を(1)から(4)に示す．

(1) 未実装の変換機能の実装

業務フロー図から BPMN 図への変換のためのコード生成テンプレート及び BPMN 図を詳細化するパターンを実装する．

(2) 未実装のトレース機能の適用

トレース機能を実装し，SOA を用いる開発においてモデル間の関係を理解し易くすることを目指す．

(3) カスタマイズ能力向上と評価

モデル定義及びパターン変換のテンプレートの作成には複数のファイルを作成する必要がある．またコード生成のテンプレートの仕様は Apache Anakia の仕様に同じである．

今後は，モデル定義とパターンテンプレートとコード生成テンプレートとの作成効率を向上するためのカスタマイズ機能を業種特化型設計開発環境構築基盤に実装する．

機能の評価のために，機能を用いたカスタマイズに掛かる時間を測定し，本基盤を用いない場合及び，カスタマイズ機能を用いない場合に必要とする時間と比較する．

モデル定義のカスタマイズ方法としては，複数の要素で属性及び関係が同じものを同一種類とみなすことにより，既存の図面からモデル定義を抽出する方法を検討中である．

(4) SOA 開発プロセスへの適用の評価

説明した SOA 開発プロセス及び業種特化型設計開発環境構築基盤を用いて実際にシステムを構築し，基盤を用いない場合との工数を比較し，モデルエディタとパターン変換機能とコード生成機能とを評価する．

また，システム構築後に，特定の仕様を変更した場合の，影響範囲の把握と影響箇所の変更に掛かる工数を比較し，パターン変換機能の評価を行う．

5. まとめ

システムインテグレータの競争優位のためには，特定ビジネス分野に特化した設計開発能力の向上が重要である．設計開発能力を向上するためには，分野特有の情報を扱い易い形式で編集可能なツールを用いると効率的であるが，必要なツールを全て揃えるには時間が掛かる．本論文では，短期に分野向けの設計開発環境を構築するとともに，構築された環境を用いシステムの開発時間を短縮する業種特化型設計開発環境構築の方法論を考案した．

同時に，方法論を実現するための業種特化型設計開発環境構築基盤のプロトタイプを構築した．プロトタイプは，図面に応じてカスタマイズ可能なモデルエディタと，パターンに基づきモデルを変換する機能と，モデルからコードを生成する機能を有する．

また，基盤の SOA を用いるシステム開発への適用方法を示した．

今後は，未実装の変換及びトレースの機能の実装と，基盤のカスタマイズ性の向上とに取り組む．また，実装した基盤を用いてシステムを開発し，基盤を評価する．

参 考 文 献

1) Object Management Group: OMG Unified Modeling Language (OMG UML), Infrastructure, V2.1.2, http://www.omg.org/spec/UML/2.1.2/Infrastructure/PDF/ (2007).

2) Object Management Group: Business Process Modeling Notation Specification, http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf (2006).

3) Nordstrom G., Sztipanovits J., Karsai G., Ledeczi A.: Metamodeling - Rapid Design and Evolution of Domain-Specific Modeling Environments, Proceedings of the IEEE ECBS'99 Conference, pp. 68-74 (1999).

4) Eclipse Foundation: Eclipse.org home. http://www.eclipse.org

5) Apache Software Foundation: The Apache Velocity Project. http://velocity.apache.org/

6) Apache Software Foundation: The Apache Velocity Anakia. http://velocity.apache.org/anakia/

7) Krafziq, D., Banke, K., Slama, D.: Enterprise SOA: Service-Oriented Architecture Best Practices, Pearson Education (2004).

8) Andrews, T., Curbera, F., Dholakia, H., et al.: Business Process Execution Language for Web Services Version 1.1, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel/ws-bpel.pdf (2003).