

# エンドユーザ主導開発のためのドメイン特化型技術の 適用性に関する考察

中所 武司<sup>†</sup>

<sup>†</sup> 明治大学理工学部 〒214-0033 神奈川県川崎市多摩区東三田 1-1-1

E-mail: <sup>†</sup> chusho@meiji.ac.jp

**あらまし** 変化の激しい時代には、業務の知識を有するエンドユーザ主導のアプリケーション開発とその保守が重要になると考え、その技法を研究してきた。エンドユーザである業務の専門家が、自ら必要とするWebアプリケーションを開発するためには、フロントエンドのビジュアルなユーザインタフェースとバックエンドのデータベースの定義に加えて、アプリケーション固有のビジネスロジックの定義が重要である。ドメイン特化型フレームワークを用いたビジネスロジックの定義において、プログラミングスキルを必要としない方式は実際の適用範囲が狭くなるというトレードオフ問題がある。本稿では、プログラミングの記述粒度を要求仕様定義レベルに保ちながらビジネスロジックのより詳細な定義を可能とするテンプレートを導入することにより適用性を改善した。

**キーワード** エンドユーザコンピューティング, ビジネスロジック, 3層アーキテクチャ, テンプレート

## Applicability of domain-specific technologies for end-user development

Takeshi CHUSHO<sup>†</sup>

<sup>†</sup> School of Science and Technology, Meiji University, 1-1-1 Higashimita, Tama-ku, Kawasaki, 214-0033 Japan

E-mail: <sup>†</sup> chusho@meiji.ac.jp

**Abstract** The development of Web applications should be supported by business professionals themselves since Web applications must be modified frequently based on their needs. In our recent studies, web applications are developed by using a domain-specific application framework and visual modeling technologies. The applicability of these technologies, however, decreases if the programming skill is required for end-user development. This trade-off problem is improved by using a template for business logic description based on three-tier architecture.

**Keywords** End-user computing, Business Logic, Three-Tier Architecture, Template

### 1. はじめに

インターネット上での Web アプリケーションの普及やクラウドコンピューティングの定着とともに、ソフトウェアのサービス化[7]が促進されている。我々は、変化の激しい時代には、エンドユーザ主導のアプリケーション開発とその保守が重要になるという観点から、小さな部門や個人の業務を対象とする中小規模の Web アプリケーションに関して、低コストで短期間に開発するとともに、頻繁な機能変更を伴う保守にも対応するために、その分野の業務の専門家主導で開発・保守ができるような技法を研究してきた。

1980年代以降、エンドユーザコンピューティングが注目され、その定義、分類、管理に関する研究が行われてきたが、筆者の解説論文[3]では、業務の知識を有するエンドユーザ、すなわち業務の専門家が自らの業務を自動化するソフトウェアを自ら開発する、という意味で用いている。そこでは、ソフトウェア産業の進

化過程におけるソフトウェア開発の主要な担い手を、初期の労働集約型産業におけるプログラマから知識集約型産業における設計者を経て、知恵集約型産業における業務の専門家と位置づけ、脱プログラミングを志向するエンドユーザコンピューティングについて述べた。エンドユーザがプログラミング言語を用いないで業務のコンピュータ化を行なえるための現実的な技術として、4GL、日本語プログラミング、ビジュアルプログラミング、AI（ルール、ファジー、ニューロ）の4項目を取り上げたが、実際にはプログラミングの概念やセンスが不要というわけではなかった。

エンドユーザの数については、2012年の米国内について、コンピュータを業務に利用しているエンドユーザの60%にあたるスプレッドシートやデータベース利用者の半数近くの2500万人、および自己申告によるエンドユーザプログラマ1300万人がプログラムを作成しているとの予測[12]がある。またソフトウェアの

品質確保の観点で、エンドユーザのためのテスト・デバッグを含めたソフトウェア工学の重要性が指摘されている[1].

エンドユーザである業務の専門家が自ら必要とするWebアプリケーションを開発するためには、3層アーキテクチャを前提としたドメイン特化型フレームワークを用いるとともに、アプリケーション固有のビジネスロジックの定義方法が重要であるが、プログラミングスキルを必要としない技術は実際の適用範囲が狭くなる。そこで、今回、ケーススタディをベースに、このトレードオフ問題と解決方法について考察する。

## 2. 基本的アプローチ

本研究の基本的なアプローチを図1に示す。ビジネスレベルで、エンドユーザ（業務の専門家）はビジネスモデルを提案する。サービスレベルで、ドメインモデルが作成され、必要なサービスが決められる。ソフトウェアレベルでは、コンポーネントを用いてドメインモデルが実装される。コンポーネントとドメインモデルの間の粒度的ギャップは、ビジネスオブジェクトやデザインパターンやフレームワークで補われる。一方、サービスレベルとビジネスレベルの意味的ギャップは、ドメイン特化型技術で補われる。

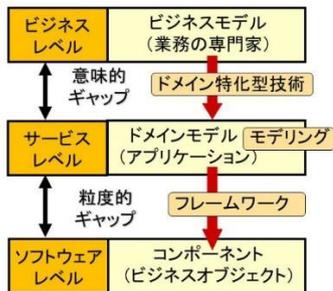


図1. エンドユーザ主導開発のアプローチ

筆者の2007年の報告[4]では、ドメイン特化型フレームワーク、ビジュアルモデリング、Webサービス連携について述べた。ドメイン特化型フレームワークの研究では、窓口業務を例題ドメインとしたフレームワークを試作し、図書管理システムなどへの適用実験により、再利用率86%という結果を得た。ビジュアルモデリングでは、業務コンポーネントを組合せた業務モデルの構築とその動的ふるまいのシミュレーション実行を繰り返しながら機能拡張していくアジャイル開発を支援するモデリングツールM-baseの試作と適用実験について述べた。Webサービス連携の研究では、WebAPIをフォームとして定義するとともに、複数の入力フォームから出力フォームへのフォーム変換によりWebサービス連携を実現できることを示した。

2012年の報告[5]では、ユーザインタフェース(UI)、

ビジネスロジック(BL)、データベース管理(DB)の3層アーキテクチャを前提としたWebアプリケーションの開発において、脱プログラミングの観点で最も重要なビジネスロジックの定義方式について、要求仕様定義段階でUI、BL、DBの組み合わせで表現するような、UI駆動型のテンプレートの導入を提案した。

2014年の報告[6]では、これまで、エンドユーザ主導開発の研究で取り上げた不用品再利用サービスなどの身近な例題をより広いドメインであるマッチングサービスととらえ、エンドユーザ主導開発技術の適用拡大の可能性を示した。特に最近は多種多様なマッチングサイトが実用化されていることから、エンドユーザ主導開発に適したアプリケーションを限定するための評価基準を導入した。

## 3. ドメイン特化型技術

### 3.1. ドメイン選択

これまで、エンドユーザ主導開発の研究のための身近な例題アプリケーションとして取り上げてきた不用品再利用支援などのサービスでは、アプリケーションで扱う物またはサービスの登録用のテーブルと、その提供者あるいは要求者となる利用者の登録用テーブルの2種類のDBテーブルを使用する。基本操作はCRUD(create, read, update, delete)であり、テーブルのレコードの各カラムは扱う物/サービスに依存するが、これらのアプリケーションは、物/サービスの提供者と要求者のマッチングの概念でまとめられる。

このようなマッチングサービスの具体的な事例は広範囲にわたるので、対象範囲を絞るためにマッチングサービスの分析と分類を実施した。まず、マッチングサイトの概要を図2のように表現し、分類要因を以下の3種類とする。

- WHO : 提供者・希望者
- WHAT : 物、サービス
- HOW : マッチング決定アルゴリズム

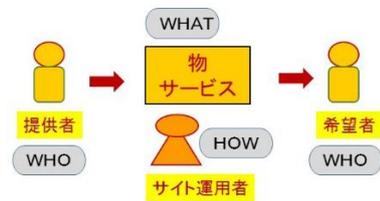


図2 マッチングサイトの概要図

WHO(提供者・希望者)については、対等な関係の一般ユーザを対象とする。ショッピングモールやホテルの検索・予約のサイトなどのWebアプリケーションはIT専門家が開発を担当し、我々の研究対象とは要求仕様レベルが異なるものなので対象外とする。

また、提供側と希望者側がともに一般ユーザであっても、営業目的のオークションサイトや中古品の買入れ・販売サイトは同様の理由で対象外とする。なお、地方自治体が窓口業務として実施している地域住民のための不用品再利用支援サービスは、現在、予算・人材不足により、Webサイトで運営しているものは少ないが、Webサイト化されれば効果が大きいと考えられるので、本研究の対象としている。

次に、WHAT（物、サービス）については、不用品再利用のような物、あるいは、災害復旧支援のボランティア活動などのサービスを対象とする。こちらも営業活動として実施されているものは対象外である。

HOW（マッチング決定アルゴリズム）については、無料で提供する不用品の再利用支援などのアルゴリズムがシンプルなものに限定する。一方、近くの空車タクシーの呼び出しや賃貸アパート・マンション紹介など、決定アルゴリズムが複雑なものは、IT専門家が開発を担当すべきものなので対象外とする。

すでに多種多様なマッチングサービスが実用になっているが、特に最近では、スマホ利用が前提のものが多く、個人間取引（CtoC, Consumer-to-Consumer）やシェアリングエコノミー（シェアリングビジネス）といわれる分野が注目されている。その代表的な分野として、自宅の空き部屋を旅行者の宿泊に貸し出す民泊サービスと自家用車を使用したライドシェアサービスがある。

基本的には、エンドユーザ主導開発に関するわれわれの研究目標は、IT技術者が開発を担当しているようなマッチングサービスとは異なる。たとえば、最近では、情報技術（IT）を応用して持続可能な社会のための環境保護に貢献する（Green-by-IT）が期待されており、地方自治体が運営する地域住民のための不用品再利用サービスに関して、もし、これらの担当者自身がWebアプリケーションを開発してWebサイトを立ち上げることができれば、扱える品物の量や利用者の数は格段に増加すると思われる。

### 3.2. マッチングサービスの分析と分類

われわれの研究目標を明確にするためには、多種多様なマッチングサービスを分析して分類する必要があると考え、まずユーザの視点から以下の2項目の評価基準を導入した。

- ・参加者の信用性への要求
- ・もの・サービスの品質への要求

マッチングサービスのサイトに関しては、提供者、希望者、サイト運用者の3種類の関与者の信用性が重要な要因となりうる。そこで、分析・分類が複雑になるのを避けるためにとりあえずサイト運用者の信頼性については考慮せず、図3に示すように、サービスへの

参加者（提供者と希望者）の信用性への要求の度合いを横軸に設定した。一方、この分野での不可欠の要因と考えられる成果物（もの、サービス）の品質の要求への度合いについては、図3の縦軸に設定した。

不用品再利用サービスの例では、無料での授受のみ扱う場合は<低, 低>、有料での授受で、希望者が提供者に直接支払う場合は信用性と品質のいずれもトラブルになりやすいので<高, 高>、決済を運用サイトが仲介する場合は、支払いに関するトラブルは避けられるので<低, 高>に位置づけられる。<高, 低>に位置づけられる例としては、独り住まいの高齢者の住宅の雪下ろしボランティアサービスがある。民泊やライドシェアの場合は、サービス提供者や希望者の信用性もサービスの質も未知なので<高, 高>に位置する。

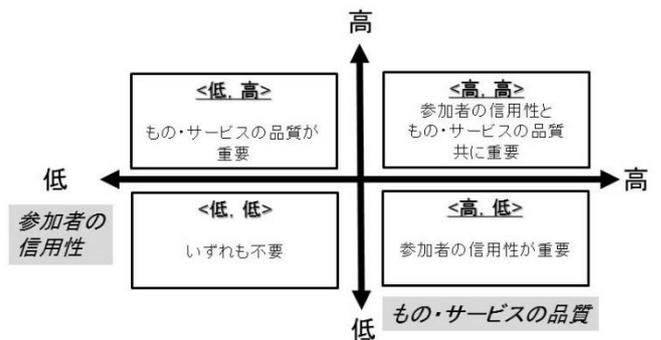


図3 ユーザビューでの分類

しかしながら、エンドユーザ主導開発技法では、業務処理をビジネスロジックとして定義する方式の難易度が重要になる。そこで、ユーザビューに基づくこれらの評価基準に加えて、システムの構築容易性の視点（システムビュー）で、新たに以下の2つの評価基準を導入する。

- ・アルゴリズムの複雑度
- ・業務ルール（ビジネスルール）の複雑度

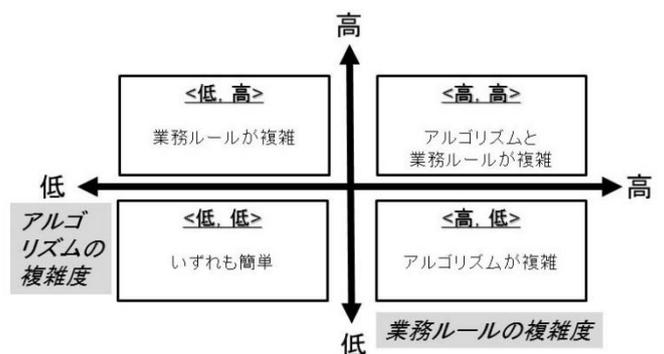


図4 システムビューでの分類

これらの基準を横軸と縦軸に対応させた図4を用いて先ほどの例を検討してみる。無料での授受のみ扱う不用品再利用サービスの場合、希望者の先着順にマ

atchingし、その後の不用品の授受は提供者と希望者の間での直接交渉とすれば<低, 低>となり、有料の場合も定額の場合は同様であるが、オークション方式の場合は処理が複雑化する。雪下ろしボランティアについては、募集定員が決まっていて先着順に受け入れる場合は<低, 低>に位置する。

民泊では、希望者自身が事前に登録された物件の中から選択するのでアルゴリズムは簡単だが、登録物件に関するルールが多いので<低, 高>に位置する。逆にライドシェアの場合は希望者の要求に合致した提供者の中から地理的に近いものを提示するなど、アルゴリズムが複雑になるので<高, 低>に位置する。

結論として、我々の研究目標とする例題は、第一義的にはシステム構築の容易さを優先して、システムビューの<低, 低>のドメインから選択すべきといえる。さらに、本研究で想定するエンドユーザは自ら使用するアプリケーションを自ら開発することから、当然その運用の担当者でもあるので、運用時のトラブルを極力避ける必要があり、ユーザビューの<低, 低>のドメインから選択すべきともいえる。

#### 4. ドメイン特化型フレームワークの適用性

われわれのこれまでの研究試作結果について考察する。図5は、縦軸にエンドユーザ主導開発技術の適用性、横軸にビジネスロジックの記述におけるプログラミング粒度をとり、5種類の研究試作の特徴を示している。一般論として、プログラムの記述の粒度を小さくすれば、プログラム言語に近い表現ができるので、適用範囲は広がる。他方、粒度を大きくして、業務に近い表現を取れば、エンドユーザが利用しやすくなるが、適用範囲は限定される。

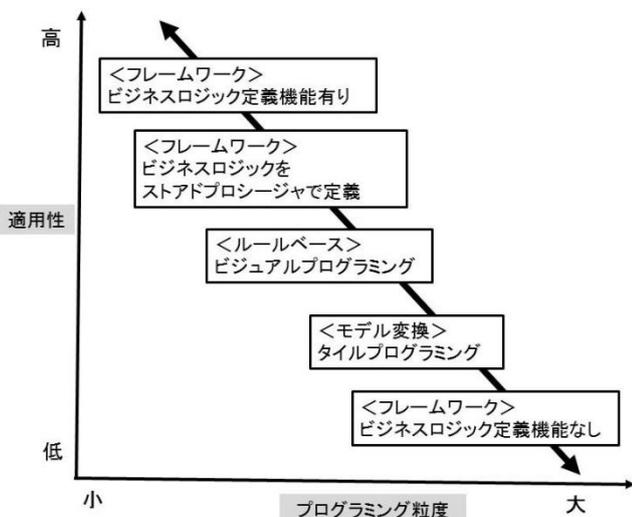


図5 エンドユーザ主導開発技術の適用性

図5の最上位は、不用品再利用システムを例題とし

て、ビジネスロジックの定義機能を有するドメイン特化型フレームワークを試作[10]したものである。エンドユーザは、ビジュアルツールを用いて、DB、GUI、ビジネスロジックを定義し、各々、JSONモデル、CSSモデル、SQL文として保存し、これらを解釈実行する方式でアプリケーションを実行する。具体的には、最初に参加者と不用品と不用品取得申請の3種類の管理テーブルのカラム名とデータ型を定義する。次に、画面の項目、属性、表示形式などの情報を定義し、JSONファイルとして保存する。ビジネスロジック機能は、バリデーション、DB操作、ページ遷移、ナビゲーションの4種類のカテゴリに分類でき、たとえば、ビジネスロジック対応のSQL文は、データモデル間の参照関係およびデータモデルとGUIとの関連から生成する。これらの関連は、オブジェクト間の結線で定義されるが、エンドユーザには、オブジェクト間の結線操作などにプログラミングスキルが求められる。

図5の2番目は、不用品再利用システムを例題として、ビジネスロジックをストアドプロシージャで定義する機能を有するドメイン特化型フレームワークを試作[11]したものである。機能は、ユーザ登録・変更、メールアドレス登録・ユーザ認証、不用品登録・更新、登録品検索・申請、登録品受領者決定プロセスなどである。実装にはASP.NETのMVCモデル用い、DBテーブルは、ユーザ、物品、物品申請、物品種別、物品授受の方法、マッチング状態管理の6種類とした。

たとえば、「もし、ユーザがすでに上限数の物品を登録している場合、ユーザは新たに物品を登録できない」というビジネスロジックでは、このユーザが登録した物品をDB検索し、数を数え、上限より少なければ登録し、上限に達していればエラー処理をする。この処理をJavaクラスからSQL文を使って実行するとDBアクセスが多く発生するので、本試作ではビジネスルールをストアドプロシージャとして実装した。エンドユーザは、ビジュアルツールを用いて、システムが提供する各テーブルの選択・追加・更新・削除操作の基本ストアドプロシージャをカスタマイズできるようにしたが、ストアドプロシージャに関するプログラミングスキルが求められる。

図5の3番目は、不用品再利用システムを例題として、ビジネスロジックをルール表現で定義するビジュアルツールを試作[9]したものである。ビジネスロジックの主要部分をルール形式で定義し、Androidアプリを生成する。不用品提供者は、登録する不用品をスマホで撮影し、簡単にアップロードでき、希望者が写真を見ながら必要なものを探せるようにしている。

実際のルール定義には3種類のコンポーネントを用いる。まず7種類の条件コンポーネントの中から適

切なものを選択し、次に条件の真偽に応じたアクションに対応する処理コンポーネントを選択する。これらの詳細な定義部分では値コンポーネントが使用される。条件には、二つの値の比較、入力値のバリデーション、DB検索結果の判定、申請ルール違反の有無などがある。処理には、画面遷移、DB操作、ダイアログ表示、メール送信などがある。値には、テキストボックス入力情報、画面遷移情報、ラジオボタン選択情報、DBレコードのカラム値、現在時刻などがある。

エンドユーザはビジュアルツールを用いてこれらのコンポーネントを組合せながらルールを定義することになるが、通常のプログラミングにおける各要素の粒度と同程度の概念の理解が必要になる。

図5の4番目は、小規模の図書管理システムを例題として、タイルプログラミングで定義したWebアプリケーションモデルを設計モデル(Struts2)に変換後、Javaプログラムを生成するモデル変換システムを試作[2]したものである。画面作成や画面遷移に用いる一般的なGUI部品に加えて、タイルプログラミングに用いられる図書管理業務用のテンプレートコマンドは事前に記述実験を通じて抽出し、Webアプリケーションモデル作成時に利用できる。主なコンポーネントは、UI用、HTMLタグ用、ロジック用、DB用の4つのカテゴリに分けられる。今回の試作では、設計モデルにStruts2を用いて、ページはJSPに対応させ、ビジネスロジックはJavaクラスとStruts.xmlに対応させた。

適用実験に用いた図書管理システムは、ユーザ登録・削除、図書登録・削除、貸出・返却、一覧表示の機能を有し、フォーム用ページ25件、ビジネスロジック定義用ページ8件、DBテーブル定義3件で構成され、ほぼプログラミングスキルが不要となった。

図5の5番目は、小規模の図書管理システムを例題として、ビジネスロジックの定義機能を有しないドメイン特化型フレームワークを試作[8]したもので、実装にはRuby on Railsを用い、Railsのscaffolding機能を活用した。DBテーブルはビジュアルツールを用いて定義するとともに、DB検索機能や入力データのバリデーション機能はあらかじめ固有の機能としてフレームワークに組み込んでいるので、エンドユーザは、必要に応じてこれらの機能を選択すればよいので、プログラミングスキルは必要ない。しかしながら、任意のビジネスロジックを定義することはできないので、適用範囲は限定される。

以上、図5に示したエンドユーザ主導開発の5種類の技術に関する考察から、適用範囲とプログラミング粒度はトレードオフの関係にあるという一般性のある結論を得た。ここで使用するプログラミング粒度は概念的なものであるが、同じ機能を定義するときの行数

が多いほど粒度が小さいという意味でもあり、また、アプリケーションをその対象ドメインの業務に近い表現で定義できるほど粒度が大きいという意味である。

## 5. トレードオフ問題の解決方法

このようなトレードオフ問題を改善するためには、多様なビジネスロジックの定義方法を含めて、開発手順を定める必要があり、以下のような図式で表現されるABC開発モデルを想定した。

Application = Business logic + CRUD

すなわち、簡単なDBテーブルはビジュアルフォームで定義できることから、まず、UIとDBを定義し、それと同水準の抽象度でビジネスロジックを定義する。なお、CRUDは、DB操作の基本機能(Create, Read, Update, Delete)を意味する。

さらに、エンドユーザ主導開発における重要課題であるビジネスロジックの定義プロセスを容易化するためのテンプレートを導入する。対象とするWebアプリケーションは3層アーキテクチャを前提にしているので、要求仕様定義段階でのビジネスロジックをUI, BL, DBの組み合わせで表現することとした。具体的には、UI駆動型のアプローチをベースに、以下のようなテンプレートを考案した。

- ①UI：システムは利用者から要求を受け取る
- ②BL：システムはその要求を処理
- ③DB：システムは必要に応じてDBにアクセス
- ④BL：システムはDBアクセス結果を処理
- ⑤UI：システムは結果を表示

このテンプレートは、ビジネスロジックの典型的な処理プロセスが{UI→BL→DB→BL→UI}であることを意味する。エンドユーザ(業務の専門家)はビジュアルなフォームに慣れ親しんでいるので、このようなUI駆動型アプローチがわかりやすいと思われる。これは、ユースケースのテキスト表現のシナリオ記述におけるシステム側の処理内容をUI, BL, DBに分割して定義していることになる。このテンプレートの適用により、プログラミングの記述粒度を要求仕様定義レベルに保ちながらビジネスロジックのより詳細な定義を可能としているので、図5で示したトレードオフ問題を改善できると思われる。

このテンプレートの使用例の説明のために、ボランティア支援システムを取り上げる。大地震、津波、集中豪雨、洪水、土砂崩れなどの自然災害において、民家の後片付けや避難所での住民支援にボランティアが活躍しているニュースを見かけるが、一方で、現地ではボランティアを適材適所に派遣する業務の困難さからボランティアを断るとか、あるいは支援物資についても同様の混乱が生じているという状況がある。もし、

地方自治体の職員が現場の状況に適したWebサイトを立ち上げてボランティアや支援物資を適切に管理し必要な場所に届けられればその効果は大きい。

まず、テンプレート{UI→BL→DB→BL→UI}の最初のUIのすべての入力項目【指定したボランティアプロジェクトの識別番号,登録者識別番号,登録者氏名,ボランティア行動規則の順守の確認チェックボックス】を定義する。次にテンプレートの最後のUIのすべての出力項目【登録者氏名,指定したボランティアプロジェクトの識別番号,登録番号】を定義する。次にテンプレートのDBで用いるテーブルのレコードのすべてのカラム【指定したボランティアプロジェクトの識別番号,登録番号,登録者識別番号,日付,登録状態】を定義する。次にテンプレートの最初のビジネスロジックにおける入力項目のチェック処理【入力項目の記入漏れ,入力値】と内部処理【登録日の取得,登録番号の決定】を定義する。最後にテンプレートの後半のビジネスロジックにおける出力項目【登録者氏名,指定したボランティアプロジェクトの識別番号,登録番号】と内部処理【なし】を定義する。

業務の専門家であるエンドユーザは,このような方法で基本的にはWebアプリケーションの要求仕様を定義できるが,複雑なビジネスロジックについては,対応するコンポーネントをITの専門家に依頼する必要がある。そのため,フレームワークのアーキテクチャは,このような追加のコンポーネントを容易に組み込めるような構造にしておく必要がある。

われわれの最近の研究試作[13]では,シンプルなマッチングアルゴリズムと実用上の最小限の要求仕様を想定した。ボランティア支援プロジェクト登録後にボランティアがプロジェクトを選択する場合は,プロジェクト管理者が速やかにボランティアの受け入れ可否を決定する仕様とした。また,ボランティア希望者が登録されている状態でプロジェクト管理者がボランティアを選択する場合は,選択されたボランティアがその要請の受諾可否を速やかに回答する仕様とした。このフレームワークはJavaEEで実装され,ボランティア支援システムの開発に適用した結果,全体のプログラムサイズ10,257行のうちの85.8%にあたる8,802行はフレームワークのソースが再利用された。

## 6. まとめ

エンドユーザである業務の専門家が,自ら必要とするWebアプリケーションを開発するためには,ビジュアルなユーザインタフェースとデータベースの定義に加えて,アプリケーション固有のビジネスロジックの定義が重要であるが,ドメイン特化型フレームワークを用いたビジネスロジックの定義方法に関して,プ

ログラミングスキルをあまり必要としない技術では実際の適用範囲が狭くなるというトレードオフ問題がある。本稿では,プログラミングの記述粒度を要求仕様定義レベルに保ちながらビジネスロジックのより詳細な定義を可能として適用性を改善するために,ビジネスロジックの定義を容易化するテンプレートを導入し,その有効性を示した。

## 文 献

- [1] M. Burnett, End-user software engineering and why it matters, in End-User Computing, Development, and Software Engineering, A. Dwivedi and S. Clarke, Eds. IGI Global, pp.185-201, 2012.
- [2] T. Chusho, F. Zhou and N. Yagi : End-user-initiative development with domain-specific frameworks and visual modeling, The 10th International Conference on Software Methodologies, Tools and Techniques (SoMeT\_11), pp.57-71, Sep. 2011.
- [3] 中所武司, エンドユーザコンピューティング - ソフトウェア危機回避のシナリオ -, 情報処理, 32, 8, pp.950-960, Aug. 1991.
- [4] 中所武司, 業務の知識を有するエンドユーザ主導のアプリケーション開発技法 - フレームワーク・ドメインモデル・サービス連携 -, 電子情報通信学会 知能ソフトウェア工学研究会 KBSE2007-30, pp.19-24, Nov. 2007.
- [5] 中所武司, エンドユーザ主導開発のためのビジネスロジックの定義方式の提案, 電子情報通信学会 知能ソフトウェア工学研究会 KBSE2012-17, pp.31-36, July 2012.
- [6] 中所武司, マッチングシステムを例題としたエンドユーザ主導開発方式に関する考察, 電子情報通信学会 知能ソフトウェア工学研究会 KBSE2014-28, pp.1-6, Nov. 2014.
- [7] 中所武司, ソフトウェア工学 (第3版), 朝倉書店, 2014.
- [8] 橋本竜太, 中所武司, エンドユーザ主導 Web アプリケーション開発技法の提案と評価, 第11回情報科学技術フォーラム, B-033, Sep. 2012.
- [9] 飯山大樹, 中所武司, 特定問題領域に特化した Android アプリケーション自動生成ツールの構築と適用評価, 電子情報通信学会 知能ソフトウェア工学研究会 KBSE2014-53, pp.7-12, March 2015.
- [10] J. Li and T. Chusho, A web application framework for end-user-initiative development with a visual tool, Proc. The 2012 IAENG International Conference on Software Engineering (ICSE'12), pp.816-822, Mar. 2012.
- [11] 許 杰, 中所武司, エンドユーザ主導の Web アプリケーション開発方式の試作 - ビジネスロジックの定義と実装方式 -, 電子情報通信学会 知能ソフトウェア工学研究会 KBSE2013-81, pp.13-18, March 2014.
- [12] C. Scaffidi, et al., Estimating the numbers of end users and end user programmers, in Proc. The 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, IEEE, pp. 1-8, 2005.
- [13] 横井悟, 中所武司, エンドユーザ主導開発を前提としたマッチングシステム構築技法 - ボランティア活動支援サービスを例題として -, 情報処理学会 第77回全国大会, 1L-08, Mar. 2015.