マッチングサービスのための市民参加型システム構築技法の研究

中所武司

Takeshi Chusho

１．はじめに

インターネット上で Web アプリケーションが普及し，クラウドコンピューティングが注目されるなど，ソフトウェアのサービス化が促進されている．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．

特に本研究においては，多種多様なビジネスロジック（業務規則）に関して，サービス提供側の視点に立ったビジネスロジック定義が重要という視点から，最近需要が増大しているマッチングサービスを取り上げた．

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．

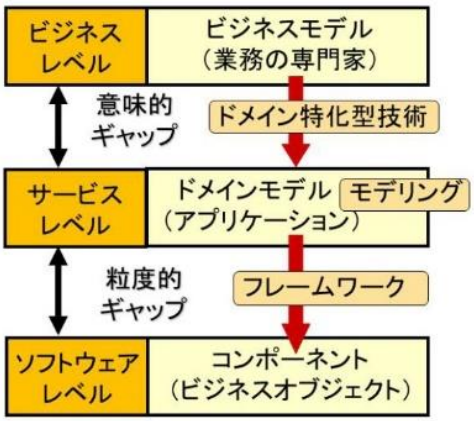

図１．エンドユーザ主導開発のアプローチ

３．ドメイン特化型技術

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．

４．ドメイン特化型フレームワークの適用性

以下の図２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，適用範囲は広くなる．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．

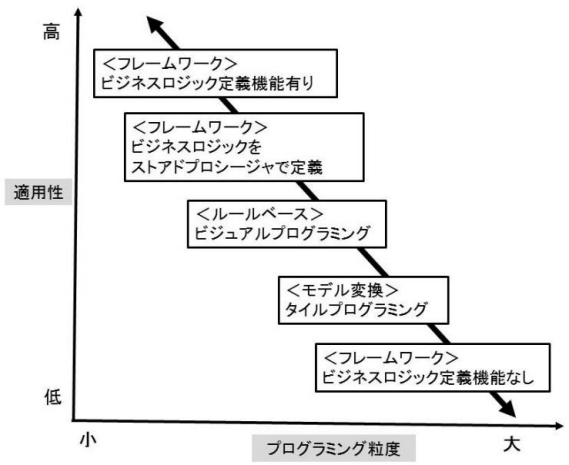

図２　エンドユーザ主導開発技術の適用性

この図２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．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２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したものである．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．

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・削除操作用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２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．

エンドユーザはビジュアルツールを用いて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２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．

適用実験に用いた図書管理システムは，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２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２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．

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る．そこで，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

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２で示したトレードオフ問題を改善できると思われる．

文献

1) Takeshi Chusho：Applicability of Domain-Specific Application Framework for End-User Development, 7th International Conference on Internet Technologies & Society 2016 (ITS 2016), pp.27-34 (Dec. 2016).
2) 中所武司：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 、 電子情報通信学会 技術研究報告 Vol.116, No.284, 知能ソフトウェア工学研究会 KBSE2016-28, pp.25-30 (Nov. 2016).