

エンドユーザ主導開発におけるビジネスロジックの実装方式の試作 The implementation of Business logic for End-User-Initiative Development

許杰 †

中所武司 †

Xu Jie

Takeshi Chusho

1. はじめに

今、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。エンドユーザになじみのある Web ブラウザをユーザインタフェースとする Web アプリケーションに関して、我々の過去の研究では、コンポーネントベースの技術としてのアプリケーションフレームワークとビジュアルモデリング技術を用いて、ユーザインタフェースおよび比較的簡単なデータベースを構築する方法を実現してきた。しかしながら、ビジネスロジックに関しては、頻繁に変更を伴う保守に対応することが重要になるという観点から、ビジネスモデルの実装方式を研究してきた。

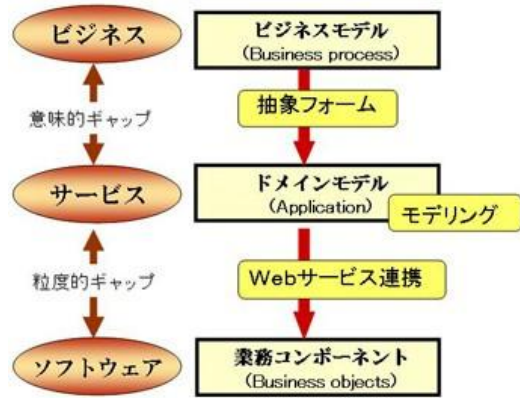


図 1. エンドユーザ主導開発のアプローチ

2. エンドユーザ主導開発

基本的なアプローチを図 1 に示す。UI、ビジネスロジック、DB を基本的な構成要素とするアー

キテクチャを前提にアプリケーションを作成する。本研究を通じてのエンドユーザ主導開発の主要な課題として、多様なビジネスロジックの定義方法とその実装方式があげられる。これまで、スクリプト言語の開発、ルール表現の導入などを試みてきたが、プログラミングの概念は必要である。最近では、特定分野対応のタイルテンプレートを組み合わせて業務処理を記述する方法で問題の解決を図ったが、十分なテンプレートの集合を用意するという課題はある。本報告では、データベースと関連付けたビジネスロジックの実装方式について述べる。

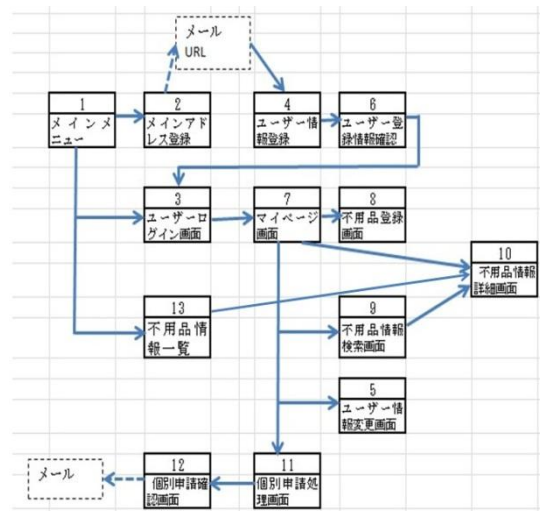


図 2. 不用品再利用システムの画面遷移図

3. 2種類の基本的アーキテクチャ

3.1 例題の概要

今回、ビジネスロジック分析のため、不用品再利用システムを例題としてとりあげ、C#で試作した。昨今、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

† 明治大学理工学研究科基礎理工専攻情報科学系ソフトウェア工学研究室、Meiji University

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やローカルに運営される不用品再利用のための中古物品の販売店に関して、もしその担当者自身が Web サイトとして立ち上げることができれば、大きな効果が期待できる。構築方式は二つ案があって、其々は従来方式と MVC モデル方式である。図 2 は不用品再利用システムの画面遷移図であり、以下の機能を含む。

- (1) メールアドレスにおける本人チェック
- (2) ユーザ登録及び情報変更
- (3) 不用品の登録及び更新
- (4) 不用品の検索及び申請
- (5) 受け取り希望者の決定処理

3.2 基本設計

従来方式は、ASP.NET の開発環境において 3 層アーキテクチャを採用しており、最も典型的な構築方式である。図 3 は従来方式を利用したシステムの構成図である。ASP.NET では HTML に出力する GUI 要素とページの制約処理は別々のファイルに分割することが推奨されている。

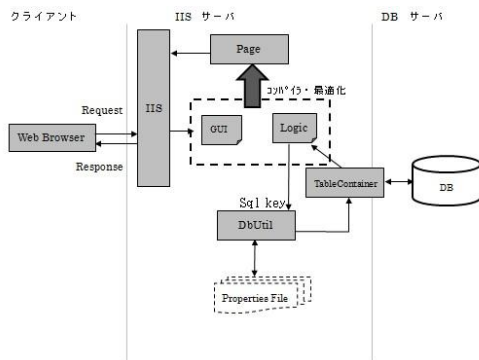


図 3. システム構成図(従来方式)

3.3 MVC モデルによるシステムの構築

近年、Web アプリケーションの開発技術が発展してきて、従来方式の ASP.NET Web Forms に代わり、ASP.NET MVC モデルをベースにした Web 開発が盛んになってきた。MVC モデルベースの不用品再利用システムのシステム構成図を図 4 に示す。

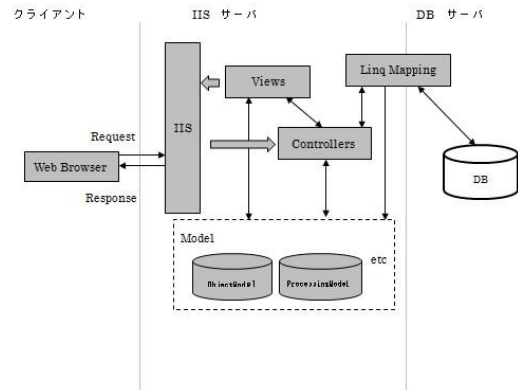


図 4. システム構成図(ASP.NET MVC)

3.4 ビジネスロジックの考察

不用品再利用システムにおいては不用品の申請に対する受付処理があり、流れ図は図 5 のようになる。この例からも、判断が多くなるとビジネスロジックの複雑さも高まっていくことがわかる。

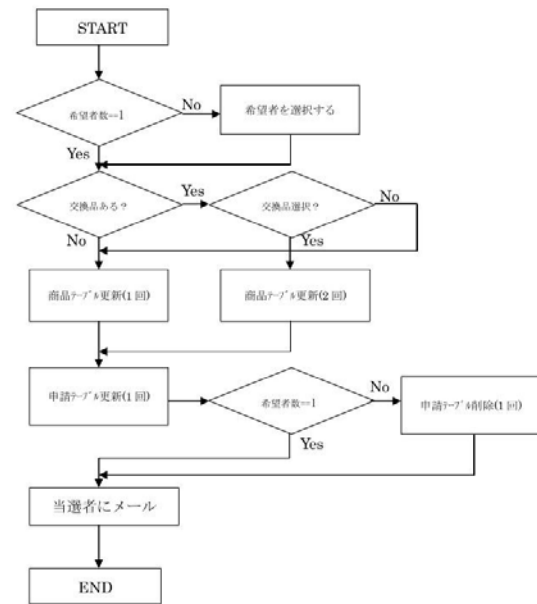


図 5. 申請及び確定の仕組

4. ストアドプロシージャによる記述

以上に述べてきた 2 種類のシステムの構成に関して、エンドユーザによるビジネスロジックの定義を支援するという観点でみると、ビジネスのロジック処理は、従来方式と MVC モデル方式のいず

れの場合も、ビジネスロジックの処理はあらゆるところに含まれている。

本研究においては、ビジネスロジックを反映しやすいデータベースの処理に注目し、データベースプログラミングを1つの支援方式として取り上げる。代表としての技術はストアードプロシージャである。ストアードプロシージャの利点は頻繁に DB のアクセスを行う処理から解放する。手続き型ステートメントとセット指向ステートメントの両方を組み合わせたものであり、極めて複雑なビジネスルールにも対応できる。複雑すぎて他の制約では実現できなかったものや、セット指向的な処理が混在しているものなども、ストアードプロシージャの形で実装できる。

4.1 典型的なビジネスロジックの説明

例題として取り上げた不用品再利用システムでのビジネスロジックの定義を考えると、システムの利用者や取り扱う対象や処理の制約などをサービス提供者（エンドユーザー）が検討していくものである。ここでは、人やシステムに対して特定の条件下でアクションを引き起こすもので、「もし～ならば～する」と表現される典型的なビジネスロジックとして、次の3つを取り上げる。

- (1) もし、登録者がすでに不用品登録数の上限(例えば5個)の不用品を登録していれば登録を断る。
- (2) もし、一定時間経過後にある品物を希望するユーザーが複数であればリストから当選者を決める処理を行う。
- (3) もし、不用品登録後、一定期間内(例えば3か月)経過したものがあれば削除処理を行う。

この三つの例について、まずはプログラムによる実装方式を考えて見よう。一番目の例はユーザーの登録制限であり、プログラムの処理においては、まずそのユーザーに関連したデータをデータベースから取り出して登録数をチェックし、5個以上の場合は登録不可の処理を行い、5個未満の場合は

登録処理を行う。DB のアクセスはステップの数と比例している。その次の2番目と3番目の例はデータベースの一括処理を要求していると考えられるので、実際にプログラムの実装手順は其々以下の通りである。

希望者のリストから当選者を決める処理

STEP1:希望者のリストを作る

STEP2:ランダム関数を利用して当選者を決める

STEP3:当選者に送信する、当選者以外の人にも送信する

STEP4:当選者の申請情報を更新、当選者以外の人
の申請情報を削除する

不用品登録後、一定期間内(例えば3か月)経過したものを削除処理は次のようになる。

STEP1:登録日付をチェックし、削除リストを作る

STEP2:削除リストを確認し、削除処理を行う。

ステップ毎にDBアクセスを行う必要があるので、ビジネスロジックによるステップ数が増えれば増えるほど、DBアクセスの頻度も上がっていく。そこで、ビジネスロジックを反映するビジネスルールの処理をデータベースに持っていった場合について、次に検討する。

4.2 ストアドプロシージャ

ストアードプロシージャを利用してビジネスロジックを実現する。4.1 に示した提案の一番目は DB 側に持って行くと、その実装内容は以下のようになる。

実装の例:

```
ALTER PROCEDURE
[dbo].[ap_CheckNumberGoodsbyUser]
@count INT = 0,
@user_No varchar(50),
@IsAlowed Bit OUTPUT
AS
BEGIN
SET NOCOUNT ON;
```

```

select @count = COUNT(*)
from dbo.User_Tbl u, dbo.Goods_Tbl g
where g.User_No = u.User_No
And u.User_No = @user_No
if @count >= 5
    begin set @IsAllowed = 0
        select * from dbo.Goods_Tbl
        where user_no = @user_no
    end
else set @IsAllowed = 1
END

```

開発環境はSQLServerであるので、CLRデータベースオブジェクトを作成してプログラムの統合を行う。CLRは(Common Language Runtime: 共通言語ランタイム)の略である。

プログラムの代わりに処理をDBに持っていくことによってプログラムの処理は軽くなってその後のメンテナンスもより容易になる。SEにとっては簡単でありがたい話だと思う。エンドユーザ主導の立場から言うと、これからいくつかの解決しなければならぬ課題が存在している。

4.3 ストアドプロシージャの自動生成

エンドユーザ(業務の専門家)は通常の業務にパソコンを利用しているが、プログラミング言語の知識はない。そこで、ストアドプロシージャの自動生成及びエンドユーザによるカスタマイズ化を実現することが必要になる。その方式として、今回のストアドプロシージャの記述例の考察に基づき、関数の概念や論理構造(判断, ループ)などを合わせてコンパイラツールを作成する方式などが考えられる。

5. おわりに

本稿では、エンドユーザ主導開発におけるビジネスロジックの実装方式の研究の一環として、2

つの実装方式を採用し、不用品再利用システムを例題とするアプリケーションを試作した。しかし、ビジネスロジックはいろいろな理由でプログラムのあちこちに存在しているため、新たにストアードプロシージャプログラミングの提案をした。そうすることによって、技術課題を特化し、エンドユーザ主導開発の実現可能に近づけると考えられる。

今後は、4.3に述べたストアードプロシージャの自動生成を実現するために、ビジュアルツールの開発を検討していく。

参考文献

- [1] 李静、中所武司、エードユーザ向け Web アプリケーションフレームワークの提案と試作 FIT2011 第10回情報科学技術フォーラム、B-013 (Sep. 2011)
- [2] 佐藤修、エンドユーザコンピューティング、日科技連出版社 (July 1996)
- [3] 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 技術研究報告 Vol.112(July 2012 予定)
- [4] 中所武司、Green-by-IT の観点からの地域活性化に関する要求分析、情報処理学会ソフトウェア工学研究会要求工学ワーキンググループ ワークショップ (Jan. 2012).
- [5] 中所武司、特定分野向けフレームワークにおけるビジネスロジックのカスタマイズ機能に関する考察、情報処理学会ソフトウェア工学研究会要求工学ワーキンググループ ワークショップ (June 2011).
- [6] Dino Esposito、プログラミングMicrosoft ASP.NET MVC ASP.NET MVC 3対応版 (マイクロソフト公式解説書) 日経BP社 (2012/5/8)
- [7] デヤン・サンデリック、SQL Server 2005 ストアドプロシージャプログラミング (SQL Server Books) 翔泳社 (2007/3/21)