

## 再利用可能な Web ビジネス情報システムの設計について

浅羽義之 松永賢次

専修大学経営学部

〒214-8580 神奈川県川崎市多摩区東三田 2-1-1

{h090021, matunaga}@isc.senshu-u.ac.jp

Web を利用したインターネットビジネス情報システムが急速に普及している。その開発には、迅速性と頑強性が強く要求されている。その要求を満たすためにコンポーネント技術を利用できるか事例研究を行ってきた。BtoC ビジネスにおいて適切な汎用コンポーネントを抽出できることがわかってきた。本稿では、そのような汎用コンポーネントを簡単に再利用できるよう、デザインパターンドキュメントを拡張したドキュメント記述法を提案した。

また、Web でのビジネス情報システムでは、HTML によるユーザインターフェース(UI)を、迅速に見栄えのよいものにする 것도重要である。UML のコラボレーション図に UI 情報を記述するように拡張し、それからHTMLファイルの自動作成を行うようにした。研究室で開発したデザインツールを用いて、ビジネスイメージに適合した UI になるように修正することができる。

## Reusable Design for Web-Based Business System

Yoshiyuki ASABA and Kenji MATSUNAGA

SENSHU University

2-1-1 Higashi-mita, Tama-ku, Kawasaki, 214-8580 JAPAN

Web-based business systems have been commonly used. Those systems require robustness and need to develop speedy. For satisfying these requirements, we have studied whether the component technology can be applied to Web-based business systems.

In this paper, we propose a method of descriptions for Generic Component catalogs. The descriptions are similar to Design Pattern and can be understood to reuse for other systems. Furthermore we show how to generate well-designed User Interface from collaboration diagrams.

## 1. はじめに

近年、インターネットの普及により、インターネットの Web を利用した、オンラインビジネス情報システムが急速に普及している。その開発には、一般的なビジネス情報システム開発と比較して、迅速性と頑強性がより強く要求されている。その要求を満たすために、コンポーネント技術[1]を利用できるか事例研究[2]を行ってきた。BtoC ビジネスを対象と情報システムを設計したところ、適切な汎用コンポーネントを作成することができた。それを再利用して新たな情報システムを作成してもらったところ、コンポーネントのインターフェース情報だけでは、開発者がどのように再利用できるかわかりにくい、という問題が生じた。

そこで、コンポーネントを再利用できるように、コンポーネントのドキュメントの作成法について研究を行った。一般的な開発者が理解できるように、自然言語で記述し、事例を含むようなものにする必要がある。また、後で検索が容易になるように、対象領域での専門用語を含む必要もある。

また、ユーザインターフェース作成コストの削減も重要なテーマである。オンラインビジネス情報システムのユーザインターフェースとなる HTML ファイルを、設計ドキュメントである UML のコラボレーション図から生成する方法について述べる。それを基に、研究室で開発したツール[3]を用いて自動的に適切なレイアウトにデザインできることを述べる。

## 2. 研究の背景

インターネット環境が整備され、企業も個人もインターネットを様々な方法で利用するようになってきている。企業においてはネットワーク上での商取引やマーケティング活動を行う機会が増大し、それがうまくいかどうかにより、企業の生死がかかっている状況となっている。そのため、オンラインビジネス情報システムの開発が急務となっている。

### 2.1. コンポーネント技術

システム開発を短縮するためには、開発工程のどこかを短縮する必要がある。しかし、人と人との間で行う必要のある要求定義の期間を変えることは難しい。そのため、コーディングなどの「対システム」の工程を短縮せざるを得ない状況になる。プログラムのモジュール化、ライブラリの提供、オブジェクト指向、そして、最近では、コンポーネント技術といった、ソフトウェアの再利用技術が取り入れられてきた。

コンポーネントは、より大きな単位でソフトウェアの再利用をはかる技術であり、その実体は複数のオブジェクトをまとめたものやアプリケーションなど幅広いものである。コンポーネントを再利用することによって工程期間の短縮を可能となると言われているが、以下にあげる問題がある。

- (1) コンポーネントを誰にでも再利用できるように、コンポーネントのカスタマイズを受け入れる設計/実装が必要である。

オンライン上の簡単な本屋システムを実際に構築し、汎用性のあるコンポーネントと開発するシステムに特化したコンポーネントを作成してみた。そして、汎用コンポーネントを、オンライン上の航空チケット販売システムを開発に再利用し、コンポーネントの調査を行った[2]。

その結果、図1のように、システムに特化した部分をコンポーネント利用者側に定義してもらい、それを汎用コンポーネントに結びつけて、ある程度のカスタマイズを実現することがわかった。つまり、コンポーネントは、フレームワークの提供と考えることもできる。

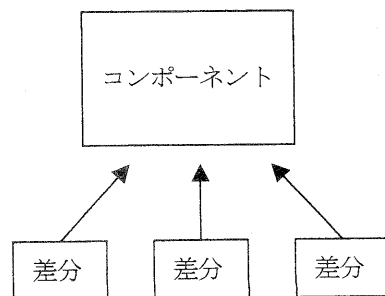


図1.コンポーネントのカスタマイズ

- (2) 専門用語に偏りのない詳細なドキュメントの作成が必要である。コンポーネントは、特定の対象向けに作成される。その対象に固有の用語のみで記述されると再利用しにくくなる。また特定技術に依存した記述に偏ることも避けなければならない。誰にでもわかるような用語で記述し、補足として専門用語を含むことが必要である。

本稿では(2)に着目し、用語の差異を吸収できるようなドキュメントの作成法に関する提案を3節で行う。

## 2.2. ユーザインターフェースの作成

BtoC のオンラインビジネス情報システムを頻繁に利用してもらうためには、利用者をひきつける、対話的で見映えのよいユーザインターフェースも不可欠となっている。その開発期間も短くしなければならない。

現在では、専門的な Web ページデザイナーが UI を作っている。情報システムのインタラクションが変更になると、デザイナーに Web ページをデザインし直してもらわなければならない。この部分をできるだけ自動化することを考える必要がある。それに関する提案を4節で行う。

## 3. コンポーネントのドキュメント形式

コンポーネントをどのようにドキュメント化するかが、よい再利用を行うために重要である。コンポーネントとは何か示すためには、コンポーネントそのものをどう呼び出すか、コンポーネントから呼び出される関数やオブジェクトは何かなどを記述しなければならない。これまでに、形式仕様を用いたドキュメント作成の研究がなされてきている[4]。形式仕様による方法は、一般的な開発現場で受け入れられるか疑問がある。

構造化されたドキュメントに、自然言語や事例を用いて記述する方法を開発したほうが普及しやすいと考える。利用する側にとっては、得たい情報をどの場所書いてあるかすぐにわかる。また、ドキュメントを書く側にとっても、どの情報をどのような形式で書くか指針を示されることによ

って、優れたドキュメントを書くことが可能になる。

このようなドキュメントとして成功しているものとして、デザインパターンのカタログ[5, 6]をあげることができる。デザインパターンのカタログは、設計上でよく起こる問題に対する設計方法を記述したもので、設計を再利用するためのドキュメントである。カタログは単なる設計の紹介だけでなく、設計の名前、どういう状況で使えばいいか、適用事例、デザインパターンを使用することによるトレードオフなども記述する。そこで、デザインパターンのカタログを参考にし、コンポーネントのドキュメントに合わせたセクションを検討し、必要なセクションを以下のものとした。

- コンポーネントの名前
- コンポーネントの概要
- 適用事例
- コンポーネントの構造
- 実行環境と使用言語
- 業種ごとの用語集
- その他

コンポーネントのドキュメントを、過度に汎用的なものすると再利用が難しくなる。そこで適用事例や業種ごとの用語集などに、専門用語を入れることにより、個別用途で利用するためにコンポーネント検索したとき、ヒットするようにできる。また、用語集は新たな業種の専門用語に対応できるようにするため、後から追加できるようにする。

以下ではそれぞれのセクションを説明する。また、資源管理を例に、それぞれのセクションにどのような記述をするのか簡単に書いていく。

### 3.1. コンポーネントの名前

コンポーネントの本質を簡潔に連想させるものを記述する。あるドメインに特化しない名前を付けることは、再利用のためには重要である。特定の人間だけにしかわからないということにはならないようにするためである。

例: 資源管理コンポーネント

### 3.2. コンポーネント概要

おおまかに、コンポーネントを使う目的とコンポーネントが解決する内容を記述する。それにより、再利用する際に、このセクションを読み、ある程度再利用に適合するかを判断する。

### 3.3. 適用事例

コンポーネントをどのような状況で再利用できるか、具体的な例を示して、利用者側にその状況を認識してもらう。基本的には、開発者がコンポーネントを作成した状況を書き、再利用する人のためにコンポーネントのメリットや問題点を具体的に記述する。

例:

- オンライン上の本屋システムでの書籍の在庫管理にコンポーネントを利用した事例
- チケット販売システムの座席管理コンポーネントを利用した事例

### 3.4. コンポーネントの構造

設計時のコンポーネントのクラス構造やインターフェースを示し、それぞれのクラスやインターフェースの動作について説明する。クラス構成について公開したくない場合は、コンポーネントのインターフェースのみ記述する。クラス構成を示すためにはUMLを用いることを標準とする。

インターフェースは2つのサブセクションに分けて記述する。1つは、コンポーネントの機能にアクセスするため(フローズスポット用)のインターフェースであり、もう1つは特化コンポーネントと結び付けるため(ホットスポット用)のインターフェースである。

### 3.5. 実行環境と使用言語

コンポーネントが動作する環境とコンポーネントを作成した言語の説明をする。動作環境には、OS (UNIX, Windows, Mac, その他) や開発環境 (Visual Basic など) を書く。使用言語には、Java, Ruby, C++ などの実装した言語を記述する。

### 3.6. 業種ごとの用語集

コンポーネントの名前に対する業種ごとの専

門用語を記述する。コンポーネントの開発者が記述するよりも、利用者が、使用した業種とその用語を追加していくような形をとるようになる。

例: 在庫管理, 座席管理, 人事管理

### 3.7. その他

コンポーネントは、基本的にソフトウェアなので、開発者の情報、ライセンスなどを記載する必要がある。そこで、どこのセクションにも当てはまらない情報を、このセクションに書く。

## 4. ユーザインターフェース(UI)の生成

Web を用いた情報システムでは、ユーザとのインタラクションは一連の Web ページでなされる。設計ドキュメントから、HTML で記述した UI プロトタイプを生成し、それを基に、デザインツールで適切な見映えとなるページに仕上げる。それにより、UI 開発期間の短縮、情報システムの変更に対する UI の自動変更、プロトタイプテストが可能になる。

### 4.1. 生成手法

UI を伴う処理を、設計段階で開発者に記述してもらうようにした。具体的には、コラボレーション図のメッセージに UI 情報を付加して明示してもらう。UI 生成ツールは、コラボレーション図を、XMI (XML MetaData Interchange) 形式に変換し、その後、それを解析して HTML による Web ページを生成する。XMI とは OMG が提唱している UML モデルをツール間で交換するフォーマットである。生成手順を図示したものが図 2 である。

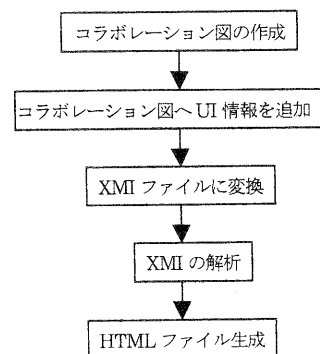


図 2. 生成手順

#### 4.1.1. XMI 解析

XMI でのコラボレーション図のデータは  
<Collaboration XMI.id="\*">

.....

</Collaboration XMI.id="\*">

のように、Collaboration タグの間に書かれる。メッセージ名は、message タグの間に書かれる。XMI を解析してこれらの情報を得る。

#### 4.1.2. UI ウィジェットのタイプ

HTML の FORM で使うウィジェットにはテキストフィールド、パスワードフィールド、チェックボックス、ラジオボタン、ポップアップ、テキストエリアがある。ウィジェットのタイプの名前を表 1 に示す。

タイプ	ウィジェット名
TF	テキストフィールド
PF	パスワードフィールド
PU	ポップアップ
CB	チェックボタン
RB	ラジオボタン
TA	テキストエリア

表 1. ウィジェット一覧

#### 4.1.3. メッセージ形式

ウィジェットを伴うメッセージをコラボレーション図に書くには、以下の形式を用いることにした。

message(#ウィジェット数ウィジェットタイプ#引数...)

ウィジェット数は、生成したいウィジェットの数を指定する。ウィジェットのタイプは、表1の中のタイプを指定する。これを基に、HTML ファイルを生成していく。

#### 4.2. UI ファイルの形式と生成

コラボレーション図ごとに1つのHTMLファイルを生成する。また、これらのHTMLを参照するための、インデックスファイルも作成する。

##### 4.2.1. UI プロトタイプファイル

コラボレーション図ごとに XMI 記述を解析し、UI プロトタイプとなる HTML によるフォームを生成する。最初にメッセージ名を抽出する。次にメッセージを参照し、ウィジェットの数、ウィジ

ェットタイプ、引数名を調べ、プロトタイプファイルに記述する。引数名はラベル名に変換する。最後にフォームの情報を送るための SUBMIT ボタンと RESET ボタンを配置する記述をする。

##### 4.2.2. インデックスファイル

コラボレーション図の名前と UI プロトタイプファイルとの表を作成し、プロトタイプファイルへのリンクを示す HTML ファイルを生成する。

#### 4.3. Web デザインツール

Web ページは、ビジネスを行う企業や対象とする相手に応じて適切なデザインを行う必要がある。本研究室では、Web ページのデザイナーが、作りたい分野の Web サイトのデザインを適切に提案し、ユーザが DynamicHTML を用いて自由にカスタマイズすることを支援するツールを開発してきた。これは、ユーザの要求を満たすレイアウトを作成するレイアウトツールと、レイアウトツールから生成されたページを加工するデザインツールの2つからなる。

4.2 で述べた UI プロトタイプを、レイアウトツールに与えることで、適切な配置、配色、マウスイベントを付加した UI に改善することができる。さらに、デザインツールを活用することで、細かい調節が可能である。

#### 5. まとめ

本稿では、Web ビジネス情報システムを事例として研究を進めてきた、再利用を可能とするためのコンポーネントドキュメント記述手法、そしてインターフェースを容易に作成するための設計記述と生成手法について述べた。

コンポーネント記述では、業種ごとの専門用語を追加できるようなドキュメント形式を使用することによって、ドキュメントの拡張が簡単になり、また管理も楽になった。その他にも、名前やキーワードだけでなく、コンポーネントの概要や適用事例を参照することによって、コンポーネントの再利用の判断ができ、ますます、再利用への道が開かれていくものとなった。

UI 生成に関しては、UI 生成による開発負担の軽減だけでなく、デザインツールを使用する

ことによって、ビジネスイメージに適合した効果的なデザインを提案するので、マーケティングの面でも支援することが可能となった。生成したファイルを基に、テストやそのまま本番用のインターフェースとして使用することもできる。

今後の課題として、ドキュメントの作成支援ソフトとドキュメントのキーワード検索支援ソフトを開発し、コンポーネントの再利用を促進する環境の整備を行いたいと考えている。ドキュメントをXMLに形式化しレポジトリに登録することによって、コンポーネントとドキュメントの管理を行いたい。まだ、再利用の技術があっても、再利用する環境が整備されていないので、再利用が行われていないのが現状である。その点を着目して引続き研究を行っていく。

#### 参考文献

- [1] D'Souza, Wills: Objects, Components and Frameworks with UML, Addison Wesley, 1998.
- [2] 浅羽義之, 松永賢次: ビジネスコンポーネントによる Web ビジネスシステムの構築, 情報処理学会第 61 回全国大会, 3S-5, Oct. 2000.
- [3] 小村文人, 松永賢次: DynamicHTML を使用する Web ページのためのデザイン支援, 情報処理学会第 61 回全国大会, 5T-5, Oct. 2000.
- [4] 吉田和樹, 本位田真一: コンポーネントベース・フレームワーク技術を利用したアプリケーションの形式的仕様記述, 情報処理学会ソフトウェア工学研究会, 129-15, Nov. 2000.
- [5] Mark Grand: Patterns in Java Volume 1, Addison-Wesley, 1998.
- [6] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.
- [7] Object Management Group: XML Metadata Interchange (XMI) V1.1, Nov. 2000.