2003－ＧＮ－49　(10)
2003／10／23

再利用性・開発容易性を考慮した入れ子可能テンプレートベース Web アプリケーション開発環境の構築と評価

田中 充　　勅使河原 可海

同期・非同期統合型 Web グループウェアを柔軟に構築するための開発環境の中で，主に非同期型の構築方法について論じ，その評価を行った．この環境の特徴は，入れ子的な HTML テンプレートの積極的な利用，メッセージ層と単語層への分離による再利用性・柔軟性の高い HTML テンプレートの提供，Web ブラウザでの編集環境，動的なページ生成を行うための豊富なカスタムタグ，データベースアクセスのための MVC の自動生成機能等の提供である．これまで設計してきたテンプレートの分析および利用者アンケートの結果から，これを活用することにより，再利用性が高く開発が容易な Web アプリケーションの構築が可能になることが分かった．

Implementation and Evaluation of a Nest-able and Template-based Web Application Development Environment for Considering Reusability and User-friendliness

Michiru Tanaka　　Yoshimi Teshigawara

In this paper, an asynchronous type Web application development environment and its methods in the development environment for building asynchronous and synchronous integrated Web groupwares flexibly are described and evaluated. The features of the environment are to promote making and using nest-able HTML templates and to provide highly reusable and flexible HTML templates by separating into a message layer and a word layer, editing tools for Web browsers, plenty of custom tags, tools for producing MVC codes automatically, and so forth. As the results of analyzing designed HTML templates and the questionnaires to users, it is found that making use of the proposed environment enables them to build Web applications with the high reusability and user-friendliness.

1. はじめに

我々は，リアルタイム系と非リアルタイム系のグループウェアの統合的環境構築を目指して同期・非同期統合型会議システム ASSIST(ASynchronous and Synchronous Integrated multimedia conferencing System for Teams) の利用対象の検討及び設計・開発を行ってきた[1][2]. 同期・非同期統合型グループウェアは，一般の会議から，ソフトウェア開発会議，教育支援システム等多様な目的に適用できると考えられる．

また，管理・保守コスト等を考慮すると提供されるシステムはクライアント側にソフトウェアのインストールを必要としない Web アプリケーションとして提供することが望ましい．しかしながら，Web アプリケーションはその通信形態が一般的にプル型であるために，同期的な処理には向かない．そこで，我々は別フレームもしくは別ウィンドウ上でポーリングを行うこと等によって同期的な処理を行うことにした．

同期・非同期統合型の Web グループウェアを柔軟に効率的に構築するためには，共通する機能や HTML ページ等のコンテンツ管理を含めた開発環境が望まれる．現在，PHP，Struts，Zope 等 Web アプリケーションを構築するための様々な言語やフレームワーク等の開発環境が登場してきているが，ほとんどが非同期型の Web アプリケーションを対象としているため，同期・非同期型統合の観点からの細かい個別の要求に答えられない可能性が大きい．

そこで，我々は独自の Web アプリケーション開発環境を構築することにした．現時点では主に非同期型用の開発支援機能が実現されている．特に再利用性と開発容易性を考慮して，入れ子可能なテンプレートが積極的に利用でき，MVC(Model-View-Controller)プログラミングモデルにおけるビューのメッセージ層と単語層への分割やデータベースアクセスのためのフォームの新規，更新，表示間での再利用を可能とし，更に Web ブラウザのみでテンプレート，データベース，ファイルの編集管理が可能な統合的開発環境を構築した．

本稿では，同期・非同期統合型グループウェアの開発環境うち主に非同期型の部分に焦点を当て，その再利用性，開発容易性を考慮した開発環境の構築と評価について論じる．

創価大学工学研究科
Graduate School of Engineering,
Soka University

2. Web アプリケーション開発環境 ADK

我々が提案している同期・非同期統合型マルチメディア会議システム ASSIST の用途は，一般の会議からソフトウェア開発，教育支援等多岐にわたり，その柔軟な開発が求められてきており，これに対応するため，図 1 に示すように開発モデルを 4 階層に分離した．

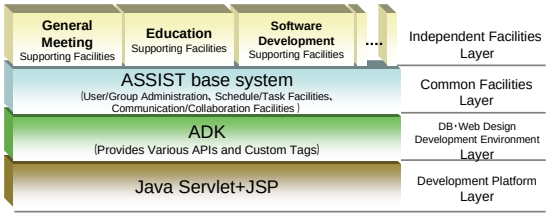

図 1　ASSIST 開発 4 階層モデル

第 1 層は，開発プラットフォーム層ということで Web アプリケーションサーバを意味している．現在の環境は Apache Tomcat4.1 がこれに該当する．第 2 層は, データベースや Web ベース開発を容易に実現するための開発環境を意味する. Java クラスライブラリによる API の提供やテンプレート言語の提供を行っている．第 3 層は，グループウェアの共通な機能を提供するものであり，ユーザ管理，グループ管理，電子メール，掲示板，スケジュール等の機能を提供する．そして，第 4 層では個別の用途に対する機能を実装していくというものである．

ADK とは主に第 2 層（広義の意味においてはユーザ管理機能等を活用するため第 3 層の一部を含む）に該当するものである．この ADK はそもそもの目的は同期・非同期統合に特化させる点であるが，通常の非同期型の Web アプリケーションを構築する目的として利用しても，その際利用性，開発利用性において多くの工夫がなされており，効率的な開発を促進することが期待される．次に，その開発環境の特徴と仕組みおよびそれを用いた開発手法について論じる．

2.1　特徴

ADK の現在実現されている特徴を表 1 に示す．最大の特徴は，Web ブラウザを用いて入れ子可能なテンプレートベースのビューを設計できる点である. また，そのビューの設計をさらに HTML，メッセージ，単語と独立分離させることができるため，多言語対応化，ビューの分業化等が可能となる点も他にはない設計開発環境との違いである．

2.2 ADK の開発プラットフォーム

ADK は，OS が Windows であれ Linux であれ J2SDK1.4，Apache Tomcat4.0，PostgreSQL6.5 以上の環境があれば動作することができる．データベースサーバに関しては，現在は PostgreSQL のみの対応であるが，設計当初から他のデータベースサーバを利用ができるように, 内部的には DBModule という独自のデータベースアクセス用クラスを利用しており，そのクラスを追加修正することによって，容易に他のデータベースへの対応ができるような工夫がされている．今後は MySQL 等への対応化を行う予定である．

表 1　現在の ADK の特徴

入れ子可能なテンプレートベースの HTML 設計
利用端末，表示スタイル別の切り替えが可能なテンプレート
HTML，メッセージ，単語への分離によるビューの設計
Web ブラウザによるファイル，テンプレート，メッセージ，単語等の管理が可能(豊富なコンテンツ管理機能の提供)
テンプレート言語 ANTL による動的な HTML の出力およびデータベースアクセスの容易化
豊富な Java クラスライブラリがそのまま利用可能
指定したデータベーステーブルから MVC のモデル, ビュー, コントローラを自動生成する機能の提供
データベースアクセス，メール送信，字句解析，ユーザ管理，グループ管理，画像管理，ファイル管理，ログ出力等を行うための豊富な Java クラスライブラリの提供

2.3 入れ子可能テンプレートと Web ブラウザによる編集環境

既存のテンプレート技術のほとんどは，ページ（ファイル）単位のテンプレートで，その中に動的な要素を配置して利用するものである．テンプレートを利用するためには，専用のツールを使うケースが多く，しかもファイルを配置し保存する手間がかかる．

一方，ADK によるテンプレートの考え方は，まず JSP(Java Server Pages)があり，その中でカスタムタグによって呼び出され展開されるものであり，タグによる入れ子的な展開が可能となっている．基本的な HTML の設計は, 設計権限を持つユーザのみにページ上に設計支援メニューを表示させ，保守モード時に各テンプレートの上部に編集指定用メニューを表示させ, Web ブラウザによる各テンプレートの編集が可能になっている(図 2).

なお, ADK の設計方針として Web ページの設計は, テンプレートを作成して行うことが前提となっている. テンプレートを用いても, 用いなくてもよいとなれば, 既存のテンプレート機能を持っている HTML エディタのように，慣れない利用者は，テンプレートを利用しない傾向になることが予想されるため，それをさけ再利用の可能性を高めるためである．

期待される効果は，次の点である．

- テンプレートベースの HTML デザインによる再利用性の向上
- 分かり易いテンプレートの配置関係
- ソースリストの見易さの向上
- 柔軟なアクセス権制御・排他制御
- テンプレート単位の領域の共有の実現
- メモリキャッシュの効率化と出力展開速度の向上

2.4　ANTL (ADK Nested Template Language)

ANTL とは，ADK によって生成される入れ子可能なテンプレートの内部の記述言語のことを指す．ANTL は，通常の HTML のタグの表記に加え，多くのカスタムタグを持つ．表 2 に現在実装されているカ

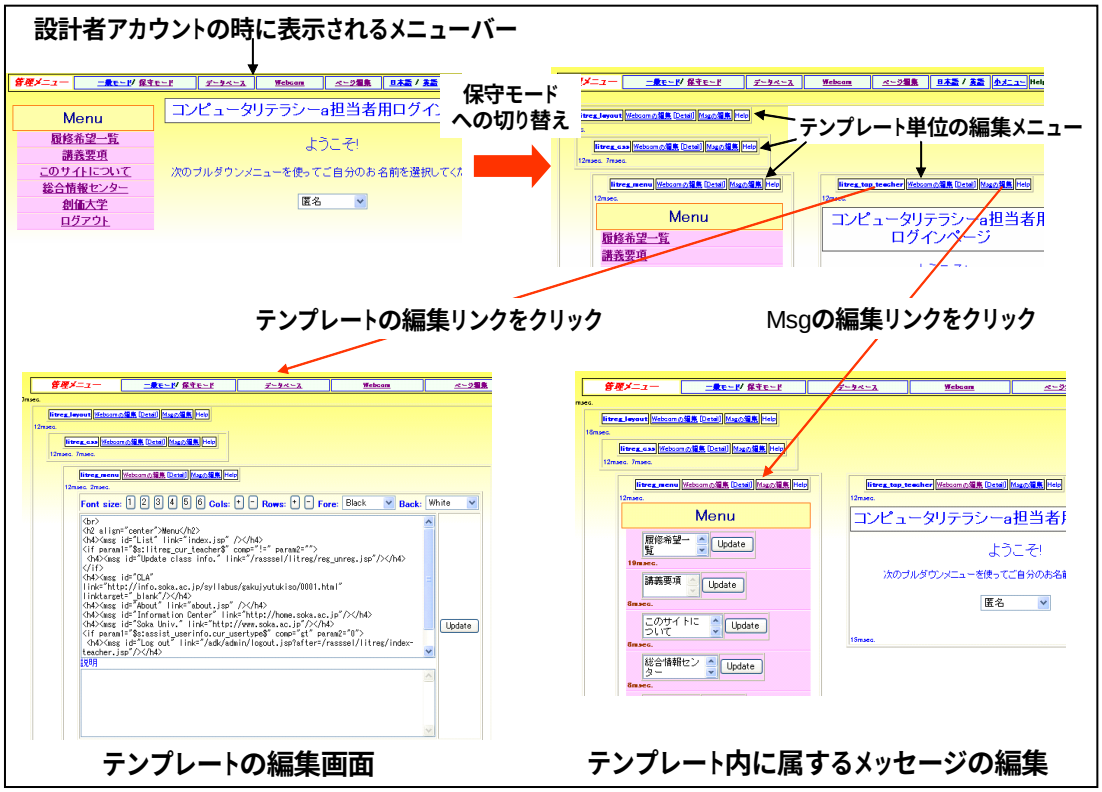

図 2 入れ子可能テンプレートとメッセージの Web ブラウザによる編集

スタムタグの一覧を示す．また，ANTL は，単語の辞書参照やタグのパラメータを動的に記述するための変数展開を行うことができる．

（1）<webcom />

テンプレートを参照するタグ．id 名でテンプレート名を指定する．このタグによって参照されるテンプレートは，その内部にさらに<webcom />というタグを用いて入れ子的に参照できる。一番上位レベルのテンプレートは，JSP ファイルのカスタムタグ<assist:webcom />によって呼び出される．テンプレート内部への動的要素の明示的な引渡しは param1～param10 というパラメータを用いることによって行うことができる． なお，もし対象の id 名が存在しない場合，自動的に新たなテンプレートを生成する．

（2）<webcomdb />

データベースの SQL の結果を代入するクエリオブジェクトを生成したり，そのオブジェクトを参照してテーブル表示を行ったり，テンプレートを元にレコード表示を行うためのタグ．

（3）<msg />

文章内容を参照するタグ．言語ごとにその内容を切り替えることができる．また query パラメータを使うとクエリオブジェクトの参照をすることができる．また変数内容を表示するパラメータも存在する．メッセージは図 2 のように Web ブラウザで編集可能である．

（4）変数やリクエストパラメータの内部展開

動的なページを生成するには，変数を展開して出力することが重要である．ADK では，ページ属性，セッション属性，アプリケーション属性，リクエストパラメータの変数展開ができるようになることが望ましい．そこでタグ内部のパラメータ上に”$.変数名$”という文字シーケンスがきたらその内容を変数として展開するようにした．

表 2 ANTL のタグ

タグ	用　途
<webcom />	テンプレートを入れ子的に参照する．
<msg />	テンプレートに関連付けられたメッセージ，各種属性変数やリクエストパラメータの内容，データベースの SQL クエリーの結果（一つのフィールド）を出力する．
<webcomdb />	データベースのクエリオブジェクトを生成し，クエリー結果のテーブル出力およびレコード出力をテンプレートに関連付けて行う．
<for>～</for>	タグの要素の繰り返し出力処理を行う．
<if>～</if>	条件に一致する時のみに処理展開する．
<set />	変数への値を代入する．
<calc />	簡単な四則演算．
<calendar />	月単位のカレンダー表示．
<timeform />	時間指定フォームの表示．
<dateform />	日付指定フォームの表示．
<selectform />	単純な表現で select タグへと展開する．
<filelist />	フォルダ内のファイルリストの表示をする．
<htmlfilter>	空要素であれば指定されたテンプレートの内部文字列を非空要素であれば内部要素をパラメータやタグ等を着色して見易く出力する．
<javafilter/>	id で示されたテンプレートを Java のソースリストとして，パラメータ，コメント，予約語等を着色して見やすく出力する．

（5）単語の辞書参照

辞書からの単語の参照を”[[単語文字列]”によって行えるようにした．テンプレートおよびメッセージタグで参照される文字列の内部に記述することによって，展開することができる．

2.5 ADK と MVC プログラミングモデル

（1）ビューの分離

MVC(Model-View-Controller)プログラミングモデルは，SmallTalk の GUI 設計に用いられた概念で，最近では J2EE 等の Web アプリケーション開発にその概念を応用させて広く適用されている．実際の処理を行うモデル，表示や出力を行うビュー，入力を受け取ってビューとモデルの遷移を制御するコントローラの 3 要素の組み合わせでシステムを実装するものである。Web アプリケーション開発では，実際の処理の記述を行う開発者と表示デザインを行うデザイナとの分業化を行う意味においてもその方式が利用されていることが多いが，特にビューに関して着目してみると，JSP ファイル上でデザイナがコントローラから引き渡されたモデルオブジェクトを用いてその結果を出力させるビューを作成しなければならず，結局開発者との協力依存関係が強く問題となっている[3][4]．一方，ADK では入れ子可能なテンプレートを積極的に用いることによって粒度の高いビューを作ることができるため, より作業を分担しやすくなることが考えられる．

また, Web ページをデザインし, 公開をする際には, 実際のテキスト文章の内容が重要である場合が多く, HTML を知らないユーザでも, 見栄え等に関する表示デザインを別としてメッセージの更新や書き換えができることが望ましい．また，多言語対応にする場合，表示デザインはそのままで，その文章内容を切り替えるだけの要求も多いものと考えられる．そこで文章内容を msg タグで参照し，言語切り替えを可能にした．

さらに，文章内容に着目すると単語自体にハイパーリンクを付与したり，その意味を説明するツールチップを表示させたりするケースがあるが，それを毎回 HTML で記述することは面倒である．そこで意味 id に関連付けて単語登録できる辞書機能を実装した．意味に基づくことにより，言語の切り替え，そして，同じ意味でも表記方法が利用者のコンテキストによって切り替えられるように適用語を作ることによってその再利用性の向上を目指した．図 3 にこれらの実現するための概念図を示す．

データベースからテンプレートの参照は，id によって指定がされるが，このとき，その時の端末情報，表示スタイル情報によってそれに該当する HTML テンプレートを選択するようにしている．そして，得られた HTML の内容の解析を行い，必要に応じてその時

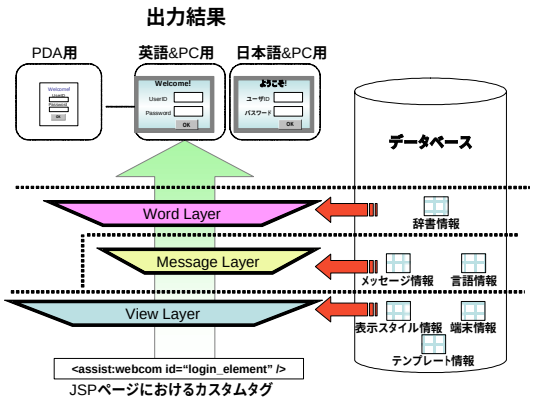

図 3 ADK の柔軟なビューの生成

の言語情報によってメッセージや単語の展開出力を行う．このようにすることによって再利用性が高く柔軟に動的なページを生成することができる．

（2）データベースの処理とフォームの記述

Web ページからのデータベースレコードの追加，更新/削除，表示の処理は，通常ビューとしてのフォームをそれぞれの処理用記述にしなければならないばかりか，データベースへの追加登録のためには，単純かつ多量なデータベースへの追加処理記述が必要であり，もし再利用性が高い記述を行おうとするとさらにモデルおよびコントローラの処理記述が必要となる．

個人的な経験から言うのであれば，JSP のページ上でこの処理記述を行う場合，データベースの一つのテーブルのレコードに関する一連の操作を行うだけでも相当な労力が必要であり，手動での記述はあまりにも面倒であった．

そこでこれらの処理記述の効率化を図るため次の 2 つの機能を実装した．

・MVC のスケルトンコードの自動生成

データベースとテーブルを指定することにより，それに関連したモデルクラス，ビューテンプレート，JSP コントローラのスケルトンコードを自動生成することができる．また，データベースレコードの追加，更新処理は，ビュー上で次のように表現することによってフォームの表示とコントローラの指定を行うだけで利用できる．

```
<webcomdb
  type=”recordout”  query=”クエリオブジェクト名”
  template_id=”ビューテンプレートの id”
  formaction=”コントローラのパス”
  postaction=”処理が成功した場合のパス”
  mode=”view か edit か new” />
```

・データベースレコードのためのフォーム表示用テンプレートの共有化支援機能

データベースレコードの追加，更新，表示を同じテンプレートで利用できるように ANTL のタグ<webcomdb />のパラメータ mode によって指定できるようにした．特に，レコードの結果を表示させるだけの場合，リストボックスによる選択を表示はせずに結果のみを表示することがで

きる．また．フィールドに登録されている値が内部処理のために数値になっている場合があるが，テンプレート内でデータベースフィールドの結果表示をする<msg/>タグのlabels，values というパラメータ用いて，対応する文字列に変換できるようにした．

2.6 コンテンツ管理ツール

Web ページ上に直ちに編集できる機能に加えて，コンテンツ（HTML テンプレート，ファイル，単語参照用辞書，メッセージ，画像）を管理するためのツールを Web アプリケーションとして提供している．このツールにより，検索機能，コピー機能や，各コンテンツの詳細なアクセス権設定，初期パラメータ，分類項目の指定等を行うことができる．

2.7 動的ページの問題とメモリキャッシュ

動的ページを出力する場合，静的ページと比較してサーバに対する負荷が圧倒的に高くなる傾向にある．ADK の場合，テンプレートのメタ情報およびメッセージ等データベースサーバにアクセスして取得しており，テンプレートの実体であるテキスト文字列は，ファイルに保存してあり，処理を行う際には，テンプレートメタ情報のアクセス→テンプレートの字句解析→タグ解析→タグ展開を行っており，さらに入れ子レベルが深くなればなるほどこの処理時間は長くなる傾向にある．そこで，テンプレートのメタ情報，メッセージ情報，テンプレートの実体ファイルとその処理結果をメモリキャッシュの活用によって高速化を行っている．これにより現存するページの中で最大数百倍の高速化が実現されている．

3. 評価と考察

ADK による開発の再利用性および開発容易性を行うため，これまで構築してきた Web アプリケーションのテンプレート数とその参照回数の計測および，アンケートを行った．また具体例として，コンピュータリテラシー科目における履修希望支援機能を時間計測しながら開発した．以下に，これらに関する評価と考察について述べる．

3.1 テンプレートの再利用数

これまで開発してきた ASSIST パッケージの内部のテンプレートの再利用性について解析した結果を表 3 に，上位参照数のテンプレートの種類を表 4 に示す．なお非テンプレート参照数に関しては，テンプレート参照 id のところが<webcom id=”$変数名$”/>のように動的な変数展開で利用されている場合があり，JSP から直接参照されるテンプレート代表的なものは解析したが，それ以外のものはカウントしていない．したがって，少なくとも示された数値以上の参照テンプレート数が存在するはずである．

テンプレートの再利用性の解析結果より，次のように考察することができる．

- 警告メッセージや基本レイアウトおよびその内部に含まれる固定的に参照されるテンプレート数が圧倒的に多い．
- 全体の 16.8%以上ものテンプレートが 2 回以上の参照すなわち再利用されており，設計する HTML のほぼすべてがテンプレートとして登録されていることを考慮すると，それなりの再利用性が高いことが言える．

表 3 テンプレートの再利用性と参照数

JSP ファイル数	243
登録されたテンプレート総数	511
テンプレート非参照数	347 以下
テンプレート総参照数	1660
参照数が 1 以上のテンプレートの平均参照数	10.18
参照数が 1 のテンプレート数	78(15.3%)
参照数が 2～5 のテンプレート数	46(9.0%)
参照数が 6～10 のテンプレート数	18(3.5%)
参照数が 11～20 のテンプレート数	7(1.4%)
参照数が 21 以上のテンプレート数	15(2.9%)

表 4 参照数上位のテンプレートとその種類

テンプレート名	参照数	種類・説明
alert	142	警告メッセージ．自動生成される JSP コントローラ上で用いられている．
assist_dialect_vl	112	単語参照用の適用語の一覧情報の取得．基本レイアウトや辞書登録ページ等に含まれる．
assist_dialect_form	104	単語参照用の適用語の選択フォーム．基本レイアウトや辞書登録ページ等に含まれる．
current_members, locale_switch, assist_logo_small, title2,windowname, logininfo_mini	87	基本レイアウトに含まれるユーザ情報の概要説明，ロゴ，タイトル等．
assist_main_menu	84	ASSIST 管理ツールのメインメニュー．
layout	84	基本レイアウト．
dateform, timeform	42	日時情報の選択フォーム
assist_tab	35	タブインタフェース用．
navibar	30	データベースアクセス用のナビゲーションバー．

3.2 アンケート結果による評価

ADK の利用に関して，創価大学工学部情報システム学科に属しており，多少かそれ以上の Web アプリケーション開発経験がある学生（大学院生を含む）8 名に，その利用法に関するチュートリアルを行い，再利用性と開発容易性に関してのアンケートを行った．その結果の 1 部を表 5 に示す．再利用性，開発容易性に対して比較的高い結果が得られた．また，コメントとして，テンプレートの id が増大するにつれ，誤参照や競合が起きるので名前空間を用いる等の改善を期待する内容が多かった．

表 5　ADK の利用に関するアンケートの結果

質 問 項 目	5段階評価（1~5）	
	平均	標準偏差
既存のテンプレート技術と比較して，入れ子可能なテンプレートベースの HTML デザインによって，テンプレート(Webcom)の再利用性が向上されると思いますか？	4.4	0.5
保守モードの時に，入れ子化されたテンプレートの枠とメニューで示されますが，このレイアウトの位置関係，配置関係は分かり易いですか？	3.8	0.4
JSP によるビューの設計と比較して入れ子可能テンプレートの設計手法によるビューのソースリスト（テンプレートの内容等）は見易く，その内容を理解し易いですか？	4.3	0.7
テンプレートとメッセージの追加はタグで id を指定して存在しない id であれば自動的に新しいテンプレートやメッセージを追加するという仕組みをとっていますが，先にテンプレートの登録作業を明示した上で後にそれを参照するという既存のテンプレート技術/機能と比較して，その設計手法をどのように感じましたか？	3.8	0.7

3.3 開発事例：コンピュータリテラシー科目の履修希望支援サイト

コンピュータリテラシー科目の履修にはコンピュータの台数の関係で人数制限があるが，これまで担当教員間での履修希望人数の状況把握が問題となっていた．この問題を解決するために，担当者が初回授業終了後に Web ページ上に履修希望人数を入力していき，学生と他の教員との履修希望情報を提示するサイトを ADK を用いて開発した．教員が情報登録するためにユーザ認証も行っている．

要求仕様の整理から実際の完成までにかかった時間は 5.5 時間であった．手動で 1 から記述した JSP ファイル数は 5 で総行数は 101，テンプレート数は 10 で総行数は 268 行．自動生成した JSP ファイル数は 1 で行数は 189 行，自動生成されたモデルクラスは 1 つで行数は 782 行，コピーしたテンプレートは 1 つで行数は 134 であった．またテンプレートの参照数は，利用した 13 のテンプレート数に対して，合計 37 回，1 つあたり平均 2.8 回の利用が行われた．

手動で記述した JSP ファイルおよびテンプレートの総行数は，503 行であるのに対して，自動生成した JSP ファイル，テンプレート，Java クラスの総行数は 1094 となり，2/3 が自動生成されたことになる．

これらのことから ADK による開発は，再利用性が高く効率的な開発が行えたと言える．

4. 関連技術・研究

Web アプリケーション開発環境はかなり多く存在しており個別の機能に関しては類似の製品が多く存在する．Web ブラウザのみでの編集環境を提供するものとしては Wiki[5]や Zope[6]等が挙げられ，HTML テンプレート技術や HTML コンポーネント技術としては，Velocity, WebMacro, FreeMarker, JavaServer Faces 等が挙げられる[4]．しかしながら，ADK のように入れ子可能なテンプレートや JSP ファイルを Web ブラウザのみで編集登録でき，メッセージと単語を分離して設計でき，データベースアクセスのフォームの再利用性が考慮され，MVC 用スケルトンコードの自動生成を行っている統合的開発環境は皆無である．

また，本来の ADK の開発目標は，同期・非同期統合型グループウェアの構築であり，現在，Web ベースインスタントメッセージング機能等，Web ブラウザを利用してもリアルタイム系の処理ができるような機能が実装されつつあり，他の開発環境と比較した場合に大きな特異性を持っている．

5. まとめと今後の課題

同期･非同期統合型 Web グループウェアを柔軟に構築するための開発環境 ADK の中で，主に非同期型の構築方法について論じ，その評価を行った．この ADK の特徴は，入れ子的な HTML テンプレートの積極的な利用，Web ブラウザでの編集環境，動的なページ生成を行うための豊富なカスタムタグ，データベースアクセスのための MVC のスケルトンコードの自動生成機能を提供するものであり．これを活用することにより，再利用性が高く開発が容易な Web アプリケーションの構築が可能となることが分かった．

今後は，同期的機能との統合を考慮しながら，テンプレートに名前空間やオブジェクト指向的な機能を追加する等により，更に再利用性，開発容易性の高い同期･非同期統合型 Web グループウェアの開発環境の構築を目指す予定である．また，その発展により同期・非同期統合型 Web グループウェア開発フレームワークへと進化することが期待される．

参考文献

[1]M. Tanaka, M. Fusuki, Y. Nishibori, Y. Teshigawara: Framework and System Flow of ASSIST, in Proc. of ICOIN-13, Vol.2, 6C-4, January 1999.

[2]M. Tanaka, Y. Teshigawara: Design of a Development Environment for Web-based Asynchronous and Synchronous Integrated Groupware Systems, in Proc. of MNSA2003, pp.582-587., May 2003.

[3]Gregory Murray（石原 直樹訳）: 最新！ スーパーガイド「正しい」Web システムの設計法を学ぶ，WEB+DB PRESS Vol.8，技術評論社，2002.5

[4]岸田 ゆき枝：これでカンペキ MVC，WEB+DB PRESS Vol.8，技術評論社，2002.5

[5] Bo Leuf and Ward Cunningham: *The Wiki Way*, Addison-Wesley, 2001.

[6] Zope (Z object publishing environment) community: http://www.zope.org/