

## カード型 Web アプリケーション/ Web サービス開発環境

嶋 幸太郎 四野見 秀明  
日本 IBM

### 概要

近年、プログラミングスキルの高くないエンドユーザにも Web アプリケーションを容易に開発・カスタマイズ可能にする手法が望まれている。本稿では、ウェブブラウザ上で動作する、カード型インターフェースを用いた新しい Web アプリケーション開発環境を説明する。Web アプリケーションは、「カード」と呼ばれる入力フォームに必要な値を入力し、それらをスプレッドシートと同様な形で参照して組み合わせることによって開発されるため、Java, JavaScript, PHP 等のコードを記述する必要がない。さらに、同開発環境において、Web サービスも容易に活用および公開できることを示す。

## Card-based Web Application/Web Service Development Environment

Kotaro Shima Hideaki Shinomi  
IBM Japan, Ltd.

### Abstract

Recently, a method that enables end-users with less programming skills to easily develop/customize Web applications is desired. In this paper, we describe a new card-based Web application development environment that runs through Web browsers. In this environment, Web applications are developed by inputting values to a form called “card” and combining such cards by referencing fields of other cards in similar manner as spreadsheets. As a result, end users do not need to write programming codes such as Java, JavaScript, and PHP. Additionally, we show that Web services can be easily utilized or published in such environment.

### 1. はじめに

近年のウェブの発達で、従来デスクトップのアプリケーションで行っていた作業をウェブブラウザで実現させる Web アプリケーションが注目を集めている。その理由の一つにインストールが不要なことが挙げられ、ユーザーは Web アプリケーションの URL にアクセスすれば、インターネット接続を介してどこか

らでもアプリケーションを使用可能である。また、アップグレードやパッチの適用が不要な点も利点の一つと言える。管理者がサーバー上のアプリケーションを更新すれば、ユーザーは全員最新バージョンを使用することができ、IT 部門の管理者が一台一台アプリケーションを更新する手間とコストが省ける。

そして、このような Web アプリケーション

では特に迅速性が重要となるが、アプリケーションを実際に使用するエンドユーザの多種多様な要求を満たすために新たなアプリケーションを開発あるいは既存のアプリケーションに変更を加えるのに、いちいち IT 部門の人に依頼していたのでは時間・コストの面で問題が生じる。そこで、エンドユーザ自身がアプリを簡単に開発・修正するエンドユーザ・プログラミングの重要性が高まってきている。実際に、2005 年にはエンドユーザ・プログラマーの数は 5500 万人にも上るとの試算もある [1]。しかしながら、エンドユーザの多くはプログラミングスキルが高くないため、Java, JavaScript, PHP 等のプログラミングスキルを要求しないでアプリケーション開発を可能にする方法が望まれている。

そこで、我々は新しいカード型インターフェースを用いた新しい Web アプリケーション開発環境を開発した。本システムにおいて、アプリケーションの開発はウェブブラウザ上で行うが、アプリケーションの修正は即座に反映され、デブロイ等の面倒な作業も不要である。本システムの大きな特徴としては、アプリケーションを構成する全ての要素は「カード」と呼ばれるウィンドウに似たインターフェースで表現されていることが挙げられる。アプリケーションは、各カードに必要な値を入力し、スプレッドシートにおけるセルの参照と似た形で他カードの値を参照することでカードを組み合わせる形で開発される。したがって、ユーザーは Java, JavaScript, PHP 等のコードを記述する必要が無い。そして、本システムのカード型インターフェースを用いて Web サービスも容易に活用・公開できることを示す。

## 2. カード型 Web アプリケーション開発環境

### 2.1 関連研究

まず、関連するエンドユーザ向けのアプリケーション開発環境について述べる。初期のエンドユーザによる開発環境としては、Apple 社の HyperCard [2]が挙げられる。そして、Web アプリケーションのエンドユーザ向けの開発環境としては、WebFormulate [3], FAR [4], BioPro [5], Click [6] などがある。WebFormulate [3]は、ウェブベースのアプリケーション開発環境であるが、ネットワークを通して連続的にデータを取得して表示させることが可能な点に特徴がある。FAR [4]は、スプレッドシート、ルールベース・プログラミングのアイデアに基づく開発環境であり、ドラッグ&ドロップによりウェブページのデザインも可能である。BioPro [5]では、ユーザーが頭に描いたイメージを元にアプリケーションを開発するイメージ指向プログラミングを実現している。Click [6] は、WYSIWIG モードでウェブページのデザインができ、またプログラミング用コードを書かずにデータベースの接続、アクションの定義、ページ遷移なども実装できる。これらに対して本システムは、ウェブページの表示、データベースへの接続、電子メールの送信、コントロールフロー、さらには次章で述べる Web サービスの活用・公開まで、アプリケーションを構成する全ての要素が「カード」と呼ばれるインターフェースで表現されているという点に特徴がある。この特徴のため、編集画面では開発されたアプリケーションで使用されているコンポーネントを一括して見ることができ、また、互いに関係のあるカードの間が矢印で結ばれるので、カードが互いにどう関連付いているのかを把握することができる。また、ユ

ユーザーが新たにツールを使用する際にはツールの使用法の習得が問題となるが、このカードという単一のインターフェースを採用することにより、本システムの使用法の習得も容易であると期待される。

## 2.2 システム構成

本システムの構成を図1に示す。本システムは、IBM WebSphere Application Server もしくは Apache Tomcat にインストール可能で、アプリケーション開発者はウェブブラウザで開発環境の URL にアクセスする。開発環境のユーザーID とパスワードを入力してログインし、開発を開始する。本システムにおけるアプリケーションに相当する単位を「シート」と呼ぶが、シートを構成するコンポーネントを「カード」と呼ぶ。そして、アプリケーションの開発は、ブラウザ上で JSP、および JavaScript で記述されたユーザーインターフェース上でカードを編集して組み合わせることで開発される。カードについては次節で詳しく述べる。ブラウザ上での操作は Java アプレットを通してソケット通信によりアプリケーション・サーバーに送信され、各カードで入力・編集された値は card.xml という XML ファイルに保存される。そして、開発環境からログオフし再度ログインし直した際には、この card.xml が読み込まれ、編集を再開することが可能となっている。

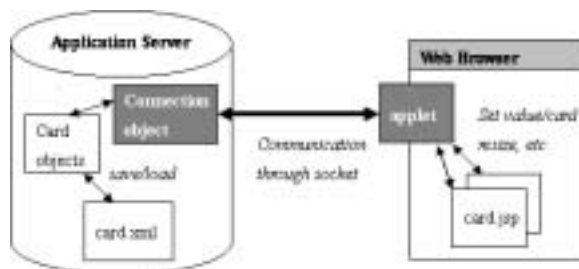


図 1: システム構成

## 2.3 カード

次に、カードの構造について説明する。カードは入力あるいは出力フィールドを複数持つことができ、フィールドは値を保持することができる。そして、実行可能なカードは実行ボタンを押すことで実行され、実行結果が出力フィールドに表示される。図2に、システムが提供しているカードの一つである DateTime カードの実行後の図を示す。このカードは、入力フィールドは無く出力フィールドのみであるが、実行ボタンが押された時刻がフィールドに表示されている。

カードはウィンドウのようなインターフェースを持ち、ドラッグ&ドロップで移動、リサイズができ、アイコン化が可能なので、開発者はアプリケーションが理解しやすいようにカードを配置することができる。

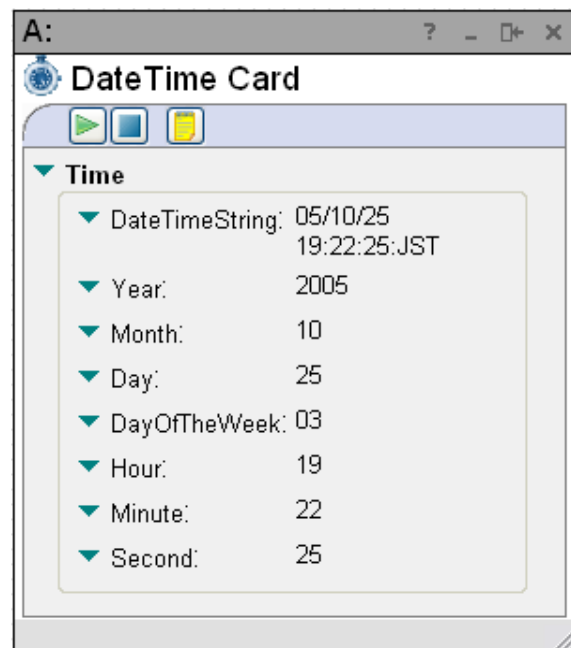


図 2: DateTime カード

そして、他カードのフィールドの値を入力として与えてカードを実行させたい場合には、カードに付与されたシート中の固有の ID とフィールド名を用い、[=カード ID.フィールド

名] という式を入力することにより、スプレッドシートのセルの参照と同様な形式で他フィールドの値を参照することができる。図3に、フィールドに値を格納するカードである Variable カード中の variable1 というフィールドで、図2で使用した DateTime カードの Year フィールドを参照させた場合の例を示す。参照式を入力すると即座に、参照先のフィールドの値が variable1 フィールドにセットされ、また参照先の値が更新されると同時にこの variable1 フィールドの値も更新される。

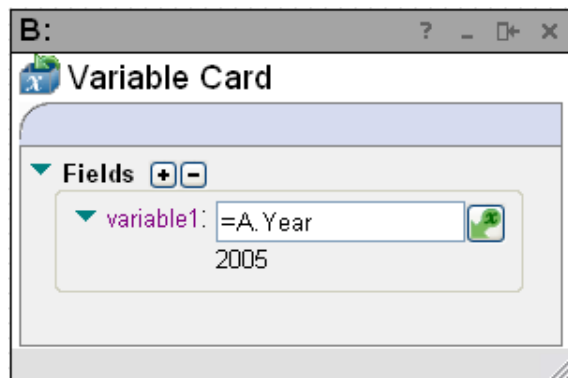


図 3: 他カードのフィールドの参照

システムが提供している主なカードの種類としては、上記の DateTime カード、Variable カードの他に、ウェブページを表示する Web Page カード、ページ遷移を記述する Page Forward カード、他フィールドに値を代入する Assignment カード、リソースファイルをサーバーにアップロードする Resource カード、条件分岐を記述する If カード、繰り返し処理を行う Repeat カード、データベースの SQL 文を実行する SQL カード、JavaScript を実行する JavaScript カードなどがある。システムが提供している種類以外のカードを使用したい場合には、自ら Java のクラスを記述してユーザー独自のカードを新たに追加することもできる。

## 2.4 Web アプリケーションの開発

前節のようにカードを組み合わせる形でアプリケーションは開発されるが、本節ではエンドユーザが実際に使用する Web アプリケーションとして公開する方法を説明する。簡単な例として、ウェブページのフォームに入力された値が正か負かを判別するアプリケーションで説明する。図4にこのアプリケーションの開発画面の図を示す。この Web アプリケーションでは、入力と出力用のウェブページに相当する Web Page カードが1枚ずつ、そして正負を判別する If カード1枚の、合計3枚のカードで構成されている。入力用の Web Page カードには入力フォームの HTML 文を入力してあり、出力用の Web Page カードには If カードの実行結果を表示させるので、2.3 節で説明したフィールド参照式を用いて If カードの実行結果を参照する式が入力してある。また、出力用ページがアクセスされた際に If カードが実行されるよう、cardsToRun フィールドには If カードの ID を指定してある。ここで使用されている Web Page カード間にはフォームの送信という関係、出力用 Web Page カードから If カードの実行の呼び出しという関係があるので、それらのカード間は矢印で結ばれている。出力用 Web Page カードには入力フォームから送信された値を格納するフィールドを追加しておき、If カードにはその値を参照して条件を評価する式、そしてその結果が真の場合、負の場合それぞれに対して出力で表示させる HTML が入力してある。ここで HTML のソースが入力してあるが、HTML の知識が無いユーザーには WYSIWIG 型の HTML エディタである ホームページ・ビルダーとの連携機能が提供されているので、それを用いることで直感的にページデザインを行うことも可能である。

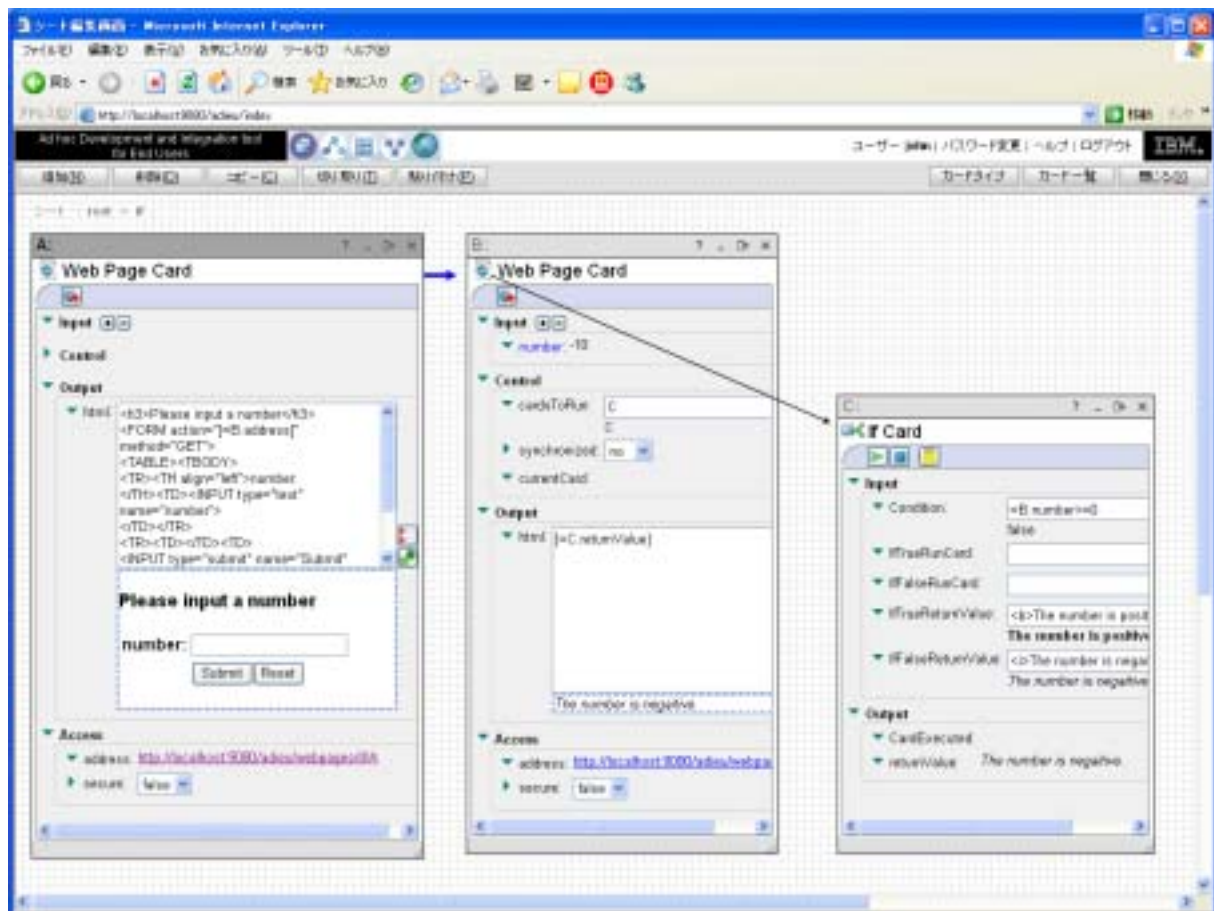


図 4: 正負判別アプリケーション

この例では If 文に相当する If カードを使用した例を紹介したが、他にも繰り返し処理を行う Repeat カード、他フィールドへ値を代入する Assignment カードなどが提供されており、これらのカードを利用することで従来のプログラミングのようにロジックをコードで記述するのではなく、フィールドに値もしくは参照式を入力してカードを組み合わせる形でアプリケーションが開発される。開発されたアプリケーションは使用するには、Web Page カードの address フィールドに表示されている URL へのリンクをクリックすることで実行することができる。

### 3. Web サービスの活用/公開

Web 上のサービスを連携する技術として

Web サービスが注目されてきている。しかし、Web サービスの標準的なプロトコルである SOAP (Simple Object Access Protocol) が複雑過ぎて、サービスの呼び出しおよび公開が難しいという問題があった。そこで、本システムでは Web サービス記述言語である WSDL (Web Service Description Language) の URL さえ指定すれば容易にカードという形で Web サービスを呼び出すことができ、また、開発したアプリケーションを容易に Web サービスとして公開せることを示す。

#### 3.1 外部 Web サービスの活用

例として、XMethods のサイト[7]で公開されている米ドル建て株価取得 Web サービスをアプリケーションの中で利用する場合を考える。

外部 Web サービスのインポート機能を使用し、WSDL の URL を指定してウィザードを進めると、WSDL ファイルがアプリケーション・サーバー上にインポートされる。そして、インポートされた WSDL ファイルはパースされ、その中で定義された各 operation に相当するカードが生成されるとともに、サービスの入出力に対応するフィールドが各カードに追加される。図 5 に、前述した XMethods のサイト[7]で公開されている StockQuote の Web サービスをインポートして生成されたカードの例を示す。

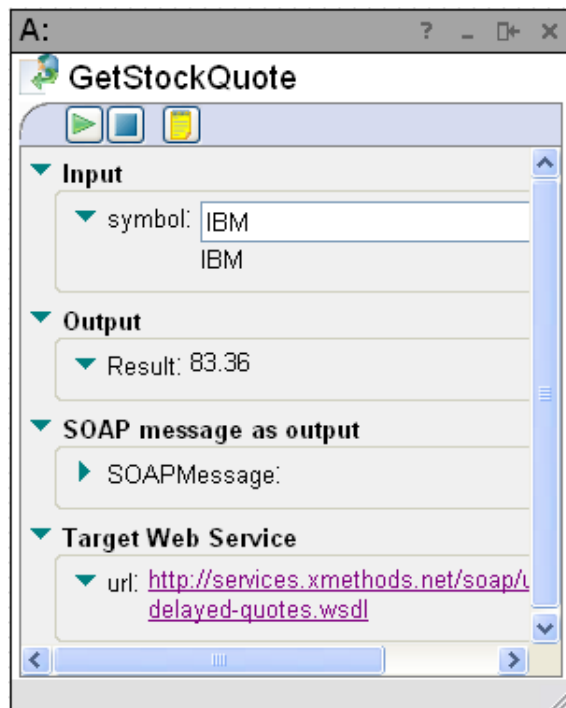


図 5: 外部 Web サービスを呼び出すカード

このカードの symbol フィールドに取得したい株価のシンボルを入力して実行ボタンを押すと、Web サービスからドル建ての株価の値を取

得することができる。そして、2.3 節で述べたフィールド参照式を用いることで、本システムで開発されたアプリケーション中の値を Web サービスの入力として与えることも、Web サービスの取得結果をアプリケーション中で利用することも容易に可能である。

### 3.2 アプリケーションの Web サービスとしての公開

また、本システムで開発されたアプリケーションを Web サービスとして公開するには、次の手順で行う。2.4 節で使用した入力値の正負判別アプリケーションを Web サービスで公開する例で説明する(図 6)。使用するカードは、Web Service カード、および前章の例と同じ If カードである。Web Service カードには、サービスの入出力用のフィールドを追加しておく。そして、Web Service カードの入力に対する参照を含む条件式を If カードに指定し、If カードの出力を Web Service カードの出力フィールドで参照する。そして、サービスがアクセスされた時点で If カードが実行されるように、Web Service カードの cardsToRun フィールドに If カードのカード ID を指定する。後は、カテゴリ名とカードタイプ名の両方を指定すると、このサービスに対応する WSDL ファイルが自動生成され、Web サービスが公開される。この Web サービスの公開はカード 2 枚と参照式、簡単な条件文を記述することのみで済み、複雑なコードを記述する必要はない。



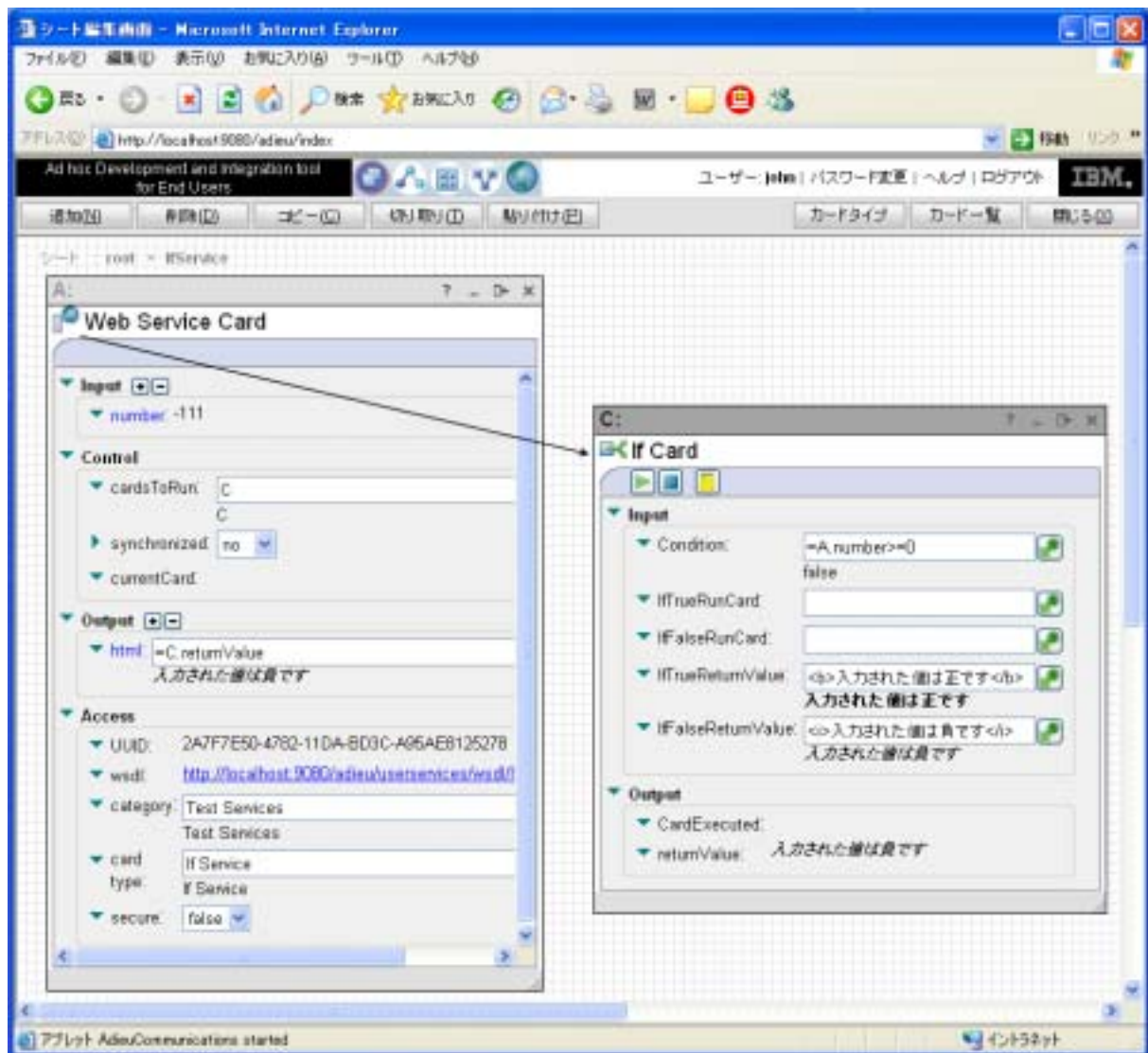


図 6: 正負判別 Web サービスの公開

そして、定義された Web サービスはもちろん外部アプリケーションからも呼び出し可能であるが、本システム中の別アプリケーション（シート）から呼び出すことも可能である。「カードの追加」ダイアログにおいて指定した名前を持つカードタイプが追加されているので、別シートにこのカードを追加すると、図 7 に示されるように Input には number という入力フィールド、出力に条件判別の結果の html を出力されるフィールドをもつ正負判別 Web サービスを呼び出すカードが追加される。適

当な数値を number フィールドに入力して実行ボタンを押すことで本 Web サービスを実行でき、他のカードと同様の方法で使うことができる。



図 7: 正負判別 Web サービスを呼び出すカード

#### 4. まとめ

我々は、プログラミングスキルを必要としないエンドユーザ向けの Web アプリケーション/Web サービス開発環境を開発した。本システムでは Web アプリケーションは、カードと呼ばれるコンポーネントに値を入力し、スプレッドシートのセルの参照に似た形でカードを組み合わせる形で開発されるため、Java, JavaScript, PHP 等のコードを書く必要が無い。さらに、カード型インターフェースを採用することにより、SOAP あるいは WSDL の知識が無くとも容易に Web サービスを活用または公開できることを示した。

現状の課題としては、ちょっとした規模のアプリケーションを開発するのにも比較的多くのカードが必要となってしまうので、よりアプリケーション寄りのカードの提供が挙げられる。また、Web アプリケーションではデータベースの活用が重要だが、本システムで提供されているデータベース用のカードは SQL の知識が無いと使うのは難しい。したがって、

SQL の知識が無くても簡単に使えるデータベース用カードの開発も課題である。

本稿で述べたシステム Ad hoc Development and Integration tool for End Users (ADIEU) [8, 9] は IBM alphaWorks のサイト [8]から無償にてダウンロード可能なので、興味のある方は使用していただければ幸いである。

#### 参考文献

- [1] B. Boehm et al., Cost Model for Future Software Life Cycle Process: COCOMO 2.0, Annals of Software Engineering Special Volume on Software Process and Product Management (J. Arthur and S. Henry eds.), J.C. Balzer AZ, Science Publishers, Amsterdam, Netherlands (1995)
- [2] Apple Computers Inc., HyperCard User's Guide (1987)
- [3] A. Ambler and J. LeoPold, Public Programming in a Web World, Visual Languages, Nova Scotia, Canada (1998)
- [4] M. Burnett et al., FAR: An End-User Language to Support Cottage E-Service, In *HCC-2001 IEEE Symposia on Human-Centric Computing Languages and Environment*, Stresa, Italy (2001)
- [5] T. Shimomura, Visual Design and Programming for Web Applications, *Journal of Visual Language and Computing*, Vol.16, pp.213-230 (2004)
- [6] J. Rode et al., As Easy as "Click": End-User Web Engineering, In *Proc. of International Conference on Web Engineering* (2005)
- [7] XMethods, <http://www.xmethods.net/>
- [8] <http://www.alphaworks.ibm.com/tech/adieu>
- [9] H. Shinomi and S. Adams, Say Goodbye to Complexity When Developing Web Services, <http://www-128.ibm.com/developerworks/webservices/library/ws-adieu/>