

IBMが注力するオープンソース・プロジェクト

— EclipseとDojo Toolkit —

昨今のソフトウェア開発における重要な要素であるオープンソース・ソフトウェア。中でも、IBMが注力するオープンソース・プロジェクトとして、EclipseとDojo Toolkitがあります。Eclipseは、Java™ ベースの統合開発環境としての地位を確立し、さらにリッチ・クライアントのプラットフォームとして、また、サーバー・サイドのプラグイン技術として利用範囲を広げています。一方、Dojo Toolkitは、Webブラウザ上で動作するリッチ・インターネット・アプリケーションの開発に欠かせない、高機能 JavaScript™ ライブラリーとして、製品での利用が進んでいる注目株のオープンソース・プロジェクトです。本稿では、この二つのオープンソース・プロジェクトの動向をご紹介します。

① ますま重要になるオープンソース

Linux®をはじめとしたオープンソースの潮流は、ソフトウェア開発の現場で注目され始めた黎明期から、質・種類ともに飛躍的な進歩を遂げています。オープンソースの利用は、システム開発期間の短縮やコミュニティによる品質向上といったコスト削減のメリットに加えて、特定のベンダーの技術に対する依存を避け、将来にわたる柔軟なシステム構築のための重要な選択肢となっています。

本稿では、そのようなオープンソース・プロジェクトの中から、IBMが注力する「Eclipse」と「Dojo Toolkit」の二つのプロジェクトの動向をご紹介します。

② Eclipse

2001年にIBMがEclipse 1.0をオープンソースとして公開してから7年が過ぎ、今やJavaの統合開発環境(IDE)の世界では確固たる地位を築いていることは誰もが認めるところでしょう。今、EclipseはJava IDEとしてだけでなく、さまざまなエリアで活用されています。開発環境としてはJavaアプリケーションだけでなく、Webアプリケーション開発や、モデリング環境、組み込みソフトウ

Open Source Projects that IBM Invests in - Eclipse and Dojo Toolkit -

Open source software is an important factor in software development today. Among many open source projects, IBM is investing in the Eclipse and the Dojo Toolkit. The Eclipse is dominant in the Java IDE area and continues to evolve as well as a rich client platform and a server-side plug-in technology, while the Dojo Toolkit is an emerging open source project that provides JavaScript libraries for developing rich internet applications. This article introduces the latest activities and features on both open source projects.

ア開発へとそのターゲット・エリアを広げています。さらには、開発環境としてだけでなく、例えばリッチ・クライアントの基盤としての活用や、サーバー環境での活用もすでに始まっているのです(図1)。

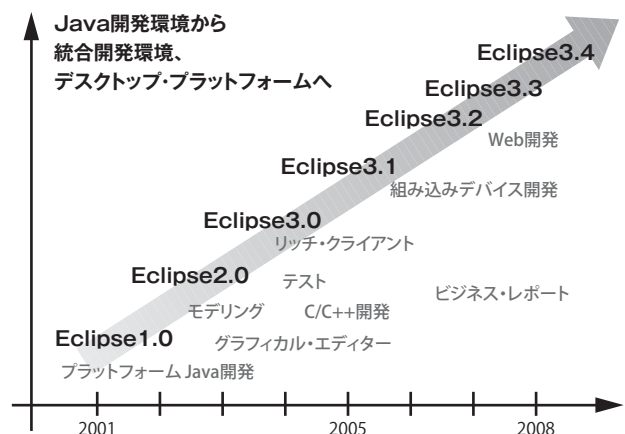


図1. Eclipseの発展

Eclipseの最大の特長はその拡張性と拡張する仕組みであるプラグイン・アーキテクチャー(図2)です。もともIBMが問題としていた、ツール間での互換性やツールを作る上での共通基盤の欠如を解決するために、疎結合で再利用性の高いコンポーネントを作成するための

プラグイン・アーキテクチャーが Eclipse をここまで広めた原動力だと考えられます。そのアーキテクチャーによって、IBM 社内のツールだけでなく、さまざまなツールが Eclipse の上に構築されるようになり、オープンソースの潮流と結びついて、現在最も成功しているオープンソース・プロジェクトの一つに成長しています。

このプラグイン・アーキテクチャーですが、バージョン 3.0 からその実装は OSGi [1] という標準に基づいています。従って、OSGi で定義された機能を活用することが可能です。

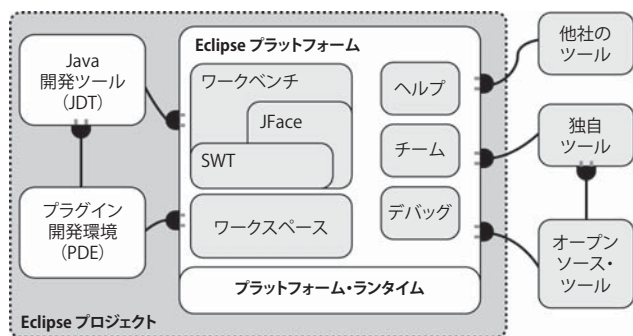


図 2. IBM のプラグイン・アーキテクチャー (Eclipse Help より引用)

IBM は Eclipse をオープンソースにし、その管理を完全に第三者である Eclipse Foundation に渡して以降も Eclipse に対して積極的にプロジェクトを提供し続けています。Eclipse Modeling Framework (EMF) や Graphical Editing Framework (GEF) などのフレームワーク、Rich Client Platform (RCP) や Web Tools Platform (WTP) などのプラットフォーム・レベルのものまで多岐にわたっています。ベース部分を公開し、頑強なプラットフォームを構築した上で、付加価値部分を製品として提供していくという方針の元で、単に提供するだけでなく、自社の製品開発に生かすというよいサイクルを生み出しています。実際に、現在多くの IBM の製品が Eclipse を基盤として活用しています。開発環境としての IBM Rational® 開発ツール製品群、リッチ・クライアントとして IBM Lotus Notes® や Sametime® などのコラボレーション製品群、さらには、Lotus® Mashups [2] や Jazz™ (Rational Team Concert) [3] などのサーバー製品群の基盤として活用されています。

ここでは、今 Eclipse で注目されるいくつかのプロジェクトを紹介します。

2.1 Rich Client Platform (RCP)

SWT (Standard Widget Toolkit) の利点である高

速性や使いやすさと、プラグインによる拡張のしやすさという利点を持ったリッチ・クライアント基盤として RCP は 2004 年に Eclipse 3.0 のハイライトの一つとして登場しました。SWT は Eclipse のために開発された Java のためのユーザー・インターフェース (以下、UI) ツールキットですが、その長は OS 標準の機能を最大限活用している点です。そのため、ネイティブな UI と比較しても遜色のない使用感とパフォーマンスを提供します。SWT が登場したばかりのころは、なぜ Java に標準で付属する Swing があるのに、新しい UI ツールキットが必要なのかという議論もありましたが、今ではそのような話はすっかり聞かなくなりました。それだけ、SWT というテクノロジーが認められたということだと考えられます。

RCP のもう一つの利点はプラグイン・アーキテクチャーにのっっているということでしょう。これは開発者にとっては Eclipse と同じプログラミング・モデルでリッチ・クライアントを開発できることや、プラグイン化することによってコードの再利用性が高まるなどのメリットがありますし、利用者にとっても、後で動的に機能を追加することができることなどの利点があります。もちろん、オープンなテクノロジーですから、第三者が提供するプラグインを追加できる可能性もあります。

IBM は RCP を基盤として Lotus Notes の拡張を行いました。Lotus Notes はバージョン 7 から RCP を基盤として活用しています。Notes® を幾つかのプラグインの集合体として提供し、ほかのプラグインと協調動作することができるようになっていきます。

Sametime もバージョン 7 から RCP ベースになっています。プラグインによる機能拡張が可能になっています。同

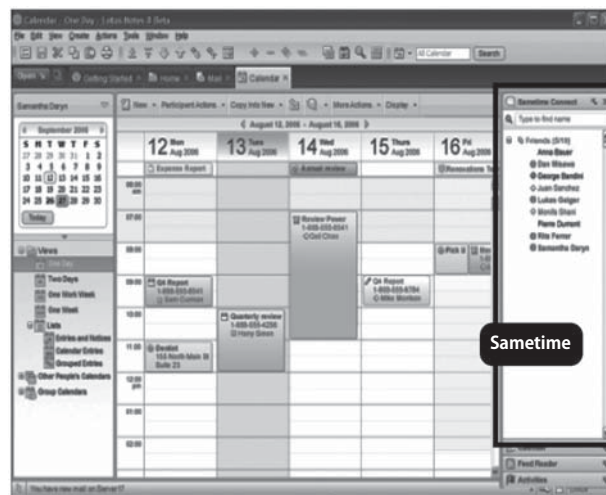


図 3. Notes に Sametime を組み込む

じテクノロジー上に実装された Lotus Notes 上に組み込むことも可能です (図 3)。例えば、Notes 上で表示されたユーザーとすぐにチャットを開始することができるなどの機能を提供しています。これも、プラグイン・アーキテクチャーを活用している利点です。

なお、対象を携帯端末にまで広げた eRCP (Embedded RCP) [4] も提案され、今後さらに広がることが期待されます。

2.2 Server-Side Equinox

Eclipse のプラグイン・アーキテクチャーの実装は OSGi 標準に基づき、Equinox という名前のプロジェクトで開発が進んでいますが、考えてみればプラグイン・アーキテクチャーはクライアント・サイドにのみ利点があるわけではありません。コンポーネント化、動的な配置、サービス間の連携など、どこでも役に立つものです。IDE を越え、PC から飛び出し、携帯端末にまで広めたこの実装はサーバー・サイドでも活用され始めています。Eclipse 内では Server-Side Equinox [5] というプロジェクトが立ち上がっています。

Server-Side Equinox は簡単にいえば、Java EE Web アプリケーション・サーバー内でプラグインを動かす仕組みです。クライアント・サイドのプログラミング・モデルと同じ方法でサーバー・サイドのコンポーネントを開発することができます。

IBM はこのテクノロジーを活用した製品をすでに出荷しています。Lotus Mashups と Rational Team Concert (RTC) です。RTC のサーバーである Jazz チーム・サーバーは Equinox を基盤として利用しています (図 4)。例えば、EMF をベースにしたモデル・ベースのコンポーネントや、シンプルな Web API である REST (Representational State Transfer) サービスを提供するサービス、Web 画面のページなどが拡張ポイントとして定義され、新しいコンポーネント、サービス、ページを追加するときにはそれらの拡張ポイントを拡張していくことになります。これは Eclipse のプラグイン開発と同じです。従って、Jazz のコンポーネントの開発環境は Eclipse のプラグイン開発環境が活用できますし、Eclipse のプラグインを開発するのと同じプログラミング・モデルで開発することができるわけです。

2.3 Eclipse 4.0 (e4)

次世代の Eclipse を考えるために、e4 というプロジェクト [6] が立ち上がっています。まだ始まったばかりですが、内容的には「よりよいデスクトップ環境の構築」と「Eclipse

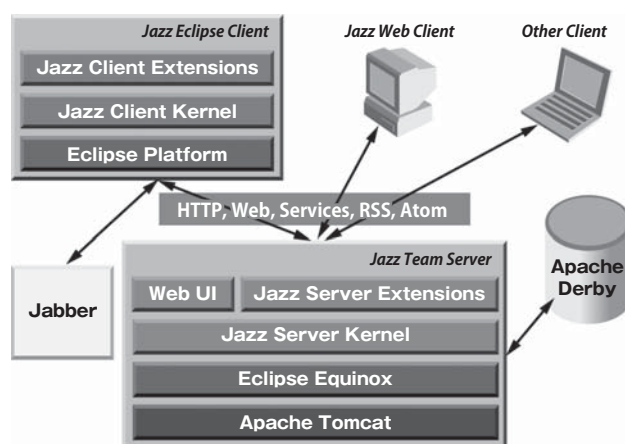


図 4. Jazz のアーキテクチャー

を Web へ」という二つのテーマを中心にコミュニティの中で話が進められています。今後 Web の世界との融合がますます進むと考えられます。例えば、Eclipse の UI 環境を Web からアクセスするなども考えられています。

e4 プロジェクトを見ていると、今後もしばらくは Eclipse が IDE やクライアント環境、サーバー環境などにおいて、発展を続けていくのは間違いのないでしょう。IBM も e4 に対して積極的な貢献を行っていますし、Eclipse に対する貢献を続けていき、自社製品での活用を進めていくことになりそうです。

3 Dojo Toolkit

Ajax やダイナミック HTML などの Web 2.0 の要素技術を利用した、リッチ・インターネット・アプリケーション (RIA) が企業システムのフロント・エンドとしても注目されていますが、Web ブラウザー上で稼働する RIA 開発に欠かせないのが、JavaScript™ ライブラリーです。JavaScript ライブラリーは一般的に、ブラウザーの仕様や動作の差異を吸収し、ダイナミック HTML の実装を簡略化する機能を提供して、RIA 開発の生産性を向上します。ここで取り上げる Dojo Toolkit [7] は、そのような数ある JavaScript ライブラリーの中のひとつで、IBM をはじめとした、Sun™、BEA システムズといった Java EE ベンダーもスポンサー、パートナーとして参画しているオープンソース・プロジェクトにより開発されています (図 5)。

Dojo Toolkit は、UI 部品であるウィジェットやライブラリーとしての機能の豊富さもさることながら、企業の大規模なアプリケーション開発を想定して設計されている点が大きな長点であるといえます。一貫性のあるモジュール化と拡張性が考慮された設計により、保守性やスケラビリティ

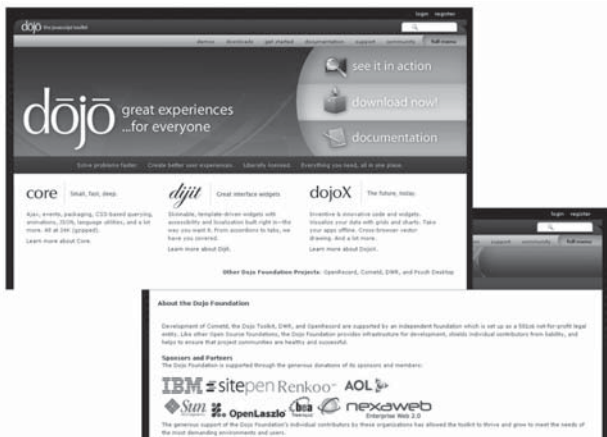


図5. Dojo Toolkit の Web サイト

ティーに優れています。また、国際化やアクセシビリティのサポートは、企業の製品やアプリケーションの重要な要件を満たします。

3.1 モジュール・フレームワーク、言語サポート、ユーティリティ

Dojo Toolkit では、JavaScript のコードはモジュールごとに定義され、モジュール間の依存をコード内に記述することで、必要なモジュールを実行時に動的に読み込みます。これにより、必要になるまでコードをダウンロードしない遅延ローディングが可能になり、大規模アプリケーションの起動時間を短縮できます。逆にアプリケーション起動時に必要なモジュールは一つのファイルに圧縮して、複数ファイルのダウンロードに伴うパフォーマンスの問題を回避するビルド機能もサポートされています。

また、クラスの多重継承や継承メソッドの呼び出しなどの言語機能も充実しており、Dojo Toolkit に含まれるクラスの拡張や大規模アプリケーションでのコードの再利用性を高めることができます。

さらに、配列処理やイベント・ハンドラーなどの基本的なものから、ドラッグ・アンド・ドロップやアニメーションなどの高度なダイナミック HTML のサポートまで、幅広いユーティリティが用意されており、これまでの JavaScript、HTML、CSS による RIA 開発の生産性を飛躍的に高めます。

3.2 ウィジェット

拡張可能なウィジェット・フレームワークと多彩なウィジェットにより、リッチな UI をより簡単に構築することができます(図6)。ウィジェットは JavaScript から動的に作成するほかに、あらかじめ HTML 内にマイクロ・フォーマットを使って記述

することができるため、画面を柔軟かつ効率よく実装できます。コンボ・ボックスやカレンダーなどの入力部品、タブやアコーディオンなどのレイアウト部品、さらにはダイアログ、メニュー、ツールバーなどの基本ウィジェットから、ツリーやグリッドなどの高度なウィジェットまでがあらかじめ提供されていますので、デスクトップ・アプリケーションに匹敵する、多様で使いやすい UI を実現できます。さらに、これらのウィジェットの継承や集約により、アプリケーションの要求に応じたカスタム・ウィジェットを作成することも可能です。

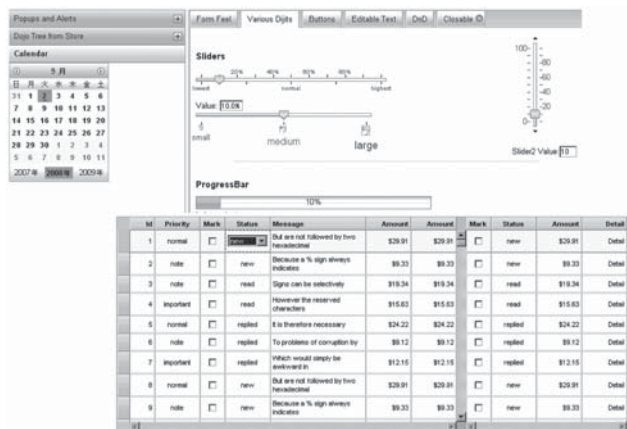


図6. ウィジェットの例

3.3 サービス・インターフェース

Ajax による REST ベースの Web API の呼び出しに加えて、JSON (JavaScript Object Notation) ベースの RPC 呼び出しや cometd と呼ばれる Web メッセージングなどもサポートされており、多彩なバックエンド・サービスへの接続が可能です。また、バックエンド・サービスの実装を隠ぺいするデータ API も定義されていますので、この API に従ってデータ・ソースへのアクセスを行えば、グリッド、ツリー、コンボ・ボックスなどのウィジェットに簡単にデータをバインディングすることができます。

3.4 グラフィックス

通常の HTML や CSS を利用したアニメーションやウィジェットの表現だけではなく、ブラウザ非依存の 2 次元グラフィックスの API がサポートされており、この API を利用することで、フロー・ダイアグラムをはじめとしたベクター・グラフィックスをブラウザ上で表示することが可能です。また、このグラフィックス API を利用したチャート描画機能も提供されているので、データを基にブラウザ上で動的に、折れ線グラフ、棒グラフ、円グラフなどを表示できます(図7)。

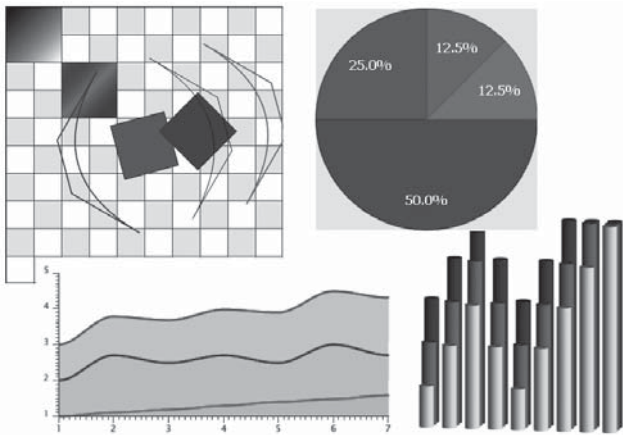


図7. グラフィックスの例

3.5 Dojo Toolkit の利用

Dojo Toolkit のコードは、プロジェクトの Web サイト (<http://dojotoolkit.org/>) からダウンロード可能で、Web アプリケーションからアクセスできるディレクトリーに展開すれば、すぐに HTML やほかの JavaScript コードから呼び出して使えます。IBM のソフトウェア製品でも、IBM WebSphere® Application Server (WAS) Feature Pack for Web 2.0 や WebSphere sMash はライブラリーとして Dojo Toolkit を出荷していますので、これらの Web アプリケーション・プラットフォーム上で Dojo Toolkit を使ったリッチな Web アプリケーション・フロントエンドを開発することが可能です。また、IBM Mashup Center や WebSphere Commerce など多くのソフトウェア製品の UI で、すでに Dojo Toolkit が使われています。

上述の Web サイトからは、「The Book of Dojo」 [8] と呼ばれる詳細な解説をはじめとして、デモ、チュートリアル、API リファレンスなど充実したドキュメンテーションも提供されていますし、Dojo Toolkit に関する書籍も発行され始めています。

現時点の最新バージョンは、1.1.1 ですが、すでに 1.2 のベータ版リリースが開始されており、世界中に広がる開発者の参画により、日々進化を続けています。Web アプリケーションのリッチなフロントエンドのための強力なプラットフォームとして、今後の発展が期待されるオープンソース・プロジェクトです。

4 広がるオープンソースの利用

すでに述べたように、Eclipse と Dojo Toolkit は、UI の中核技術として、多くの IBM 製品で使われています。

本稿では、クライアント技術関連の二つのプロジェクトを取り上げましたが、サーバー・サイドの技術としても多数のオープンソースの利用が進んでいます。例えば、軽量プログラミング・モデルを採用した新しい Web アプリケーション・プラットフォームである、WebSphere sMash でも、Groovy や Ivy といったオープンソースが基本技術として採用されています。今後も、オープン戦略に基づいたオープンソースへの投資とその利用の促進が、ますます重要になるでしょう。

[参考文献]

- [1] OSGi, <http://www.osgi.org/>
- [2] Lotus Mashups, <http://www.ibm.com/jp/software/lotus/products/mashups/>
- [3] Jazz, <http://jazz.net/>
- [4] eRCP, <http://www.eclipse.org/ercp/>
- [5] Server-Side Equinox, <http://www.eclipse.org/equinox/server/>
- [6] E4 project, <http://wiki.eclipse.org/E4>
- [7] The Dojo Toolkit: <http://dojotoolkit.org/>
- [8] The Book of Dojo: <http://dojotoolkit.org/book/dojo-book-1-0>



日本アイ・ビー・エム株式会社
ソフトウェア開発研究所
シニア・テクニカル・スタッフ・メンバー

野口 雅人 Masato Noguchi

[プロフィール]

1990年、日本 IBM 入社。ホームページ・ビルダー®、Rational Application Developer などの製品開発に従事。現在は Web 2.0/RIA 関連のソフトウェア開発を担当。



日本アイ・ビー・エム株式会社
ソフトウェア開発研究所
シニア・テクニカル・スタッフ・メンバー

若尾 正樹 Masaki Wakao

[プロフィール]

1990年、日本 IBM 入社。ホームページ・ビルダー、Rational Application Developer などの製品開発に従事。現在は Rational チーム開発製品の開発に携わる。