

エンタープライズ向けソフトウェア開発でのOSS活用

Utilization of OSS in Development of Software for Enterprise Use

田中 誠一郎

今村 大輔

■ TANAKA Seiichiro

■ IMAMURA Daisuke

企業向け情報システム（以下、エンタープライズシステムと記す）が動作するハードウェアがオープン化されて久しい。オープンアーキテクチャ上でのシステム構築の普及に伴い、OS（オペレーティングシステム）、ミドルウェア、アプリケーション・フレームワーク、コンポーネント、及び各種ツールといったソフトウェアもオープン化されている。このような背景の下、エンタープライズシステムのソフトウェア開発では、ソースコードが公開されているOSS（オープンソースソフトウェア）の積極的利用が進んでいる。

東芝ソリューショングループは、“東芝ソリューショングループ共通基盤 CommonStyle™”という、エンタープライズシステムを開発するための基盤技術を体系化しており、CommonStyle™を構成する基盤ソフトウェアにOSSを活用している。

Many years have passed since enterprise information systems were first built on open architecture computers. In tandem with the expanding development of enterprise information systems on open architecture computers, the development of operating systems, middleware products, application frameworks, components, and development tools in the open architecture environment has been taking place as well. As a result, open source software (OSS) is being progressively utilized in the development of software for enterprise information systems.

Toshiba Solutions Group is systematizing a set of fundamental technologies for the development of enterprise information systems called the CommonStyle™ Toshiba Solutions Group System Development Standard, utilizing OSS for the construction of its basic software.

1 まえがき

1998年に“Open Source”ということばが生まれてから10年余りが過ぎ⁽¹⁾、商用利用では“提供者による保守が担保されない”という課題がある一方で、エンタープライズシステムの開発では、OSS（オープンソースソフトウェア）が必須と言ってもよいほどその活用が浸透している。

東芝ソリューショングループは、東芝グループのICT（情報通信技術）事業を行う役割を担っており、エンタープライズシステムの開発は主要な事業内容の一つである。

ここでは、エンタープライズ向けソフトウェアを開発するための、東芝ソリューショングループ共通基盤 CommonStyle™におけるOSSの活用について述べる。

2 CommonStyle™の体系

2.1 CommonStyle™の構成要素

ICTの技術革新はビジネスの変化を加速させ、企業価値を高めるチャンスを広げる。このチャンスを生かすために、新たなビジネス要件に応えるシステムをタイムリーに実現することが必要である。急速な技術革新の下で高品質なシステムを短時間で開発するには、システム開発のノウハウの標準化とその再利用が重要である。

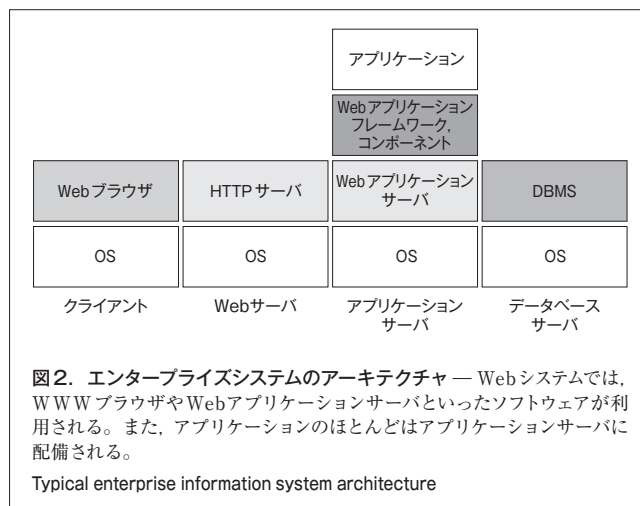
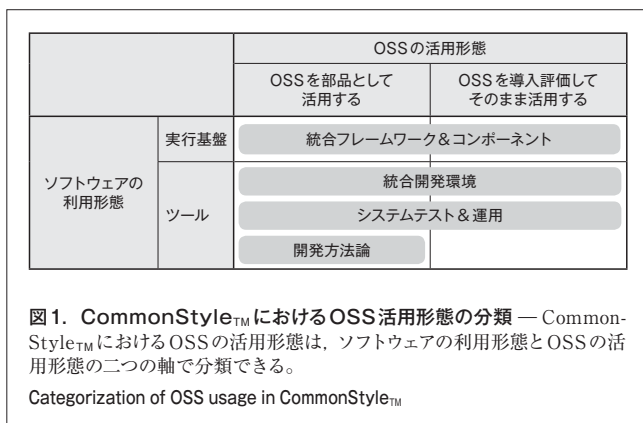
東芝ソリューショングループでは、システム開発のノウハウを“開発プロセス”と“開発基盤”の二つの観点で標準化しており、開発基盤を標準化したものがCommonStyle™である⁽²⁾。

CommonStyle™は、以下の四つのカテゴリから成る。

- (1) 開発方法論 (CommonStyle Methodology) システム開発を行ううえでの作業標準である。作業の定義と流れを示したプロセス、開発成果物のドキュメントを規定したテンプレート、及び開発のプロセスに沿ってポイントを示したガイドから構成される。
- (2) 統合開発環境 (CommonStyle IDE) プログラムを作成しテストするための統一的な環境を提供するためのツールである。
- (3) 統合フレームワーク&コンポーネント (CommonStyle Framework and Components) アプリケーションの構造を標準化し、設計・実装上の規約を提供するフレームワークと、共通な処理を部品化したコンポーネントである。
- (4) システムテスト&運用 (CommonStyle System Test and Management) 大量の負荷を発生させる負荷試験、及び人間の操作に沿って行う運用試験を支援するためのツールや、運用中の基礎データを採取し可視化するためのツールである。

2.2 CommonStyle™におけるOSSの活用形態

CommonStyle™のソフトウェアは、利用形態により次の



2種類に分かれる。

- (1) 実行基盤 システムに組み込んで利用するソフトウェアを指す。
- (2) ツール アプリケーションには組み込まれずに、開発や運用で利用するソフトウェアを指す。

実行基盤には“統合フレームワーク&コンポーネント”が該当し、ツールには“開発方法論”、“統合開発環境”、及び“システムテスト&運用”が該当する。

また、それぞれに対し、OSSの活用形態は次の2種類に分かれる。

- (1) OSSを部品として活用する形態 CommonStyle™として提供するソフトウェアの基本部や共通部として活用する。
 - (2) OSSを導入評価してそのまま活用する形態 OSSが目的にかなった利用ができるかを評価し、標準的なソフトウェアとして認定し活用する。
- これらのOSS活用形態の分類をまとめて図1に示す。

3 実行基盤としての活用

ここでは、CommonStyle™の実行基盤におけるOSSの活用事例について述べる。CommonStyle™では、エンタープライズシステムに対してアーキテクチャをパターン化して、そのアーキテクチャ上の役割ごとにOSSを活用している。

3.1 エンタープライズシステムのアーキテクチャのパターン化

近年ではWebシステムが主流であり、基本的なハードウェア構成は次のとおりである。

- (1) クライアント
- (2) Webサーバ
- (3) アプリケーションサーバ
- (4) データベースサーバ

各ハードウェアには、システムを構成するソフトウェアが役割に応じて配備されている(図2)。

CommonStyle™では、標準的なシステムアーキテクチャを、構成要素となるソフトウェアの特質を次の二つの軸で整理してパターン化している。

- (1) 論理層 ソフトウェアの機能的な役割による分類⁽³⁾
 - (a) クライアント層
 - (b) プレゼンテーション層
 - (c) ビジネス層
 - (d) インテグレーション層
 - (e) リソース層
- (2) コンポーネント層 ソフトウェアの汎用性による分類⁽⁴⁾
 - (a) システムソフトウェアコンポーネント層
 - (b) ミドルウェアコンポーネント層
 - (c) ビジネス非依存コンポーネント層
 - (d) ビジネス特化コンポーネント層
 - (e) アプリケーションシステム層

エンタープライズシステムのアーキテクチャパターンの例を、図3に示す。CommonStyle™のソフトウェアの主なカバー範囲は、ミドルウェアコンポーネント層とビジネス非依存コンポーネント層である。

以下に、CommonStyle™におけるこれらの層での主なOSSの活用状況について述べる。

3.2 ミドルウェアコンポーネント層におけるOSS活用

ミドルウェアコンポーネント層には、Webブラウザや、HTTP(Hypertext Transfer Protocol)サーバ、Webアプリケーションサーバ、データベース管理システム(DBMS)といったソフトウェアが位置する。この層におけるOSSの活用は、導入評価してそのまま活用する形態となる。CommonStyle™では、次のOSSのミドルウェアを標準的なソフトウェアとして認定している。

- (1) HTTPサーバ：Apache HTTP Server⁽⁵⁾
- (2) Webアプリケーションサーバ：Apache Tomcat⁽⁶⁾
- (3) DBMS：PostgreSQL⁽⁷⁾

	クライアント層	プレゼンテーション層	ビジネス層	インテグレーション層	リソース層
アプリケーションシステム層	コンタクトセンタソリューション				
ビジネス特化コンポーネント層	CRM共通フレームワーク				CRM共通データモデル
ビジネス非依存コンポーネント層			Drools		
	Rugthyme ワークフロー				
	Staveware Core				
ミドルウェアコンポーネント層	Internet Explorer ^(注1) 8/9	Tomcat, WebCube, WebSphere ^(注2) , WebLogic ^(注3)			PostgreSQL 8/9 Oracle ^(注4) Database 11g
システムソフトウェアコンポーネント層	Windows ^(注5) XP/7	Red Hat ^(注6) Enterprise Linux ^(注7) 5/6, CentOS 5, Windows ^(注7) Server 2008			RHEL, Windows ^(注7) Server 2008

RHEL : Red Hat Enterprise Linux

図3. CommonStyle™のアーキテクチャパターンの例 — CRM (カスタマー リレーションシップ マネジメント) のシステムを例にしたアーキテクチャパターンの例である。ミドルウェアコンポーネント層に図2に示したWebアプリケーションサーバやDBMSを具体的な製品で挙げている。また、ビジネス非依存コンポーネント層にWebアプリケーション フレームワークやコンポーネントを具体的な製品で挙げている。

Example of CommonStyle™ architecture pattern

DBMSは、エンタープライズシステムでは企業活動の結果のデータを蓄積し管理するものであり、非常に重要な位置を占めるとともに、大規模で複雑なソフトウェアである。このため、従来はOracle Database⁽⁸⁾のような商用製品を利用する場合はほとんどであった。しかし近年では、クラウドにおけるOSSのDBMSの活用が進んでおり、東芝ソリューション(株)のクラウド基盤にもPostgreSQLを採用している。

3.3 ビジネス非依存コンポーネント層におけるOSS活用

ビジネス非依存コンポーネント層には、Webアプリケーション・フレームワークや、ログ出力、メール送信といった各種コンポーネントが位置する。この層におけるOSSの活用は、OSSを部品として活用する形態と、導入評価してそのまま活用する形態の両方がある。

(1) OSSを部品として活用する形態 クライアント層からインテグレーション層までをカバーするフレームワークとしてStaveware™を開発し、整備している。Staveware™はWebアプリケーションを開発するためのフレームワークで、複数のOSSを組み合わせて機能を拡張している。以下に、Staveware™がベースとしているOSSを論理層の層ごとに述べる。

(a) クライアント層

- Dojo Toolkit⁽⁹⁾ GUI部品のライブラリとして活用

(注1), (注5) Internet Explorer, Windowsは、Microsoft Corporationの米国及びその他の国における商標。

(注2) WebSphereは、世界の多くの国で登録されたInternational Business Machines Corporationの商標。

(注3), (注4), (注8) WebLogic, Oracle, Javaは、Oracle Corporation及びその子会社、関連会社の米国及びその他の国における登録商標。

(注6) RedHat及びRed Hatをベースとした全ての商標は、米国及びその他の国におけるRed Hat, Inc.社の商標又は登録商標。

(注7) Linuxは、Linus Torvalds氏の米国及びその他の国における登録商標。

(b) プレゼンテーション層

- Apache Struts⁽¹⁰⁾ ユーザーのリクエストを処理する制御部として活用
- DWR⁽¹¹⁾ クライアントとの非同期通信のための通信部として活用

(c) ビジネス層

- Spring Framework⁽¹²⁾ 拡張性を高めるためのオブジェクトの管理部及びトランザクションの制御部として活用

(d) インテグレーション層

- Apache iBatis (現在はmyBatis)⁽¹³⁾ DBMSへの問合せ文 (SQL文) の管理容易性と拡張性を高めるための機構として活用

その他にも、log4j⁽¹⁴⁾やXerces⁽¹⁵⁾といったThe Apache Software Foundation⁽¹⁶⁾で開発され公開されているOSSを利用している。

(2) OSSを導入評価してそのまま活用する形態

CommonStyle™では、有用なOSSをそのままアプリケーション開発に活用するために、導入評価を行い、マニュアル及びサンプルアプリケーションを整備し、標準として認定したうえでサポートを実施している。次に例を示す。

(a) BI帳票ツール Eclipse BIRT⁽¹⁷⁾ HTML (Hypertext Markup Language) 及びPDF (Portable Document Format) 形式の帳票を出力する機能と、帳票をデザインするエディタから成る帳票ツール

(b) ビジネスルールエンジン Drools⁽¹⁸⁾ ビジネスルールをプログラムから切り離して定義することにより、頻繁なビジネスルールの変更を可能とするソフトウェア

4 開発ツールとしての活用

ここでは、CommonStyle™の開発ツールにおけるOSSの活用事例について述べる。

統合開発環境CommonStyle™ Studio for Java™(注8)は、OSSの統合開発環境Eclipse®をベースに、開発に有用な機能を追加したものである。Eclipseは開発環境としての機能統合基盤であり、追加機能はプラグインという形態で追加される。プラグインもOSSとして提供されているものが多く、CommonStyle™ Studio for Java™はこの仕組みを利用して機能追加をしている。機能追加には次の形態がある。

- (1) OSSのプラグインをそのまま追加
 - (2) OSSのプラグインに適切な設定変更を加えて追加
 - (3) プラグインを新たに開発して追加
- 各形態の代表的なものを表1に示す。

表1. CommonStyle™ Studio for Java™におけるOSS利用
OSS usage in CommonStyle™ Studio for Java™

形態	ツール名	OSS活用の説明
OSSのプラグインをそのまま追加	Javaカバレッジツールセット	単体テストを実行した際に、そのテストがどの程度テスト対象を網羅したかを計測する機能。JUnit®というOSSをそのまま追加した。
OSSのプラグインを設定変更して追加	Java静的解析ツールセット	開発者が作成したプログラムソースコードをルールに基づいてチェックし、品質的に問題がある箇所を警告する機能。FindBugs™(注9)、CheckStyle™といったOSSを活用している。ルールの設定を変更し、有効に作用するようにルールを調整した。
プラグインを開発して追加	Java画面作成ツール (ViewComposer™)、Javaコード生成ツール (AppComposer™)	ViewComposer™は、部品をキャンバスに配置して画面プログラムを作成する機能。AppComposer™は、設計情報を元にプログラムを自動生成する機能。ともに、Eclipseの描画ライブラリなどのOSSを利用して開発した。

5 あとがき

エンタープライズ向けソフトウェアを開発するために構築した、東芝ソリューショングループ共通基盤CommonStyle™におけるOSSの活用について述べた。

CommonStyle™では、顧客に高品質のシステムをタイムリーに提供するために、OSSを活用して開発基盤を整備している。利用形態には、目的とする機能を提供するためにOSSを部品として用いる場合と、OSSを導入評価してそのまま利用する場合がある。

OSSの進化は非常に速く、生産性及び品質の向上にとって、より有用なOSSが新たに公開されたり、現在利用しているOSSがバージョンアップされたりすることが頻繁にある。東芝ソリューショングループは、最新技術を用いたソリューションを顧客に提供するため、今後もOSSの評価並びに活用を継続していく。

(注9) FindBugsは、メリーランド大学の商標。

文献

- (1) Open Source Initiative. "History of the OSI". <<http://opensource.org/history/>>, (accessed 2012-05-18).
- (2) 東芝ソリューション(株). "東芝ソリューショングループ共通基盤CommonStyle™". <http://www.toshiba-sol.co.jp/core_tech/commonstyle/index_j.htm>, (参照2012-05-18).
- (3) Alur, D. et al. "Core J2EE Patterns: Best Practices and Design Strategies". Sun Microsystems Press, 2001, 496p.
- (4) ヤコブソン, I. 他. "ソフトウェア再利用ガイドブック". 杉本宜男 他監訳. シイエム・シイ出版部, 2000, 534p.
- (5) Apache Software Foundation. "The Apache HTTP Server Project". <<http://httpd.apache.org/>>, (accessed 2012-05-18).
- (6) Apache Software Foundation. "Apache Tomcat". <<http://tomcat.apache.org/>>, (accessed 2012-05-18).
- (7) PostgreSQL Global Development Group. "PostgreSQL". <<http://www.postgresql.org/>>, (accessed 2012-05-18).
- (8) 日本オラクル. "Oracle Database". <<http://www.oracle.com/jp/products/database/>>, (参照2012-05-18).
- (9) Dojo Foundation. "Dojo Toolkit". <<http://dojotoolkit.org/>>, (accessed 2012-05-18).
- (10) Apache Software Foundation. "Apache Struts". <<http://struts.apache.org/>>, (accessed 2012-05-18).
- (11) DWR-Easy Ajax for JAVA "DWR". <<http://directwebremoting.org/dwr/index.html>>, (accessed 2012-05-18).
- (12) SpringSource. "Spring Framework". <<http://www.springsource.org/spring-framework>>, (accessed 2012-05-18).
- (13) MyBaTis Blog. "iBatis". <<http://www.mybatis.org>>, (accessed 2012-05-18).
- (14) Apache Software Foundation. "Apache log4j". <<http://logging.apache.org/log4j/>>, (accessed 2012-05-18).
- (15) Apache Software Foundation. "Apache Xerces". <<http://xerces.apache.org/>>, (accessed 2012-05-18).
- (16) Apache Software Foundation. "The Apache Software Foundation". <<http://www.apache.org/>>, (accessed 2012-05-18).
- (17) Eclipse Foundation. "Eclipse BIRT Project". <<http://www.eclipse.org/birt/phoenix/>>, (accessed 2012-05-18).
- (18) Drools-JBoss Community. "Drools, Business Logic Integration Platform". <<http://www.jboss.org/drools>>, (accessed 2012-05-18).
- (19) Eclipse Foundation. "The Eclipse Foundation open source community website". <<http://www.eclipse.org/>>, (accessed 2012-05-18).
- (20) djUnit. "djUnit plugin for Eclipse". <<http://works.dgic.co.jp/djwiki/Viewpage.do?pid=@646A556E6974>>, (accessed 2012-05-18).
- (21) Pugh, B. et al. "Find Bugs in Java Program". <<http://findbugs.sourceforge.net/>>, (accessed 2012-05-18).
- (22) Schneider, D. et al. "The CheckStyle Plug-in for Eclipse". <<http://eclipse-cs.sourceforge.net>>, (accessed 2012-05-18).



田中 誠一郎 TANAKA Seiichiro

東芝ソリューション(株) 生産技術センター 生産技術担当主任研究員。東芝ソリューショングループ共通基盤CommonStyle™の開発に従事。
Toshiba Solutions Corp.



今村 大輔 IMAMURA Daisuke

東芝ソリューション(株) 生産技術センター 生産技術担当グループ長。東芝ソリューショングループ共通基盤CommonStyle™の開発及び普及と、生産技術の社内標準化に従事。
Toshiba Solutions Corp.