Webシステム開発効率化の取り組み

製造・流通システム開発本部
製造・流通システム開発部
水本　誠一
Seiichi　Mizumoto

製造・流通システム開発本部
製造・流通システム開発部
加藤　正也
Masaya　Kato

当社は、Web開発プロジェクトに対して、そのプロジェクトを確実に成功に導くために、IDAFと称する「Web開発フレームワーク」の整備活動を進めている。IDAFは、Webアプリケーションフレームワークと開発プロセスから構成される。主たる狙いは、Webシステム開発の効率化であるが、IDAFでは２つの側面からのアプローチによりその目標を実現しようとしている。１つは、開発プロセス自体の効率化、もう一つは、IDAFを利用する側、すなわち組織の成熟度向上による開発効率化である。本論にて、IDAFのWebシステム開発効率化実現コンセプトの詳細、およびプロジェクトへの適用事例を紹介する。

1. はじめに

Web関連技術は、その進歩の早さや市場規模の拡大によって注目を集めている。一方で、短工期化とベンダー間の競合による低価格化が進んでいるのも事実であり、われわれには、効率的な開発実施が求められるようにしている。

当社、製造流通システム開発本部では約８年間に渡りWeb系開発プロジェクトを実施してきたが、更なるシステム開発効率化実現のため、Webシステム開発プラットフォームとしての、アプリケーションフレームワークと開発プロセス整備活動に着手した。システム開発における、アプリケーションフレームワークの利用は、開発効率および保守性の向上に特に効果が高い。とくに大規模システムの開発におけるアプリケーションフレームワークの利用は、人に依存する開発スタイルの違いを吸収し、プログラムの生産性・可読性を向上させるため不可欠である。また、開発プロセスの整備は、これまで人の知識や技術に頼って行われた作業を視覚化することで、作業の効率化を図るとともに、人に依存するリスクを軽減できる。

当本部では、この開発プラットフォームをIDAF(Industrial and Distribution Application development Framework)と称し、Web系システム開発効率化実現のソリューションと位置付け整備活動を進めている。

本稿では、IDAFのコンセプト、構成および適用事例について紹介する。

2. 整備活動の背景

整備活動を開始した背景には以下の点があげられる。

（１）短納期化

先に述べたとおり、当本部では約８年間にわたってWeb系システム開発を行っているが、近年は短工期を要求されるケースが多い。B to B向けのシステムでもこの傾向が見られるが、B to C向けでは、競合他社と比べてどれだけ早いタイミングでサービスを立ち上げられるかがサービス成功の重要なポイントであるため、特にこの傾向が顕著である。

（２）低コスト化

近年、システム規模に対しての開発コストは景気の低迷やオフショアリングの普及など様々な要因により低下傾向がみられ、常にコスト削減を求められている。こうした状に悪影響を与える。更にひとつのプロジェクトの損失が会社全体の収益に大きな影響を与えるケースも少なくない。

（３）流動的な要員体制

ある程度の規模のプロジェクトを不定期に行っていれば、必然的にプロジェクトを構成する要員は流動的となる。特定のお客様向けのプロジェクトを継続していたとしても、必ずしもその流儀に精通した要員が確保出来るとは限らない。短工期プロジェクトでは新たな要員をどれだけ早く立ち上げるかといった点も成功の大きな要因となる。

このようにシステム開発を取り巻く状況は厳しさを増している。こうした状況においてもプロジェクトを実施する上で品質の確保は大前提であり、当社でもより効率的でより確実な開発が求められている。この、効率的で確実な開発作業の遂行を実現するための手段として、当本部では開発プラットフォームIDAFの整備活動に着手した。

IDAFは、部品群を含めたアプリケーションフレームワークと開発プロセスで構成されるWebシステム開発プラットフォームである。IDAFによって、実装フェーズを含めた各フェーズの成果物と作業手順が明確化され、システム開発者はその内容に従った開発を行う事で、品質・効率の両面での効果が期待できる。次の章でそのコンセプトの詳細について説明する。

3. IDAFコンセプトモデル

3.1. IDAF構成概要

IDAFの概要構成図を図１に示す。

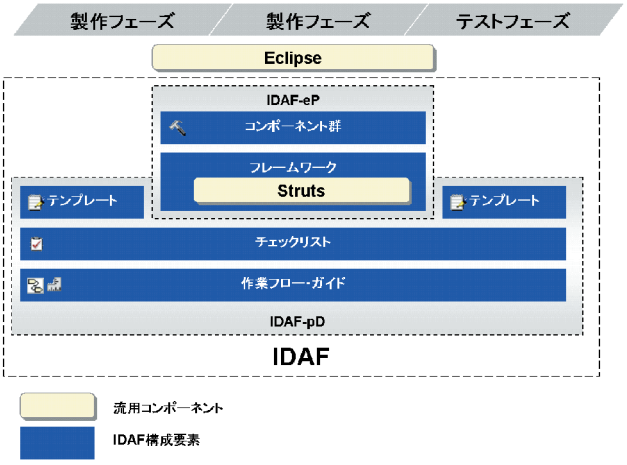

図１　DAF概要構成図

IDAFは、アプリケーション実行フレームワークIDAF-eP(IDAF Executable Platform)と作業フローや成果物テンプレート群を定義したIDAP-pD(IDAF Process Definition) から構成される Webシステム開発プラットフォームである。IDAF-ePは、J2EEに準拠した環境で稼働可能なWebアプリケーションフレームワーク、IDAF-pDは、IDAF-eP上に構築されるアプリケーションの開発プロセス定義文書である。

本章では、アプリケーションフレームワークと開発プロセスがシステム開発にもたらす一般的な効果を整理し、IDAFのコンセプトをまとめる。

3.2. アプリケーションフレームワークがシステム開発にもたらす効果

アプリケーションフレームワークは、アプリケーションの稼働に必要な処理の中で、共通的な処理を担当することで、個別のアプリケーションで開発すべき対象を減少させる。このアプリケーションフレームワークは、システム開発に以下の効果をもたらす。

（1）開発の効率化

アプリケーションの基本となる処理をフレームワークで吸収する事で、アプリケーションごとに実装するコード量が減少し、開発生産性の向上が期待できる。

（2）品質の向上、均一化

セッション管理やDBのコネクション管理などの共通的な処理がアプリケーションフレームワークに吸収されるため、個々の開発者の実装範囲およびテスト範囲が減少し、結果として品質の向上が期待できる。

また、アプリケーションフレームワークによって規定されたレイヤーリングポリシーは、アプリケーションに対してクラス／メソッドの分割指針を与え、作業スタイルの違いによる実装物の揺らぎを吸収する。これによりソースの可読性が向上し、問題および変更の影響範囲分析がより正確になる。

3.3. 開発プロセスがシステム開発にもたらす効果

一方、開発プロセスはシステム開発の手順を示すものであるが、システム開発に以下の効果をもたらす。

（1）作業の効率化

成果物およびその他文書のテンプレート、作業フローやガイドの利用により、作業が効率化される。

（2）成果物品質の底上げ

成果物間の整合性確認、実装時に配慮すべきセキュリティ確認事項など、成果物に対しての一般的な確認項目はチェックリスト化されている。このチェックリストの利用はアプリケーションにおける問題の早期発見につながる。また、こうしたアプリケーションロジックに直接関わらない問題を事前に解決しておくことで、レビューアーは本来実施すべきアプリケーション視点のレビューに集中できるため、品質の向上が期待できる。

3.4. IDAFのねらい

アプリケーションフレームワーク・開発プロセス個別で見た場合の効果は上記のとおりである。さらにわれわれは、IDAFを整備する上で①アプリケーションフレームワーク（IDAF-eP）と開発プロセス（IDAF-pD）を組み合わせで考えること、②組織内で継続利用し段階的改善を加えることに重点を置いた。以下に上記２点のねらいについて説明する。

3.4.1. 組み合わせによるねらい

アプリケーションフレームワークと開発プロセスを組み合わせで考える事の狙いは、以下の２点である。

（1）実装物と設計書の対応の明確化

設計書を特定のアプリケーションフレームワーク上で開発される実装物に絞る事によって、設計書と実装物との対応がより明確化される。対応の明確化によって、設計書から実装物を作成する際の曖昧さが減少し、品質の安定化が期待できる。また、対応をより明確化する事で、設計書からソースの自動生成なども期待できる。

（2）実用性の高いプロセスの定義

より実用性の高いプロセスを定義するために、対象とするアプリケーションフレームワークを特定した。これには、次のようなねらいが含まれる。

まず、対象とするアプリケーションフレームワークを絞ることにより、構造の異なる様々なフレームワークへの適用の妥当性を検討する必要がなくなり、整備活動自体のス

ピードを向上させることができる。さらに、フレームワークやレイヤーリングの考え方を特定する事で、より曖昧性の小さい、実用的なプロセスの定義が出来る。

3.4.2. 組織で継続利用する事のねらい

IDAFは組織内で継続利用する事で一段と効果を発揮する。そのねらいは以下の２点である。

（１）IDAFの成熟度向上

IDAFを特定プロジェクト用ではなく、組織で整備し、組織内の様々なプロジェクトに適用・評価を繰り返す事で、IDAFに段階的改善を重ね、IDAF自体の成熟度を向上させる。

（２）組織の成熟度向上

IDAFを組織内で繰り返し利用する事で、使う側のIDAFに対する成熟度を向上させる。

誰がやっても全てのプロジェクトを成功に導くアプリケーションフレームワーク、および開発プロセスの定義は現実的には不可能である。システム開発プロジェクトでは、毎回その要件・スケジュール・要員構成など多くの点で異なっており、様々な場面での状況判断を人が下さなければならない。つまり、システム開発を成功のためには、開発プロセスやアプリケーションフレームワークだけでなく、そのプロセスやフレームワークに対する使う側の成熟度も非常に重要な要素となる。IDAFでは、プロジェクトを超えた組織内での継続利用によってIDAFの利用機会を増加させ、IDAFと人の両方の成熟度を高めることによって、両者がバランスを取りながら成長してゆくことをねらいとしている。

3.5. IDAFのプロジェクトへの適用と段階的改善のサイクル

図２に、IDAFのプロジェクトへの適用とその改善活動の概要を示す。

１）適用

整備されたIDAFを、あるプロジェクトに適用する。適用されたプロジェクトにおいては、IDAFから恩恵を受ける一方で、多くの課題も発生する。これらの課題はプロジェクトメンバーの新たな実践によって解決されるが、一部の課題は解決されずに課題として残される。これらの実践や残された課題は、IDAFの今後の改善要素となる。

２）評価・改善

プロジェクトで発生した課題や実践された解決策・新たな試みは、プロセス改善検討会に持ち込まれ、IDAFの改善要素として検討される。残された課題に対してはその解決策を検討し、課題に対して実践された解決策に対しては妥当性やその課題自体の一般性を検討した上でIDAFにフィードバックする。

4. IDAF-ePの構成

本章では、IDAF-ePの構成について説明する。

4.1. IDAF-ePの特徴

IDAF-ePは、Strutsをベースとしたjavaアプリケーションフレームワークである。特徴としては、以下の４点が挙

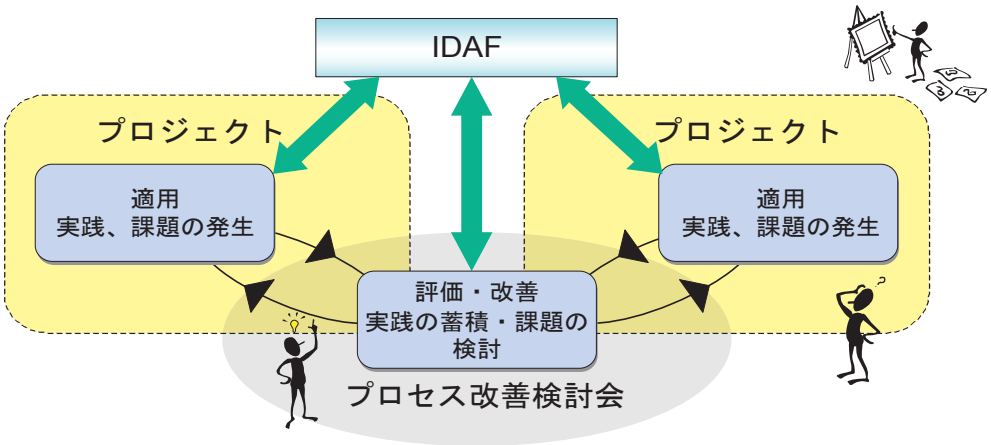

図２　プロジェクトへの適用イメージ

げられる。

（1）□他システム連携部品の充実
（2）□プラットフォームに依存しない動作環境
（3）□オープンソース環境での開発が可能
（4）□フォールトトレラント構成の実現

以下にその特徴を説明する。

（1）他システム連携部品の充実

図３に、IDAF-ePの概要構成図を示す。IDAF上で構築したサービスは、通常HTTPを経由し、HTMLベースで提供されるが、Apache AxisAXISなどのWebServiceコンポーネントの利用によりWebService経由での提供も可能である。またMQクライアントを利用することにより、MQを経由した他システムとの連携も可能で、多彩なサービス提供が可能である。

（2）プラットフォームに依存しない動作環境

IDAF-ePでは、特定のアプリケーションサーバに依存するパッケージなどは利用していないため、JBOSSやWAS（WebSphere Application Server）などJ2EEに準拠したアプリケーションサーバ上であればプラットフォームに依存せず稼働可能である。

（3）オープンソース環境での開発が可能

IDAF-eP上で稼働するアプリケーションの開発は、オープンソースであるEclipseを想定している。またテストツールもオープンソースのCactus、JUnitを想定しており、フリーソフトウェア環境での開発が可能である。

（4）フォールトトレラント構成の実現

EJBが実行されるAPサーバは、APサーバのクラスタリング機能を利用してフォールトトレラント構成の実現が可能である。APサーバ側をクラスタリング構成にすることで、障害発生時のAPサーバ切り離し／復旧時の切り戻しが自動の自動化が可能である。また、AP層のクラスタリング構成によって、水平負荷分散対応も容易で可用性の高いシステム構成が実現出来る。

IDAF-ePの特徴としては、上記の点が挙げられるが、こ

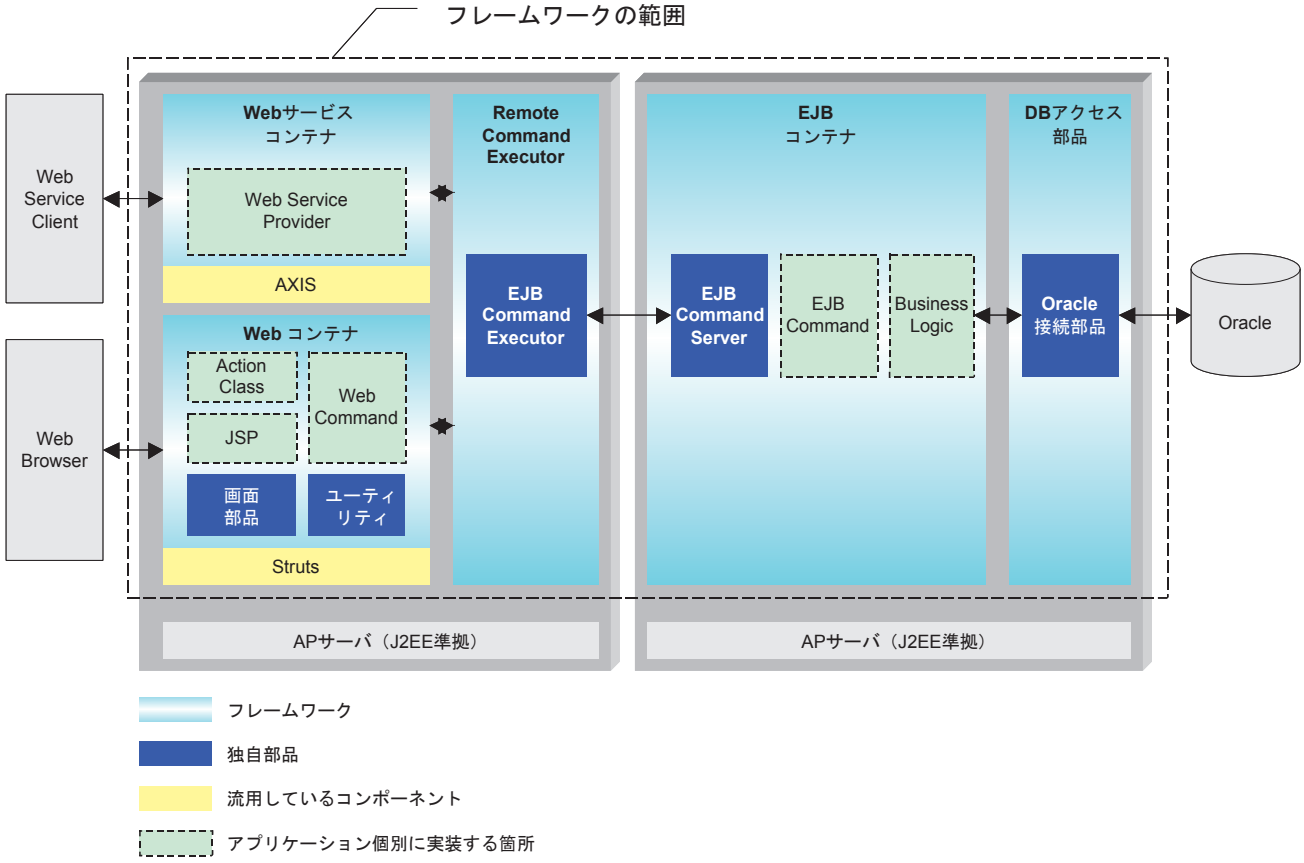

図３　DAF-eP構成図

れらの点に加えて、使用／流用する技術は、StrutsやEclipseなど、業界標準のものである。また開発ガイド、テスト実施ガイドなどIDAF-eP上でのアプリケーション開発に必要な文書も整備されており、IDAF-eP上での開発時のキャッチアップの容易性にも配慮している。

4.2. IDAFの機能

IDAF-ePはStrutsをベースとしたWebアプリケーションフレームワークである。Strutsのソースに修正を加えることなく、機能を追加している。以下にIDAFによって追加されている機能の一部を紹介する。表１

4.2.1. Webコンポーネント

StrutsではアプリケーションロジックをActionクラスのメソッドに記述する。アプリケーション実装者はActionクラスメソッド内に入力チェックや、業務処理、SQLなどまで記述する事ができ、設計及び実装の自由度が非常に高い。

IDAF-ePではフレームワーク上に実装されるアプリケーションに機能分割の指針を与える為、Actionクラスを含めて４つの階層を定めている。この指針によって実装物の可読性を上げている。

4.2.2. DBアクセスコンポーネント

Strutsにはデータベースにアクセスするためのコンポーネントは含まれていない。したがって、Strutsを使ってアプリケーションを構築する場合何らかのデータベースアクセス手段を用意する必要がある。IDAFではEntity BeanやO/Rマッピングのようなオブジェクト指向的な手段をとらず、JDBCを利用して直接SQLを記述する方法を採用した。Entity BeanやO/Rマッピング自体が比較的新しい技術であり、まだ成長過程にある点と、これまでのSQLの知識を利用したパフォーマンスチューニングが実施できる点が主な採用理由である。

4.2.3. EJBコンポーネント

IDAFではAP層のロジックの実装にEJBを採用している。EJBにはトランザクション管理の自動化などのメリットがある反面、クラス作成やデプロイがわずらわしいといったデメリットもある。IDAFではEJBCommandパターン(図４)を用いることによりこの問題を回避した。EJBCommandパターンの採用によって、EJB特有のわずらわしい処理をフレームワークに隠蔽し、個別のアプリケーションでは、本来処理すべき業務処理のみを実装すれば良い。これによって、アプリケーション実装者にかかる負担を軽減している。

4.2.4. 画面部品

HTMLの画面部品はStrutsが提供しているタグライブラリをそのまま利用可能である。さらにIDAF-ePでは、Strutsが提供する画面部品をより使いやすくするためのユーティリティや、ページ操作を行う際のカスタムタグライブラリなどを追加している。

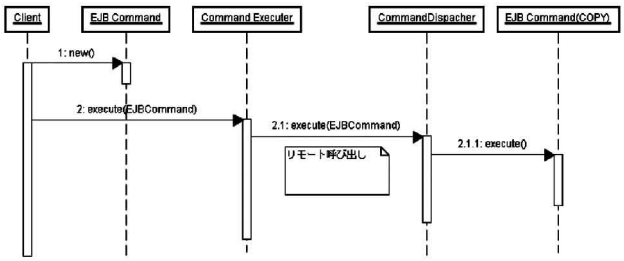

図４　EJBコマンドパターンのシーケンス図

表１　DAFでの追加機能

機能名	説明
Webコンポーネント	StrutsのActionクラスの拡張およびAction以降の階層を定義するクラス
DBアクセスコンポーネント	ORACLE接続用の共通関数
EJBコンポーネント	EJB CommandパターンにおけるCommandExecutorとCommandServer
画面部品	カスタムTAG Lib および画面表示用のデータBean
ユーティリティ関数	文字列操作、入力チェック、リソース取得などのユーティリティ関数

4.2.5. ユーティリティ関数

IDAF-ePでは日本語対応の文字列操作関数や入力チェック関数など、ビジネスアプリケーションを開発する上で必要となるユーティリティ関数を提供している。

4.3. IDAF-ePのレイヤー構成

IDAF-ePでは、サーバ側のロジックを「Action」「WebCommand」「EJBCommand」「BusinessLogic」の４階層に分割し、それぞれの役割を定義している（図５）。以下にそれぞれの役割について説明する。

4.3.1. Action

Actionは画面(JSP)と１対１で作成する。Actionクラスにはセッションタイムアウトや権限などの共通的な事前実行条件チェックと、画面ごとの入力チェックを実装する。

4.3.2. WebCommand

WebCommandは機能ごとに作成する。たとえば、１つの画面に「変更」と「削除」の機能がある場合は、変更処理用のWebCommandと削除処理用のWebCommandを作成する。また、複数の画面に同じ機能がある場合などは、１つのWebCommandを複数のActionで共用することができる。

4.3.3. EJBCommand

EJBCommandはEJBコンテナで実行されるCommandクラスで、トランザクション単位で作成する。トランザクションの管理はEJBのCMT(Container Managed Transaction)を利用する。EJBCommandは複数のBLを組み合わせてひとつの業務を完成させる、ワークフローを定義するクラスである。

4.3.4. BusinessLogic(BL)

BLはデータベースアクセスロジックを含む処理で、機能分割の最小単位である。BLは１つのテーブルを更新する、または１本のSQLの検索結果を返すなど、データベースに対するシンプルな処理を実装し業務処理などは実装しない。

5. IDAF-pD構成

5.1. IDAF-pDの構成とプロジェクトでの利用

IDAF-pDは、システム開発で実施する作業を定義する文書で構成される。それらの文書は、作業フロー、テンプレート、チェックリスト、ガイドの４種類に分類され、プロジェクトメンバーは、これらの文書を作業実施の際に適宜参照可能である。図６にこれら４種類の文書とプロジェクトで実施する作業との関連を定義する。

作業フローには、作業を行う際に利用するテンプレートやチェックリストが明記されており、プロジェクトでの作業者は、この作業フローをベースとして作業を進める。実際にプロジェクトにおける作業の進め方の概要を以下に示す。

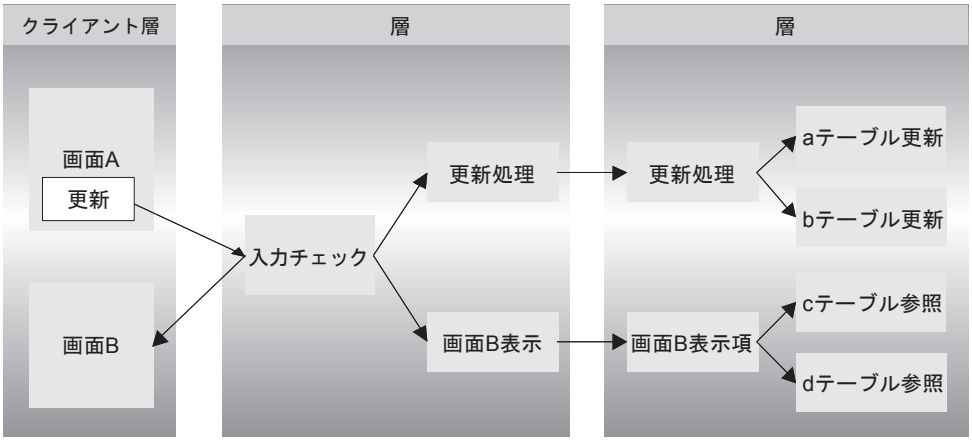

図５　DAF-ePレイヤ構成

① 作業者は、作業フローに定義されている作業順序にしたがって作業を進める。作業フローには、プロジェクトで実施する作業の順序および作業で使用するテンプレートやチェックリストが定義されている。

② ガイドには、作業内容の指針が定義されている。作業者は、ガイドで作業指針を確認後、テンプレートをベースとして成果物を作成する。

③ チェックリストには、作成した成果物に対する確認項目が定義されている。作業者はこのチェックリストを自身が作成した成果物に対してのセルフチェックやレビューチェックリストとして利用する。

作業実施者は、上記定義文書の内容にしたがって作業を実施する。それぞれの文書の構成概要と実際にIDAF-pDで用意されている文書の抜粋を表２に示す。

5.2. IDAF-pDの特徴

IDAF-pDの特徴について、以降で説明する。

（1）Webシステム開発に特化した設計書テンプレート、チェックリスト

IDAFは、Web上でのシステム開発プロジェクトを対象

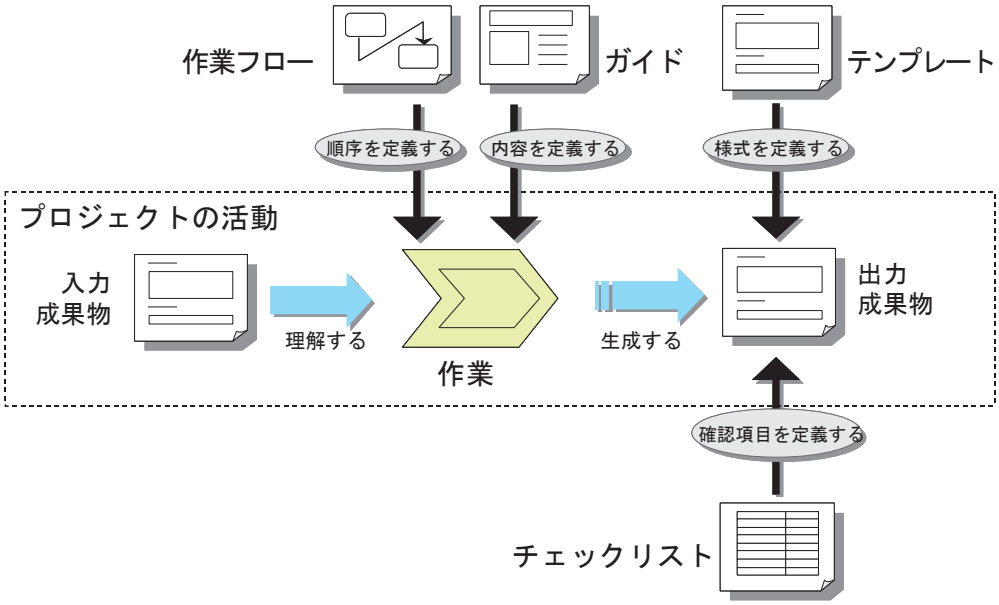

図６　定義文書と作業との関連

表 2　定義文書の構成概要と抜粋

文書種別	構成の概要	定義文書抜粋
作業フロー	全体概要フローとフェーズごとの詳細フローから構成される	・全体概要フロー ・外部設計フェーズ作業フロー
テンプレート	成果物に対するテンプレートと議事録などプロジェクトを通して必要となるテンプレートから構成される。	・外部設計書成果物テンプレート ・内部設計書成果物テンプレート ・標準系テンプレート（画面レイアウト・操作標準テンプレート、設計標準テンプレート）
チェックリスト	各成果物に対するチェックリストから構成される。	・議事録テンプレート ・外部設計書成果物チェックリスト ・内部設計書成果物チェックリスト
ガイド	プロジェクトで特に注意する事項に対してガイドが作成されている	・実装物セキュリティチェックリスト ・テスト実施ガイド・レイヤーリングガイド

としている。そのため、Webシステム開発で注意すべき事項をまとめたテンプレートやチェックリストが整備されている。表３にそれらの中のいくつかを簡単に紹介する。

（２）IDAF-ePに特化した設計書テンプレート､作業ガイド

IDAF-pDは、IDAF-eP上で稼働するアプリケーションの開発を前提としている。各レイヤーの機能分割方針は、レイヤーリングガイドに定義され、その結果生成された設計書は、IDAF-ePの各レイヤーの実装物と対応付いている。

図７に、IDAFで定義されている主な設計書と実装物との対応を示す。図中の太線は、現状で自動生成を実現している作業である。

各レイヤー上の実装は、Javaクラスメソッド実装となるが、各レイヤーで実装する処理の特徴に応じて最適化された設計書が用意されている。BL層はDBアクセスのためのDBアクセス仕様書が入力資料となり、Validation系および整合性の事前チェックが主な処理となるWeb層では入

表 3　Webシステム開発向けの定義文書

文書名	概要	項目例
Web画面レイアウト・操作標準テンプレート	Webシステムを開発する際に忘れがちな画面制御のルール事項をまとめたテンプレート	・カーソル形状制御 ・JavaScriptの処理範囲方針 ・ボタン２度押し制御 ・Web禁止文字対策など
セキュリティチェックリスト	Webシステムの設計・実装時にセキュリティ面から注意すべき事項をまとめたチェックリスト	・Web公開フォルダへの一時ファイル保存 ・Cookieの設定など

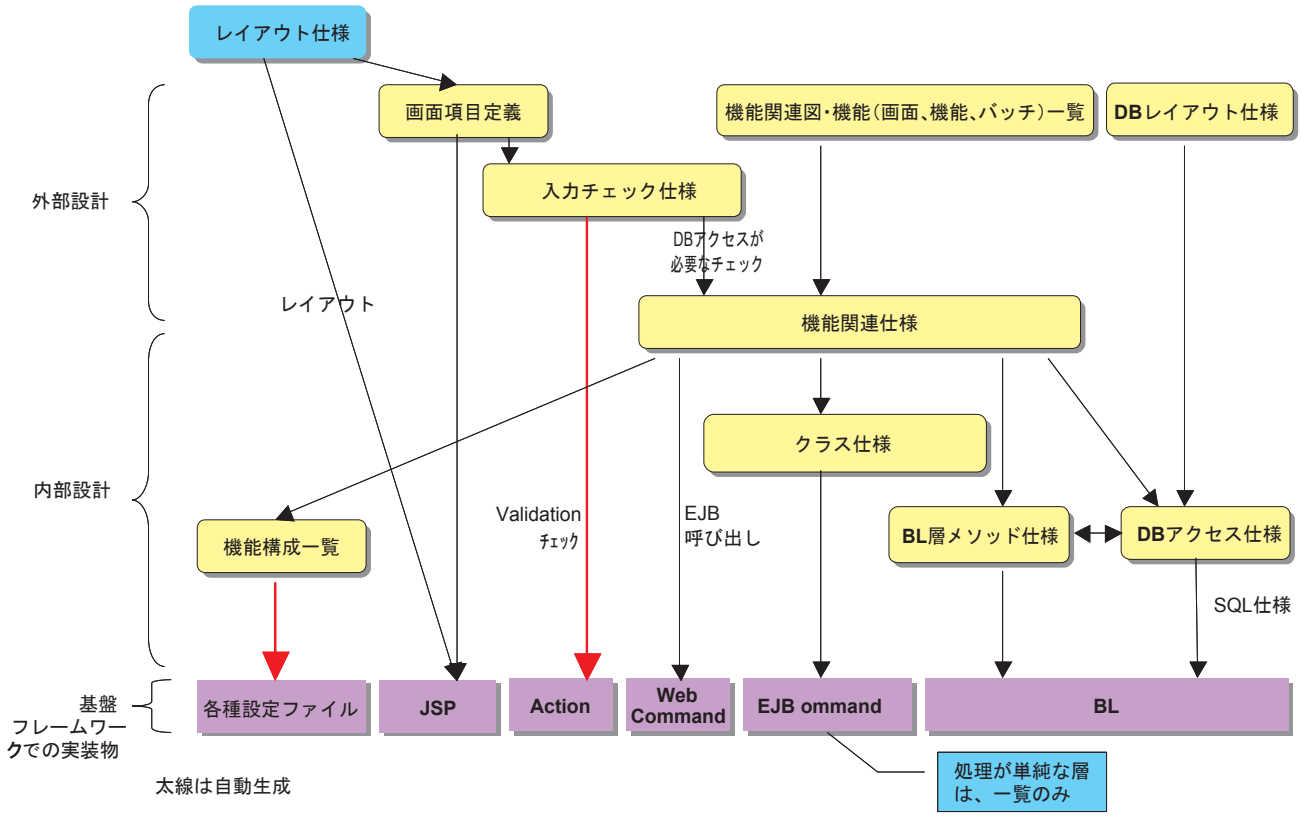

図７　プロセスで定義されている主な設計書と実装物との対応

力チェック仕様が実装作業の主な入力資料となる。

5.3. 改善例

IDAFでは、組織内での継続的な改善を続け、組織全体での成長をねらいとしている。IDAFプロセスの初版を整備してから、約１年間が経過するが、その間にも改善が繰り返されている。以下の表に具体的にこれまでに実施してきた改善内容を紹介する。（表４）

6. 適用事例

DAFは、これまでに既にいくつかのプロジェクトに適用されている。本章では、適用プロジェクトの一つであるNet商品販売サイト連携システムについて、その概要と特徴について簡単に紹介する。

6.1. システム概要

既に稼働中のNet商品検索、注文サービスを他の販売サイトに公開し、公開先のサイト経由で注文を受け付けるシステムである。図８にシステム構成の概要を示す。

当サイトでは、商品検索／予約サービスが構築されており、HTTPで一般ユーザに公開されていた。開発したシステムでは、このサイトが保有する商品検索／予約機能をWebサービスとして他のサイトに公開した。

表４　DAF改善例

課題	対応
設計フェーズのセキュリティ面で配慮すべき点が整理されていない（チェックリスト化されているのは実装フェーズのみ）	設計フェーズのセキュリティチェックリストを追加整備
詳細設計にかかる時間が大きすぎる	設計書間で重複する記述を削除
画面、機能の関連がつかみにくいため、レビューが難しい	画面、イベント、機能の関連を定義する機能関連図を成果物に追加

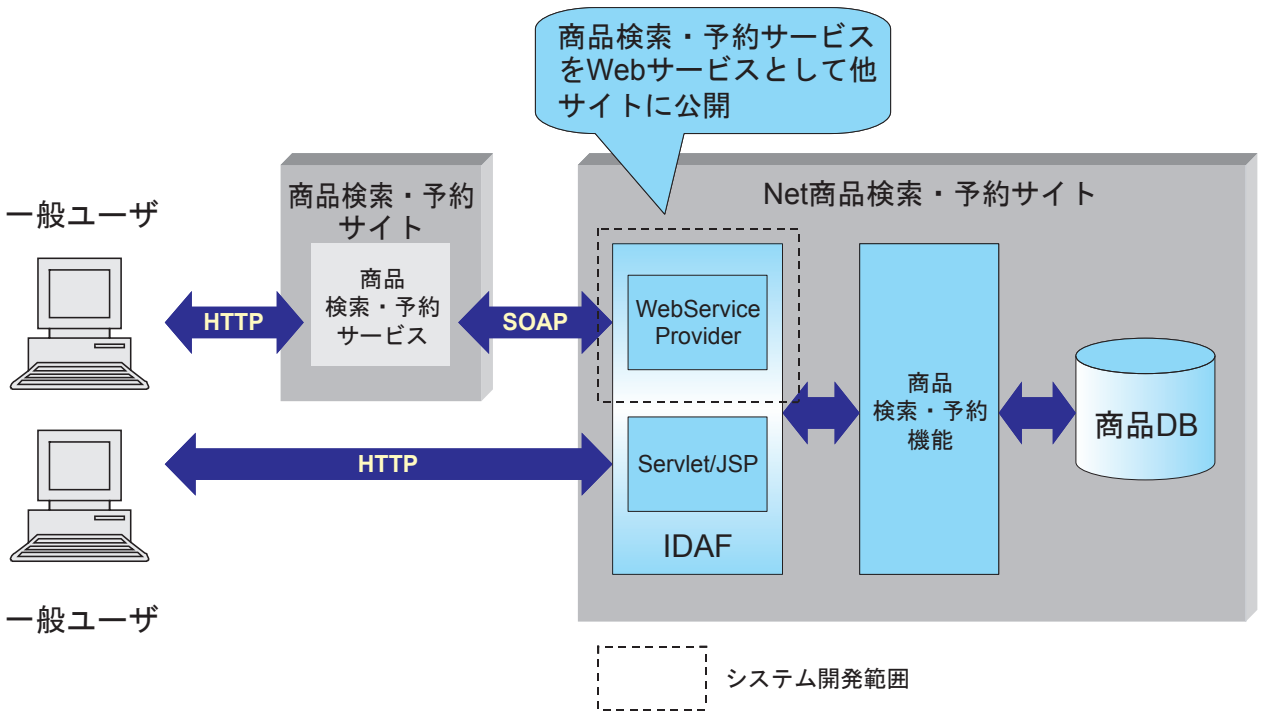

図８　システム概要構成図

6.2. システムの特徴

当システムの特徴は、以下の２点である。

（１）Webサービスを使ったサービス提供

（２）稼働中の商品検索／予約機能の流用

当システムでは、既に構築が完了し、HTTP経由で一般ユーザに公開されている信頼性の高い機能をそのまま流用している。この方法によって、低コストでのシステム構築を実現している。

6.3. その他適用事例

IDAFは、これまで中・小規模のプロジェクト中心に適用を行ってきた。その中には１日当たり５万アクセスを超えるサービスや、１００人月を超える規模のシステムも含まれる。表５は、これまでに適用してきたプロジェクトの抜粋である。

7. 評価

7.1. 改善点

IDAFを実プロジェクトに適用した結果、これまで行ってきた開発と比べ、以下のような改善点が見られた。

（１）開発工数の削減

外部設計～サブシステム内連携テストで約30%程度(IDAF適用前にわれわれ組織で使ってきたフレームワーク上での開発と比較した場合の試算値)の工数削減を実現した。IDAFでは、これまで不足していた詳細設計書の記述内容を追加している。この結果、詳細設計にかかる工数には若干増加が見られたが、一方でコーディング～UTにかかる工数が大幅に削減され、結果として30%程度の工数削減を実現した。

（２）開発スピードの向上

開発手順、ガイド、テンプレートの整備によって、一番工数が必要となるコーディング・UTフェーズに投入可能な人数が増加し、主にこのフェーズでのスピード向上が実現出来た。

（３）問題対応スピードの向上

レイヤー別の処理範囲、粒度の定義によって影響調査のスピードが向上し、結果として、問題対応スピードが向上した

7.2. 今後の改善に向けて

IDAFは組織での継続利用を想定しているが、継続利用してゆく上で以下のような課題が挙げられる。

（１）効率的なプロジェクト経験収集

実プロジェクトへの適用を通して得られる課題や実践は、プロセスやアプリケーションフレームワークにとって非常に有益な情報である。こうした情報をより効率的に収集する為の仕組み作りが必要である。

（２）新技術の取り込み

現在は、実プロジェクトを通して得られた課題や実践を改善ポイントとして主に検討している。しかしながら、Web系の新技術も同様に改善ポイントとして検討する必要がある。新技術の取り込みには、既存システムへの影響やその技術自体の成熟度や将来性など充分な考慮が必要となる。フレームワーク自体を陳腐化させない為に、今後こうした技術に常に目を配り改善ポイントとして検討してゆかなければならない。

表５　DAF適用事例

システム概要	規模	概要および特徴
共用認証システム	60人月	5万／日を越えるアクセス数
Net商品情報登録システム	100人月	Net販売商品情報の登録システム 商品情報の登録と検索用テーブルへの展開 100人月規模のシステム
リリース管理システム	15人月	テスト環境にリリースされているモジュールを本番環境へ移行するツール
商品購入システム	50人月	オンライン決済による商品購入システム

8. まとめ

本稿では、製造流通システム開発本部におけるWebシステム開発の効率化への取り組みについて紹介した。

システム開発を取り巻く環境は、年々厳しさを増している。こうした環境に適応するため、われわれは長年のWebシステム開発経験を元にアプリケーションフレームワークと開発プロセスの整備を行った。アプリケーションフレームワークは、個別アプリケーションでの実装範囲を減少させることによりアプリケーション実装作業を効率化できる。また、開発プロセスは作業手順や成果物を明確化する事で、主に設計作業を中心に作業を効率化できる。

われわれは、整備したアプリケーションフレームワークと開発プロセス実際のプロジェクトに適用することで、システム開発が効率化される事を確認できた。しかしながら、われわれはシステム開発効率の一時的な改善だけでなく、組織内での継続利用と段階的改善による組織の継続成長もねらいとしている。

IDAFは最初の整備活動が完了してからまだ1年と日が浅い。しかし、実プロジェクトへの適用を通して得られた課題・実践を元に改善を行った。今後も更なる生産性・品質の向上のため、こうした活動を今後も継続していきたい。

＜問い合わせ先＞
製造・流通システム開発本部
製造・流通システム開発部
Tel 044-540-2857　水本 誠一
E-mail:seiichi-mizumoto@exa-corp.co.jp
