JFE 技報 No. 14
（2006 年 11 月）p. 40–44

EA（Enterprise Architecture）手法による製鉄所システム構造評価と刷新への取り組み

Evaluation and Renovation of the Information System Architecture of Steelworks with an EA Methodology

大森　清生　OOMORI Sugao　JFE スチール　西日本製鉄所　企画部システム室長
釣井　　豊　TSURII Yutaka　エクサ　福山システム開発部　技術推進チームマネージャー

要旨

製鉄所システムは技術革新や業界動向の変化に追随しながら，30 年以上にわたって所の生産管理業務を支えてきた。その一方で，システムの巨大化・複雑化・多様化は，経営ニーズへの対応の遅れなどの遠因となりつつある。JFE スチール　西日本製鉄所では，このようなシステムの現状を，米国や日本の大手企業で実績のある EA（enterprise architecture）手法を用いて分析し，とくに老朽化更新の必要性が高いオープン系サーバについては，更新の指針となる技術標準および基盤評価基準を策定した。

Abstract:

The information system in West Japan Works of JFE Steel has provided an important function on its production control operations for past over 30 years, while following both technical innovations and market trends. On the other hand the system became enormous and complex, and it might give a negative impact on the business, such as a slow response to business needs. JFE Steel analyzed the current state of the system with a methodology called enterprise architecture (EA), and established policies and standards of the future system architecture, especially for the aging servers which require immediate replacement with those adopted advanced technologies.

1. はじめに

製鉄所システムは稼働後 30 年以上経過しているものも多く，システムも巨大化・複雑化・多様化している。西日本製鉄所ではシステムの刷新にあたり全体最適の視点から EA 手法を用いて現状分析を行うとともに，中でも老朽化更新の緊急度が高いオープン系サーバの更新についての技術標準・基盤評価基準を策定した。

また，この基準に基づきメインフレームのオープン環境上へのサーバ統合を行い，安価で変化に強いシステム化に成功した。

2. 製鉄所のシステム化の歴史

2.1 システム化の経緯

製鉄所のシステム化の歴史は，古くは 1960 年代後半に遡る。当初のプロセスコンピュータ設置にともなう実績集計システムから始まり，1970 年代の要員・歩留まり・省エネルギーなどの合理化を目的とした各種オンラインシステムの開発，1980 年代は設備リフレッシュとプロセス連続化・同期化を設計思想とする総合システムの開発に至った。その後も，リードタイム短縮などの顧客サービスと設備資産の有効活用を両立させる一貫計画システム化を推進し，生産管理システムのレベルアップを実施してきた。1990 年代に入ると，ダウンサイジングの流れにより従来のメインフレーム上の確立されたオンラインおよびバッチ方式から，オープン環境によるシステム構築が増加した。

2.2 ダウンサイジングの波

ダウンサイジング化の流れは，ハード・ソフトおよび目的に合わせたツール・言語を導入し，より早くシステムを構築することが特長であった。急激なダウンサイジングの波と日進月歩の技術革新により標準技術が確立せず，ハードウェア（以下，H/W），オペレーティングシステム（以下，OS），ミドルウェア（以下，M/W），言語，データベースなどさまざまな製品が乱立した。システム化の都度最適な製品を選択した結果，複数アーキテクチャが混在した構造となった。システムが巨大化するにつれアーキテクチャ相互間のインターフェースが複雑化し，経営ニーズへの対

応遅延やシステムトラブルの遠因になりつつある。

3. EA について

3.1 米国における EA の発展の経緯 [1)]

EA は米国政府の IT 関連調達改革のなかで大きく発展してきた。米国では，各府省が個別に IT 化を推進してきたことによる税金の無駄遣いが指摘されたことより，EA の有効性が検討され適用されてきた。すなわち，システム開発費用やアプリケーション保守費用が適正かどうかを見極めるために，EA の導入が行われてきたのである。日本では経済産業省を中心に EA の進展が図られており，先進的な企業では実質的な取り組みが本格化している。代表的な EA のフレームワークとしては 1999 年に CIO Council（各府省情報化統括責任者 連絡会議）が発表した FEAF（federal enterprise architecture framework）がある。

3.2 EA の 4 階層モデル

Fig. 1 に代表的な EA のフレームワークに共通する 4 階層モデルを示す。EA では現行アーキテクチャ（as-is）をビジネス，アプリケーション，データ，テクノロジーの 4 つの視点で可視化し，ターゲット・アーキテクチャ（to-be）を策定する。一方で，システム全体のビジョンやプリンシプルを明確にし，また各種スタンダードを定義する。そして，現行とターゲットの間のギャップを明確にした上で移行計画を立てて，ターゲット・アーキテクチャへ移行するというアプローチを取る。

3.3 西日本製鉄所への EA 適用

西日本製鉄所では，従来よりシステム構築において次の開発理念を掲げてきた。

(1) 将来にわたって変化に強いシステム構造
(2) IT 投資の適正化

オープン系システムの導入から 10 年が経過した昨今，数百台にも及ぶサーバの中には，製造中止から数年（5〜7 年）を経て保守サービス終了となる機械が顕在化し始めた。システム刷新が急務となったこの機会を契機に，オープン系システムについても改めて上記理念をプリンシプルとして定め，アーキテクチャの標準・基準の基礎とした。具体的には，上位アーキテクチャ（BA，AA，DA）を変えずに，プリンシプルに準じた TA の見直し・変更を図り，ターゲットアーキテクチャへ移行させたのである。

4 章ではプリンシプルを具現化するための実現方法，5 章ではターゲットアーキテクチャに移行するためのシステム構造評価と刷新方法について記述する。

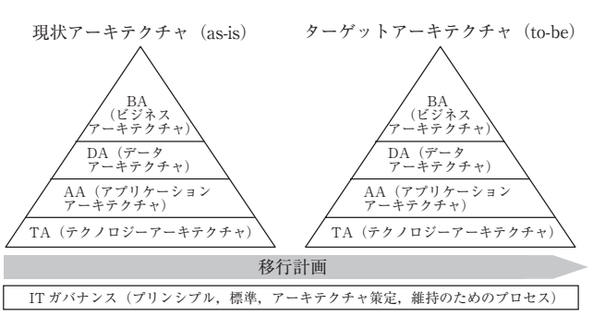

Fig. 1 4-layered model of EA

4. TA におけるプリンシプルの具現化

4.1 変化に強いシステム基盤

サーバは H/W，その上で稼働する OS，OS の上で稼働する M/W，M/W の提供するアプリケーションプログラムインターフェース（以下，API）を使用するアプリケーション（以下，AP）から構成され階層構造となっている。導入後 10 年以上経過したシステムでは，**Fig. 2** の各階層の関係から H/W だけでなく OS や M/W をバージョンアップあるいは異なるソフトウェアに変更する必要がある。

バージョンアップの場合は API が大幅に変更されることは少ないため，AP から見たインターフェースを **Fig. 3** のラッピングにより保証し変更を最小化できる。

一方，製品の販売中止や同等の機能を持つ製品がないなどの理由により発生する，異なる OS や M/W への変更は AP に甚大な影響を与える可能性がある。AP への影響を最小化するためには，

(1) 複数 H/W で稼働する業界標準に従った OS や M/W を採用すること
(2) 特定のプラットフォーム（H/W と OS の組み合わせを意味する）に依存する API を使用しないこと

が重要である。オープン系システムの更新では，Fig. 3 のラッピングにより標準 M/W に置き換えを継続することに

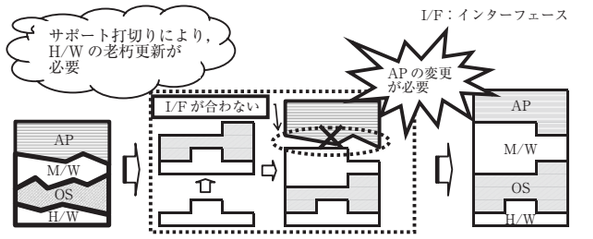

Fig. 2 Impact of hardware change to applications

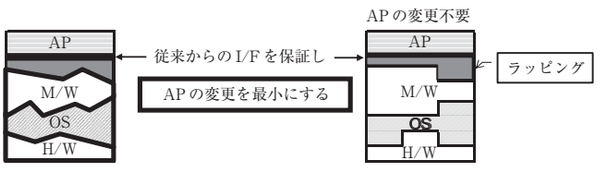

Fig. 3 Minimization of application change with wrapping technique

より，H/W や M/W などのテクノロジー面の変化に強いシステム基盤を実現できる。

4.2　サーバ統合による IT 投資の適正化

オープン系システムでは，1つのシステムが1台以上のサーバによって構成されることが多い。メインフレームと異なり，各システムがサーバを占有できるこの形態は，安定したレスポンスの保証や障害範囲の局所化などの点で効果的であった。

その反面，バックアップや監視，障害対応などの運用負荷の増加，老朽化したサーバの更新負荷の増加など，運用期間が長期化するにつれて問題が顕在化してきた。それらの要因はサーバ台数の増加にともなって増大する傾向にあるため，サーバ台数を削減することのできるサーバ統合がその有効な解決手段となる。

サーバ統合は，台数の削減，基盤の標準化という2つの要素がある。具体的には，ソフトウェアライセンス数の削減や運用負荷の削減，基盤標準化によるテスト期間の短縮などがあげられる。（**Fig. 4**）

このように，IT 投資の適正化を実現する上で，サーバ統合を意識した設計を行っていくことは重要である。

5.　オープン系システムの構造評価と刷新方法

5.1　システムの現状分析（as-is）

サーバ，クライアント，メインフレームなど他システムとのインターフェースを含めた構成図を作成する。サーバやクライアントに導入されている M/W 間の連携を把握することによりシステムの方式設計が明らかになる。現状分析は以下の手順により行う。

(1) プロセスの配置と起動方法

サーバ，クライアント，メインフレームについてプロセスの配置場所と起動方法を把握することにより，必要な M/W の機能を洗い出す。

(2) データの配置

上記プロセスとデータの関係を把握する。

(3) システムの運用制約

サーバ統合の検討を行う際に必要な運用上の制約条件を抽出する。具体的にはシステムの起動・停止，バックアップのタイミングなどを調査し複数システムの統合が可能か判断する材料とする。

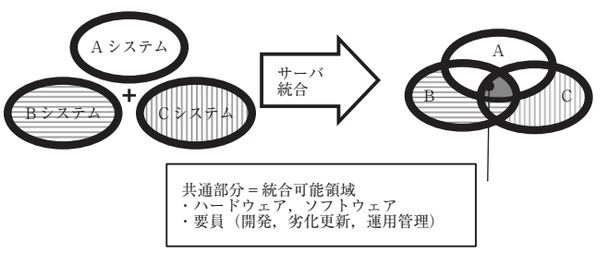

Fig. 4　Concept of server consolidation

5.2　評価基準によるプラットフォーム選定

システムを類似する機能要件を持つグループに分類し，各グループの機能要件から最も評価点の高いプラットフォームを以下の手順により導出する。

(1) グループから機能要件を抽出

例：「操業系システム」は「信頼性が高い」，「シミュレーション」系は「大量の計算を行う」など

(2) 機能要件を成立させる非機能要件を定義

例：信頼性を高くするためには「H/W 故障率が低い」，大量の計算を行うためには「CPU を占有できる」など

(3) 非機能要件について各プラットフォームを相対評価

例：「メインフレーム」は「サーバ」より H/W 故障率が低い，「サーバ」は「メインフレーム」に比べ CPU を占有できるなど

(4) グループに対して各プラットフォームの評点を計算

(1) から (3) の評価基準を **Fig. 5** の A・B・C のマトリックスで表現し掛け算する。評点結果（マトリックス P）から各アプリケーショングループに適したプラットフォームを選択する。

5.3　プラットフォームとシステム構造の決定（to-be）

更新後のシステムのプラットフォームと M/W の決定，サーバ統合の可否判断を行う。

(1) Fig. 5 により決定したプラットフォームに更新可能か判断

必要な M/W が稼働する，あるいは同様の API を提供する M/W が存在するかチェックする。OS が提供する API も同様のチェックを行う。

(2) 標準 M/W への置き換え

Fig. 3 のラッピングにより同一の API を提供できる場合，標準 OS や標準 M/W への置き換えを行う。こ

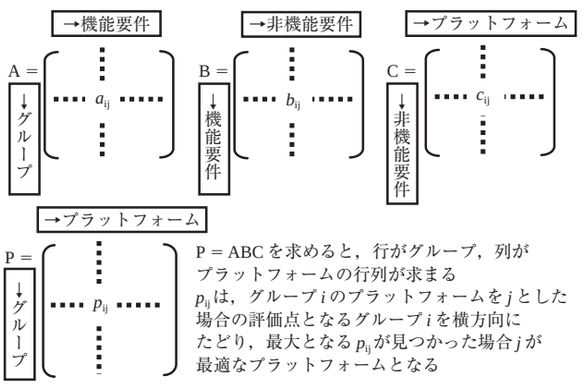

Fig. 5　Platform evaluation criteria

れにより将来的なプラットフォーム変更に追随できるシステム構成とする。

(3) サーバ統合の可否判断

機能面，性能面，運用面からサーバ統合が可能か判断する。

(4) 移行計画の策定

サーバの更新プランを作成し，各年度の更新費用と必要開発要員を算定する。

6. 刷新事例

5章で述べたプラットフォームの選定基準と構造評価手法により，UNIX* 機の操業系システムをメインフレームのオープン系 OS に移行し，2004 年 8 月より現在までトラブルなしで稼働している。また，オープン系 OS の API への更新により，4 章で記述したプラットフォームの変更が容易な構造が実現できた。

Windows** 上の標準 M/W を使用している管理系システムを，数週間でオープン系 OS に移行でき，プラットフォーム変更が容易であることが実証できた。

上記 2 システムに新規構築の物流系システムを加えたサーバ統合の刷新事例について，更新前後におけるサーバ数，基盤の種類，ソフトウェアライセンス数の比較を **Table 1** に示す。

7. 考察

7.1 EA 手法の有効性

EA は IT 投資最適化の枠組みを提供するものであって，その実現手段を定義するものではない。西日本製鉄所では，4 章，5 章で述べたとおり，EA を実現する手法を確立した。以下，その有効性について考察したい。

(1) 変化に強いシステム基盤への更新

刷新事例に見られるように，標準的な M/W の採用によりプラットフォームの一元化や変更が容易となる。この結果，システム要件に合ったコスト効率の高いプラットフォームを，将来にわたって選択できるようになる。

(2) 評価基準によるプラットフォーム選定

プラットフォームの選定にあたっては，システム開発の都度妥当性を検討するのではなく，あらかじめ合意された評価基準を適用するほうが効率的かつ客観的である。また，この基準は技術動向や，稼働後のシステムの評価をもって見直しを図っていくため，先進性の維持にも貢献する。

7.2 刷新による適正化

EA の概念を取り入れた今回のシステム刷新により，以下の適正化が図られた。

(1) ハードウェアの削減

6 台のサーバを 1 台のサーバに圧縮できた。

(2) ソフトウェアライセンス数の削減

CPU 数に依存するソフトウェアライセンス数を 4 から 2 に削減できた。

(3) AP 開発期間の短縮

操業系システムの事例では，移行先のオープン系 OS で UNIX と同等の API 機能を提供した。開発期間は API 開発とテストに短縮され，新規再構築時の 1/5 で実現できた。

サーバ数およびライセンス数を削減でき，従来より AP の開発および改造期間が短縮できたことから，IT 投資の適正化に寄与したといえる。

7.3 EA におけるガバナンスの重要性

サーバの更新は一過性のプロジェクトではなく，所を取り巻くビジネス環境や技術動向に適合させて，継続的に最適化を図るための恒常的なプロセスでなくてはならない。EA でいうところのガバナンスは，サーバ更新の維持・管理に必要なプロセス，ルール，スタンダード，評価基準，役割と体制を定義するものである。IT の最適化を実現するには，ガバナンスの重要性を認識し，**Fig. 6** に示す PDCA のマネージメントサイクルを実施することが重要である。

Table 1 Comparison of quantity

	更新前			更新後		
	サーバ数	基盤	ライセンス数	サーバ数	基盤	ライセンス数
操業系システム	3	UNIX	0	1	オープン系 OS	2
物流系システム	1	オープン系 OS	2		オープン系 OS	
管理系システム	2	Windows	2		オープン系 OS	
合計	6	3 種類	4	1	1 種類	2

* UNIX は，米国およびその他の国におけるオープン・グループの登録商標である。
** Windows はマイクロソフトの登録商標である。

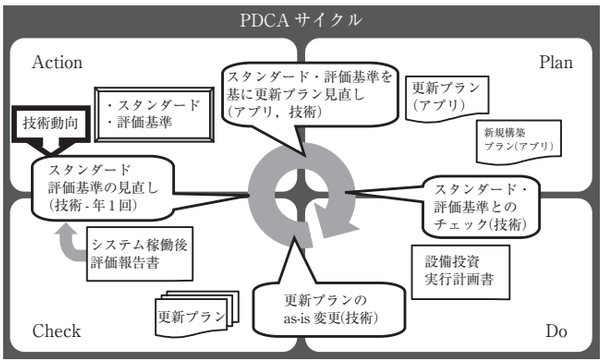

Fig. 6　PDCA cycle of systems update

8.　おわりに

EA 手法により老朽更新の緊急性の高い一部オープン系基幹システムの刷新が実現できた。さらに EA の全体最適の見地から，

(1) 数百台のサーバについて老朽更新プランの策定

(2) ガバナンスと呼ばれる管理プロセスの確立

ができたことは以降のオープン系システムの構築・運用にとって有用である。今後はメインフレーム上の基幹系システムの刷新においても本手法で得られた知見を活かし発展させていきたい。

参考文献

1) IBCS IT 戦略グループ．エンタープライズ・アーキテクチャ．2003, p. 32–39.

大森　清生

釣井　　豊