

SOAを活用したRapid Enterprise Renovation(SOA RER)のアーキテクチャ

山下 眞澄

SOA-driven Rapid Enterprise Renovation Architecture (SOA RER)

Masumi Yamashita

金融機関のCIO(最高情報責任者)や情報システム(IS)部門長には、システムが社会のインフラとして堅牢性を維持でき、かつ長期的な視野を持った全社全体最適を目指すエンタープライズ・アーキテクチャ(EA)視点のアーキテクチャを維持することが課せられている。その一方、昨今の激動する経営環境の中で、部分最適であっても市場への即応性や早い投資対効果(ROI: Return on Investment)を目指すプロジェクトへの投資を決断しなければならない。本論文ではその解決の方向性を示すべく、SOA(Service Oriented Architecture)を活用したRapid Enterprise Renovation(SOA RER)アーキテクチャを新たに構築した。その構造はSOAラッパーを活用したシンプルなものにもかかわらず、堅牢な基幹系システムと即応性を持ったシステムを同時に進化させ、かつインテグレーションすることができる。現在SOA RERアーキテクチャは正式にプレス・リリースされ、IBM Yamato Lab.で実装され稼働実証されている。

Chief Information Officers (CIOs) and directors of Information System (IS) divisions in financial institutions are being burdened with maintaining their architecture from the point of view of Enterprise Architecture (EA), which aims at company-wide total optimization with a long-term perspective that can preserve the robustness of systems as social infrastructure. On the other hand, they have to decide on investment in projects which aim at Return on Investment (ROI) and readiness in the market, despite partial optimization, in the rapidly changing business environment of recent years. This paper describes SOA (Service Oriented Architecture)-driven Rapid Enterprise Renovation (SOA RER) which is able to solve the above conflict by leveraging SOA wrapping technology. SOA RER is currently being formally announced in press releases, and has been implemented as a demonstration system in the IBM Yamato Lab.

Key Words & Phrases : サービス指向アーキテクチャ ,エンタープライズ・アーキテクチャ ,エンタープライズ・テクノロジー・フレームワーク ,ラッパー ,リファクタリング
SOA, enterprise architecture (EA), enterprise technology framework (ETF), wrapper, refactoring

1 .はじめに

日本の金融機関のCIO(最高情報責任者)や情報システム(IS)部門長の典型的な「悩み」を次の言葉に集約することができる。「業務案件は目白押しであるが、基幹系システムの設計開発担当者には限りがあり、その実現は法制度対応優先、かつ逐次化されてしまう。しかし要員調達が可能だからといって、膨大な機能を作り込んできた基幹系システムを分散系で一気に再構築するには、要件定義すらおぼつかない。

さらに顧客接点におけるサービスの高度化対応として新チャネル開設や、場当たりに構築してきた分散系システムの老朽化対応も同時に行う必要があるが相互影響が心配だ」。CIOやIS部門長には、金融機関システムが社会のインフラとして堅牢性を維持でき、かつ長期的な視野を持った全社全体最適を目指すエンタープライズ・アーキテクチャ(EA)視点のアーキテクチャを維持することが課せられている[1][2]。その一方、昨今の激動する経営環境の中で、部分最適であっても市場変化への即応性や早い投資対効果(ROI: Return on Investment)を目指すプロジェクトへの投資を決断しなければならない。過去から少

提出日: 2007年2月21日 再提出日: 2007年6月27日

しずつ増大してきたこのような危機に対応するために行ってきた日本の金融機関システムのアーキテクチャの歴史的特徴は以下の2点にある。

- ① 1980年代中ごろから基幹の勘定系をスクラップ・アンド・ビルドしIMS™ (Information Management System)を中心としたデータ・システムズ環境 (DSE: Data Systems Environment)を用いた第3次オンラインを構築。
- ② 1990年代後半からは分散システムや営業チャネルの多様化への対応としてハブ・アンド・スポーク・アーキテクチャを実装。この概念を適用したアーキテクチャはFHNE(Financial Hub Network Environment) [3]とよばれ、MQ(Message Queuing)やHTTPプロトコルをベースとしたメッセージ・ブローカーやゲートウェイで実装されてきた。

しかし近年さらに急激に迫ってくるCIOやIS部門長の課題を解くために、金融機関のシステム・アーキテクチャはDSEやFHNEの次を目指してどのように変化していけばよいのであろうか。本論文ではその解決の方向性を示すべく、既存の基幹系の特性維持と市場変化への即応性を両立させるエンタープライズ・レベルでのアーキテクチャの革新として、SOA(Service Oriented Architecture)を用いた Rapid Enterprise Renovation(SOA RER)アーキテクチャを新たに構築した。SOA RERとは本論文で述べるアーキテクチャの愛称であり、その構造はSOAラッパーを活用したシンプルなものであるにもかかわらず、堅牢な基幹系システムと即応性を持ったシステムを同時に進化さ

せ、かつインテグレーションすることができる。

2章では解決策と代替案、3章ではEAのモデル化手法を用いてSOA RERアーキテクチャの設計思想を述べ、4章で結論と今後の課題を述べる。

2. SOA RERアーキテクチャによる解決と代替策

SOA RERアーキテクチャの基本コンセプトは、現在の大量の事務を支える堅牢な基幹系を維持し徹底再利用しつつ、SOAにより新商品や新サービスを俊敏に実現させることである。つまり、堅牢かつ高速大量処理に最適化され、確立されたアーキテクチャを持つ現行システムと、新商品・サービス開発のスピード化・柔軟性を実現できる新システムを明確に2つのワールド(現行ワールドとSOAワールド)として分離し、その間を業界標準に準拠した「サービス」の単位で柔軟に連携でき、最適に融合することである。

2.1 SOA RERアーキテクチャによる解決

図1にSOA RERの概念レベル・アーキテクチャを示す。SOA RERアーキテクチャの主たる設計思想は「ラッパーを経由し、他のワールドのアーキテクチャを徹底仮想化し、相互の影響を最小限におさえること」にある。SOAワールド(2階部分)のサービス・リクエスターが現行ワールド(1階部分)のバックエンド・システムをアクセスするとき、ラッパーはSOAワールドのサービス・プロバイダーとして扱われるが、現行ワールドでは通常のフロント・システムとして仮想

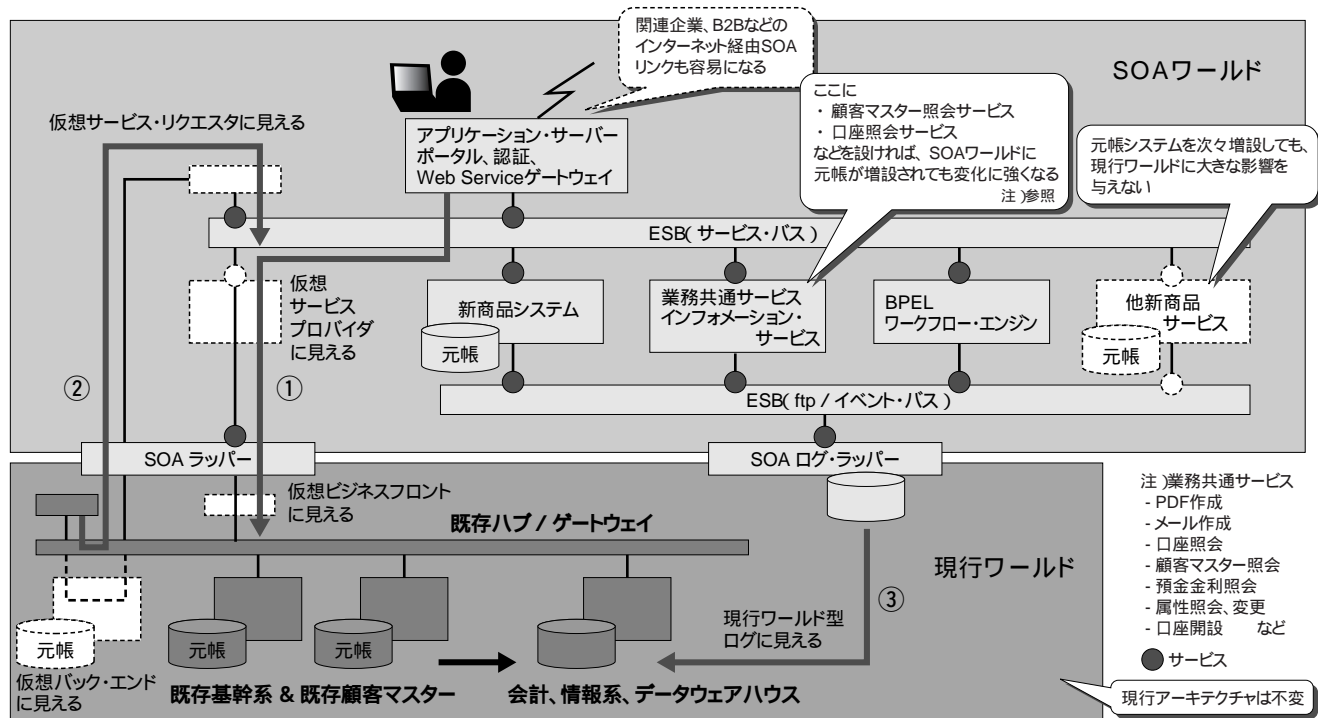


図1. SOA RERの概念レベル・アーキテクチャ(参照アーキテクチャ)

的に扱われる(図1①矢印)。逆に、現行ワールド(1階部分)のフロント・システムがSOAワールド(2階部分)をアクセスするとき、ラッパーは現行ワールドでは仮想バックエンド・サーバーの一つとしてしか見えないため従来の技術とスキル要員が活用できる。そしてSOAワールドではそのラッパーは仮想的にSOAサービス・リクエスターとしてしか見えない(図1②矢印)ためここではSOA技術とそのスキル要員だけで対応が可能である。加えて、SOAワールドで発生した業務ログはログ・ラッパーで現行ワールドのログの形式に可能な限り変換する(図1③矢印)ため、現行ワールドのデータウェアハウスへの影響を抑えることができる。これは「コロンブスの卵」的でシンプルすぎる設計思想かもしれない。しかしこれらのラッパーは単なるプロトコル変換器ではない。明確に定義されたサービスという業界標準のインターフェースを介して相互に仮想化することにより、それぞれの進化や変化を各ワールドの技術の中だけで充足させる絶縁装置の役割を担っている。つまり、現行ワールドは現在活用されているDSEやハブやゲートウェイを維持したままSOAワールドへ影響を与えずに活用、進化することができる。またSOAワールドの変化による現行ワールドへの影響を最小化することができる。

SOA RERアーキテクチャの主たる設計思想はこの仮想化が主題であるが、第3次オンラインで成功した以下6点の勘定系設計思想を温故知新的に企業レベルに発展させることで、より強固なものとすることができた。

- ① 勘定系中心のハブ・アーキテクチャから、企業全体をハブ構造で統治するEAの視点への進化(3章)
- ② 個別API(Application Programming Interface)によるエニ・ツー・エニから、標準化されたサービスによるエニ・ツー・エニの実現
- ③ 業務共通コンポーネント再利用の考え方から、業務共通サービス再利用への進化(3.3章)
- ④ IBM IMSを基盤としたトランザクション・フレームワーク製品(IBM SAIL, IBM CAP-A)による開発生産性向上から、分散アプリケーション・フレームワーク(IBM NEFSSライブラリー製品等)とBEPL4WS(Business Process Execution Language for Web Services)による開発生産性向上への進化
- ⑤ 顧客管理ファイル(CMF/CIF)の構築から、物理複数元帳を許す仮想顧客マスターへの進化(3.1章)
- ⑥ 社会基盤としての大量処理、事務の堅確化と堅牢性は維持しつつ、この既存のアーキテクチャには影響を与えず、柔軟性、タイム・ツー・マーケット、商品サービスの多様性の実現

2.2 代替策との比較

CIOやIS部門長の課題はこのSOA RERアーキテクチャによる2つのワールド構造以外に、以下のような解決策も考えられる。

① 基幹系への組み込みや、基幹系の再構築

この施策は前述のように、基幹系システムの設計開発担当者には限りがあり、その実現は法制度対応優先かつ逐次化されてしまうことに加え、少顧客小型案件でも現行機能への無影響確認に多大な作業負荷が必要となる。さらに再構築になるとROIリスク、要件定義リスク、要員調達リスク、移行リスク、運用リスクの観点から採用できるケースは非常に少ないと考えられる。現行システムがかなりひどい状態であっても、SOA RERアーキテクチャの考え方であれば、2つのワールドが互いにサービスで疎結合化しており相互の影響を最小限に抑えることができる。このため、現行の更改をおだやかに進めつつ、新業務にもすばやく対応できる可能性が大きい。これは、サービスによる疎結合化を徹底したため基幹系の再構築でも周辺システムには影響を与えなかった英国 IF.com社のSOA事例[4]に見ることができる。

② 既存のハブ・アンド・スポーク、ゲートウェイ、アダプターの延長

これは、すでにフロント・システム、バック・システムと2層化されているFHNEに基づいた現行システムをそのまま延長し、その設計思想を踏襲していく方法である。SOAにかかわる新たな投資を必要とせず、またSOAなどの新たな知識の習得も必要がない。しかし、今後は基幹系システム中心からサービス中心に向かい、またグループ企業や他社との柔軟な連携が戦略方針になっていくことを考えると、アーキテクチャ・トポロジーは類似していても独自のAPIでしかアクセスできない表記の仕組みだけでは柔軟性や俊敏性への対応能力は低いといわざるを得ない。また特段のアーキテクチャ設計思想を作らず、SOAを個別ソリューション構築から小さく始め、徐々に大きくしていく考え方もある。しかし最も多くの情報量とその操作機能を持つ現行基幹系の再利用を最初から考慮するSOA RERアーキテクチャではサービスの反復再利用のROIをより早く得ることができると考えられる[5]。

SOA RERアーキテクチャは上記2つの案に比して前述のCIOやIS部門長の課題をより良く解決することができる。その反面FHNE時代以上にマスター・データベースやプロセスが分散することで、後述するいくつかの課題(3.2章)も顕在化しやすい。しかしSOA RERアーキテクチャは基本のコンセプトであり、状況によって例外が許されないわけではない。あくまでもEAにおけるプリンシプルの解釈すべきで、現実解はス

テークホルダー合意の下での例外とするなど EAガバナンスを用いて判断すべきである。

3. エンタープライズ・テクノロジー・フレームワーク(ETF)

EAでは企業レベルのITアーキテクチャを表現する方法の一つにエンタープライズ・テクノロジー・フレームワーク(ETF)がある[6]。ETFとはEAレベルでITアーキテクチャ構成要素をモデル化したものである。これはIBM GS Method[7]に従っておりファンクショナル(機能)ETFとオペレーショナル(運用基盤)ETFから構成される。前者は機能的なコンポーネントを発見し、その間の静的、動的な関係を構築手法に従ってモデル化して構築する。また後者は論理的な処理装置(ノード)を想定し、その構成を物理的な配置(ロケーション)図の中に置き、基盤配置手法に従ってコンポーネントを配置していく。SOA RERアーキテクチャはソリューション単位だけでなく企業レベルのアーキテクチャであることから、本論文ではSOA RERアーキテクチャの設計思想を詳細化するためETFを活用した(図2)。本論文では、IBM GS Methodにおけるコンポーネントをサービスと置き換えることでETFの考え方を発展させることができた。またサービス候補特定、仕様化も含めたボトムアップの詳細化の手法も新たに確立した(3.3章)。

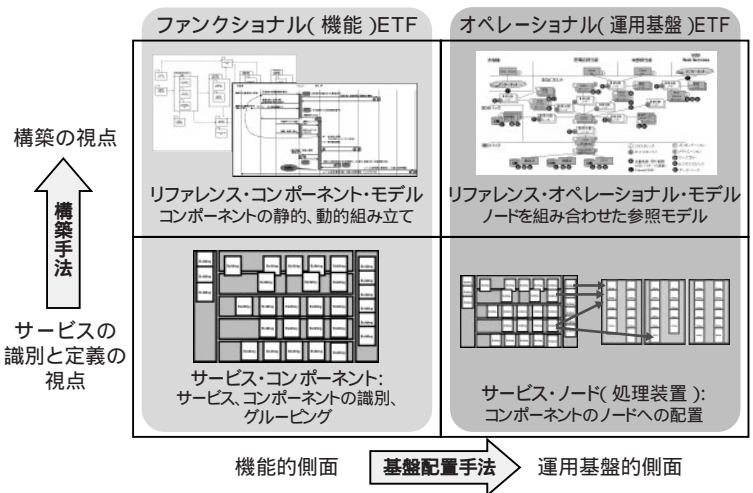


図2. SOA RERアーキテクチャのエンタープライズ・テクノロジー・フレームワーク(ETF)

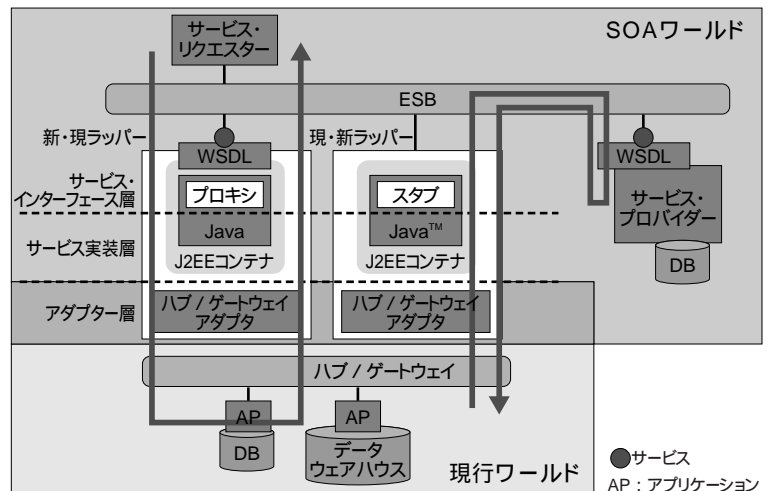


図3. SOAラッパー

3.1 オペレーショナルETF

図1は概念レベルのSOA RERアーキテクチャ参照モデルでもある。前述のように2つのワールド間でラッパーを用いた仮想化による疎結合が主たる特徴である。しかしSOAワールドと現行ワールドを結合するラッパーは単純な「コロンブスの卵」だけではない。なぜならサービスの流れにも双方向性があり、また情報アクセスの仮想化も必要だからである。そこで以下のような4つのリファレンス・アーキテクチャを新たに構築した。

① SOAラッパー(図3)

1) 新現ラッパー

SOAワールド起動のラッパー

2) 現新ラッパー

現行ワールド起動のラッパー

双方向のオンライン・ラッパーが必要であるが、投資順序としては1)新現ラッパーが先行すると考えられる。なぜなら、新商品・サービスを実現するに当たっ

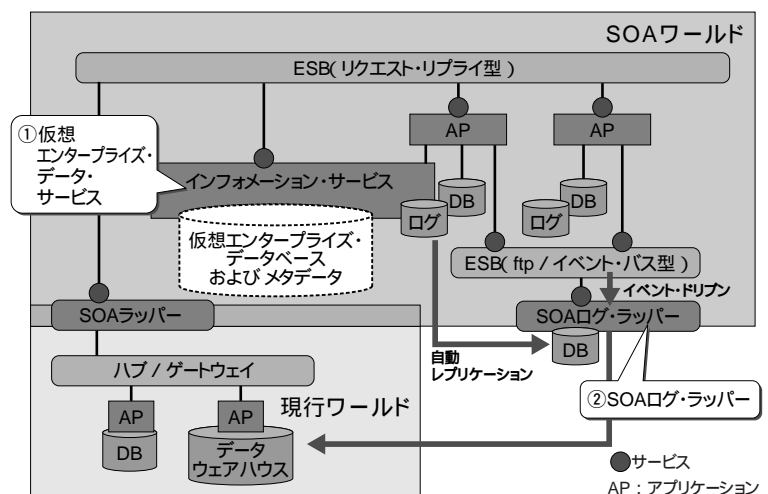


図4. インフォメーション・ラッパー

て、現行システムを改造するより、まったく新規に構築するフロント・システムがサービス・リクエスターとなって、既存システムをサービスとして再利用するほうがリスク低減やROIの視点で有利であろうと考えられるからである。

② インフォメーション・ラッパー(図4)

1) 仮想エンタープライズ・データ・サービス

Xquery(XML文書を検索する標準)やSQLによるデータ・フェデレーション、またはインフォメーション・サービス(データ収集用ウェブ・サービス)による仮想データ・ベースへのアクセスで実現する。サービスの例としては、複数に分散されたマスター・データ・アクセス・サービスや分析データ・アクセス・サービスなどが考えられる。製品としてはIBM WebSphere® Information Integratorなどが活用できる。

2) SOAログ・ラッパー

SOAワールドで記録された業務ログは、変換することであたかも現行ワールドで記録されたかのように扱えると現行への影響を最小化できる。自動レプリケーション、ftp、イベント・パスなどで収集したデータを市販のETL(Extract / Transform / Load)ツールやTRLX(IBM 5655-G47 金融インダストリー製品、MQベースのログ収集基盤)などで振り分け、編集し、現行で処理可能なフォーマットへ変換するアーキテクチャが必要となる。

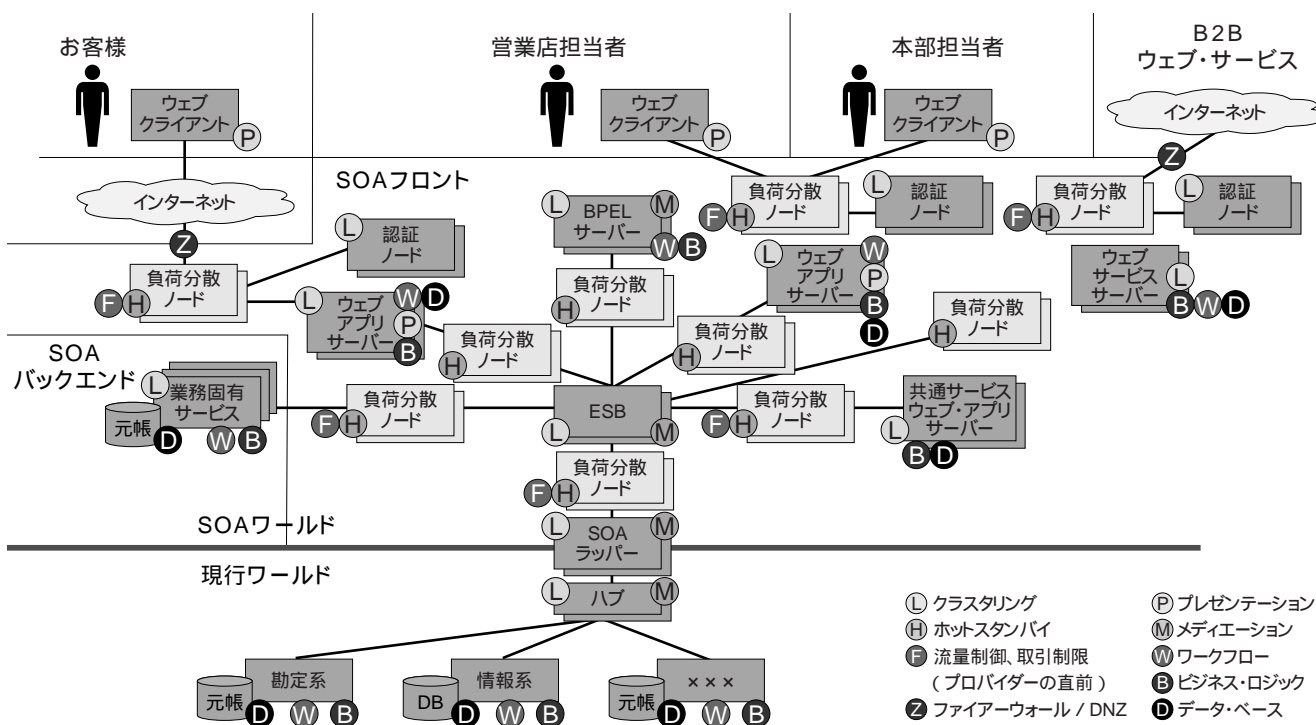
インフォメーション ラッパーは情報の流れがSOAワールドから現行ワールドの方向に偏った思想とした。これは、大量の情報とプロセスを抱えた現行ワ

ールドに極力手をいれず、可能な限り新しいISOA ワールド側で対処すべきと判断したためである。

IBM GS Methodに従って、SOA ラッパーの概念モデル(COM: Conceptual Operational Model)の考えを仕様モデル(SOM: Specified Operational Model)のリファレンス・アーキテクチャに一段詳細化したものを図5に示す。これは基本構成要素を仮定のロケーションに配置したモデルであり、丸記号で示すSOA RERアーキテクチャ設計で定義した9種類の業務機能、テクニカル機能とともに個別ソリューション・システム設計のテンプレートとすることができる。この後、物理モデル(POM: Physical Operational Model)を構築する際に、運用やコストを考慮し、物理的にサーバーなどの集約をはかることになる。

3.2 ファンクショナルETF

業務機能を定義する場合にリファレンスとすることができるのは、すでに長年構築維持されているIBMの銀行モデルであるIFW(The Information Framework)や保険モデルであるIAA(Insurance Application Architecture)、さらには昨年発表されたWSBF(WebSphere Business Service Fabric)の損害保険サービス機能、顧客マスター機能であるWCC(WebSphere Customer Center)などを参考にすることができる。また、B2B(Business to Business)領域のメッセージ・テンプレートは昨今充実しつつあるISO 20022[8]を活用することもできる。しかしSOA RERアーキテクチャでは2つのワールドでマスター・データ・ベースや処



1. 元帳が多数存在しうることに起因
 - 口座名寄せ、顧客属性情報の整合性
 - 巨大な物理名寄せDB、顧客マスターDBを構築するのではなく、サービスとして仮想名寄せDBを構築
 - 移管処理
 - 管理店は必要であるが、システムでは把握でき変更できるが、お客様には見えないように仮想マッピングする
 2. 新規サブシステムが複数存在することに起因
 - 金利、営業日などの照会、計算
 - 共通のコンポーネントまたは照会サービスを構築する
 - 還元資料の作成(業務ログやバッチ元帳との関係)
 - 業務ログや静態DBを現行の形に変換できれば現行処理にマージできる可能性はある
 - 総勘への反映
 - 総勘への入力インターフェース(勘定科目とフォーマット)は通常標準化されていることが多いため、加工して投入しやすい
 - 情報系への反映
 - 現行の入力業務ログに編集することができれば、新商品・サービスのコード追加で対応できる可能性がある
 3. 連携が複数システムにわたることに起因
 - 勘定や取引の精査、突合(計算照合)
 - それぞれのシステムでの処理件数、勘定がおたがいゴメイになっているかを検証する標準化が必要
 - 期日管理、現行センター一括処理、連携処理の未処理管理、再処理との関係
 - 大量でなければ、個別に記述管理しオンライン連携する未処理管理や、期日管理だけを共通サービス化できる可能性がある
 4. BPELを活用したヒューマン・ワークフローに起因
 - ワーキング・パターンの多様化により、権限の設定も多様化
 - BPELが参照するLDAPをエンタープライズ・レベルと部門やサブシステム・レベルとで疎結合化して管理
- “容易 > > > 困難”

図6. 金融機関におけるSOA RERアーキテクチャの業務問題

理が分離されていることによる課題もいくつか存在する。本設計の過程で洗い出した業務的な課題を図6に示し、これらの解決に向けて、現時点での検討による難易度を示した。この課題は今後アーキテクチャと業務の両視点で現実解を検討していく必要がある。

3.3 仮想サービス・リファクタリング手法

SOA RERアーキテクチャのETFを構築、成熟させるためのボトムアップ的な手法として、仮想サービス・リファクタリング手法[9] (VSRM: Virtual Services Refactoring Method) を本プロジェクトで新たに確立した。これはビジネスのあるべき論から開始するトップダウン手法とは異なり、既存システムの再利用に重点をおいた短期間でサービスが発見できる方法である。具体的にはMartin Fowlerのオブジェクト指向プログラムへのリファクタリング[10]をSOAアーキテクチャへのリファクタリングへ応用した考え方である。複数、できれば3種類以上の既存のフロント・エンド資産を分析し、フロント、バックを含むサービスの候補、粒度を検討したのち、仮想的にSOA化によるリファクタリングを机上で試みその再利用効果を予測する。図7に手順の概要を成果物の流れで示す。リファクタリングの前後の機能等価性はユースケースが維持されているか否かで判断する。

トップダウン手法に比較してVSRMは以下のメリットがある。
 ① 既存のアーキテクチャ設計が存在し、サービスの具体的なイメージがわかりやすい

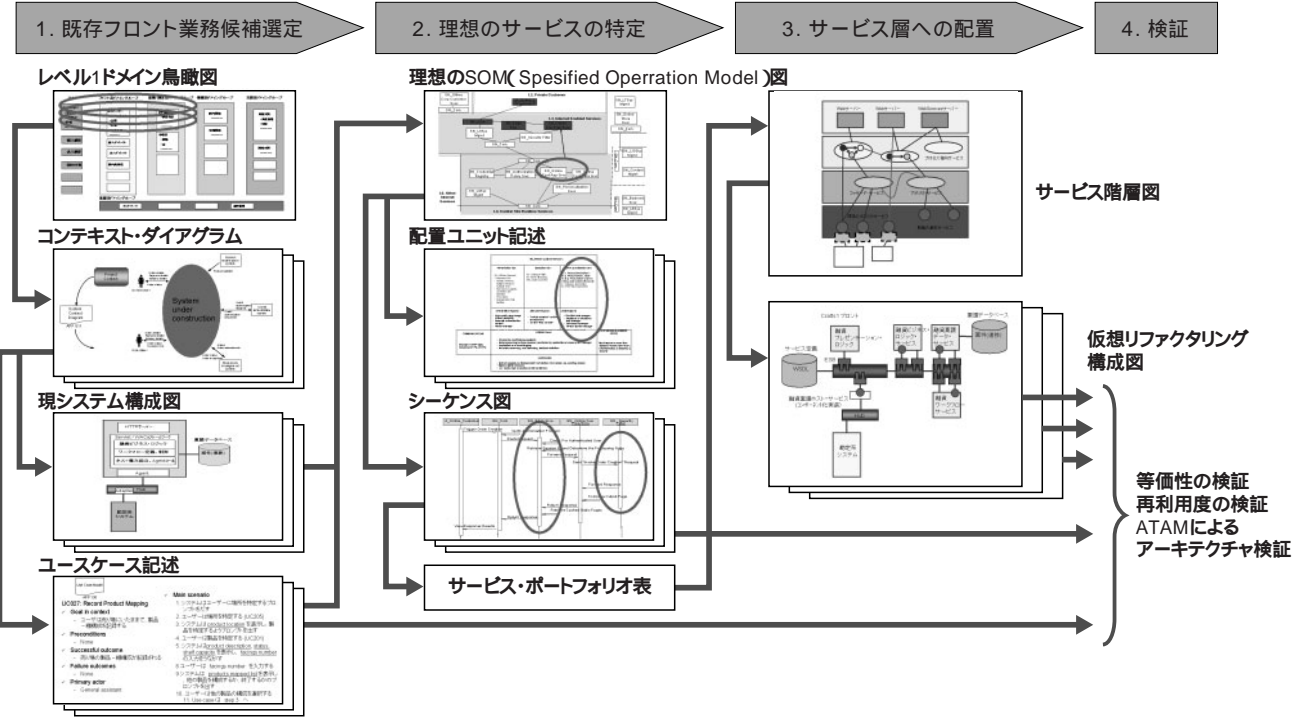


図7. 仮想サービス・リファクタリング手法(VSRM)の成果物の流れ

- ② 作業負荷の大きなトップダウン手順をふまず、サービスの理解や検討が容易
 - ③ IBM GS Methodの考え方により、アーキテクチャと実装を明確に分離でき、実装依存のないアーキテクチャを目指せる
 - ④ 具体的な効果(再利用の回数、確率)予測が可能
 - ⑤ 手法として具体的な手順(WBS: Work Breakdown Structure)、成果物イメージ(Work products)、技術的ガイド(Technique Paper)が準備されている
 - ⑥ 更改の迫る次の案件から即役に立つサービスが把握できる(小さく始める「パイロット」の検討が容易)
- VSRMは複数の実績を積みつつ、アーキテクチャ検証手法の一つであるATAM[Architecture Tradeoff Analysis Method] [11]によるアーキテクチャ評価と洗練を行いながら成熟を続けている。この手法によりすでに、ファンクショナルETFの業務サービスとして、口座開設、残高照会、取引明細照会、振込・振替、取引履歴照会、利用料明細照会、ポートフォリオ紹介、顧客属性照会変更、利率紹介変更、一部繰り上げ返済、条件変更、本人確認、ステートメント作成、還元資料、未処理管理、取引精査、勘定照合などが発見できている。また共通システム・サービスとしては、共通認証、セッション制御、取引制限、電子メール送付、統一エラーメッセージ、統一電文・コード変換、業務ログ収集、ftp機能、PDF文書作成機能、取引ログ出力などを洗い出すことができた。またオペレーショナルETFとして図5のSOA参照アーキテクチャを導出することができた。

4. おわりに

SOA RERアーキテクチャでは特に現行システムの並存と再利用による開発生産性に強く着目している。そのためEAで活用されるITの要素と経営価値の関係を表現するSCN(Strategic Capability Network) [6] (図8)をSOA視点で新たに描き、それらの関係を確認した。既存基幹系の再利用に元を發し、並行開発、コンポーネント再利用、仮想化が重要な視点となり企業価値全体に貢献していることを示している。

本論文では、金融機関を題材にETFを活用しSOA RERアーキテクチャの設計思想を論じた。これは日本の金融機関だけでなくグローバルでも活用が可能である。SOA事例としてすでに公開されている英国 The Standard Life Assurance Company [12]や米国 Charles Schwab [9]のSOAアーキテクチャもESBトポロジー表現だけでなくさらにSOAラッパー、SOAログ・ラッパー、業務機能やテクニカル機能の配置などを具体的に定義し発展させると、本アーキテクチャに近づくと考えられる。さらには金融機関にとどまらず、他のインダストリーで利用可能であろう。なぜなら、現行の基幹系が充実しているか否かにかかわらず、現行ワールドも新規のSOAワールドも平行して同時に進化、変更することが可能なシンプルかつ理解しやすいアーキテクチャだからである。

SOA RERアーキテクチャとそれを支えるVSRMはIBM Japan 金融ソリューションとして正式にプレス・リリースを行った [13]。IBM Yamato Lab.ではSOA

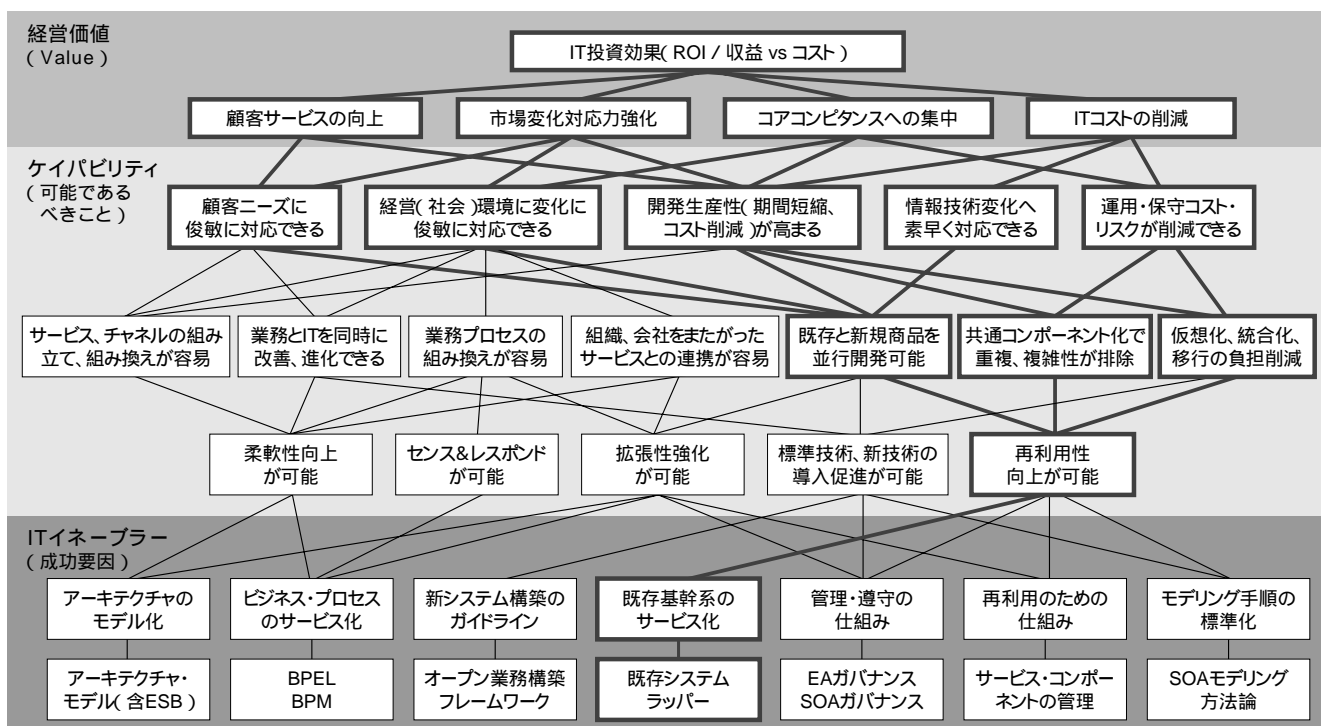


図8. SOA RERアーキテクチャのストラテジック・ケイパビリティ・ネットワーク(SCN)

RERアーキテクチャの実装環境を構築し実際に近い金融業務の一部を実機上で稼働させることに成功し、そのデモンストレーションも可能である。実施評価としては、既存DSE勘定系システムを連携活用し短期間に融資新商品をSOAワールド上に構築できることを実証できた。また、金融機関のお客様による実証プロジェクト(PoC: Proof of Concept)も開始した。

今後は現在IBM Yamato Labで蓄積されつつあるベスト・プラクティスを活用し、さらに価値ある企業レベルのアーキテクチャへ成熟させていきたい。そのためにはSOA RERアーキテクチャの強化に必要な十分なパフォーマンス、キャパシティ、セキュリティ、可用性、運用監視などの非機能要件のモデル詳細化、サービスのライフサイクルの管理、EA視点によるSOA効果試算や現実的なガバナンス・プロセス、にフォーカスする必要があると考えている。

謝辞

本論文の執筆に当たって、日本アイ・ビー・エムの社内コミュニティ Financial Technology Study(FTS) 2006における金融ソリューション岡田正雄さんを始めたメンバーの方々やIBM YamatoLab.ソフトウェア開発研究所の皆様との議論を参考にさせていただきました。あらためて深謝いたします。

参考文献

- [1] 特集 エンタープライズ・アーキテクチャ, 株式会社東京三菱銀行システムの全体最適を目指し、いち早くEAを導入, ProVISION, No.41 pp.20-27, <http://www.ibm.com/jp/provision/no41/index.html> (2007.6.20) .
- [2] インタビュー 住友信託銀行株式会社 SOAとEA, BPMの三位一体で、ビジネスのニーズに柔軟に 대응するシステム, ProVISION, No.51, pp.32-39,

- http://www.ibm.com/jp/provision/no51/pdf/51_interview3.pdf (2007.6.27) .
- [3] 山下真澄: “銀行システムにおけるSOAの考え方,” <http://www.ibm.com/jp/software/events/sodc/session/pdf/A-3.pdf> (2007.6.20) .
- [4] Dirk Krafzig 他2名: ENTERPRISE SOA, PRINTICE HALL, ISBN 0-13-146575-9(2004) .
- [5] IBM IBCS: サービス指向アーキテクチャ SOA投資の収益率測定のための実用ガイド, IBMブローチャー G510-6320-00(2006) .
- [6] IBMビジネスコンサルティングサービスIT戦略グループ: エンタープライズ・アーキテクチャ, 日経BP, ISBN4-8222-1873-2(2003) .
- [7] 内藤裕史: “IBM Global Service Method概要,” PROVISION, No.43, pp.38-41, http://www.ibm.com/jp/provision/no43/pdf/43_article1.pdf (2007.6.20) .
- [8] International Organization for Standardization: ISO 20022 Universal Financial Industry message scheme, <http://www.iso20022.org/> (2007.6.20) .
- [9] 山下真澄: “金融機関におけるSOA化に向けた考慮,” <http://www.ibm.com/jp/software/events/summit/pdf/a3.pdf> (2007.6.20) .
- [10] Martin Fowler, 児玉公信 訳: Refactoring, ピアソン・エデュケーション, ISBN 4-89471-228-8(2000) .
- [11] P.Clements 他2名: Evaluating Software Architecture, Addison-Wesley, ISBN 0321154959 (2002) .
- [12] Norbert Bieberstein他4名: Service-Oriented Architecture(SOA) Compass, IBM Press, ISBN 0-13-187002-5(2005) .
- [13] 日本IBM: “2006年12月6日SOAを活用した金融機関向け新ソリューション体系を発表,” <http://www.ibm.com/jp/press/20061206001.html> (2007.6.20) .



日本アイ・ビー・エム株式会社
 金融事業
 ディスティングイッシュト・エンジニア(技術理事)
 金融インダストリーCTO
 ITアーキテクト
山下 真澄 Masumi Yamashita

[プロフィール]

1978年、日本IBM入社。都市銀行や地方銀行のお客様を担当するSE、SE課長、プロジェクト・リーダーを長期に経験。現職は、金融機関のお客様に直接、アーキテクティングやエンタープライズ・アーキテクチャの構築支援を行うとともに、金事業部の技術戦略を担当。日経BP社「SOA大全」監訳。情報処理学会正会員。MASUMIY@jp.ibm.com