

## 企業独自のリファレンス・アーキテクチャー構築とその運用方法

加藤 洋\* 杉山 真一

## About the Reference Architecture construction in a company

Hiroshi Katoh\* Shinichi Sugiyama

近年、短期間で高品質のシステムを構築する要求が強くなっている。この解決策として、アセットの再利用があげられる。実績のあるアーキテクチャーを参照(リファレンス)することにより、短期間で品質の確保されたシステムの構築が可能となる。筆者は、アーキテクチャー・パターンの蓄積であるリファレンス・アーキテクチャー(RA)の構築が企業において必要であると考え、当論文は、お客様企業で蓄積していくRAのドキュメント体系とそのメンテナンス体系について、ITベンダーの視点とは異なった実装方法を記述し、お客様企業における独自のRA構築を提案する。

In these years, Customers require us to construct high quality systems in a short period. Reuse of asset is one of solutions to such requests. Making reference of the experienced architecture enables to construct high quality systems in a short period. I have judged that the Reference Architecture (RA) is needed for a company to store up architecture patterns. This paper proposes original RA construction in a customer, describing the different implementation method from the viewpoint of IT-Vender about the documentation and the maintenance of RA accumulated in the customer.

Key Words & Phrases : リファレンス・アーキテクチャー ,アーキテクチャー ,アーキテクチャー・パターン ,エンタープライズ・アーキテクチャー ,再利用  
Reference Architecture, RA, Architecture, Architecture Pattern, Enterprise Architecture, Reuse

## 1. はじめに

eビジネスの発展、浸透に伴い、短期間で高品質のシステム構築の要求が強くなっている。5年、10年前であれば、プロジェクト期間が2、3年のものが多かったが、今日では、半年～1年が主流である。同じ企業環境において、短期の小さなプロジェクトが多数並行して走るケースも少なくない。また、上流工程のデザインがドキュメント化されておらず、システム構成図だけが残っている場合もある。その結果、システムの拡張・変更時に、アーキテクチャーの検討過程がわからないために、変更作業に多くのワークロードが必要になったり、アーキテクチャーを変更できなかったりするケースもある。

近年では、アーキテクチャー間の高い整合性維持

が求められる。このため、ITシステムを長期的な観点、ビジネス的観点でITの全体が見渡せる企業におけるIT都市計画図であるEnterprise Architecture(EA)の必要性が高まってきている。EAとは、組織の目標とミッション、ビジネス、それを支える情報、ビジネス・アプリケーション・システムとテクノロジーを定義する業務・システム全体の設計図である[1][2]

こうした状況において、短期間で品質の確保されたシステムの構築を推進する方法として、上流工程のアーキテクチャー・デザインにおけるアセットの再利用が有効である。これは、実績のあるアーキテクチャー・パターンや、デザイン・パターンを蓄積し再利用することである。再利用を推進するための体系として、リファレンス・アーキテクチャー(以降RA)と呼ばれるものがある[3]

企業においても、RAによりアーキテクチャー・パターンの再利用をすべきである。実績のあるアーキテクチャー・パターンを蓄積し、再利用することが有効で

提出日：2003年08月29日

\* HIROSHIK@jp.ibm.com

ある。企業内でのアーキテクチャーの統一化も図れる。しかし、多くの企業において、RA構築にあたり、以下のような課題がある。

- (1) RAの蓄積に作業負荷をかけられない
- (2) 蓄積したRAが陳腐化して利用されなくなってしまう。

(1)についてであるが、再利用するために実績のあるアーキテクチャー・パターンを蓄積するには、そのために作業負荷がかかってしまうという課題がある。この課題を解決するために、企業内に蓄積されたシステム設計書とRAを組み合わせることにより、作業負担をかけずに企業独自のRAを蓄積できると考えた。

また、RAの記述レベルは、ITベンダーや文献[4]で提供される、抽象化されたアーキテクチャー・パターンのドキュメントより、具体的な内容にすべきである。その理由は、同一企業内では同一のシステム制約や要件下でのシステム構築となるため、ギャップ分析作業が簡素化されるからである。つまり、企業毎に独自性を持ったドキュメントを蓄積にすべきということである。

次に、課題の(2)についてである。蓄積したRAを新規プロジェクト案件で活用し、陳腐化を防ぐメンテナンス体系の仕組み作りの課題がある。これは、RAの蓄積/公開、活用、見直しという3つのサイクルでメンテナンスできる。

本論文では、企業独自のRAを構築、維持していく上で、必要なドキュメント体系とメンテナンス体系を記述する。

## 2. RA構築の必要性和RA構築時の課題

### 2.1 システム開発上の課題とRAの必要性

eビジネスの展開に伴い、アプリケーション単位、サービス・チャンネル単位にと、新たなシステムが次々に短期間で構築されている。また、システム開発には複数ベンダーの製品が使用されるケースも多くなっている。このようなIT環境の中においても、高いシステムの品質を確保する必要がある。その後の業務拡張、サービス・チャンネルの統合など、システム変更を意識することも重要である。これらを実現する際、下記のような問題にぶつかることがある。

- ・ 類似したシステムでありながら、異なるアーキテクチャーで構築されており整合性が取れていない。
- ・ 成功したプロジェクト資産の活用や再利用ができないため、無駄な投資が発生している。

上記の問題を解決するためには、システム構築の上流工程、アーキテクチャー・デザインにおけるアセットの再利用が有効である。過去のプロジェクトで実績のあるアーキテクチャー、デザインを蓄積し再利用すれ

ば、短期間に高品質のシステム構築が可能になる。これにより類似したシステム環境においても、アーキテクチャー・デザインが統一され将来の拡張性および統合性も確保できる。したがってRAを構築し運用することが、上記の問題を解決するために必要である。

### 2.2 RA構築の課題

実際にお客様企業においてRAを構築するためには、解決しなければならない大きな課題が2つある。

- ・ RAの蓄積に作業負荷をかけられない。

ほとんどのプロジェクトにおいてシステム設計書を作成しているが、アーキテクチャー・デザインとして再利用するには記述内容が詳細すぎる。このため、RAのために新たなドキュメントが必要になる。記述内容も含めて、なるべく簡易なものでないと多大な作業負担がかかってしまう。

- ・ 蓄積したRAが陳腐化するため利用されなくなる

蓄積されたRAは、IT技術の進歩に伴い陳腐化していく可能性がある。蓄積したRAを定期的に見直しするような運用ルールが必要である。

上記の課題を解決すべく、3章以降でお客様企業独自のRAを蓄積するドキュメントの体系、メンテナンスの体系について記述する。

## 3. 企業独自のRAに求められるもの

### 3.1 RAの適用手順

システム・デザインでは、システム要件(機能要件、非機能要件、システム制約)を定義し、論理モデル、物理モデルという手順で構築する。RAを使用するとその手順が短縮でき、RAのアーキテクチャー・モデル(論理モデル、物理モデル)とシステム要件、制約とのギャップ分析となる。これによりデザイン作業のワークロード削減、時間短縮が実現される。

### 3.2 企業独自のRA

企業独自のRAを構築する場合、どのようなものを蓄積するか、どのように運用していくかという観点について考える必要がある。

ITベンダーから提供されるRA、市販書のアーキテクチャー・パターンの記述は、どの企業、システムにも適用できるように汎化されているが、逆に企業独自のRAは汎化しすぎないほうが使いやすい。なぜなら、企業においては同一のシステム制約や要件下でのシステム構築であるため、ギャップ分析作業が簡素化されるからである。また、企業内のアーキテクチャーはできるだけ統一化を図りたいというニーズがある。この目的を達成するため企業独自のRAのドキュメント体系を構築することが求められる。また、構築され

た企業独自のRAをメンテナンスする仕組みを考えることが求められる。

(1) RAドキュメント体系

RAドキュメントは、負担をかけずにドキュメントを記述し、蓄積することができなければならない。筆者らが考える企業独自のRAに必要なドキュメントは下記の3つである。

- a. システム・コンテキスト  
( System Context )
  - ・ システム・サマリ
  - ・ システム要件サマリ
  - ・ 主要テクノロジー
- b. アーキテクチャー・ダイアグラム  
( Architecture Diagram )
  - ・ アーキテクチャー全体図
  - ・ 論理構成図
  - ・ 物理構成図
- c. アーキテクチャー・デシジョン  
( Architecture Decision )

最低限この3つのドキュメントがあれば、アーキテクチャーの選択が可能である。詳細な情報が必要な場合、RAの元となるシステムのプロジェクト設計書を参照すれば良いため、これ以外のドキュメントは不要と考えた。RAと既存設計書の重複を避け、メンテナンス負荷を軽減させることも可能となる。

記述方法、表記法は、企業内で利用する人がわかりやすい方法を採用すれば良い。システム設計などで使用している標準化、記述ツールで記述すれば良い。UML[ 5 ]で記述しても良い。

3つのドキュメントの記述レベルには、基本的に統一したものにすべきである。企業内での再利用性を考えると、アーキテクチャー・デザインを行う場合、システム要件、システム制約の記述は詳細化のレベルを揃えるべきである。そのため、RAとしては、アー

キテクチャーに関する記述に加えて、企業内の要件、制約に対応したアーキテクチャー・デシジョンなどを記述したもので、中でも事例的なものが有効である。特に、ネットワーク環境、使用端末環境、認証システム、運用管理システム、システム運用( 正常運用、障害運用 )などは、同じ要件であることが多い。アーキテクチャーを再利用することで省力化できるとともに、お客様内で統一の取れたアーキテクチャーとなり品質の確保ができる。

(2) RAメンテナンス体系

RAの運用は、蓄積したアーキテクチャー・パターンの陳腐化を防ぎ、再利用される仕組みにしなければならない。筆者らが考えるメンテナンス体系に求められるものを以下に記述する。

RAのメンテナンスを考える場合に重要な要件は、技術動向を踏まえてリフレッシュするサイクルとRAを維持する体制である。

RAメンテナンス体系の具体化には、以下の3つのプロセスから成るメンテナンス・サイクルが必要である。

- ・ RAの構築・公開・メンテナンス・プロセス
- ・ RAの活用プロセス
- ・ RAの見直しプロセス

次に、RAの維持体制であるが、企業内のシステムに横断的なガバナンスを効かせることのできる組織でメンテナンスする必要がある。具体的には、EA推進チームもしくは、共通基盤チームのような組織でメンテナンスしていくべきである。RAを管理するシステムとしては、文書管理システム、Lotus NotesのDBもしくは、イントラネットのWebページなどのシステム開発者が参照できる既存のシステムを利用すれば良い。

4. RAの構築と運用について

本章では、作業負担をかけず再利用しやすいRA

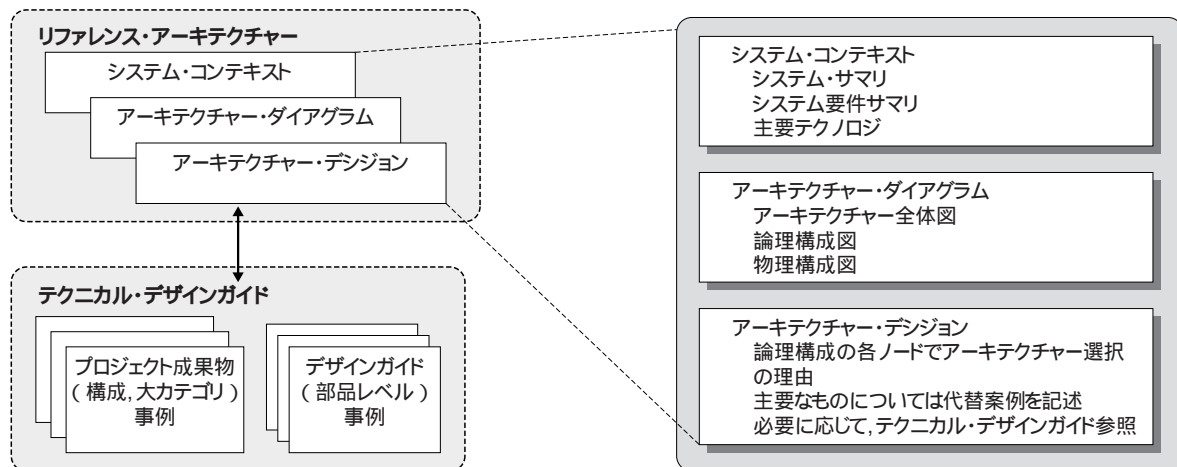


図1 .RAのドキュメント体系

表1 .サンプル:システム・コンテキスト(システム・サマリ)

NO.	カテゴリ	サマリ
1	インフラ	<ul style="list-style-type: none"> <li>・ 入力系中心のイントラネット系Webシステム (3層構造)</li> <li>・ 共通認証システムを使用し,他のイントラネット業務間はシングルサインオンを実現</li> </ul>
2	アプリケーション	<ul style="list-style-type: none"> <li>・ 融資業務(融資,格付,資産調査)の審査部分対象</li> <li>・ 個人ローン,外部の審査システムとの連携</li> </ul>

ドキュメント体系と陳腐化を防ぐメンテナンス体系的な内容を記述する。

#### 4.1 RAの構築

##### (1)RAドキュメント

RAは,3つのドキュメントを中心に構成される(図1参照)

##### a. システム・コンテキスト

システム・サマリ,システム要件サマリ,主要テクノロジーの3種類について記述する。

- ・ システム・サマリ  
インフラ面,アプリケーション面のシステム特性を記述する
- ・ システム要件サマリ  
主に非機能要件を中心に記述する(対象ユーザー,接続クライアント,サービスレベル,キャパシティ要件など)
- ・ 主要テクノロジー  
利用したハードウェア,ソフトウェア製品および接続プロトコルなどの主要製品を記述する(表1参照)

##### b. アーキテクチャー・ダイアグラム

アーキテクチャー全体図,論理構成図,物理構成図の3レベルでシステム概要を記述する。

- ・ アーキテクチャー全体図  
3層,2層構造などアーキテクチャー全体がわかるように記述する。
- ・ 論理構成図  
ノード構成,ノード内の主要機能(特にワークフロー,DBアクセス機能など再利用できる共通部品を記述)を記述する。
- ・ 物理構成図  
採用したハードウェア,ソフトウェア製品を記述する(図2参照)

##### c. アーキテクチャー・デザイン

ノード毎に,採用したアーキテクチャーのデザインのポイントを記述する。デザインポイントとは,採用したアーキテクチャー,採用理由,実装方法,代替案などである(表2参照)

##### (2)記述レベルについて

- ・ 記述に関しては抽象化せず,見本となる事例紹介という形式で情報を提供することが重要である。  
例:入力業務系Webシステム  
:参照業務系Webシステム
- ・ 詳細を知りたい場合は,既存のプロジェクト設計書やテクニカル・デザインガイドを参照させることとする。RAとの重複を避け,メンテナンス負荷を軽減させる。

##### (3)テクニカル・デザインガイドについて

テクニカル・デザインガイドは,再利用できる部品群,例えばサーバ構成からホスト通信,外字のように企業内で利用できる大きな項目から小さな項目まで集められたガイド。RAの補助として使用するとさらに効果的である。

#### 4.2 RAのメンテナンス・サイクルについて

蓄積されたRAは陳腐化していくことが懸念される。したがって蓄積されているRAそのもの(RAリスト),

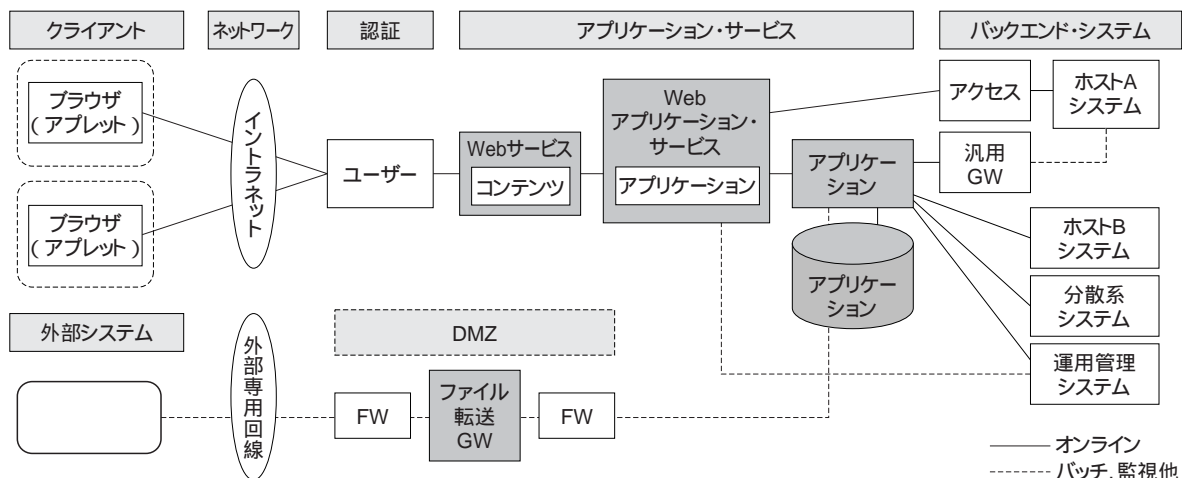
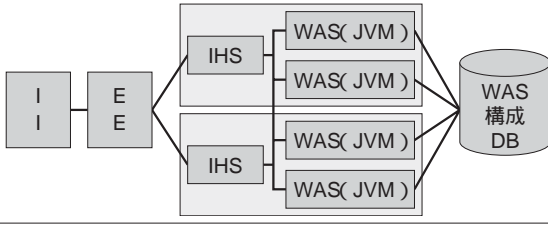


図2 .サンプル:アーキテクチャー・ダイアグラム(アーキテクチャー全体図)

表2 .サンプル:アーキテクチャー・デザイン(アプリケーション・サーバード)

NO.	カテゴリ	サマリ
1	エリア	IHS-WASの構成
2	アーキテクチャー・デザイン	(1)PD-IHS間は ,stateless構成 (2)IHS-WAS間は ,Affinity構成
	採用理由	(1)WAS障害時に融資業務アプリケーションのみの再起動 .State fullの場合 ,IE再起動での回復となり ,インパクト大のため (2)1WAS上に2JVMをたてるため ,Affinity構成必須 .初期ランザクシオン振り分けによる負荷分散
3	代替策	(1)State full構成 .WAS障害時にIE再起動が必要 .Windows系APサーバの場合 ,State full構成のみ可能 (2)Affinityなしの1:1構成 .セッション情報を引継ぐ業務には不向き
4	実装概要	
5	既存環境への影響	PD間のタイムアウト値

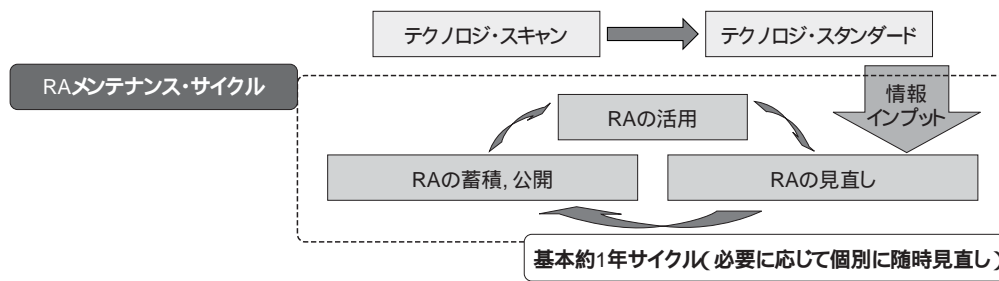


図3 .RAのメンテナンス体系

および個々の記述内容については ,定期的にメンテナンスしていくことが必要である ( 図3参照 )

(1)RAの蓄積 ,公開

- ・ 選択された見本的なアーキテクチャーをRAとしてドキュメント化し ,企業内に公開する .

(2)RAの活用

- ・ 企業内で新規システム構築時に活用する .
- ・ テクニカル・デザインガイドとともに活用し ,アーキテクチャー・デザインを実施する .

(3)RAの見直し

- ・ テクノジ・スキャン( 定期的な技術動向調査 )より見直されたスタンダードの情報に基づき ,RAリストの見直し( 追加 ,削除 )を行う .テクノジ・スキャンはEA活動の1つで技術動向を調査するものである .実際にはネットワーク ,セキュリティ ,基盤 ,言語などにカテゴリを分けて ,技術の陳腐化 ,先進技術を調査する .情報はIBMのようなITベンダーから提供されることもある .スキャン結果を元に ,企業内で使用する技術 ,製品の基準をテクノジ・スタンダードという .RAはエンタープライズ・アーキテクチャー活動と密接に関係し ,使用する技術につい

て定期的に動向にあわせてアップデートする必要がある .現在の技術の進歩 ,メンテナンス負荷を考慮すると年1回程度のサイクルが妥当である .

5 .適用と評価

筆者らは ,企業独自に構築し ,公開されたRAドキュメントをあるプロジェクトに活用し ,アーキテクチャー・デザインを実施した .適用したプロジェクトは ,Lotus Notesベースで構築された現行システムを ,Webシステムに再構築するプロジェクトである .システムの特性よりRAとして ,イントラネット系Webシステム( 入力系業務 )を参照した .

5.1 アーキテクチャー論理モデル作成

RAのドキュメントであるアーキテクチャー・ダイアグラム( 特にアーキテクチャー全体図 ,論理構成図 )を中心に参照し ,プロジェクトの要件とのギャップ分析作業を実施した .

ギャップ分析作業の結果 ,イントラネット系WebシステムのRAより不必要なノードを削除する形で作業で

きたため、短時間にアーキテクチャー論理モデルの作成ができた。新システムのアーキテクチャーの適用にあたりノード毎のアーキテクチャー・デザインを参照し選択した。当ケースでは、ワークフロー(アクセス権限)機能、ファイルアクセス機能、帳票作成機能、DBコネクション機能などはそのまま再利用することとした。詳細な内容が必要な機能のみ、RAの元プロジェクトの設計書を参照すればアーキテクチャー適用有無の判断ができた。結果として、効率的に作業できた。

## 5.2 アーキテクチャー物理モデル作成

同様にRAのアーキテクチャー・ダイアグラム(物理構成図)を参照し、ギャップ分析作業を実施した。アプリケーション・サーバ、DBサーバのノード構成、システム間連携などは、製品選択、サーバ構成形態など、実績のあるものが選択でき高い品質が確保できた。

## 6. 結論

RAを適用することにより、短期間で高品質のアーキテクチャーをデザインできた。

特に、ギャップ分析作業において企業特有のアーキテクチャー・デザインが記述されていることが非常に有効であった。実際の開発プロジェクトのアーキテクチャー・デザインに、RAを活用したことで、期間

の短縮および高い品質の確保という観点での効果について、十分な検証ができた。また、企業独自のRAを構築することが重要であることを検証できた。しかしメンテナンス体系については、RAの構築、公開と活用については検証できたが、RAの見直しのサイクルについては検証できておらず、今後も継続して検証していく必要がある。

企業独自で構築したRAの陳腐化を防ぎ、企業内で活用するには、メンテナンス・サイクルや推進体制などについて、お客様とさらに検討を進めて成熟させていく必要がある。

## 参考文献

- [1] 経済産業省( ITアソシエイツ協議会 ),  
<http://www.meti.go.jp/feedack> , '03.06.06
- [2] John A.Zachman, *A Framework for Information Systems Architecture*, the *IBM Systems Journal*, Vol.26, No3, '87
- [3] *IBM Systems Journal* Vol.38, No.1, 1999  
<http://www.research.ibm.com/journal/sj38-1.html>
- [4] F.Bushmann/R.Meunier/H.Rohnert/P.Sommerlad/M.Stal ソフトウェアアーキテクチャ, 近代科学社, ISBN4-7649-0283-4, 2000年
- [5] Object Management Group( UML Resource Page ),  
<http://www.omg.org/uml/>



日本アイ・ピー・エム株式会社  
金融システム事業部  
金融クライアントIT推進  
ITアーキテクト

加藤 洋 Hiroshi Katoh

### [プロフィール]

1990年日本アイ・ピー・エムに入社。銀行担当のSEとして、分散システムを中心に担当。東京三菱銀行様の勘定システムの営業店サーバを設計・構築した新営業店プロジェクト、融資稟議/個人ローンのBPRプロジェクト(イントラネットWeb)など大規模なプロジェクトに参画。現在は、東京三菱銀行様担当のCITA(クライアントITアーキテクト)として、お客様のエンタープライズ・アーキテクチャー構築のサポートを行っている。



日本アイ・ピー・エム株式会社  
金融システム事業部  
金融クライアントIT推進  
主任ITアーキテクト

杉山 真一 Shinichi Sugiyama

### [プロフィール]

1991年日本アイ・ピー・エムに入社。金融のお客様担当SEとして東京三菱銀行様を担当。主に東京三菱銀行様の勘定システムを核とした、他システムとの連携を中心に各種プロジェクトに携わる。メッセージ・ブローカーを構築した新営業店プロジェクト、融資稟議/個人ローンのBPRプロジェクト(イントラネットWeb)など大規模なプロジェクトに参画。現在、東京三菱銀行様担当のCITA(クライアントITアーキテクト)として、お客様のエンタープライズ・アーキテクチャー構築のサポートを行っている。