

# グラスボックス化により地方自治体での 利用者主導の開発を実現するAIST包括フレームワーク

林浩一<sup>†</sup> 和泉憲明<sup>‡</sup>  
Koichi Hayashi<sup>†</sup> Noriaki Izumi<sup>‡</sup>

<sup>†</sup>ピースミール・テクノロジー株式会社  
<sup>‡</sup>産業技術総合研究所 社会知能技術研究ラボ

## 要旨

地方自治体のシステム開発は、アーキテクチャ・仕様ともに開発業者に強く依存していることが多く、コスト増加や利便性の低下の原因にもなってきた。AIST 包括フレームワークは、利用者主導でのシステム開発を可能にするために、アーキテクチャと仕様の透明性を高めるためのソフトウェアとノウハウを体系化したものである。産業技術総合研究所（産総研）での開発を通じて構築された AIST 包括フレームワークは、現在複数の地方自治体のシステム開発に適用され進化を続けている。

## 1. はじめに

業務に最適化された基幹システムを導入して業務を効率化することは、民間、公共を問わずいずれの事業体においても重要な課題のひとつである。しかし、システム開発に失敗し、リリースの大幅遅延や予算超過、重要な機能の欠落などが発生してしまう例が後を絶たない。この問題の主要因は発注者の開発に関する統制能力、いわゆるガバナンスの不足にある。一般に利用者である業務担当者はシステムの知識はあまりなく、逆に開発業者の担当者は業務の知識が十分ではない。そのため業務システムの要件を正しく合意しながら進めることは本質的に難しい作業であるが、かつて両者を媒介し開発を統制してきた情報システム部門が、システムの複雑化によりその役割を担えなくなったことが大きい。

発注側の統制力が弱くなり、開発業者への依存性が高くなると、アーキテクチャや仕様についての方針確定の主導的な役割が開発業者の側に移ることが起きる。このとき業者固有の製品や技術を導入してしまうと、以降の発注ではその業者が有利になる。こうしたいわゆるベンダーロックインの構造ができると競争原理が働かなくなり、追加開発に際しての費用が不透明で、割高なものになる。この構図は様々な事業体で見られるが、公共機関、中でも地方自治体において特に顕著である。

本稿では、システム開発を開発業者に依存しすぎる状況を改め、利用者・発注者が本来の役割を果たし主導的に開発を進められるようにするために作られたAIST包括フレームワークについて述べる(図1)。

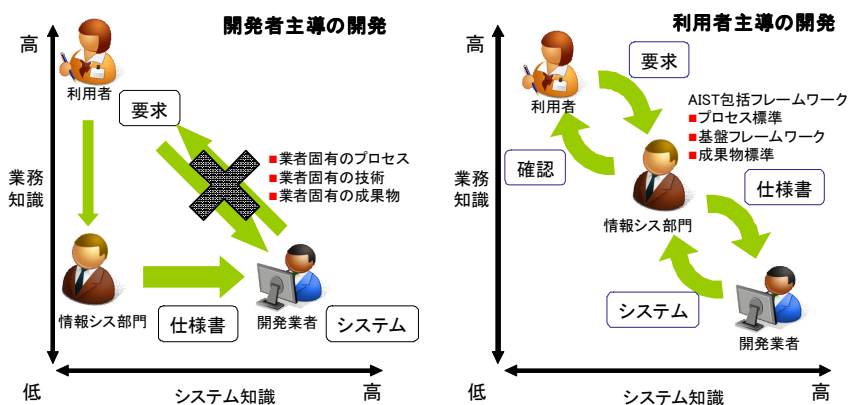


図1 利用者主導の開発

## 2. 関連する取り組み

公共機関や地方自治体のシステム調達および開発を効率的なものにするための取り組みで、特に重要なものとしては、電子政府を目指した「業務・システム最適化」と呼ばれる標準化の活動が挙げられる。この一環として、たとえば、経済産業省ではEA (Enterprise Architecture) の観点から開発の進め方や成果物を規定し[1]、総務省でも自治体EAのガイドラインを定めている[2]。しかし、これらガイドラインが定める成果物は、従来ホストコンピュータでの開発で使われてきたDFD (Data Flow Diagram) などの図表が中心であり、必ずしも利用者にとって理解しやすいものではない。また、具体的なアーキテクチャが示されていないので、業者による固有の技術の導入を回避する手段とはならない。

AIST 包括フレームワークは、これら EA ガイドラインを具体化し、実働可能とすることを目的とし、利用者の開発への積極的な関わりを前提とするユースケース駆動[3]の考え方を取り入れることで、利用者主導のシステム開発が可能になるように成果物の拡張・整備を行ったものである。また、オープンソースを中心とした具体的なアーキテクチャを定めることで開発業者への依存性の最小化を図っている。

### 3. AIST 包括フレームワークのねらい

#### 3.1. 利用者主導の開発とガラスボックス化

AIST 包括フレームワークは、利用者主導のシステム開発を実現するためのソフトウェアとノウハウを体系化したものである。AIST とは産総研の英文名称 (Advanced Industrial Science and Technology) の頭文字であり、産総研の基幹業務システムの構築を通じて概念化されたことにちなんでいる。「包括」フレームワークとは、ソフトウェアだけでなくそれを活用するためのノウハウまで包括的に規定していることを表している。AIST 包括フレームワークにより、利用者・発注者は開発業者の見積りを鵜呑みにすることなく自ら開発規模を把握し、コンサルタントに頼ることなく発注仕様書を作成し、開発側のプロジェクトマネージャに丸投げすることなく開発管理に主体的に関わることを可能にする。

AIST 包括フレームワークは、ガラスボックス化という独自のコンセプトに基づいて策定されたもので、プロセス標準・基盤フレームワーク・成果物標準により構成される。ガラスボックス化とは、利用者には「専門知識がなくても使える」というブラックボックスの利点を提供すると同時に、開発者には「必要な設計情報がすべて開示される」というホワイトボックスの特徴を要求するという考え方である。このガラスボックス化を実現するために、AIST 包括フレームワークは以下の指針で設計されている。

- **業務改革なき業務改善・効率化**：現行業務の明確化をシステム見直しの起点とする。民間企業の場合、コンサルタントによって提言された抜本的な業務改革や横断的な組織改組が起点となることもあるが、法令などの制約の多い自治体業務では、対象業務の漏れ抜けの発生リスクが高い。このため、現行の業務を漏れなく正確に捉え、利用者自身が改善を積み重ねて全体の効率化を図るほうが現実的である。このアプローチでも、全体業務を俯瞰することで、共通化による開発コスト削減やその削減されたコストを原資とするシステム化範囲の拡大といった効果的な改善が可能である。
- **ユースケース単位での開発管理**：開発の管理を利用者の業務視点を表すユースケース単位で行う。業務視点の単位で成果物を管理することで、利用者の管理負荷を軽減できる。開発業者によっては各成果物の品質責任を明確化するために、種類ごとに再委託先に作業依頼することも多く、利用者側が複数種類の成果物を業務視点で捉え直す負担を強いられていた。ユースケースは受け入れテストの単位にもなるので、この点でも発注側の負荷を軽減できる。
- **標準技術としてのオープンソース活用**：開発業者への依存性の解消のためにオープンソースを利用する。オープンソースの活用メリットとして、ライセンスフリーであることによるコスト削減効果が強調されることがあるが、ここでは、仕様が公開されていることによる標準技術としての位置づけを重要視している。具体的には、オープンソースのライブラリとの接続を可能とするようなスキルや技術文書の理解力が、開発業者に要求する技術レベルとして設定できることが重要である。

#### 3.2. 地方自治体への適用の意義

AIST 包括フレームワークによる利用者主導の開発は、多数の部門やステークホルダーが関わる大規模な基幹システムで、個別の業務システムごとに分けて構築する場合に特に有用である。民間の大企業をはじめ様々な種類の事業体を対象にできるが、個別システムごとの分割発注を前提としている地方自治体が現在の重点対象である。開発のあり方を転換することにより以下の観点での革新を目指している。

- **システム資源の一元化**：異なる業務システム間で、ハードウェアやミドルウェア、データベース、OSなどを一元化することで、情報システム部門の管理負荷と導入・運用コストを軽減する。
- **競争市場の形成**：利用者による仕様の正確な把握と、Java を中心とするオープンソースをはじめ LDAP、ポートレットといった仕様が公開された標準技術をベースとするアーキテクチャの採用により、特定業者への強い依存性を解消し、技術競争による適正な市場を形成する。

- **地場 IT 産業の振興**：特定業者への依存性の解消をさらに進め、首都圏の大手業者に一括発注していたシステム開発を地場や中小の開発業者に分割発注することで、公共事業としての意義を強化する。
- **共同利用の促進**：業務要件成果物を自治体間で流通させ、業務の標準化を進めることによって、システムの共同利用を促進し、長期的にはクラウドサービスとしての提供を可能にする。

## 4. AIST 包括フレームワークの構成

AIST 包括フレームワークの主な構成要素は、ガラスボックス化を可能にするプロセス標準と基盤フレームワーク、成果物標準である。基盤フレームワークによりアーキテクチャを共通化し、その上で業務視点を起点とした開発を行うためのプロセス標準と成果物標準を定めている。以下に示す全体構成はフルスクラッチ開発に対応する場合のものだが、選択的に採用することによってパッケージ活用、ホストマイグレーションの開発形態にも対応できる。

### 4.1. プロセス標準

プロセス標準では、要件分析、基本設計、開発、保守運用の各プロセスを定める。全工程を通じて業務視点からの成果物によるトレーサビリティを確立できることが重要な特徴である。

- **要件分析プロセス**：現行 (AsIs) の業務とシステムを精査することにより、次期 (ToBe) の業務要件とシステム要件をとりまとめる。帳票と決裁の流れを中心に記述した業務フロー図を採用することで、業務全体を正確に理解でき、システム化範囲を利用者主導で決定することが可能になる。
- **基本設計プロセス**：「要件分析プロセス」の成果物に従い、システムの外部設計を行う。業務フローのうちシステムとユーザの界面となる部分について、システムの挙動をユースケースシナリオとして書き下していく。自然言語での記述であるため利用者に理解できる。さらに、ユースケース単位で整理された画面・帳票・業務ルールを利用者が管理する成果物とする。開発業者主体での外部設計では、想定したサブシステム相互の外部インターフェースの仕様と機能概要しか明確にできないことが多いが、ユースケース単位での成果物管理により利用者自らが機能要件を具現化できる。
- **開発プロセス**：「基本設計プロセス」の成果物に従って、フィーチャーを設計し、プログラミングとテストを行う。フィーチャーとは FDD (Feature Driven Development) と呼ばれる開発手法[4]に基づく考え方で、ユーザにとって意味のあるプログラム単位のことである。ユースケースの単位をプログラム単位であるフィーチャーに対応づけることで、トレーサビリティが確立し、ユースケース単位での受け入れテストやデータ移行などが行えるようになる。
- **保守運用プロセス**：リリースしたシステムに対して保守・運用を行う。全工程を通じて業務視点からの成果物のトレーサビリティが維持されているため、制度改正への対応やシステム化範囲の拡張についても、ユースケースを単位として利用者主導で実施することが可能になる。

### 4.2. 基盤フレームワーク

基盤フレームワークは、各業務システムの基盤となるアーキテクチャを定めるものである。主要な構成要素としてオープンソースのライブラリと仕様が開示された技術を採用しているところに特徴がある。

プログラム言語と動作プラットフォームは、Java を採用している。OS や AP サーバは Java が動作するものであれば良いが、Linux と JBoss というオープンソースの組み合わせを標準のモデルとする。ソフトウェアフレームワークとしては、Web アプリケーションフレームワークとして Struts, OR マッピングツールとして Hibernate と iBatis, DI コンテナとして Spring を採用している。いずれも広く利用されてい

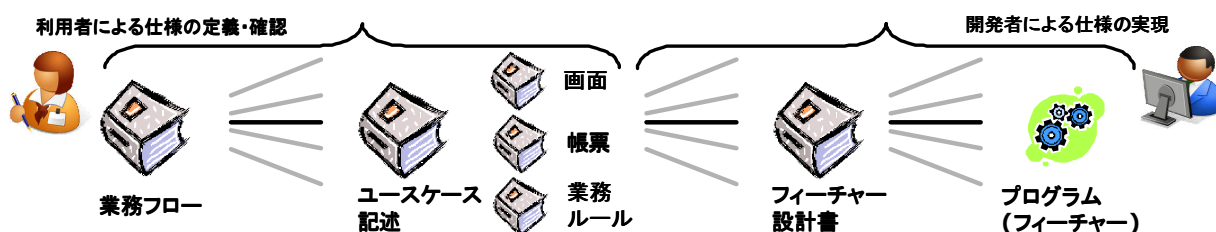


図2 成果物のトレーサビリティ

るオープンソースのライブラリであり、対応できる開発業者は多い。

ユーザ管理の基盤としては、標準的なディレクトリサービスのプロトコルである LDAP を採用している。これにより特定の製品に依存することなくユーザ管理の一元化を行うことができる。また、Web 画面上に複数の機能を集約したポータルを作成するために、ポータルのコンテンツを Java で作成する標準的な仕様であるポートレットを採用している。こちらも仕様が公開されており、採用している製品も多いため特定の開発業者に依存する必要がなくなる。

### 4.3. 成果物標準

成果物標準とは、品質を維持する為に、成果物を作成する際に守らなければならないルールのことである。主要なものとしては、コーディング規約（ソースコードの記述ルール）、命名規約（ソースコードや設定ファイルで用いられる名称の命名ルール）、メッセージ・ログ規約（システム利用者向けに出力するメッセージやシステム運用者向けに出力するログの出力方法の定義）、UI 設計ポリシー（画面や帳票等の表示ルール）などを定めている。これ以外にも、各プロセスの成果物ドキュメントのテンプレートを提供し、品質の向上とレビューの効率を高める。

## 5. 適用事例

これまでに表 1 に示すシステム構築プロジェクトに AIST 包括フレームワークが採用されている。いずれの事例においても、利用者自らが業務要件の定義に深く関与することで、既存の業者に限定されることなく調達による業者選定を行い、利用者主導でのプロジェクト管理が進められている。

表 1 AIST 包括フレームワークの採用状況

事業体	プロジェクト（期間）	開発状況
産総研	次期情報システム (2005 年～2010 年)	AIST 包括フレームワーク上に財務会計、人事給与などの一般管理業務に加え、研究員や研究テーマなど特有の管理業務のためのシステムが構築された。2010 年からシステムの稼働を開始している。これによりアーキテクチャの一元化が行われ運用コストの削減が達成された。
横浜市	福祉関連システム (2006 年～)	福祉 5 法システムをはじめ、複数の福祉系システムの開発に適用されている。2012 年に最初のシステムが稼働する予定である。
新潟県	基幹システムオープン化 (2007 年～)	財務会計、給与、税のシステムの構築に適用されている。2010 年には財務会計システムが稼働している。
札幌市	基幹システムオープン化 (2010 年～)	住民記録、国民保険、税など多数のシステムの構築に適用を進めている。2012 年から順次稼働が開始する予定である。このプロジェクトでは特に地場業者の参入機会を増やすことを目標にしている。

## 6. まとめ

AIST 包括フレームワークのねらいとして、利用者主導の開発とそれを実現するガラスボックス化の考え方を示した。また、その概略の構成を説明し、採用している開発事例を紹介した。今後の展開としては、これまでの横浜市、札幌市といった規模の大きな自治体での経験を元に、より規模の小さな自治体へ広く適用する中で、地方自治体のシステム開発や制度改正対応などの運用・保守について社会的な運用スキームを確立していく。併せて、サービス提供形態の汎用化を進め、中央省庁、独立行政法人、民間企業など同じ課題を持つ様々な事業体への適用を進めていく予定である。

### 参考文献

- [1] IT アソシエイト評議会, “業務・システム最適化計画について (Ver.1.1) ~Enterprise Architecture 策定ガイドライン~, [http://www.meti.go.jp/policy/it\\_policy/ea/data/report/r2/r2.pdf](http://www.meti.go.jp/policy/it_policy/ea/data/report/r2/r2.pdf), 2003.
- [2] 総務省, “自治体 EA 業務・システム刷新化の手引き”, <http://www.soumu.go.jp/denshijiti/eatebiki/index.html>.
- [3] アリスター・コーバーン, ユースケース実践ガイド, 東洋経済新報社, 1996.
- [4] スティーブ・バルマー 他, アジャイル開発手法 FDD, ピアソンエデュケーション, 2003.