

次世代ソフトウェアアーキテクチャへの展望

青山 幹雄（新潟工科大学），小笠原 秀人（東芝），金谷 延幸（富士通研究所），中所 武司（明治大学）
佃 軍治（日立），中島 震（NEC），中谷 多哉子（Slagoon），深澤 良彰（早稲田大学）

1. はじめに

E-ビジネスやE-サービスの広がりによって、企業や社会がインターネットやモバイルコンピューティングを媒介とする新しい構造へ転換を迫られている[2]。この新たな構造を実現するためには、従来のコンピューティングを中心としたアーキテクチャやアプリケーション毎に異なるアーキテクチャによる個別開発技術では限界がある。ネットワーク上で動的に進化するビジネスやサービス全体をモデル化し、実現する方法が求められている。本稿では、このような構造を形作るアーキテクチャを次世代ソフトウェアアーキテクチャと呼ぶ。

さらに、ネットワーク上では、ソフトウェアを開発・所有することなく、そのサービスを利用できる。図-1に示すように、ネットワーク化はソフトウェアの開発・利用形態にも変化をもたらした。

本稿では、このような観点から次世代ソフトウェアアーキテクチャへの展望を述べる。

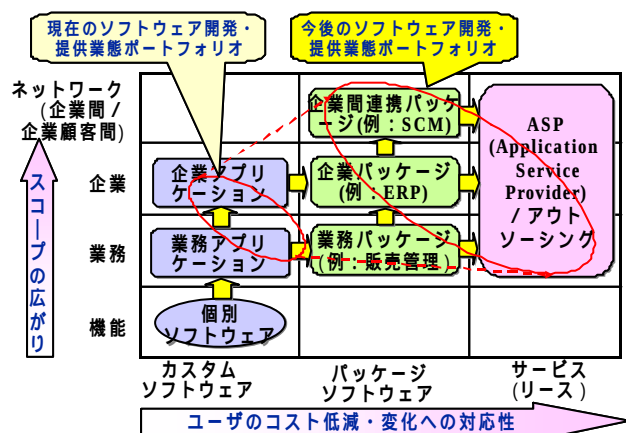


図-1 ソフトウェア開発・利用ポートフォリオの変化

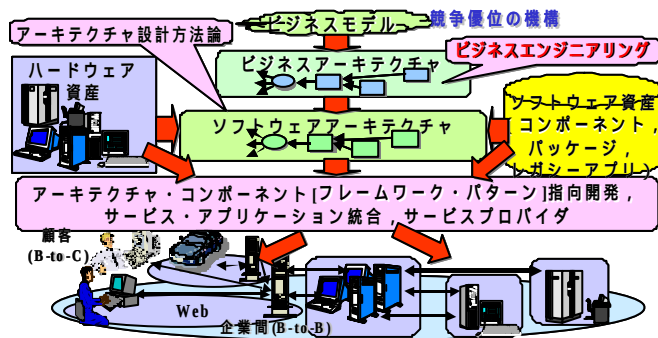


図-2 次世代アーキテクチャ技術の枠組み

2. 次世代ソフトウェアアーキテクチャへのアプローチ

図-2はビジネスモデルを出発点として、ソフトウェアによるビジネスやサービスの支援にいたるプロセスを示す。従来のソフトウェアアーキテクチャは、ソフトウェアの設計段階から出発していた。しかし、E-ビジネスやE-サービスでは、企業間あるいは企業と顧客にいたるビジネス全体の最適化が競争力の鍵であることから、次世代ソフトウェアアーキテクチャでは、ビジネス構造などのより高水準なアーキテクチャを重視すべきである。これを高水準アーキテクチャと呼ぶ。

ここで、ビジネスのモデル化、設計、評価技術の体系をビジネスエンジニアリングと呼ぼう。したがって、図-2のプロセスは、ビジネスエンジニアリングとソフトウェアエンジニアリングからなる。

- 1) ビジネスエンジニアリング：ビジネスアーキテクチャとその分析方法論、ネットワークベースのソフトウェアエンジニアリングへの展開
- 2) ソフトウェアエンジニアリング：ソフトウェアアーキテクチャとその実現技術

3. ビジネスエンジニアリングとアーキテクチャ

E-ビジネス全体のプロセス、リソース組織などの構造モデルをビジネスアーキテクチャと呼ぶ。ビジネスアーキテクチャは、次のようなビジネスに関する要素とその関係の構造をモデル化する。

- 1) ビジネス環境：状況、目的、要求、成果
- 2) ビジネス境界：責務、職務
- 3) ビジネス主体：資源、場所、機能、実行

ビジネスアーキテクチャの分析・設計は難しい。また、ビジネスアーキテクチャの決定はビジネスの成否を決めるほどの戦略的決定事項である。そのため、優れたビジネスアーキテクチャの基本的な構造をパターン化するE-ビジネスパターンなどの再利用をベースとする方法論が提案されている。

4. ビジネスアーキテクチャの実現

- (1) ビジネスアーキテクチャからソフトウェアアーキテクチャへ

ビジネスアーキテクチャをソフトウェアアーキテクチャへ展開するためには、要求条件、実現の制約条件などに応じたソフトウェアアーキテクチャを設計あるいは選択する。規模、スケーラビリティ、

ネットワーク拡張性、モビリティ、進化特性などビジネスとソフトウェアに対する多様な非機能的特性が考慮されなければならない。

ソフトウェアアーキテクチャには、サービスブローカ(メディエータ)、エージェントなどのネットワーク指向アーキテクチャ、あるいは既存ソフトウェアアーキテクチャパターンの組合がある。

さらに、EAI (Enterprise Application Integration)などのネットワーク上のアーキテクチャ統合機構、ASPなどのサービスプロバイダ、分散オブジェクト環境などのミドルウェア技術への展開も必要である。

(2)ソフトウェアアーキテクチャの設計

ソフトウェアアーキテクチャは性能や信頼性などのビジネスやシステム全体の非機能的特性を決定づけることから、非機能的特性を考慮したソフトウェアアーキテクチャ設計方法論が提案されるようになった。

ソフトウェアアーキテクチャ設計方法論は、次の2つのアプローチに分類できる。

(a)分析的設計アプローチ

1) アーキテクチャ&コンポーネント指向開発

ビジネスプロセスの分析から始まりアーキテクチャを中心としてコンポーネントへ展開する分析的アプローチ。例えば、Catalysis[3]がある。

2) 非機能的特性

非機能的特性をアスペクト(aspect)とも呼ぶ。アスペクトに着目したプログラミング方法論としてアスペクト指向(Aspect-Oriented)プログラミングがある。また、非機能的特性を考慮したアーキテクチャ設計方法論に ABAS (Attribute-Based Architecture Styles)がある。

(b)構成的アプローチ

ソフトウェアの分析、設計、プログラミングなどで繰返し出現する典型的な解をソフトウェアパターンと呼ぶ。ソフトウェア開発の各段階でパターンを組み合わせて構築する方法が試みられている。ESS (Enterprise Solution Structure)などのビジネスパターン、アナリシスパターン、デザインパターンなどが提案されている。

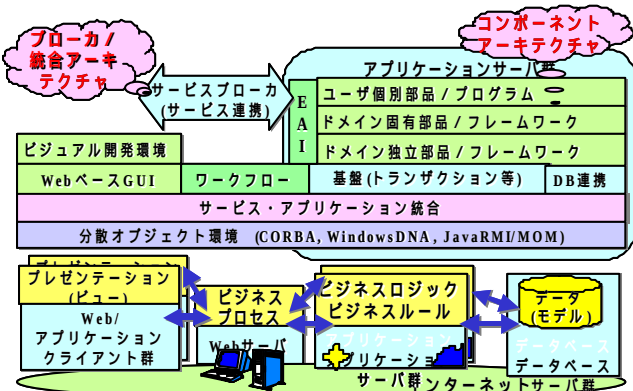


図-3 アーキテクチャの実現

5. ソフトウェアアーキテクチャの実現

フレームワークはソフトウェアアーキテクチャを実現したものである。複雑なソフトウェアアーキテクチャを実現するために、図-3に示すような、階層化されたアプリケーションフレームワークシステムが提案されている。また、フレームワークの実現には各種パターンも利用されている。

フレームワークは、それ自体が制御機構をもつので、従来の開発方法とは異なるアプローチが必要である。さらに、フレームワークを中心にしてコンポーネントを組み合わせてアプリケーションを構築する方法が開発されている。

また、携帯端末などの組み込みソフトウェアのアーキテクチャもネットワークのエンドポイントとして重要である。特に、組み込みソフトウェア開発では開発期間が競争力の鍵となるので、プロダクトラインアーキテクチャの再利用などのアプローチが期待できる。

6. まとめと提言

E-ビジネスの進展に伴い、高水準ソフトウェアアーキテクチャ技術は今後のソフトウェア開発・提供・利用の戦略的技術である。その核技術として次の技術の研究・開発を推進すべきである。さらに、このような技術を実践で活かすためにソフトウェアアーキテクトの育成を図るべきである。

- 1) 高水準アーキテクチャのモデリング技術：ビジネスアーキテクチャ分析・設計方法論とソフトウェアアーキテクチャへの展開
- 2) ソフトウェアアーキテクチャ実現技術とそれに基づく E-ビジネス実現技術：フレームワーク、パターン、コンポーネントと統合技術
- 3) アーキテクチャの動的進化機構とその支援[1]

なお、本調査研究の内容は Web ページ (<http://ies045.iee.niit.ac.jp/NISE/>)を参照下さい。謝辞：本調査研究にアドバイスを頂いている東京大学の玉井哲雄教授に感謝します。また、議論頂いた上原三八(富士通研究所)、佐伯元司(東京工業大学)の両氏に感謝します。

参考文献

- [1] 青山幹雄ほか、次世代コンポーネントウェアと今後のソフトウェア研究開発戦略、SSR '99, Feb. 1999, <http://www.iisf.or.jp>.
- [2] D. Amor, *The E-business (R)Evolution*, Prentice Hall PTR, 2000.
- [3] D. F. DSouza and A. C. Wills, *Object. Components, and Frameworks: The Catalysis Approach*, Addison Wesley, 1998.