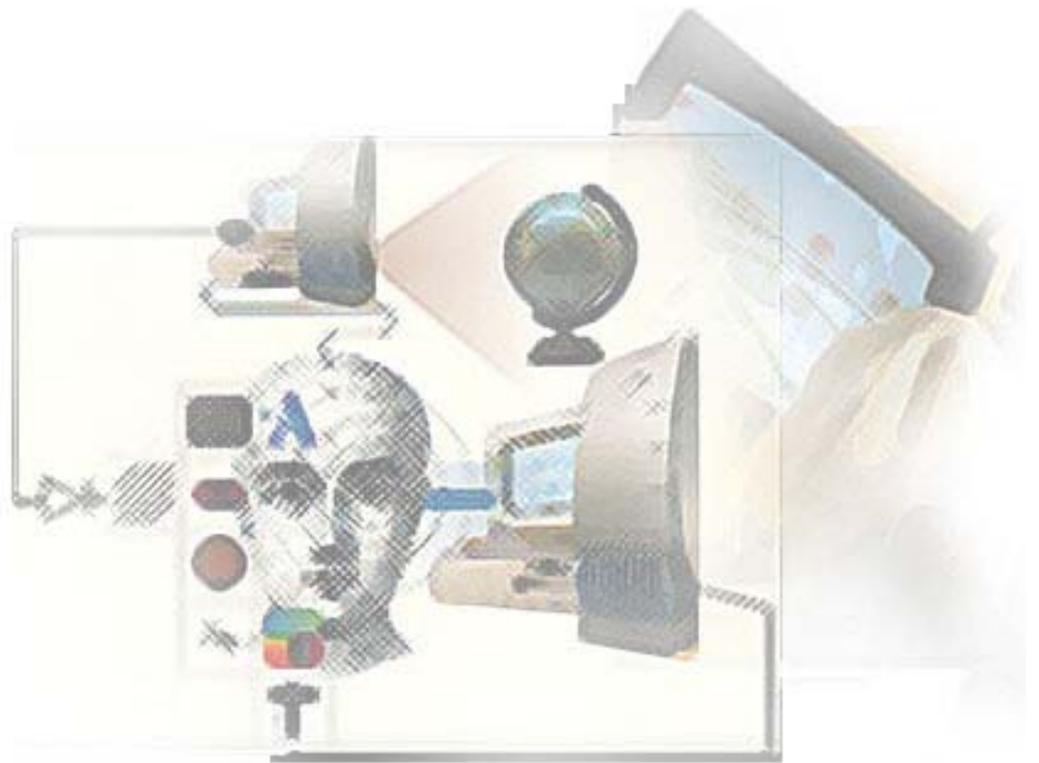


ビジネス コンポーネント アーキテクチャ SOA と EDA の統一



Fiorano
Enabling change at the speed of thought

Fiorano Software, Inc.
日本オフィス

東京都千代田区外神田 3-13-2
03-6777-7530

メール: info_jp@fiorano.com

Entire contents © 2005 Fiorano Software, Inc. All rights reserved.

この文書は、書面による事前の許可なくいかなる形態においても複製を作成することを禁止されています。

この文書に記載されている情報は、信頼がおけると信じるに足る供給元から得ています。

Fiorano Software 社は、この文書の内容について、正確性および完全性の保障をするものではありません。Fiorano Software 社は、この文書に記載されている情報およびその翻訳物の誤記、脱落または不十分性について責任を
いません。

▶ はじめに

ビジネス環境の急激な変化に迅速に対応していける柔軟でコンポーネント化された次世代のビジネスシステムを開発するためのアプローチとして SOA (サービス指向アーキテクチャ) と EDA (イベント ドリブン アーキテクチャ) が注目されています。しかしながら、SOA や EDA は、複雑なビジネスアプリケーションの分散化にはたいへん役立ちますが、ビジネス ソリューション自体のモジュール化にはあまり期待できません。分散されたコード モジュールを管理することも重要なことですが、アジャイルなビジネスシステムや拡張性に富んだシステムを開発するためにはソリューションのコンポーネント化がたいへん重要です。

ビジネス コンポーネント アーキテクチャ (BCA) とは、再利用可能なビジネス コンポーネントの集合としてのビジネスシステムを開発することです。ここでいうコンポーネントの集合とは、リクエスト - リプライ (SOA) や イベント ドリブン (EDA) によってコンポーネント同士が相互に連携しあっている状態を言います。BCA は、サービスやイベントを組み合わせることでビジネスシステムを構築し、再利用の活性化、管理の容易化、ダイナミックな拡張性の促進、ビジネスプロセス管理の容易化を図るものです。

▶ SOA (サービス指向アーキテクチャ)

SOA とは、コンポーネント (しばしば “サービス” と言及されます) を用いたアプリケーション デザインのことを指しています。ここでいうコンポーネント (サービス) とは、他のクライアント アプリケーションから呼び出すことができるインターフェースを有しているものを指し、また複数のコンポーネント (サービス) を鎖のようにつなげて (リクエスト - リプライ方式の呼び出しの連鎖によって) より大きな『コンポジット アプリケーション』とすることも可能なコンポーネントのこととも指しています。このように複数のコンポーネントの連鎖からなるコンポジット アプリケーションは、大規模なビジネス プロセスを形成するロジカル モジュールとして機能することが期待されています。図 1 は、典型的な SOA のインプリメンテーションを表しています。

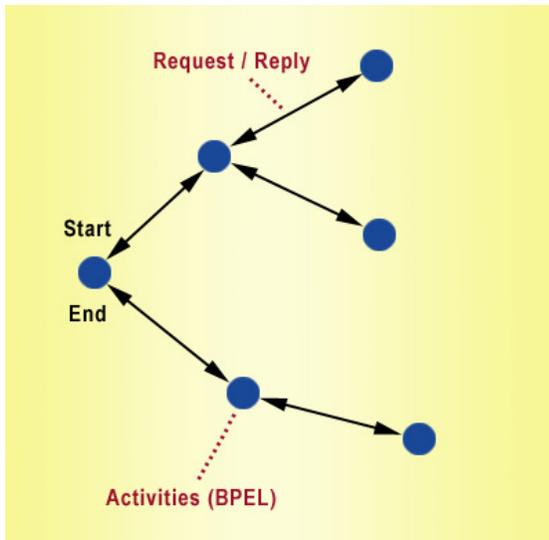


図 1-サービス指向アーキテクチャ

しかしながら、SOA では、リモートに在るコンポーネントの機能にアクセスするためのリクエスト - リプライ方式の分散アプリケーションを構築することにその主眼が置かれており、各コンポーネントを構築するためのコンポーネントモデルを必須のものとして定めてきませんでした。コンポーネントモデルとは、『粒度の大きさに基づいたソフトウェアモジュール』の開発を支援するためのものです。ここでいう『粒度の大きさに基づく』とは、ビジネス業務の大きさに適合していることを指し、技術的な意味でのソフトウェアモジュールの機能の大きさを指していません。SOA では典型的に、ソフトウェアモジュール間のインターアクションという分散環境での振舞いに焦点をあてており、ソフトウェアモジュールそのものの設計やセマンティクスにあまり注目していません。実装済みの多くの SOA でも、ソフトウェアコードとしてのモジュール間のリクエスト - リプライ方式のインターアクションの効率性を誇っていますが、モジュールそのものの設計はあまり良いものとは言えません。”大き過ぎる”か“小さ過ぎる”のどちらかであり、技術的なセマンティクスばかりを実現し、ビジネス業務上のセマンティクスには注意をはらっていません。このように、再利用性が高くモジュラーの利点を持つビジネスコンポーネントの構築方法にあまり力点をおいてこなかった結果として、SOA システムの開発やデプロイメント、出来上がった SOA システムの機能拡張や変更など、最新のシステムといわれる SOA が旧メインフレーム時代と同じような難しさを持つものとなってしまっています。

▶EDA (イベント ドリブン アーキテクチャ)

EDA とは、複数のコンポーネント同士がイベントを介して連携しながら業務機能を実行していくアプリケーションを構築するための方法を指しています。SOA との大きな違いは、SOA では連鎖の開始点や途中にあるすべてのビジネスコンポーネントがリクエスト - リプライによる呼び出し先からのリターンが返ってくるまで待機していなければならない点にあります。EDA では、どのコンポーネントもリターンを待たずに次の処理を続けられます。EDA のコンポーネントは、受信したメッセージを処理し、続いて送信メッセージの配信を行うだけで、リターン結果を待ちません。EDA では、(呼び出し先からのリターンを待つ必要がないため) 鎖状につながった複数のソフトウェア モジュールによるイベントのパイプライン処理、並列処理が可能で、その結果として SOA よりも効率がよくなります。図 2 は、EDA のインターアクションを表しています。

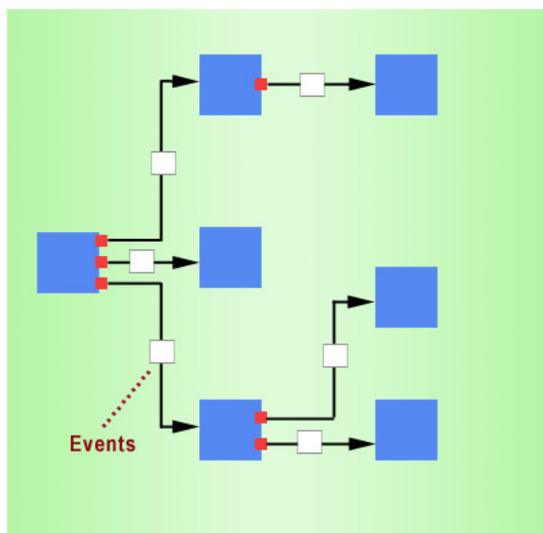


図 2 - イベント ドリブン アーキテクチャ

しかしながら残念なことに、現在の EDA アプローチでは SOA と同じ問題を抱えています。すなわち、EDA のプロセス連鎖に参加するコンポーネントのモジュラリティや粒度よりも、リモートへ分散されたソフトウェア コンポーネント同士のイベントの交換方式に焦点があたってしまっています。EDA モジュールのビジネス セマンティクスに基づいた設計が足りないため、アプリケーションのメンテナンス、デバッグ、変更拡張が難しいものとなってしまっています。

▶ ビジネス コンポーネント アーキテクチャ - SOA と EDA の統一

BCA とはアーキテクチャに対するアプローチのことで、ビジネス要件や課題をより小さなモジュールへ細分化するという考え方に基づいています。細分化された各モジュールはカプセル化されたコンポーネントとして実装され、アクセスと実行が可能な形で記述された (well-defined) ビジネス上の業務機能がこの BCA というアーキテクチャで実行されます。ビジネス コンポーネント間の連携は、リクエスト - リプライ (SOA) によるものでも、イベントを介した (EDA) ものもよく、両方を使用することができます。BCA はこのように、アプリケーションの設計方法を分散コンピューティングのコンセプトからビジネス コンポーネント モジュールのインテリジェントな設計へと転換させてくれます。図 3 に示すように、一つのアプリケーション内に複数のリクエスト - リプライ方式と複数のイベント交換方式の両方を含んで構築することが、BCA では可能です。

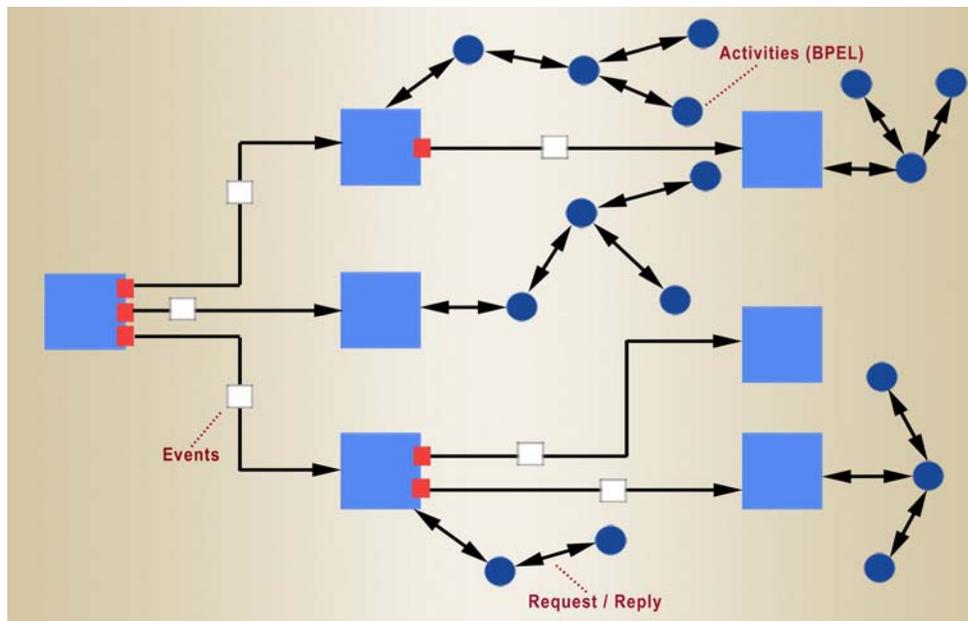


図 3 - ビジネス コンポーネント アーキテクチャ - SOA と EDA の統一

BCA では SOA と EDA を論理的に単一のフレームワークで統一して使うことを可能としています。その結果、アプリケーション内の各ビジネス コンポーネント間をどのように結びつけていくかという分散環境上の問題はほぼ解決でき、ソフトウェアモジュラリティをどう実現していくかという問題により注力できるようになります。ビジネス コンポーネント間の連携方式の決定 (リクエスト - リプライ、イベント交換

のどちらにするか)よりも、ビジネスコンポーネントを実装する粒度の正しい度合いを見つけることのほうがはるかに重要です。

▶ ビジネスコンポーネント: SOA と EDA のビルディングブロック

ビジネスコンポーネントとは、ビジネスファンクション(業務機能)のセマンティクスに従ったソフトウェアモジュールのことです。単一の技術的な機能を実装するプログラミング上のコンポーネントとは別のものです。例えば、“顧客住所の変更”はビジネスコンテキスト上の意味を持ったビジネスコンポーネントであります。一方、“データベースのテーブル変更”は技術的な機能を実装したコンポーネントであり、ビジネス上の機能とは直接的な関係はありません。リクエスト-リプライ方式またはイベントドリブン方式で情報交換を行うビジネスコンポーネントを組み合わせることでアプリケーションを開発する方法を、ビジネスコンポーネントアーキテクチャ(BCA)と呼んでいます。BCAアプリケーションでは、プログラミングの労力をほとんど要しないでアプリケーションの変更、管理が行えます。そのため、従来のモノリシックなアプリケーションのデザインと異なり、BCAのアプローチはビジネスシステムを開発する上でたいへん魅力的なものとなっています。

▶ BCA の特長

ビジネスコンポーネントには、『カプセル化』と『モジュラリティ』という2つの重要な特徴があります。

カプセル化：70年代後半に出てきたオブジェクト指向プログラミングによってよく知られるようになった機能です。これは、ソフトウェアコンポーネントのインタフェースと処理ロジックを分けるためのものです。ビジネスコンポーネントの外部インタフェースは、コンポーネントによって提供される機能を外部へ提供するためのものです。一方、内部の実装は、外部からの認められていないアクセスに対して隠されています。ビジネスコンポーネントは、“リクエスト-リプライ”方式もしくは“イベントドリブン”方式のいずれかを採ります。“リクエスト-リプライ”方式では、外部プログラムから呼び出すことができるサービスインタフェースを有し、このインタフェースを通じて内部機能をインボークします。“イベントドリブン”方式では、イベントディスクリプタを持っており、コンポーネントの実行時にそのインプットポートに入ってくるイベントのうちディスクリプタとマッチ

するイベントをコンシュームします。このように、ビジネス コンポーネントにアクセスできるのは、その外部インタフェースを通じてだけとなります。コンポーネントの内部実装の変更は、コンポーネントを利用している外部のアプリケーションに影響しません。BCA アプリケーションにおいても、インタフェースに違いがなければ、コンポーネントを内部実装が異なる他のコンポーネントと置き換えることができます。

業務ロジックの実装からインタフェースを区別することは、インタフェースの規則を守っている限りにおいて、各コンポーネントの開発を地理的に離れた別々の開発チームで行えることを意味します。カプセル化は、また、バージョン コントロール、コンフィグレーション管理、ダイナミックなデプロイメントなど、現代の分散ビジネス アプリケーションの開発に有用な多くの利点を提供してくれます。

モジュラリティ：モジュラリティとは、問題をいくつかの小さな問題に分解する過程を指します。各ビジネス コンポーネントはそれぞれ単一の業務機能を実装しているという点から、ビジネス コンポーネントはモジュラーであるといえます。また、各ビジネス コンポーネントは、他のビジネス ソリューションで再利用することが可能です。単一のビジネス コンポーネント モジュールの内部実装は、典型的に一連のステップまたは "アクティビティ" からなっています。ステップやアクティビティがチェーンのように繋がることで、コンポーネント内の情報フローを実現しています。

▶ BCA の利点

従来のモノリシックなアプリケーションが一つのもので全てを実現するよう設計されるのとは異なり、BCA アプリケーションは、イベント (EDA) または リクエスト - リプライ (SOA) で通信し合う複数のビジネス コンポーネントを組み合わせることで実現されます。BCA には、モノリシックなアプリケーションとしてデザインする従来の方法に比べて、大きな利点があります。

- **開発の柔軟性**：ビジネス コンポーネントでは、開発がより容易なものとなります。すべてをまとめて実装するモノリシックなアプリケーションは複雑で大きなものとなってしまいますが、個々のビジネス コンポーネントのビジネス ロジックは大幅に単純なものとなります。ビジネス コンポーネントでは、複数の開発チームで分担して個々のコンポーネントを開発でき、他のチームが開発しているコンポーネントの内部実装の詳細を知る必要がありません。

- **再利用性**：各コンポーネントはプログラムからアクセスするためのインターフェースが定義 (well-defined interface) されているため、他のコンポーネントとは独立して開発、テスト、デバッグを行えます。これは、プロジェクトの開発生産性を高めるだけでなく、ビジネスコンポーネントの設計が適切に為されている場合には再利用を大きく促進します。
- **ダイナミックなデプロイメントと実行時の変更 / 置き換え**：ビジネスコンポーネントは、リモートのノードに実行時にダイナミックにデプロイできます。また、プロセス内のコンポーネントを、新規のコンポーネントや変更を加えたコンポーネントでダイナミックに置き換えることもできます。ひいては、頻繁に変化するビジネス環境に応じて既存プロセスを変更していくためのコストと時間の削減につながります。
- **コンフィグレーション管理とバージョンコントロール**：ビジネスコンポーネントにはバージョンコントロールおよびダイナミックなコンフィグレーション管理の機能が備わっており、企業全体にわたったデプロイメントの管理が可能となります。

▶ まとめ

SOA や EDA が分散システムを構築するためのアプローチとして普及してきていますが、どちらのアプローチもビジネスコンポーネントのカプセル化やモジュラー化に焦点を当てておらず、真に拡張性に富んだビジネスシステムを導き出すものではありません。ビジネスコンポーネントは、同期型のリクエスト-リプライ方式、非同期型のイベント交換をどのように組み合わせても連携できるものです。したがって、ビジネスコンポーネントアーキテクチャは、モジュラ ビジネスコンポーネントに基づく単一のフレームワーク上にサービスとイベントを統一化するものであり、プログラマーの手を煩わせることなく変更、管理が行えるアプリケーションを実現します。BCA は、アジャイルなビジネスシステムを実現する次世代のアーキテクチャであると言えます。

▶ Fiorano Software について

Fiorano Software は、カリフォルニアに本社を置く、エンタープライズ インテグレーション ミドルウェアの業界をリードしている企業で、メッセージング インフラ

トラクチャ技術において数多くのお客様から高い信頼をよせられています。Fiorano のソリューションは、インターオペラビリティ、パフォーマンス、スケーラビリティ、ROI などの面で新たなパラダイムをもたらしています。アメリカン エクスプレス、AT&T ワイヤレス、ボーイング、BP (旧ブリティッシュ ペトロリアム)、エリクソン、FedEx、ロッキード マーチン、モーガン スタンレイ、モトローラ、POSCO、シュルンベルグなどの世界的なリーダー企業で Fiorano の技術が採用され、企業のバックボーン システムとして稼動しております。

Fiorano Software に関する詳細な情報は、弊社のホームページ (www.fiorano.com/jp) をご参照くださるか、info_jp@fiorano.com 宛てに電子メールでお問い合わせください。