



アドバンスセッションー3

Web 系システム開発の困難を乗り越える

[講演要約]

CAC が提供する Web 開発ソリューション CWeS

株式会社シーエーシー 金融ソリューション本部 プロデューサー

吉山 潔

1. 新しい Web 開発ソリューション体系

Web システムは、企業における現在または次期の標準システム・インフラという位置付けにまでなっている。しかしながら、現状の課題は多い。数多くのお客様にヒアリングを行ったが、わずか1～2年で寿命となり作り直しが必要となる Web システムも少なくない。

その要因として、J2EE や Web 負荷分散システムといった技術的なハードルの高さ、これに対する設計手法、開発方法論の欠如、将来的な業務要件の追加・拡張までを見込んだシステム構築の難しさ、そして再利用性を無視したプランニングなどが挙げられるだろう。その結果、開発時点で必要となっている業務要件にしか対応できない硬直したシステムになってしまい、TCO 削減などは夢のまた夢だ。

求められるポイントは2つある。1つは素早いサービス・インの実現、もう1つは再利用性の高いシステムの実現。経営的な視点から見れば、この2つによって、システムのトータルコストの削減や、高いビジネス競争力をもたらす高品質なシステムが実現されるのだ。そのためには、明確な開発方法論に裏付けられた開発基盤が必須となる。

こうした課題に対するソリューションとして弊社が提案するのが CWeS だ。さまざまなお客様に対し Web ベース



の企業システムを提供してきた CAC が考える、新しい Web 開発ソリューション体系である。CWeS では、システム開発や運用など Web システムの全ライフサイクルにわたってコストを削減し、迅速なサービス・インと高い品質を実現する。

2. 3層構造が可能にする高い性能と開発生産性

CWeS の特徴は、Base、Service、Tailor という3層構造だ(図1)。これにより Web システム特有の基本機能の開発が不要となり、業務共通部分もコンポーネントとして提供されるため、開発生産性とその精度が格段に向上する。

CWeS 3層概念図

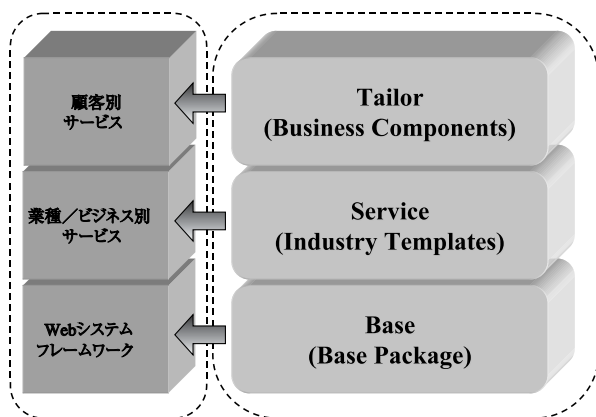


図1 CWeSによる3層ソリューション

図1の一番下にあるBaseは、Webシステムにおけるベースとなる機能を提供するレイヤーだ。たとえば、Webシステムが苦手とする連続処理やセッション管理、また処理の安全性を確保するセキュリティ機能などを提供する。これらは、Webシステム実現の際に基本となる機能だ。

第2層のServiceは、業務という視点からシステム機能を考えて見えてくる、各業務にとって普遍的に必要な機能を実現するものだ。たとえば、金融システムにおける銘柄管理、日々の取引における注文約定、お客様ごとに必須な口座管理などの機能を、各金融機関に共通する普遍的な機能として提供できる。

最上層のTailorでは、お客様ごとに固有となる業務機能を構成する。言い換えれば、お客様の競争力を高めるためのシステム機能を提供するレイヤーだ。お客様企業の内部で共通に使えるコンポーネントとテ일러メイドで構築する機能から成る。

CWeSでは、この3階層によって早いサービス・インと高い再利用性を実現している。

そこには具体的なプラットフォーム製品から方法論、そしてコンサルティング・サービスまでが含まれる。CWeSを導入する場合は、オブジェクト指向開発方法論をベースとした開発標準、モデリング・ガイドライン、UMLベースのドキュメント標準などが体系化されたWeb開発技術ガイドラインを用いて開発が行われる。

これに加え、ユーザー部門の開発標準策定コンサルティングから自社システム開発まで、幅広く適用を行っていく。これらの方法論、ガイドラインが3層の仕組みを背後から支援し、CWeSの提供ソリューションをより確かなものにしていくのだ。

3. 適用の広がる CWeS のアプローチ

CWeSのコンセプトに賛同いただいている優秀なパートナー企業が多数おられる。その1社、ファイテック ラボラトリーズ社には、このあと、金融システム・ソリューション「xTrade」についてご紹介いただく。また、オブジェクト指向開発環境を提供するトゥゲザーソフト・ジャパン株式会社、そしてSEICMMのレベル5認定企業であるインドのNIIT社など、これらのパートナーとのコラボレーションにより、CWeSはより具体的で有効なソリューションを提供する。

CWeSは金融業務システムの開発を通じて誕生したが、その適用範囲は金融業務だけにとどまらず、Webベースの企業システム全体に及んでいる。CACでは、Webシステム開発における標準としてCWeSを活用し、今後のさまざまなWebシステム開発に適用していく予定だ。

CWeSを使ったWebシステム構築の事例としては、4大金融グループの1社のシステム、また、ある地銀の大規模CRMシステムなどが挙げられる。前者の場合、CWeSを使わない場合の見積もりでは1年以上かかると見られた開発工程が、半年ほどに短縮できた。注目していただきたいのは、いずれの場合も、業務要件の縮小など作業量の圧縮が行われたのではなく、当初の要件を満たした上で早期のシステム構築が実現できたという点だ。決して技術的リスクの低くないWebシステム開発において、当初の業務要件を満たしつつ迅速なサービス・インを迎えられたのは、CWeSによるものが大きいとの評価を受けている。

CACではCWeSのBaseレイヤーを構成する製品としてWebmentorを提供する。これは、長年携わってきた企業向けWebアプリケーション開発のノウハウが集約された、CWeSの中核を構成するWebアプリケーション・フレームワークだ。

WebmentorはJ2EE標準に完全準拠し、Webシステム開発者に求められる専門知識を大幅に低減する。その特徴として、デザイン・パターンを用いた再利用性の高い設計方法、ユースケース・ベースの開発に最適なシステム・デザイン、J2EE準拠のデザイン・フレームワークの活用、そしてオブジェクト指向MVCの採用によるパフォーマンス重視の設計などが挙げられ、これらの活用により、開発コストを大幅に削減し、速いサービス・インを実現する。

弊社の実績では開発効率が30~40%向上し、その結果、開発期間が短縮され、開発コストも削減された。後続の開発作業の効率にも好影響をもたらし、システムの品質向上

にもつながっている。

改めて作る必要のないものは再利用し、必要不可欠な機能の開発に集中する、というアプローチによって、顧客満

足度の高い Web システムを迅速かつ容易に構築することが可能になる。今後の CAC の Web システム開発に、CWeS と Webmentor はなくてはならないものとなるだろう。

[講演要約]

次世代金融プラットフォーム xTrade

ファイテック ラボラトリーズ 社長 兼 CEO

松島利幸氏

1. 金融業界を中心としたトレーディング・システム構築のための Java コンポーネント・フレームワーク

xTrade は、主に金融系の高機能なトランザクション・システムを構築するフレームワーク製品だ (図2 参照)。その特徴は大きく5つにまとめられる。

第1に、非常に高いパフォーマンスを実現できること。Java のアプリケーション・サーバー上のプラットフォー

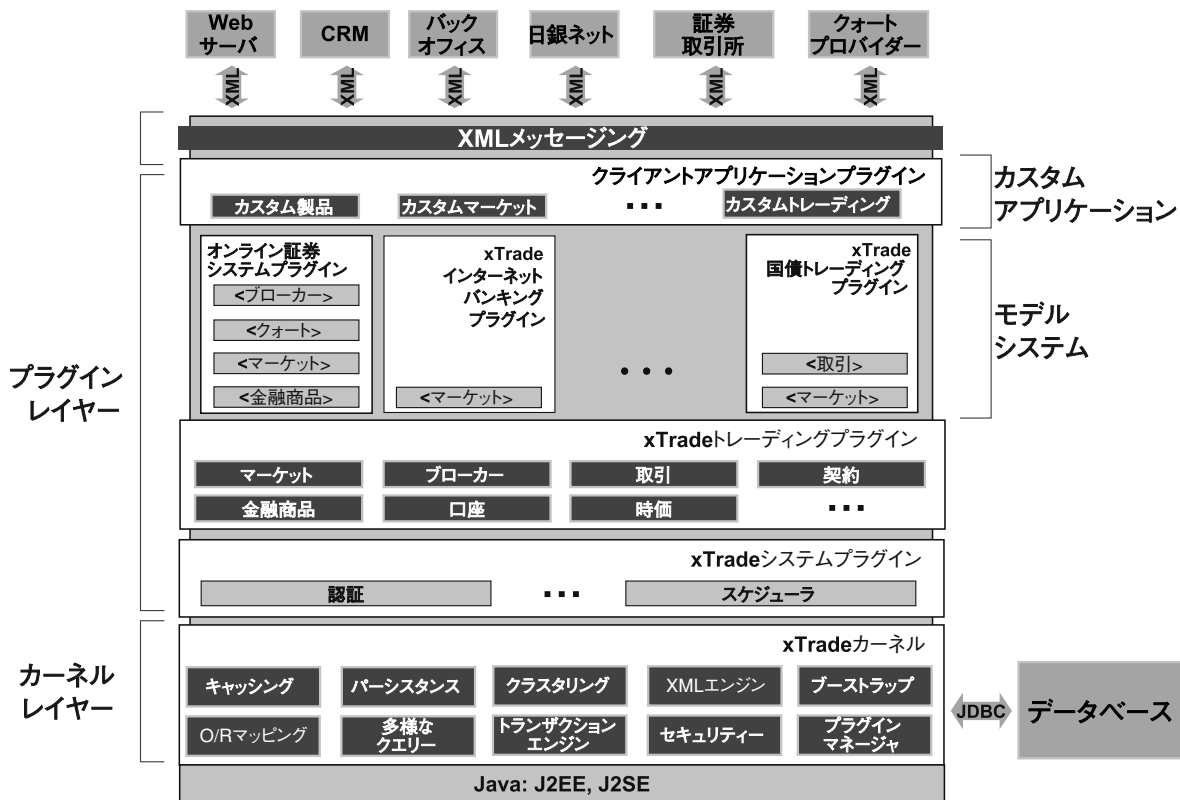


図2 xTrade コンポーネント・アーキテクチャ

ムとしては、現時点でおそらく最高速と考えられる。

第2に、再利用可能なコンポーネント・アーキテクチャであること。

第3に、本来、金融システムを念頭に作られているため、高い信頼性とセキュリティを実現していること。セキュリティレベルに関しては、たとえば、あるアカウント・グループごとに個々の操作、つまりクラスのメソッドを起動できるか否かというような、非常に細かいレベルの設定もできるようになっている。それらも、当然ながら Web ベースのアドミニツールなどでダイナミックに制御できる。

第4に実績。オンライン銀行であるイーバンク銀行のメールベースの送金システムを、xTrade で構築して、2002年7月から稼働した。電子メールで送金できる米国のオンライン決済サービス「ペイバル」とほぼ同等レベルのシステムである。

第5の特徴は、最近特に EAI の標準になりつつある XML を採用して、他のサブシステムとの融合性を高めていること。企業の既存のシステム・インフラにすんなり入って行かなければならないわけだから、この点は非常に重要だ。

弊社には、松井証券株式会社という、非常にパフォーマンスを要求されるお客様をクライアントとしてきた歴史がある。この松井証券のネットストックをはじめ、数々のミッション・クリティカルな金融系オンライン・システム構築で培ったサーバーサイド Java のテクノロジーをベースに3年間に及ぶ独自の技術開発を集結し、金融業務のノウハウを活かし完成させたのが、xTrade だ。xTrade では、データベースのスキーマから自動的にバインディング・クラスを生成する O/R マッピング、または、クラスタ環境の中でデータベースからオブジェクトがキャッシュされた場合に各オブジェクト間の整合性を取る分散キャッシュ、リクエスト・レスポンスのモデルに基づくシングル・トランザクション、アカウント・グループごとにデータベースを分割管理するアカウント・セグメンテーションなどの独自技術を駆使している。

こうして、Web 系、特にハイパフォーマンスを要求される金融系オンライン・システムにとって必要最小限となる機能を効率的に絞り込み、アーキテクチャおよびインプリメンテーションの双方に徹底的にチューニングをかけて、システムの構築に十分足りるプリミティブを提供した。

それが図2の下部にある「xTrade カーネル」というレイヤーだ。ここに、ハイパフォーマンスで、スケーラブルで、リライアブルな金融系のシステム構築のための知恵が詰まっている。さらにその上に、コンポーネント・アーキ

テクチャのシステムを作った。

2. 柔軟性の高いコンポーネントベース・システム

弊社ではコンポーネントのことを「プラグイン」と呼んでいる。基本的に、メッセージングのメタ・リプレゼンテーションとデータベースのメタ・リプレゼンテーション（スキーマ）およびビジネス・ロジックなどをバンドルしたものをプラグインと呼ぶ。このプラグインは、xTrade カーネル上で非常に高いパフォーマンスで稼働する。

プラグインには複数のレイヤーがあり、システムプラグインと呼んでいるレイヤーでは、ホリゾンタルな機能を提供する。ここでは、ログイン・パスワードや PKI などに対応した認証のプラグインやバッチジョブなどを組み込めるスケジューリングのプラグインがある。

そして、金融トレーディングに特化している弊社の場合、さらにその上に、トレーディングのための汎用的なプラグイン・レイヤーを設けている。

つまり「注文とは、ある商品のある数量、ある条件で取引するリクエストである」という概念に基づき、実際にインプリメンテーションを行った汎用的なレイヤーを1階層設けているのだ。その上に、各商品を株や指数オプションなどに特化させればオンラインの証券取引システムができる。また、商品をキャッシュ・プロダクトやフィックス・インカム・プロダクトに特化させると、それはオンライン・バンキング・システムになる。さらに、航空会社のマイレージやネットマイルのようなポイントに特化させると、ポイント・システムができる。そうしたシステム構築ができるようなフレームワークを xTrade で構成した。つまり、オンライン・トレーディングやオンライン・バンキングの基本システムを提供している。このレベルまでがちょうどビジネス・ロジックを実現する部分だ。

また、外部とのやりとりのため、標準で XML をサポートしている。このメッセージング・レイヤーはかなりフレキシブルだ。当然、XML のさまざまなオーバーヘッドが大きい場合に備え、たとえばオブジェクトをそのまま投げるといったことも可能な、非常に速いプロトコルもサポートできる構造になっている。

xTrade のサーバーのプレゼンテーション・レイヤーは、さまざまなプレゼンテーション・スキームをサポートできる。PC の Web ブラウザや VB で作られた Windows 上のアプリケーションもあり、DoCoMo などの携帯端末もある。これらも XML を使っているので、プレゼンテーショ

ン・レイヤーで構造化されたメッセージを変換することにより、自在に対応することが可能だ。たとえばXMLをHTMLに変換すればWebブラウザ上で動くアプリケーションができるし、CHTMLに変換すればモバイル端末で動くオンライン・トレーディング・システムができる。

このように、アプリケーション・サーバー側のインタフェースを特に変えることなく、さまざまなプレゼンテーション・デバイスに対応し、またさまざまなサブシステムに融合していくことが可能だ。これは、XMLという構造化されたメッセージングの強みである。証券系のシステムでは、マーケットとのやり取りやデータの取り込みなども、XMLベースで一種のアダプタを構築することにより、既存のシステムにどんどん入れていくことが可能だ。

xTradeのアーキテクチャは将来的に非常に有効だ。現在、米国でもお客様から評価を得ている。今後、日本でもいろいろなかたちで金融の垣根が取り払われ、さまざまなサービスを提供しなければならなくなる。たとえば銀行ではETF（指数連動投信）などを扱うようになると思うが、その際、全く別システムで提供するよりは、コンポーネントベースの柔軟性の高いシステムで統合されたサービスを提供できることは非常に有用だろう。

プラグイン・アーキテクチャの特徴は、コンポーネントの入れ替えで、新しい業務への対応や、別システムとのインタフェースが可能になることだ。弊社は金融系に特化して、いろいろなコンポーネントを提供している。またオブ

ジェクト指向型のコンポーネント・アーキテクチャであるため、既存のコンポーネントをカスタマイズするのもかなり自由だ。XMLベースで他システムとのインタフェースを柔軟なかたちで提供できるので、Webサービスとしても使っていただける。

3. 圧倒的に優れたxTradeのパフォーマンス

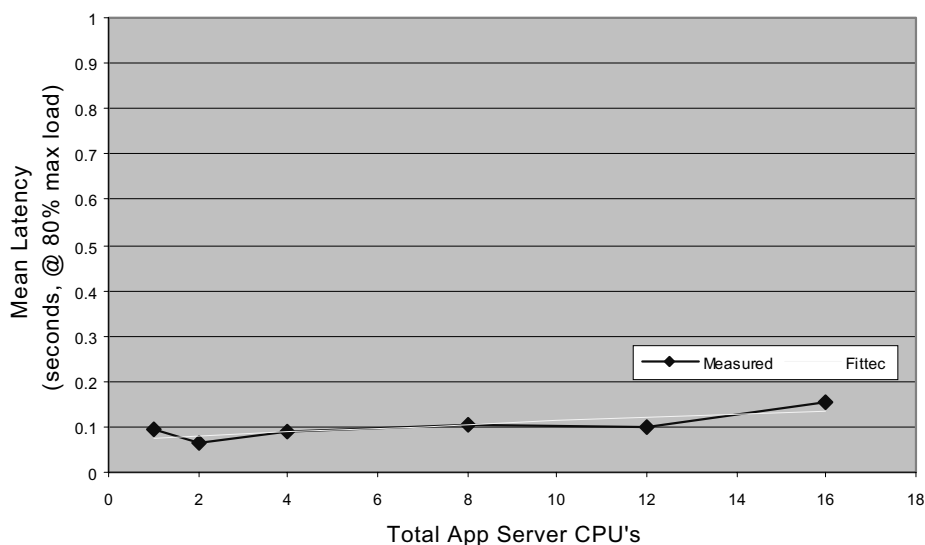
以上がxTradeの概要である。パフォーマンスに徹底的にこだわり、J2EEのサブセットをあえて使って処理速度を上げた。ではどれほどの速度が出るのか。2002年6月に米国サン・マイクロシステムズのベンチマークセンターでテストさせていただいたが、かなり良い数字が出た。

弊社の開発したオンライン・トレーディング・システムを使い、現実的なシナリオでテストした結果、2,000万～3,000万円程度で構築できるようなワークグループ・サーバー・レベルのハードウェアで、1時間あたり500万ヒットまでこなすことができた。米国のオンライン証券会社チャールズシュワブ社でも1日あたり2,500万ヒットだ。

応答時間も、システムがスケールアップしても常に0.2秒以下にとどまった（図3）。スループット、応答時間ともにきちんとスケールアップしていることを示すデータだ。

なぜそれほど速いか。基本的にはやはり弊社独自のテクノロジーがある。たとえば分散キャッシュ。先述したよう

システムの拡張後も、高速レスポンスを確保



➡ レスポンスタイムはサーバー台数にかかわらずに常に0.2秒以下を計測。エントリーレベルのアプリケーションをエンタープライズ規模に拡張しても、常に高速レスポンスを保てる分散キャッシング技術、スケラビリティがあることが実証されました。

➡ より大規模なエンタープライズシステムにおいても、xTradeベースのアプリケーションであればスケールアップが可能です。

xTradeであれば高負荷システムにも低コストで高いサービスをユーザに提供できます。

図3 xTrade ベンチマーク検証

に、分散環境下で本当の意味でのキャッシングをして、キャッシュ・コヒーレンスも保証している状況なら、メモリーにアクセスしていろいろなデータを整合的に取ってこられるため、圧倒的に速い処理ができる。詳しくは弊社ウェブサイトをご参照いただきたい (<http://www.fitechlabs.co.jp/products>)。

EJB との比較も気になると思う。BEA Systems 社の WebLogic や IBM の WebSphere などは、かなりパフォーマンス・チューニングがされている。当然キャッシュ機能も、WebLogic7.0J などからは備えている。ただ、先述したように、本当の意味での分散トランザクションをサポートしているので、どうしてもオーバーヘッドが高くなってしまふ。

あるお客様の要望で EJB と xTrade のパフォーマンスをテストした。これには、WebLogic7.0J のキャッシュなどの技術も含めた状況で比較してみたが、やはり xTrade のほうが100倍くらい速い (図4)。

- 検索中心のデータベースアクセスにおいて、100倍以上のパフォーマンスを計測
 - EJBコンポーネント: 7,385/1秒
 - xTradeプラグイン: 1,000,000件/1秒



図4 xTrade vs EJB with キャッシング パフォーマンス比較

弊社は EJB を否定しているのではない。EJB でなければ対応できない部分もある。しかし、シングル・トランザクションのモデルでシステムを構築できるようなオンライン・アプリケーションの場合は、xTrade を使っていたければ、非常にコストパフォーマンスがよくなるはずだ。

4. 「Win-Win-Win」の関係を目指す

パフォーマンスにこだわった背景には、金融系に特化してきた弊社の歴史もあるが、ネットバブルの崩壊を目の当たりにしたエンジニアとしての考えもある。

「スケーラブルなシステム」を考えてみよう。当然、J2EE もスケーラブルなシステム・フレームワークだが、スケーラブルなだけでは実は十分ではない。

インターネット系のビジネスやシステムは、多額のお金

が入ってきたせいもあり、とにかくスケーラブルに、という考えで作ってしまった。つまり、取引がどんどん増えて売り上げが大きくなると、システムも増強してどんどん大きくなる。すると、取引量はきちんと管理できてもパフォーマンスがそれほど高くなり、結局システムコストが売り上げを上回ってしまう。そのため、いつまでたっても損益分岐点をクリアできず、結局儲からない。ハードウェア/ソフトウェア・ベンダーの場合も、最初は儲かっていたが、結局ネットバブルが崩壊したため売り上げが見込めなくなった。それに対し、xTrade のハイパフォーマンス・システムでは、コストの上昇がずっと緩やかで、必ずどこかで損益分岐点をクリアして儲かるようになる (図5)。

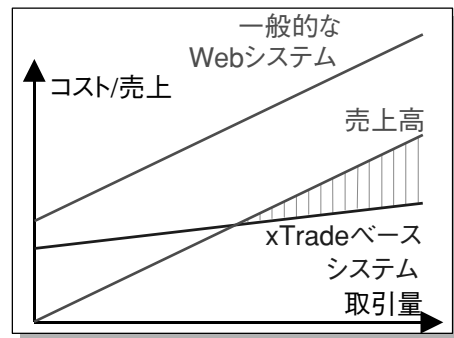


図5 xTrade を採用した場合のメリット

我々のビジネスの基本は、作ったシステムでお客様に儲けていただき、その儲けの一部を頂戴して自分たちも儲かることだ。ネットバブルでは、たまたま顧客がお金をどんどんつぎ込んでくれたので、ベンダーやシステム・デベロッパーもそれに乗じて儲けていた。ところが、結局顧客のビジネスが成り立たなくなり、全員困ったことになった。よって、弊社ではやはり徹底的にパフォーマンスにこだわろうと考え、xTrade というシステムを作ったのだ。

お客様のビジネスには、同業他社と本質的には変わらない「基本的な部分」と、会社によって「差別化を図るべき部分」とがある。お客様にとって、本当に重要なのはこの差別化の部分であり、基本的部分はきちんと動いていて当たり前のはずだ。が、実は、基本部分のシステム構築に時間とコストがかかってしまうのが常で、本当に重要な、他社との差別化を図る部分に、なかなか予算もエネルギーも回らない。それを何とか解決するのが弊社の役割だ。

オブジェクト指向のコンポーネント・アーキテクチャの長所は、ベースのシステムを変えるのではなく、カスタム・コンポーネントを付け加えることによってシステムのビヘイビアを変えていくことだ。それにより、お客様と我々と

の上手な役割分担が可能になる。

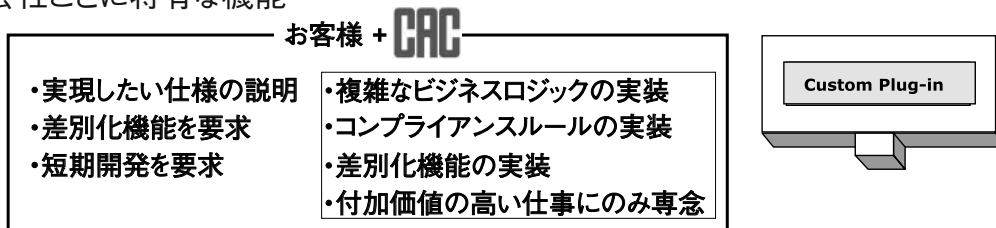
我々のパートナーとなるシステム・インテグレータには、個々のお客様に特有なビジネス・ロジックやコンプライアンス等に関して、お客様とともに仕様をきちんと把握し、カスタム・コンポーネントを作っていただく。これにより、ビジネスの核となる部分に非常に高い価値を提供できる。お客様には、きちんとできて当たり前の基本部分は完全にごちらにお任せいただき、ビジネスにとって本当に重要な差別化の部分にエネルギーを集中して、システム・インテグレータとともにシステムを構築していただく。つまり三者とも「Win-Win-Win」の関係になれる。これが弊社の基本的なビジネス・スキームだ（図6）。

これまで、基本的な部分にばかり時間とお金を取られていたケースでも、xTradeのようなシステムを使ってその

基本部分に高いパフォーマンスを提供できれば、企業が独自の差別化を図るべきサービスの構築に、予算も人もかけることができるはずだ。

CACのように弊社とシナジーの高い関係にあるシステム・インテグレータに加え、サン・マイクロシステムズ株式会社、日本オラクル株式会社、日本BEAシステムズ株式会社のベンダー各社にも、非常に強力なパートナーとして協力していただいている。こうしたの方々とも緊密な関係を保ちながら、お客様に儲けていただけるシステムを提供していきたい。本来なら何十億とかかるようなシステムをその10分の1のコストで構築したり、さまざまな別々の機能を1つのシステム上に統合することなども、我々の製品によって実現できるようになったら非常に幸いだ。

会社ごとに特有な機能



特定業務にて汎用的な機能

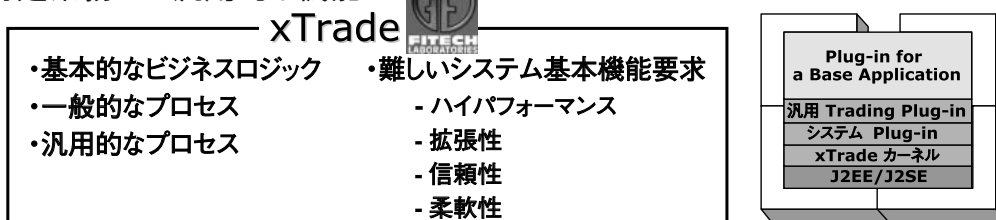


図6 xTradeに関する開発役割分担