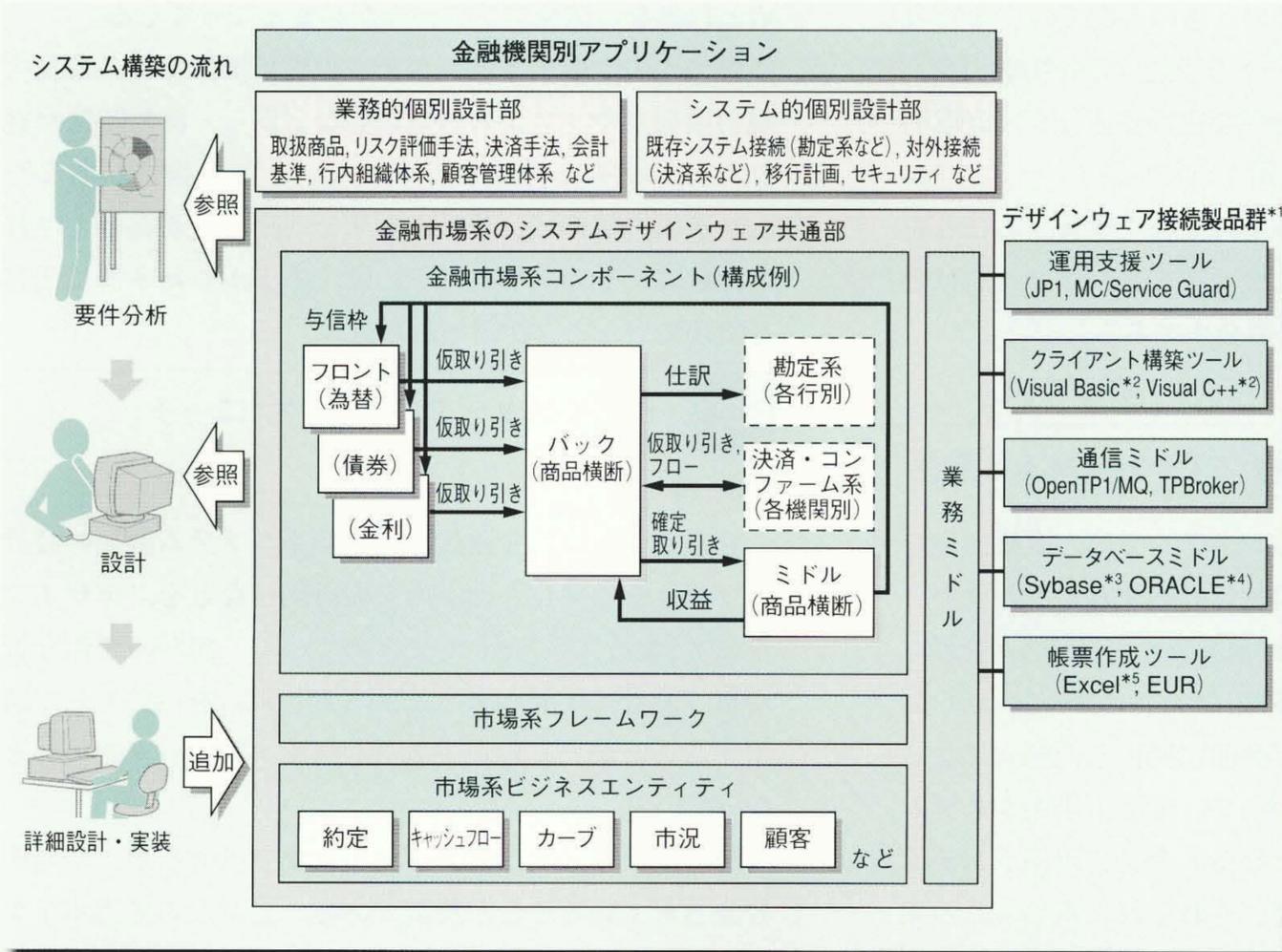


デリバティブ商品の進化に柔軟に対応する 金融市場系システム向けデザインウェア

Designware for Financial Marketing Systems Extensible with Derivative Products

二木誠司 Seiji Futatsugi 山橋哲也 Tetsuya Yamahashi
石川 明 Akira Ishikawa 仲 勇 Isamu Naka



注：
*1 「デザインウェア」および「Designware」は、シーキュー出版株式会社の登録商標である。
*2 Visual BasicおよびVisual C++は、米国およびその他の国における米国Microsoft Corp.の登録商標である。
*3 Sybase, Sybaseのロゴは、米国法人Sybase, Inc.の登録商標である。
*4 ORACLEは、米国Oracle Corporationの登録商標である。
*5 Excelは、米国Microsoft Corp.の商品名称である。

金融市場系システム向けデザインウェア

金融市場系システム向けデザインウェア共通部を中心とする、個別設計部や各種製品の適用を含めた設計体系を示す。このデザインウェアは、要件分析と設計時に参照され、その成果が蓄積されていく。デザインウェア共通部の最下層であるビジネスエンティティから、デリバティブ商品^(*)の拡張性を考慮した構造としている。フレームワーク層とコンポーネント層でも、商品の拡張性と各金融機関の個別事情に適應できるように、選択が可能なパターン集としている。

金融ビッグバンの進展に伴い、為替先物や資金取り引きなど、オプション条件の付いたデリバティブ商品に取り組むことは、グローバル展開を図る金融機関はもちろんのこと、リスクヘッジだけを目的とする金融機関や事業法人にとっても必須条件となっている。これらの金融機関などにとっては、みずからの健全性確保のため、市場環境の変化による影響を把握、コントロールする必要がある。

この金融市場系業務を支援するシステムでは、取扱商品をはじめとする業務の変化に柔軟に対応することが重要となる。ただし業務は、各取り引きの時価評価からリスク管理、勘定、決済、現物管理まで多岐にわたり、システムの対応はたいへん難しい。

これに対応するために、日立製作所は、約定などのビジネスエンティティや、これらを組み合わせて評価ロジックを実現するパターン、さらに大きなシステム構成まで、段階を設けて「デザインウェア(設計体系)」化している。このデザインウェアを指針として、要件分析段階から実装までの作業を進めることができる。パッケージを適用するアプローチと比べて初期設計段階で効率化が見込め、個別要件に柔軟に対応が可能という特徴を持つ。このデザインウェアは大規模案件に適用中であり、サービスメニューとしての提案も計画中である。

1 はじめに

金融ビッグバンの進展に伴い、資金が国際化する傾向が強まってきている。資金の移動に制約がなくなりつつあ

ることから、グローバル金融機関から輸出入に携わる一般事業法人まで、日々の為替レートの変動や金利状態、商品市況などに収益を左右されることが常態化してきた。

この金融市場系の業務では、顧客にさまざまな魅力を

※) デリバティブ商品：為替や資金、証券売買取引などに対して、これらの将来の取引レートを予約する先物条件や、市況状況によって取引成立を定めるオプション条件を付与したもの

備えた金融商品を提供すること、ディーラーの市況判断でみずから資金を移動させることなどを行っている。ここでは限られた資金で「てこ」を利かせた大量の利益を得ることもできるし、大きな損を被る危険度も高い。市場系の取引形態としては、そのときの市況で取引を行う危険を避けるため、将来のレートを予約する先物取引や、市況状況によって取引成立条件を定めるオプション取引などのデリバティブ商品が多用されている。

このように複雑な金融市場系の業務を支援することを目的として、日立製作所は、金融市場系システム向けのデザインウェア(設計体系)を開発中である。

ここではこのデザインウェアについて述べる。

2 金融市場系業務支援システムに対する要件

金融市場系業務を支援するシステムでは、取扱商品をはじめとする業務の変化に柔軟に対応することが重要となる。業務は、各取引引きの時価評価から、リスク管理、勘定、決済、現物管理まで多岐にわたっている。さらに各業務は、特にデリバティブ商品に関係した動向によって日々変化している。したがって、金融市場系システムでは、柔軟性が重要な要件となる。金融市場系システムを構成するサブシステム群と、それぞれに求められる柔軟性項目を表1に示す。

柔軟性は、単に業務を円滑に進めるためのものではない。例えば、新しい確率モデル(微分方程式)に基づくプライシングロジックに対応することは、フロントで競合

商品の収益性に直接影響を与えることになる。

このように金融市場系システムでは柔軟性が強く求められていることから、例えば預金・融資業務や証券取り次ぎ業務のように、管理手法もある程度固まっているものとは異なる開発アプローチが必要となってくる。

幾つかの金融市場系システムの構築事例では、業務実現の困難さやシステムの複雑さなどから、稼動時期が遅れたり、十分な性能が得られないなどの報告がある¹⁾。システムの構築には多額のシステム化投資が必要とされるだけに、そのアプローチには、確実かつ効率的に業務を実現できることが求められる。

3 デザインウェアによるアプローチ

3.1 特徴

日立製作所は、金融機関用市場系システム開発の設計ノウハウを標準化したものや検討項目などを、デザインウェア(設計体系)としてまとめている。開発の要件分析から、設計、実装の各フェーズでこれらを指針として活用することが、有効なシステム構築アプローチの一つとなっている。

デザインウェアは設計ノウハウ集ではあるが、階層が設定されていることが特徴である。設計上の考慮ポイントの例示から、製品の適用例を含めた実装までを体系的にまとめたものである。

金融市場系システムのデザインウェアは、デザインウェア共通部が中心であり、ここでは、(1) 約定やキャ

表1 金融市場系システムを構成するサブシステム群と柔軟性要件

金融市場系業務は一般に、取引先と約定を交わすフロント業務と、各種評価を行うミドル業務、事務処理を行うバックオフィス業務に分けられる。

サブシステム	主な業務内容	機能概要	柔軟性要件
フロント	約定構成・入力	参照する商品と条件(金額、期間、レート、オプション条件など)の構成管理	新規に設計された商品の参照と入力、約定条件の非定型入力
	プライシング	約定条件を評価し、適切な内容を構成する支援環境	プライシングロジックの新規組込み、独立したプログラムとの連携
	ポジション管理	ディーラー個人やグループ単位での収益、市場リスクなどを計測する。	商品ごとの収益算出、市場リスク計測変化への対応
	リスク枠管理	顧客信用度に応じた信用枠などの条件の設定	枠管理構造の変化への対応
ミドル	市場リスク管理	損益の確率や市況への感応度の計測	ロジックの新規組込み、独立したプログラムとの連携
	信用リスク管理	信用に応じた与信の配枠、全体での与信計測	枠管理構造の変化、与信計測手法の変化への対応
	収益管理	組織ごと、商品ごとの収益算出	商品ごとの収益算出方式への対応
バック	契約管理	相手方との基本契約内容の管理	契約規格変更への対応
	コンファーム	外部ネットワーク、ファクシミリ、郵便などを使用した相手方との個別約定内容の交互確認	コンファーム手段変更への対応、商品ごとの確認項目と内容の変化への対応
	勘定	約定内容の勘定仕訳と記帳	商品ごとの勘定仕訳ルール変更への対応
	決済	現金、有価証券の受け渡しと入金確認	決済ルール変更への対応
	担保・証拠金管理	約定の現在状況に応じた差し入れと徴求	差し入れルール変更への対応
	金利リセット	変動金利などが確定した場合の約定内容更新	商品ごとの変動要素変化への対応
	現物管理	有価証券などの預け、預かり	有価証券の種類の変化への対応

ッシュフローといったデータを表す「ビジネスエンティティ層」、(2) 業務を実現するパターンである「フレームワーク層」、および (3) より大きなシステム単位の接続形態パターンである「コンポーネント層」で構成している(33ページの図参照)。また、設計・実装経験を持つ、周辺基盤製品などとの接続を管理する「業務ミドル層」も、この中に含まれている。

各金融機関の個別業務や既存システムの実状に応じた部分は、業務・システムの個別設計部として切り出している。この中には、近い将来、標準化の見込めるリスク評価手法のようなものも含まれており、業界標準が数パターン程度に収まってきた場合には、デザインウェア共通部として取り込まれることも想定される。

3.2 関連技術・動向

デザインウェアと対照的な概念としては、パッケージやクラスライブラリがある。ここでは欧米のソフトウェアベンダのものが有力であり、例えば、米国Infinity社²⁾やSummit Systems社³⁾から統合パッケージやクラスライブラリが販売されている。デザインウェアと異なり、そのまま計算機上で実行できる形態で提供されるのが特徴である。これらの比較を表2に示す。

表2で適合性として示したとおり、デザインウェアによるアプローチは、個別ニーズに細かく対応する場合に適している。パッケージやクラスライブラリでは、個別ニーズに対応しようとして、ソースコードを改変するなどの必要性から、工数がかえって増加してしまうことがある。現在有力とされているこれらの製品でも、国内の特別な制度に対応が必要であるなどの事情に限らず、柔軟性に欠けるといって必ずしも十分ではないと考える。

関連動向として、ビジネスエンティティ層では、OMG (Object Management Group：分散オブジェクト環境の

表2 システム化アプローチの比較

デザインウェアによるアプローチは、個別ニーズに細かく対応する場合に適合する。

アプローチ	デザインウェア	パッケージ・クラスライブラリ
提供形態	設計文書、一部実装例	実行可能コード、ソースコード(カスタマイズ指針書を含む。)
要件対応	個別に、柔軟に対応	想定された要件に従うことを強制
設計効率	内容が標準的であれば効率的	カスタマイズが製品の想定内であれば効率的
実装効率	多くの場合スクラッチ開発	カスタマイズだけの対応または一部改変
適合性	個別ニーズに細かく対応する場合に適合	早期に稼働させる場合に適合

標準化団体)が金融市場系エンティティインタフェースの標準化を図るためのRFP(Request for Proposal)を発行している。ただし、現在のところ、通貨の仕様について取りまとめられた段階にとどまっている。

デザインウェアでのフレームワークとコンポーネントの考え方は、コンポーネント指向⁴⁾やデザインパターン⁵⁾の流れに沿ったものである。これらは、拡張性や再利用性を保証する解とされてきたオブジェクト指向が、考え方自体は受け入れられたものの、一般に、スクラッチからの設計の専門性が高くて困難である、との反省から生まれたものである。これらは、拡張する単位を明確に区切り、単位ごとに属するエンティティを明確化することにより、設計を個別に進めることを可能にするものである。オブジェクト指向の発展とはいえ、必ずしもオブジェクト指向設計を行わなくてもよいことも特徴の一つである。性能や作りやすさを考慮すると、単純な集計処理などではデータベースへの問合せ内で処理を記述するほうが有利であることなどから、必ずしもオブジェクト指向が適するものではない。このデザインウェアでは、設計手法をフレームワークやコンポーネントごとに変えていくことも可能である。

4 金融市場系システムの共通デザインウェア

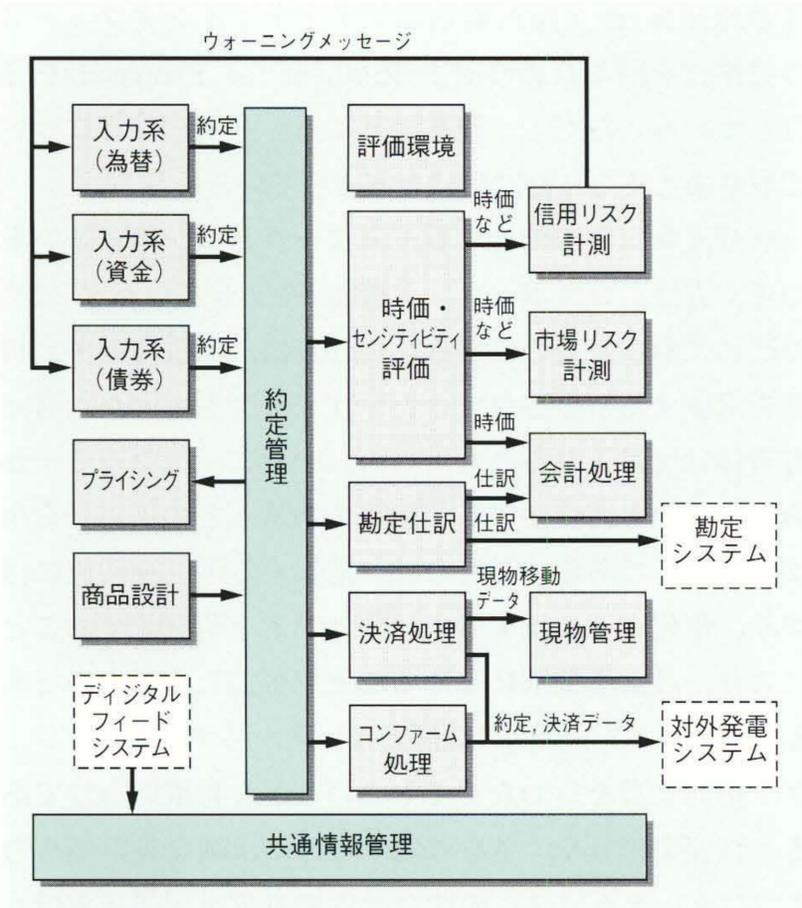
4.1 コンポーネント構成

金融市場系システムのサブシステム群であるフロント、ミドル、およびバックシステムで、どういった機能分解を実際に行うか、また、既存システムとの接続をどのようにとらえるかは、業務分析と設計の第一歩に当たる。

多くの開発で行われる機能分解では、それぞれのシステムでデータベースサーバとアプリケーションサーバを物理的に分離すると、大規模プログラムの新規開発単位(多くの場合100万ステップ以上)に影響することとなる。機能分解に際しては、業務的な考察はもちろんのこと、データやプログラムの開発単位まで考慮する必要がある。特に、金融市場系システムでは柔軟性が要求されることから、より慎重な機能分解が必要とされる。

このデザインウェアでは、効率よく、柔軟性をもって開発を進めるためには、全体の業務をどのように実現すればよいかの指針となるコンポーネント構成を例示している。これらの考慮点に配慮したコンポーネント構成例を図1に示す。

コンポーネント設計での具体的な考慮点について以下に述べる。



注：[] (外部システム)

図1 コンポーネントの構成例

コンポーネントごとにインターフェースが定義され、これを守るにより、独自の開発を拡張することができる。

4.1.1 フロントシステム

取扱商品が決まっている入力系では、商品ごとに設計が可能である。一方、商品設計を行うコンポーネントでは為替や資金など多くの種類を参照することになるため、汎用コンポーネントの作り方を考慮する必要がある。

さらに、プライシングを行うコンポーネントでは、自システム内のロジックだけでなく、ミドルなどの他システムや、遠隔地の別システム(例えば、海外拠点で運用されるもの)を取り込むことを想定しておくことが大切である。

4.1.2 ミドルシステム

市場リスクや信用リスクの計測は、商品横断的に行われる。算出のロジックは制度変更などで大きく変動することを想定し、キャッシュフローなどのデータを、管理するコンポーネントとは別にくくり出しておく。すなわち、データ管理系と計測系、および計測環境系(ロジック適用を判断するもの)に大別されるコンポーネント構成とすることを想定する。

4.1.3 バックシステム

業務の方向として、STP(Straight-Through

Processing)を実現することが求められている。これは、全体業務がワークフロー的に(事務作業を行うことなく)流れるようにすることとしてイメージされる。このためには、商品種類別にコンポーネントを構成するのではなく、業務内容ごとのコンポーネントとするのがよい。例えば、コンファーム、勘定、決済、現物管理などである。

4.2 業務処理フレームワーク

フレームワーク層では、業務を定型化した処理パターンを提供する。ビジネスエンティティが標準化されることが前提となるが、フレームワークを適用することにより、各システム(コンポーネント)で同じような処理を重複開発することなどを避けることができる。

約定の時価評価を行うフレームワークの例について以下に述べる(図2参照)。ここで、約定の時価評価は、フロントでのプライシングだけでなく、信用リスクや市場リスクの把握、時価会計処理で共通な処理である。時価評価は、以下の処理を使い分けることにより、実現するものである。

(1) マーケットプライスの参照

債券売買約定などが対象であり、マーケットプライスに取得分額面を掛け合わせたものが時価となる。

(2) キャッシュフロー割引

金利スワップなどで行われる。確定しているキャッシュフローは、別途計算される期間ごとの割引率で現在価値に引き直す。変動キャッシュフローは、金利予測したうえで、同様に計算する。

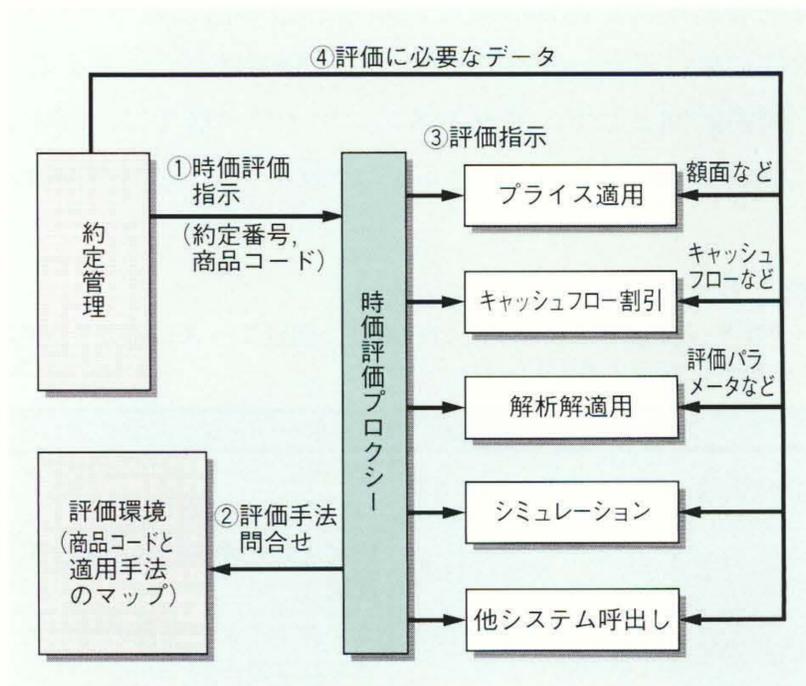


図2 約定評価フレームワーク

商品によって適用される手法が異なる場合の処理の流れを示す。時価評価プロクシーとのインターフェースを守るにより、他手法の追加や内容変更を行うことができる。

(3) 解析的な数式を適用

Cap/Floorなどの単純なオプション約定などで行われる。ブラックショールズ式などを適用して評価する。

(4) シミュレーション

複雑(エキゾチック)なオプション約定などで行われる。モンテカルロシミュレーションなど、さまざまな手法が適用される。手法に従って入力情報は変更となる。

(5) 他システム呼出し

他システムで計算された値を持って来る、または遠隔処理呼出しで相手方のロジックを使用するなど

図2に示す例では、これらの手法ごとのデータと呼出しインタフェースを共通化し、適用手法を動的に選択できるようにしている。ただし、同一商品をバッチ処理的に取り扱う場合には、性能上このフレームワークを適用する必要はない。

他のフレームワークとしては、センシティブティ評価やコンファーム、勘定仕訳などの商品ごとに手法が定義されるものが考えられる。

4.3 ビジネスエンティティ

ビジネスエンティティ層については、このデザインウェア

アでは、金融市場系業務でのビジネスエンティティとして、約定(取引内容)や顧客、有価証券銘柄、市況情報、担保などをあらかじめ設定し、これらの関係をどのように考えるかを例示している。

このうち、約定は、金融市場系システムで最も重要なエンティティである。ここでの柔軟性が、システム全体の特性として反映される。

約定を中心とするエンティティの例を図3に示す。この例では、約定は顧客などを参照する取引共通部分と、参照する汎用商品またはOTC商品で構成する。汎用商品は、債券売買取り引きや上場先物、先物オプションなど、多くの約定で参照されるエンティティを表している。OTC商品は、個別に条件設定される金利スワップや債券オプションなどを表し、個別にキャッシュフローが設定される場合もある。OTC商品は、同図中のスワップシヨンの例で示すように、さらに先の構造を持つこととなる。

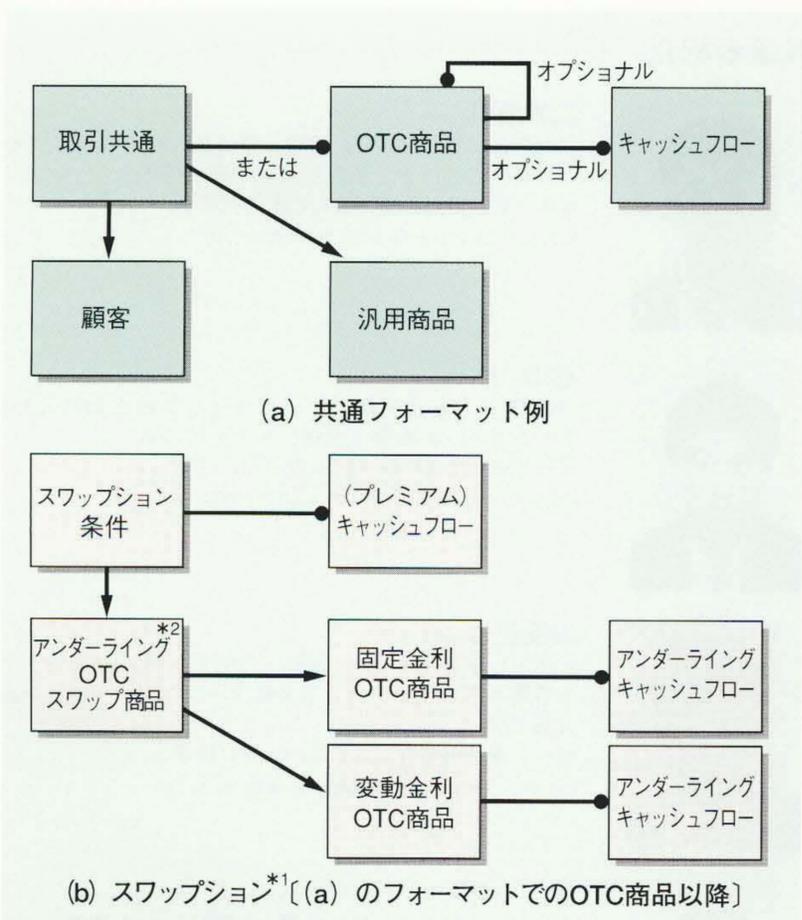
この例では約定に一つの共通フォーマットを導入しているが、ほかにも幾つかの共通フォーマットと個別商品構成が考えられる。いずれの場合でも、商品が追加された場合への対応方針として、部品をあらかじめ定義しておき、その組合せでデリバティブ商品を表現することとしている。これは、多くのデリバティブ商品で、既存の商品をオプションや先物のアンダーライングとして使用するが多いからである。

5 実開発への適用

5.1 システム構成

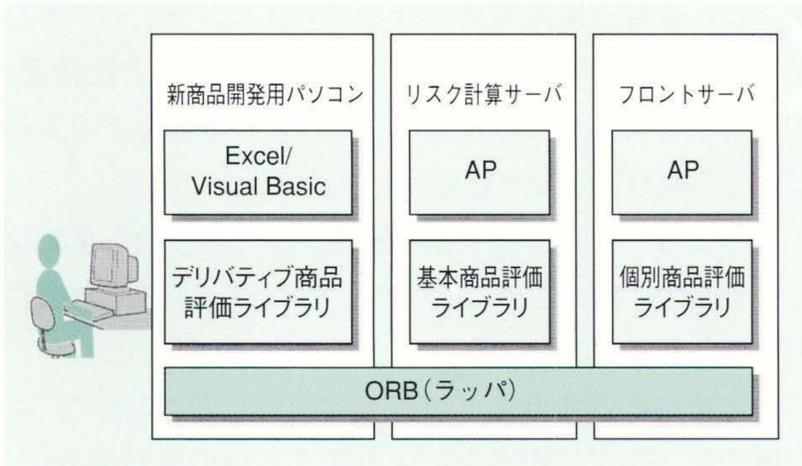
金融市場系システムには、業務を実現することと同様に、性能と耐障害性が求められる。このため、ハードウェアを二重化することや、クライアントとアプリケーション、データベースを分解した3層構造などが候補とされる。このとき、コンポーネントをハードウェアやプロセスに対応づけることや、データベースインスタンスとすることが考えられる。パッケージでは、これらのシステム構成までを柔軟に対応させることは不可能である。

フロントとミドル、バックでデータベースを共有するか、データが独立したシステムとするか、あるいは顧客や市況情報などをどのように情報共有するかも設計ポイントとなる。幾つかのシステムでデータを共有する場合、共有単位をコンポーネント化し、データインタフェースを明確に定義することが必要となる。これらの点についても、このデザインウェアでは、実装ノウハウとして対応ごとの特徴と考慮点を示している。



注：略語説明ほか OTC (Over the Counter)
 *1 金利スワップとオプションを組み合わせた金融商品
 *2 オプション実行時に発効する構造

図3 約定ビジネスエンティティの例
 デリバティブ商品を個別にエンティティ化せず、部品化して組み合わせることで汎用化する。



注：略語説明 AP (Application Program)
ORB (Object Request Broker)

図4 商品評価ロジックの共有基盤

各システムを接続し、ロジックを共有して商品評価する環境を提供する。

5.2 他システムとの接続

多くの場合、金融市場系システム全体を一気に構築することはまれであり、現行システムからの移行を考慮しながら部分的なリリースを繰り返すことになる。このとき、他システムとのデータインタフェース方式を個別に開発するのではなく、標準化することが設計ポイントとなる。

ここでは、約定などのビジネスエンティティを標準モデルとして、各システムで参照することが重要となる。データ供給側からゲートウェイシステムでのデータ標準化を経由して、受け側システムに接続することが考えられる。

さらに、金融市場系システムでは、4章でフレームワークとして例示した約定評価の例のように、他システムのロジックを呼び出すニーズも強い。このデザインウェアでは、ORBを使用した相互ロジック使用の形態などを想定している(図4参照)。

6 おわりに

ここでは、日立製作所が開発に取り組んでいる金融市場系システムのデザインウェア(設計体系)について述べた。

デザインウェアは、パッケージやクラスライブラリと比べて、適用面でより柔軟な対応を可能にする。金融市場系システムが高度で複雑な業務であることを考慮すれば、このデザインウェアによって初期設計を効率よく立ち上げ、また、そこで示される指針に従うことにより、システムの性能を、当初考えたレベルまでたどりつかせ

ることは重要である。

多くの案件で要件定義やパッケージのカスタマイズに苦勞している現状と比較して、このアプローチは、より確実性が高いと言える。

現在のところ、このデザインウェアを使用した場合の設計の効率性を定量的には計測していない。個別案件を対象として効率性を計測すること自体が難しいことから、どのようにしてデザインウェアの効果を測るかが課題となっている。

また、現在は、設計のノウハウの蓄積段階ではあるが、サービスメニュー化を進め、今後幅広く提供していく考えである。

参考文献ほか

- 1) 日経コンピュータ, 1997年3月3日号, pp.156~158
- 2) <http://www.infinity.com>
- 3) <http://www.summithq.com/>
- 4) 青山, 外: コンポーネントウェア, 共立出版(1998)
- 5) M.ファウラー: アナリシスパターン, アジソンウェスレイ(1998)
- 6) E.ガンマ, 外: デザインパターン, ソフトバンク(1995)

執筆者紹介



二木誠司

1988年日立製作所入社, 情報・通信グループ 情報システム事業本部 システム開発本部 第一部 所属
現在, 銀行市場系システムの開発に従事
E-mail: futatugi@bisd.hitachi.co.jp



石川 明

1981年日立システムエンジニアリング株式会社入社, 第一システム本部 第一金融システム部 所属
現在, 銀行市場系システムの開発に従事
E-mail: a-ishikawa@cm.hitachi-system.co.jp



山橋哲也

1986年日立製作所入社, 情報・通信グループ 情報システム事業本部 情報システム事業部 第一アプリケーション設計部 所属
現在, 銀行市場系システムの開発に従事
E-mail: yamahasi@system.hitachi.co.jp



仲 勇

1992年日立製作所入社, 情報・通信グループ 情報システム事業本部 システム開発本部 第四部 所属
現在, 銀行市場系システムの開発に従事
E-mail: i-naka@bisd.hitachi.co.jp