

オブジェクト指向による共通ミドルウェア

Generic Middleware Applying Object-Oriented Method

雨宮 隆郎
AMAMIYA Takao

金子 巧
KANEKO Takumi

野田 和生
NODA Kazuo

共通ミドルウェアとは、情報通信システムのJavaBeans^(注1)による完全なコンポーネントウェアであり、ビジュアル監視制御システムを構築するためのミドルウェアとして開発した。GUI(Graphical User Interface)ツールを利用したソフトウェアコンポーネントの組合せによるシステム構築作業時のコーディングレスを実現している。共通ミドルウェアを利用するシステム構築は、オブジェクト指向分析ともよくマッチングし、CS(顧客満足)向上に貢献する。

Generic middleware is a complete component ware for information communication systems using JavaBeans. It has been developed as a middleware to build visual supervisory control systems. This middleware provides a method to build a system by combining software components through a GUI tool without any source coding. System configuration using generic middleware matches object-oriented analysis well, and provides the benefit of enhanced customer satisfaction.

1 まえがき

製品のCS向上という課題に対し、ソフトウェアの再利用性向上による品質、生産性、開発期間短縮という生産面からのアプローチと、オブジェクト指向分析による客先ニーズの分析結果の的確な反映という仕様面からのアプローチとが考えられる。

共通ミドルウェアは、ビジュアル監視制御システムにおいて、共通基盤となる機能を含む全機能をJavaBeans規格に添ったソフトウェアコンポーネントとして開発したもので、再利用性及びカスタマイズ容易性の向上と、オブジェクト指向分析の結果のスムーズな取込みを目指したものである。システム構築に必要なすべての機能をJavaBeansとして開発したので、カスタマイズ作業を、同時に開発したシステム構築ツール(図1)のGUI操作だけで行えるという特長がある。このツール上では、共通ミドルウェアのコンポーネントを目に見える形でCASE(Computer Aided Software Engineering)ツールのように取り扱うことができる。これにより、従来にない高い再利用性と、オブジェクト指向分析との親和性を実現している。

2 共通ミドルウェアの設計コンセプト

共通ミドルウェアは、JavaBeansをベースにしたコンポーネントウェアである。これによりシステムカスタマイズ作業から、コーディング作業をいっさいなくすことと、オブジェクト指向分析との親和性を持たせることをコンセプトとしている。

(注1) Java及びその他のJavaを含む商標は、米国Sun Microsystems社の商標。

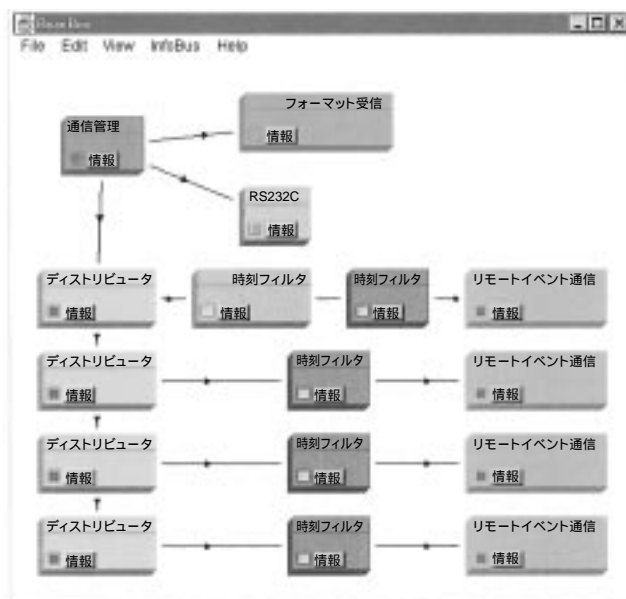


図1 共通ミドルウェア用システム構築ツール 共通ミドルウェアのソフトウェアコンポーネントを、GUI操作で組み合わせてシステム構築を行える。

System builder tool for generic middleware

従来のミドルウェアという概念は、共通的な機能を提供するソフトウェア部品群という範疇(はんちゆう)を出ないものであり、それを利用するシステムのカスタマイズ作業に、わずかであってもプログラミング言語によるコーディング作業が必要であった。また、設定テーブルによるカスタマイズ方法をとる場合もあるが、カスタマイズ可能な範囲が狭くなるという問題があった。

共通ミドルウェアは、コンポーネントウェアとして実装されており、システムカスタマイズをソフトウェア部品の組合せで実現できる。これは、カスタマイズの自由度とソフトウェアの再利用性とを同時に高めることができる。

再利用性が高くなれば、新たな機能の追加が生じた場合でも、それに関するコンポーネントを作成するだけで済むようになり、図2に示すようなソフトウェア設計プロセスの好循環を実現できる。

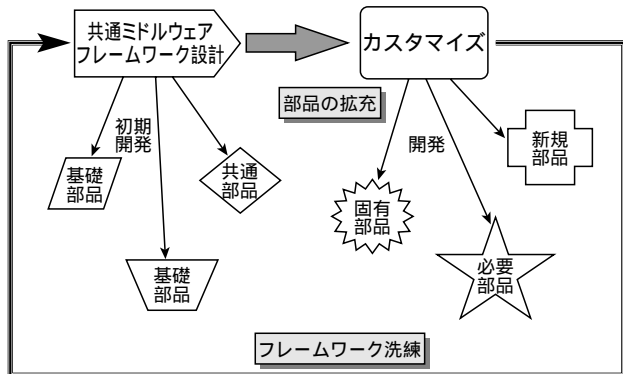


図2. 望ましい設計プロセスサイクル コンポーネントウェアは、設計プロセスに好循環をもたらす。
Desirable cycle of design process

3 共通ミドルウェアの特長

共通ミドルウェアでは、上流工程であるオブジェクト指向分析との親和性を高めるために、システムカスタマイズ作業に対してオブジェクト指向的考えである集約関係や継承関係を取り込む仕組みを考案した。これにより、オブジェクト指向分析のアウトプットを部品組立てによるカスタマイズ作業に直結することができる。

まず、集約関係の概念を取り込む仕組みについて述べる。集約関係は、大まかな機能が更に規模の小さな機能の集まりであることを表しているという関係を分析により明らかにしたものである。オブジェクト指向言語における集約関係の実装は、親にあたるオブジェクトが子にあたるオブジェクトを生成することで行われる。一方、システム構築ツールを使ってソフトウェアコンポーネントを組み合わせる作業では、オブジェクトの生成はシステム構築ツールにより行われるので、コンポーネントどうしに集約関係を持たせることができない。つまり、各Bean(ソフトウェアコンポーネントのこと)の親オブジェクトは、必ずシステム構築ツール(ビルダツール)になってしまうのである。そこで、特殊なイベントを導入することにより、コンポーネント間の集約関係の締結をイベント伝達で実現することにした。これを後天的集約関係と名づ

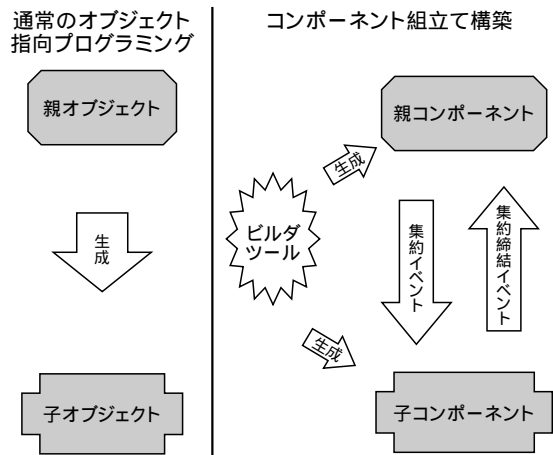


図3. 後天的集約関係 GUI操作でコンポーネント間に集約関係を作る仕組みを示す。
Acquired aggregative relationship

けた(図3)。これにより、機能の部分的な変更を、子に当たるコンポーネントの置換という作業で実装することができる。例えば、通信のプロトコルスタックにおいて、HTTP(Hyper Text Transfer Protocol)のプロトコル処理を行うBeanを親として作成し、それが用いる下位のプロトコルを行うBeanをTCP/IP(Transmission Control Protocol/Internet Protocol)用とRS232C(電気的接続規格の一種)用と2種類用意しておけば、アプリケーションレベルのプロトコル処理を行うBeanを変更することなく、下位のプロトコルBeanを付け替えるだけでRS232C上にHTTPを実装するようなカスタマイズを行うことも可能となる。

次に、継承関係の概念を取り込む仕組みについて述べる。継承関係の目的の一つとして、元になるクラスに対して機能を追加するという点がある。これを部品の組合せで実装することにした。カスタマイズの要件として、機能の追加や削除が必要と思われる処理をコンポーネント化する場合に、それらを機能分割し、その入出力となるイベントを同じクラスにする。

このようにして作られたコンポーネントは、一連の処理機能の中で、取外し自由となる(図4)。これは、連鎖応答パタ

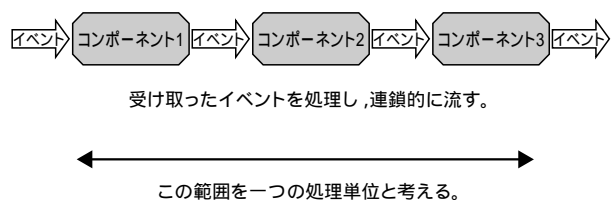


図4. 連鎖応答パターン コンポーネントの入出力が同じクラスの部品は、機能追加/削除のカスタマイズ手段を提供する。
Chain of responsibility pattern

ーンと名づけた仕組みである。この仕組みにより、実質的に継承関係がもたらすものとほぼ同等の効果を実現している。

例えば、コンポーネント2の持っている機能が不要であるシステムであれば、これを取り外し、コンポーネント1の出力のイベントを直接コンポーネント3に渡すことが可能である。また、新たな機能を追加することが必要となった場合は、コンポーネント4という新コンポーネントを作成し、コンポーネント1とコンポーネント3の間に入れることができる。こうすることで、コンポーネント1とコンポーネント3にまったく変更を加えることなく新たな機能を追加することができる。

これら、コンポーネントウェアに適した設計手法を開発、適用することにより、オブジェクト指向分析の結果をシステムカスタマイズに直結させるプロセスを実現することに成功している。

4 開発したコンポーネント

ビジュアル監視制御システムに利用するために開発したソフトウェアコンポーネントは、機能の大分類ごとに設計したフレームワーク(FW: frame work)に基づいて開発をした(図5)。

- (1) 通信FW これに属するコンポーネントは、外部装置(監視対象機器やカメラなど)との通信を行い、データを送受信する。
- (2) フォーマット別業務処理FW これに属するコンポ

ーネントは、受け取ったデータの種類ごとに適切な処理を行うコンポーネント群であり、更に、集信管理、保存編集、構成情報、データ生成、配信管理、送信編集のサブフレームワーク(SFW)に分類される。

- (3) データストレージFW これに属するコンポーネントは、データ保存を抽象的レベルで行うコンポーネント群であり、RDB(リレーショナルデータベース)や独自のストレージなど、保存形式の変更をコンポーネント交換だけで可能なように実装をしている。
- (4) 端末インタフェースFW これに属するコンポーネントは、端末装置とのデータのやり取りを行う。
- (5) GUI-FW これに属するコンポーネントは、端末画面を構成するGUIコンポーネント及びデータアクセスなどの機能を提供するコンポーネント群である。

5 共通ミドルウェアとCS

共通ミドルウェアは、自由度の高いカスタマイズ手段を実現していることから、ユーザーからの要求を満たすための柔軟な対応が可能となり、CSを高める効果がある。また、その再利用性の高さにより、高品質なソフトウェアを短納期で提供することが可能でもある。

更に、オブジェクト指向分析からシステムカスタマイズへの移行がスムーズに行えるという特長も持っている。オブジェクト指向分析作業は、CS向上のための重要な手段であり、

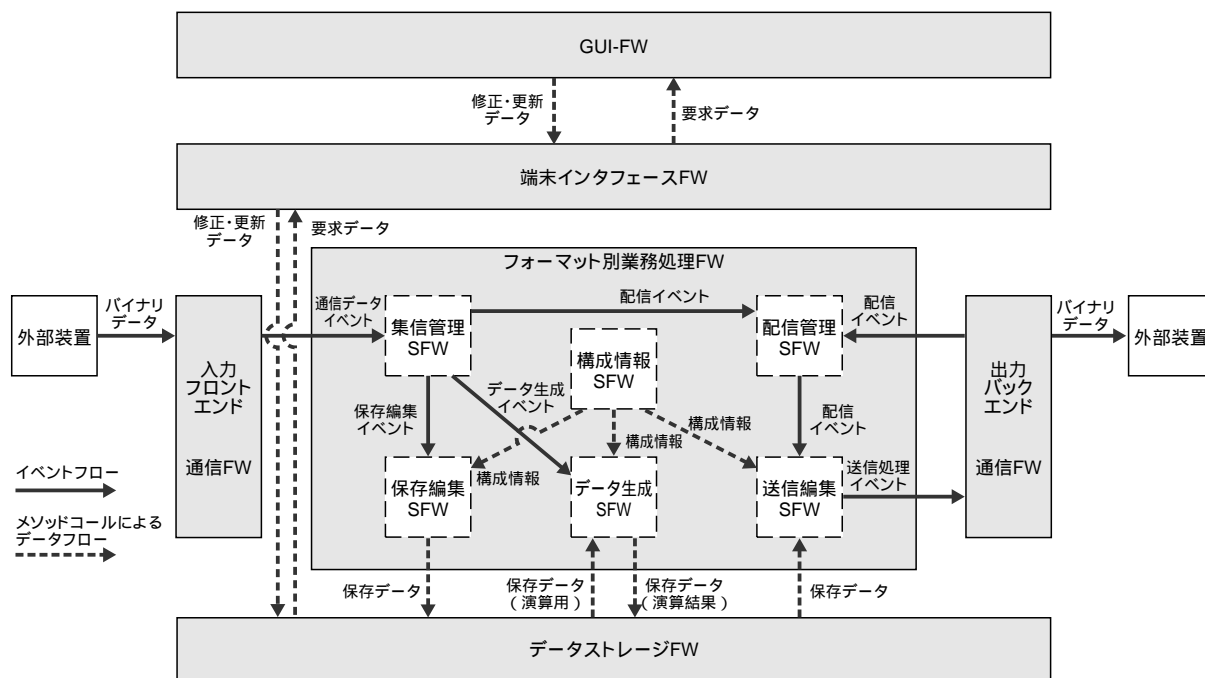


図5. コンポーネント群の分類 共通ミドルウェアを構成するソフトウェアの分類を示す。
Categories of software components

次のような作業を行う。

オブジェクト指向分析では、顧客がシステム化したいと考える問題領域を個別の要素に分解していくことにより、オブジェクトと呼ばれる“モノ”の集まりとして表現できるようにし、それら“モノ”の間の関連を定義していく。このような作業を行うことで、漠然とした問題領域を顧客やシステムエンジニア(SE)が認識できる形に整えていく(図6)。

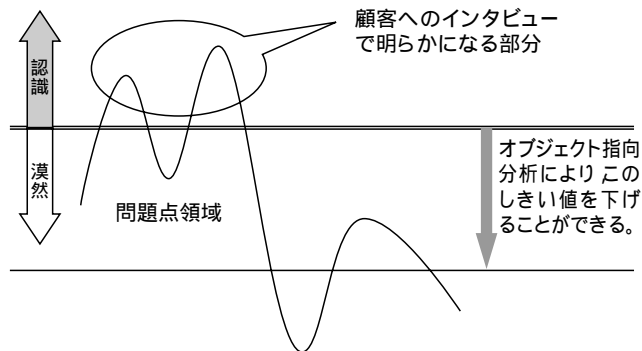


図6 . CSから見たオブジェクト指向分析の効果 良いオブジェクト指向分析はCS向上に貢献する。

Effect of object-oriented analysis from perspective of customer satisfaction

一般に、顧客自身が認識している解決すべき問題点は、個別の現象の列挙にとどまる。そのため、顧客から聞きとった問題点の解決だけを目的としたシステムでは、仕様は満足していたとしても、顧客が運用に入ると数々の不満や改善項目が発生することが多く、十分なCSを得ることができない。

オブジェクト指向分析を問題領域に対して実施すると、問題点の本質、顧客の気づいていない問題点を発見することができ、顧客に対して発展的な提案を行うことも可能となる。

すなわち、共通ミドルウェアの特長の一つである、オブジェクト指向分析の結果をスムーズにカスタマイズ作業へ反映できるという性質は、CS向上に対する有力なソリューションとして位置づけることができる。

6 あとがき

ビジュアル監視制御システムの開発に対して、再利用性の高いコンポーネントウェアである共通ミドルウェアの開発を実現でき、オブジェクト指向システム分析との親和性とあいまって、顧客の細かなニーズに柔軟に対応することが可能となった。

今後のビジネスは、顧客が明確には意識していないニーズをいかに表に引き出すかに掛かっており、引き出したニーズを共通ミドルウェアの部品群として拡充していくことが、今後のビジュアル監視制御システムの発展にとって重要な課題である。



雨宮 隆郎 AMAMIYA Takao

情報・社会システム社 日野工場 通信応用システム部主務。
通信応用システムの開発に従事。

Hino Operations - Information and Industrial Systems & Services



金子 巧 KANEKO Takumi

情報・社会システム社 日野工場 通信応用システム部参事。
通信応用システムの設計に従事。

Hino Operations - Information and Industrial Systems & Services



野田 和生 NODA Kazuo

情報・社会システム社 日野工場 通信応用システム部主幹。
通信応用システムの開発業務に従事。情報処理学会会員。

Hino Operations - Information and Industrial Systems & Services