

顧客アプリケーション構築を支援するイネーブラ

Enabler to Support Building of Customer Applications

松本幸男, 森良行

要約 オープン系システム構築では、市販ソフトウェア商品の利用が前提であり、複数のソフトウェア商品を組み合わせてその上にシステムを構築する。この為、それらの組合せ技術や利用技術が必要となる。日本ユニシスはこれらの技術をソフトウェア化することで、顧客が業務処理にのみ専念してシステム構築を目指すことができるイネーブラソフトウェアを提供している。

イネーブラソフトウェアは、市場のニーズとテクノロジーの双方向で捉えた方法論（マーケット V 手法）を利用して創出する。対象とするソリューション分野を選別し、その対象分野に応じた具体的なソリューション事例を抽出し、その機能要件に対してどの機能がイネーブラとして提供でき、どの部分に利用者専用のカスタマイズ用のエリアを埋め込むかを定義する。イネーブラでは、市販ソフトウェアが持つ機能に業務的な機能要件を付加した形で提供し、利用者は業務に則した機能として利用できる。これにより、市販ソフトウェアのバージョンアップによる非互換や組合せ技術を吸収し、利用者からのインタフェースを変更しなくても利用可能にする。

イネーブラソフトウェアによって、市販ソフトウェア商品の適用技術をソフトウェア化することで、保守性の向上やソフトウェア資産の延命化を図ることができる。

Abstract This paper discusses Enabler Software required for designing and implementing the open systems application. In open systems design methodology, it is the generally accepted approach that various ISV(independent software vendor)software products are sorted out and put together to build a complete application software on the open platform, which requires sound technological expertise to use and combine existing standard software products. Nihon Unisys has translated above technological expertise into a specific software product, and provides for customers as " Enabler Software " that allow a user to develop the application systems focusing attention on the business logic to be resolved.

Enabler Software is created using the methodology called " market V " that measures the requirement from market needs and technology for the problem solution in both directions. This software facilitates for a user to screen targeted solution categories, retrieve concrete sample solution met targeted categories, and define what features to be provided for corresponding functional requirements as the enabler and which portion of enabler to be released for customizing area for customer use only.

The Enabler Software provides the additional user application oriented features other than original features of ISV software selected, which allows users to make use of practical and effective software features met their own application requirements. By this means the loss of compatibility and the differences of software combination expertise as a result of the feature enhancement of selected ISV software products are resolved, allowing a user to run these software without changing any user interfaces.

Enabler Software translates the application technologies of ISV software products into another concrete software components, which facilitates a user to take advantage of increase of maintainability and protection of existing software assets, extending system lifecycle.

1. はじめに

1990年代のオープン化とダウンサイジングの波は、ハードウェアビジネスを基盤に固有のサービス/ソリューションソフトウェアを提供することによって収益をあげていた情報産業市場を新しいオープン市場へと変貌させた。

それまでのクローズされた市場では、製品種類の構成や製造・構築技術は固有技術をベースに確立され、開発コストや習熟曲線は予測可能であったし、買い手は実績や保証のある製品を安心して購入でき、ビジネスは計画的、長期的であった。ところが現在のオープン市場では、競争ルールも確立されず、技術や製品は不確実な要素が多く、どのような製品種類の構成が最終的にベストになるのか、どの製造・構築技術が最も効率的なのかが定かでない。業界の各社も新しく生まれて来る技術や製品を梃子に多種多様の戦略を試みている。ベストの戦略は定まらず、ベンダー各社は製品と市場のポジショニング、マーケティング、サービスなどの方法を手探りで模索している。

一方顧客にとっては、製品やサービスの初めての買い手になることが多く、信頼性や安定性、コストが重大な関心事になる。オープン市場におけるソフトウェア業界の参入障壁の低さを考えると無数の小規模企業が特定の問題を解決するソフトウェアを開発することができる。しかし、ユーザは製品間の非互換の問題と、多種多様な要求や変化に対応できない売り手からソフトウェアを買う危険は避けたい。またアプリケーションプログラムを書き直したり、要員を再教育したりする経費を削減したい。ハードウェアの低価格化に比し、オープン系プラットフォームでの開発コストは、経験もなく不慣れが故に、将来到達できるコストと比べると、一般的に高いコストになることも考えておかなければならない^[1]。このようなオープン市場における売り手や顧客にとっての問題はますます多様化していくものと考えられる。それはコンピュータ業界が位置するデジタル情報産業が、今や単一の産業というよりも融合と分離を繰り返す複数の産業の集合となってきたからである。クローズされていた市場の境界はあいまいになり、ますます複雑で多様化してくる。これからのデジタル産業ではハードウェアとソフトウェア、コンピュータと通信、業務用と家庭用、情報コンテンツと提供手段、サービスと製品、一般市場と特殊市場のような分類さえもできなくなってくるであろう。

このような複雑で多様化するオープン市場においては、売り手は単に製品やソリューションを提供するだけでなく、新しい製品やサービスの性能、機能に対する信頼性や安定性、コストといった顧客の関心事にこたえていくことがますます重要になる。

本稿では、オープン系システム構築で必須となる市販ソフトウェア商品の組合せ技術やそれらの利用技術をソフトウェア化して、顧客を本来の業務ロジックの検討・構築に専念させることを可能とするイネーブラソフトウェアについて論じる。

2. イネーブラ登場の背景

2.1 ビジネス環境の変化

1990年代にコンピュータ業界の大半の企業はその事業戦略の軸をハードウェアビジネスからサービス/ソリューションソフトウェアビジネスに転換した。すなわち、ハードウェアプロダクトに代わる新しいビジネスプラットフォームとして、生産コス

トがかからない収益の高いサービス/ソリューションソフトウェアビジネスへどう転換を図っていくかの戦略を描くことが重要となった。

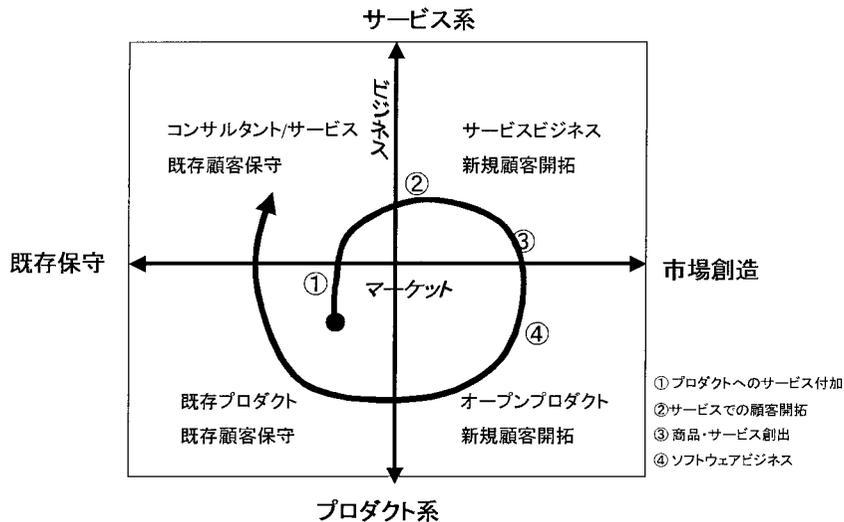


図 1 サービス/ソリューションソフトウェアビジネスへの転換

図 1 はスパイラル戦略によってサービス/ソリューションソフトウェアビジネスへの転換と拡大を目指すイノベーションモデルとして提起したものである。図 1 のスパイラルの①から④のステップを、以下に示す。

ステップ①：(プロダクトへのサービス付加) ソフトウェアの製品力を強化し、プロダクトとサービスを組み合わせることで既存市場での地位を強めるためのステップ。

ステップ①から②：(サービスビジネス転換) プロダクトを基本にしたビジネスからサービスビジネスへの転換のステップ。

ステップ②：(サービスでの顧客開拓) ソリューション構築のための付加価値サービスをビジネスとして確立する。

ステップ②から③：(新規市場開拓) コンサルタント/サービスビジネスで培ったサービス力やソリューション構築力が新たなコアコンピタンスを形成し、これを武器に新規市場を開拓する。

ステップ③：(商品・サービス創出) 現在のコアコンピタンスを動かしたり、組み合わせたりして新しい商品やサービスを創出する。

ステップ③から④：(ソリューションソフトウェアビジネスへの転換) サービスのソフトウェア化、プロダクト化によって付加価値を付け、収益性の高いソリューションソフトウェアビジネスを確立するステップである。

ステップ④：(高収益ソフトウェアビジネス) 市場性、収益性の高い付加価値ソフトウェアを大量に販売することによって高収益ソフトウェアビジネスとして確立するステップである。デファクトスタンダードソフトウェアとし

での地位を狙うための商品開発，市場戦略の実行フェーズでもある^[2]。

以上はサービス/ソリューションソフトウェアビジネスアプローチの一つのモデルである。

本稿の主題であるイネーブラはステップ③から④の分野を第一のターゲットとする商品であり，サービスのソフトウェア化によってソフトウェアに付加価値を付けると共に高品質・低コストサービスを提供する戦略ステップである。オープン市場において厳しい選択眼と評価能力を持った顧客の要求を充たすためには，単に企業がプロダクトアウトの製品を提案するというのではなく，顧客を巻き込み顧客と共同で商品を創り出すというアプローチが必要になる。その観点でこれから重要となるのは，顧客のイメージを誘発し，顧客のビジネスシーンを描けるソフトウェアやサービス商品の提案である。そのような商品を提供することによって売り手企業は顧客のニーズやその変化，商品に与えている意味や背景となるビジネスについて貴重な情報を得ることができる。

2.2 オープン化のシステム構築への影響

2.2.1 オープン系システム構築の特徴 ~ オープン系プラットフォームにおける開発 ~

情報産業市場のオープン化とサービス/ソリューションソフトウェアビジネスへの転換は，オープン系プラットフォームでのシステム構築やソフトウェア開発においても従来の方式からの転換を迫る結果になった。これは，システム構築やソフトウェアの開発方式が製品・工程のアーキテクチャに依存して異なってくるからである。このことは，製造業における製品・工程アーキテクチャと製品の開発・製造の関係で説明することができる。

製造業における製品・工程のアーキテクチャとは，どのようにして製品を構成部品や工程に分割し，そこに製品機能をどう振り分け，部品・工程間のインタフェースをどのように設計・調整するかに関する基本的な設計構想のことであり，大きくオープン型とクローズ型があると言われる。オープン型とは機能的に独立性が高く，かつインタフェースを標準化したモジュールの組合せでできた製品で，よい部品や良い工程を連結すれば高い価値を生む。他方クローズ型製品は，各部品間・工程間の相互依存性が高く，互いに微調整しながらじっくりと最適設計・連携製造することで始めて価値が生まれる^[3]。

従来のメインフレームでの独自仕様インタフェースによるシステム構築はクローズ型アーキテクチャによる開発・製造と見ることができし，複数の ISV (Independent Software Vender) ソフトウェアや IHV (Independent Hardware Vender) ハードウェアを組み合わせで行うオープン系プラットフォームでのシステム構築はオープン型アーキテクチャによる開発・製造といえる。このアーキテクチャの違いは製品・工程，インタフェースの違いのみならず，物作りに必要とされるコンピタンスなどの違いに密接に関係する。さらには企業経営や組織・人事，企業間連携においてもオープン型/クローズ型アーキテクチャの違いがあるが，本稿と直接関係しないため割愛する。表1は物作りに関連する事項における各アーキテクチャの特徴を比較したものである。

表1 クローズ型アーキテクチャとオープン型アーキテクチャの特徴

| | クローズ型アーキテクチャ | オープン型アーキテクチャ |
|---------|------------------------------|-------------------------------|
| 製品・工程 | 統合型（インテグラル型） | モジュール型 |
| インタフェース | 独自仕様インタフェース | 標準インタフェース |
| コンピタンス | 最適設計・連携製造 統合能力 日本企業が得意 | 寄せ集めの設計 アセンブリ能力 米国企業が得意 |
| 知識創造 | 暗黙知 | 形式知 |

以上の考察から、オープン系プラットフォームにおけるシステム構築では、これまでのクローズ型アーキテクチャでの構築とは異なり、オープン型アーキテクチャでの構築能力であるモジュールを統合する能力が求められる。モジュールの統合能力とは、すなわち、ソフトウェア/ハードウェアモジュールの最適アセンブリ能力、システムの機能完結能力、仕様の共通化・標準化能力およびインタフェースのシンプル化・標準化能力である。オープン系プラットフォームにおけるシステム構築では、これらの能力を組合せ技術や利用技術サービスとして顧客に提供することになる。本稿の主題であるイネーブラは、このようなサービスをソフトウェア化し、サービス/ソリューションソフトウェア商品として提供することを狙ったものである。

ソリューション構築における提供サービスは、結局これまでメインフレームメーカーのコアコンピタンスとして培われてきた統合アーキテクチャ能力をオープン型アーキテクチャの製品に対するモジュール統合能力へと組替えていくことによって競争優位の確保に繋がっていくことになる。

2.2.2 プロダクトセットとその適用範囲

プロダクトセットは、90年代初期からのUNIXを中心としたオープンビジネスやメインフレームのオープン化を契機にオープンプロダクトの組合せや連携を明示することに始まった。

「オープンプロダクトセット」は、ガートナグループのクライアントサーバシステムの分類（プレゼンテーション、アプリケーション、データベースの分類形態による5種類の分散コンピューティングモデル）に蓄積転送型モデルを付加し、形態別に個々のプロダクトの組合せを定義したものであった^[4]。

このモデル自体は、既存プロダクトの実装イメージをテクノロジーベースで分類したものであり、プロダクトの選定を容易にし、動作保証によるリスクの最小化、開発見積りの精度向上などに寄与することを目的とした。しかしながら、このプロダクトセットは、顧客の求めるソリューション構築において、充足すべきソフトウェアや開発すべきソフトウェアを明らかにするものではなかった。メインフレームについては95年のHMP（Heterogeneous Multi-Processor）の発表を契機に「HMP Clear Pathストーリー」に基づいたプロダクトセットを提供してきた。ここでは次のステップでプロダクトセットを定義し、新たなソフトウェアを開発してきた^[5]。

- 1) ネットワーク・基盤連携の進化のステップを明らかにし、その各ステップで実現されるべき情報操作系（いわゆる情報系とオフィス支援系）と基幹業務系のシ

システム像を明らかにする。

- 2) このシステム像を実現する主たる技術分野と、実現するシステム形態のマトリックスから、より具体的なシステム像を定義する。
- 3) この具体的なシステム像に対して、実現できている部分を、プロダクトセットとして定義し、実現できていない部分を開発すべき項目としていく。

従って、HMPのClear Pathストーリーは、HMPを使ってどのようなシステムをマーケットへ訴えるかというメッセージであり、同時に開発のガイドラインとしての意味を持っていた。現在、Unisys e @ction Solution コンセプトでは、情報システム基盤の技術体系と、それに基づいた自社製品および市販（ISV）製品から精選したソフトウェア製品群をBaseline プロダクトセットとして体系化している。

しかしながら、プロダクトセットによる提案・適用には、プロダクトセットからアプリケーション像が伝わらず、ユーザにとってアプリケーション的に何ができるかに応えられないという課題があった。これはプロダクトセットを定義する時に、想定しているアプリケーションシステム像が捨象されるためである。

以上の考察からプロダクトやテクノロジーをベースに、どのようなプロダクトの組合せを検証し、機能・性能を明らかにすべきかという基本動作に基づいて「ユーザの問題に対して何を答えるか」という分野への検討が必要と考えた。

通常顧客に提供するプロダクトの内容説明は、多くの場合、抽象的に機能が列記されているだけで、ユーザにはわかりにくいことが多い。商品がユーザ企業に生み出させるであろうさまざまな効果をユーザの立場から押さえて、ユーザにわかりやすい説明ができるようにすることが求められる。商品のコストではなくメリットの方に顧客の目を向けさせる努力が必要である。

3. イネーブラの位置づけと持つべき特性

Unisys e @ction Solution コンセプトでは、イネーブラ製品群を情報システム構築支援イネーブラと情報システム連携支援イネーブラに大別している。前者はパッケージ・アプリケーション短期導入支援ソフトウェアとアプリケーション短期開発支援ソフトウェアからなり、後者は制御連携支援ソフトウェアとデータ交換支援ソフトウェアからなっている。本稿では、情報システム構築支援イネーブラに焦点を当ててその位置づけと特性について説明する。

情報システムを構成するソフトウェア群を以下のように四つに分類し、その一つとしてイネーブラを位置づける。図2は、四つのソフトウェア群の役割を象徴的に示すために、四つのソフトウェア群を車と道路にアナログしたものである。

1) プラットフォームプロダクト群

この分野は、ハードウェアとオペレーティングシステム、あるいはそれ相当のソフトウェアの分野である。この分野は、エンタープライズサーバ上で実行されるすべてのコンピューティング環境の基盤になるものであり、エンタープライズサーバでは特に「メインフレーム属性を持ったプラットフォームプロダクト」を提供することが必要である。図2では、あらゆるコンピューティング環境の基盤として、この分野を「道路」にアナログしている。メインフレーム並みの平坦な

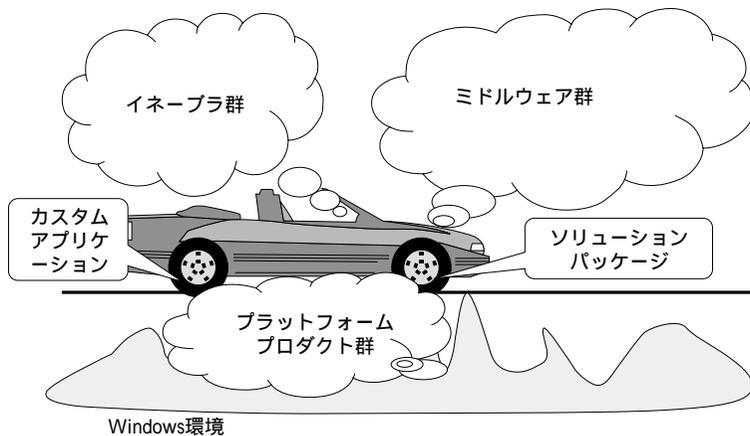


図 2 情報システムを構成する四つのプロダクト群

「道路」、すなわち、アプリケーション環境に高度な信頼性、拡張性、管理性、機密性などの機能を提供する。

2) ミドルウェアプロダクト群

いわゆるミッションクリティカル業務遂行の基盤となるデータベースやデータ通信制御ソフトウェアやメッセージブローカなどのミドルウェアソフトウェアである。図2では「車のエンジン」にアナログしている。

3) イネーブラ群

イネーブラは、ソリューション（業務アプリケーション）から見てその構成要素になっているプロダクト群の持つ機能を平準化し、機能や性能、プロダクト間の整合性を検証して提供されることによって、プロダクトの安定性や保守性を向上する。図2では、イネーブラを「ドライバ」にアナログしている。ドライバ（イネーブラ）は、平坦な道路（プラットフォームプロダクト群）と優秀なエンジン（ミドルウェアプロダクト群）を用いて、顧客のソリューションを指向するアプリケーションインフラストラクチャとして位置づけられている。

4) ソリューション・業務アプリケーション群

顧客の問題を解決するソリューション群は、図2では、「車の車輪」にアナログしている。

四つのプロダクト群に基づいて、ソリューション（業務アプリケーション）の構成を階層的に示したのが図3である。この図3を参照しながらイネーブラの持つべき属性およびその意義を以下にまとめる。

- ・イネーブラは、第一に「ソリューション」をイメージさせるものでなければならない。すなわち、ユーザがソリューションとしてのビジネスロジックに専念できることを支援するものでなければならない。そのためイネーブラはソリューションの下位層にあたるプロダクト間のインタフェースを扱う。
- ・イネーブラは、それを構成するプロダクトの、特定分野に対するソリューションとしての機能充足度の過不足を調整するものである。例えば、Exchange サーバをトランザクション処理という特定分野で適用する場合の機能を補完する

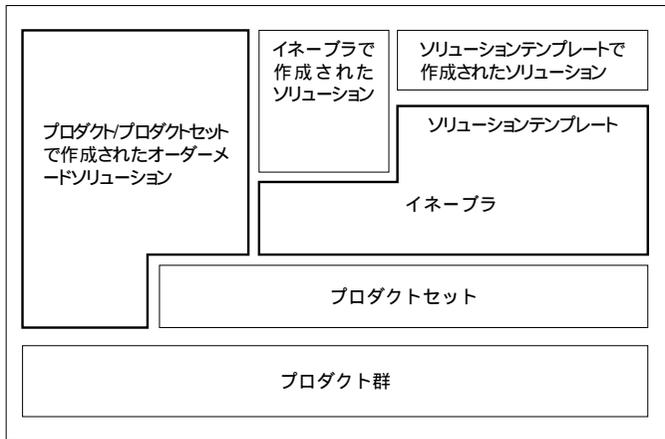


図 3 イネーブラとは？

ソフトウェア機能などがこれにあたる。

- ・イネーブラは、プロダクト間のインタフェースを扱うだけでなく、ソリューションテンプレートの機能も持たなければならない。ソリューションテンプレートは、想定するソリューションの持つべき機能（制御の流れというより、データの流れに対する機能）に対するテンプレートである（これをアプリケーションガイダンスとも呼んでいる）。具体的には、画面やデータベースアクセスに対するサンプルであったり、アプリケーションウィザードであったりする。これはデータの流れに対する機能のテンプレートであるから、カスタマイズによってソリューションの構築が容易になる。

4. イネーブラの創出（マーケットV手法）

ここでは、イネーブラを、どのようにして生み出していくかという課題に対する我々の試みを紹介する。情報技術（IT）の潮流と顧客のニーズを見通し、両者の間を結び付けて新しいイネーブラを生み出す方法論の紹介である。

この方法論は、顧客ニーズとテクノロジーをさまざまな素材から調査・分析し、最新のITとプロダクトの組合せで開発すべきイネーブラ候補を生み出すことを目的とする。この「さまざまな素材」を入力に、「方法論」を通して、イネーブラを生み出していくのである。イネーブラが生み出されると、入力となった素材は、そのイネーブラのもとに再位置づけされる。この再位置づけされた情報は、開発するイネーブラのソリューションイメージを判り易く説明するための情報に役立てることができるし、「ソリューションテンプレート」として活用できるものになる。

4.1 イネーブラ創出の基本構造

この章では、イネーブラ創出のアプローチを紹介するのに先立って、そこで使う概念・用語などを定義する。

1) Abellの戦略事業定義

イネーブラを生み出すための方法論を考えるきっかけが、エーベル（Abell）という経営学者の3軸3平面であった。もともとこの3軸3平面は、事業ドメイ

ンを定義するために考えられたものであり、新規事業を起こす時どの軸に戦略性をもって事業化するかに用いられる。

我々の方法論は、エーベルの3軸を、WHO(主要顧客はだれか)、WHAT(どういった機能を提供するか)、HOW(どういう技術で実現するか)を表わすものと捉え、3軸であるWHO、WHAT、HOWに軸間の動きとして関連性を持たせることにより、最終的に顧客にソリューションを提供するために必要となる機能、技術を关系的、連想的に明らかにしていく方法論である。軸間の関連性を図4のように定義する。

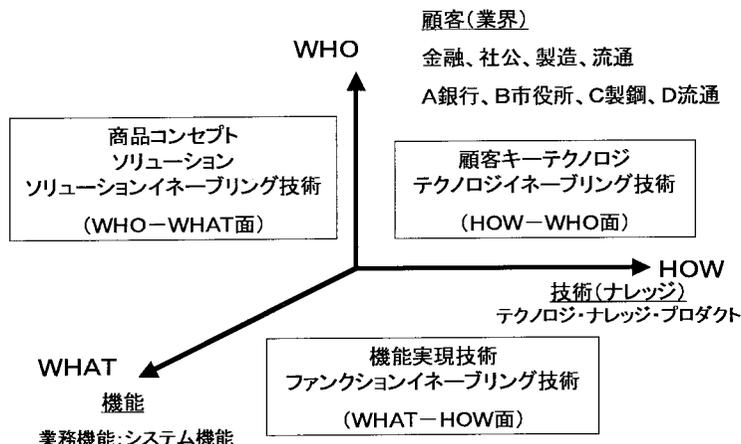


図4 イネープリング検討における3軸3平面

2) 3軸3平面の意味付け

3軸は、隣接する二つの面を持つことから、その解釈は、それぞれ2通り有り得る。ここではイネーブラに直接的に関わるWHAT軸に関してその2面性について定義する。

WHAT(“機能”)軸は二つの面を有する。WHO側の機能が、ソリューションから定義される「業務機能」(WHO WHAT)で、HOW側の機能がプロダクトやテクノロジーから定義される「システム機能」(HOW WHAT)である。業務機能を抽象化し、システム機能にマッピングすることを洗練(リファインメント)と呼ぶ。またリファインメントしたシステム機能から業務機能を再度位置づけ直し、イネーブラを生み出し、ソリューションテンプレートを創り出す。この両方向への転換技術が、イネーブラを生み出すコア技術である。

次に3平面の意味付けについて説明する。WHO WHAT面は、「どのような顧客にどのような機能を提供するか」という商品コンセプト/ソリューションを表わす面と言える。これを実現する技術を、「ソリューションイネープリング技術」と呼ぶ。ソリューションイネープリング技術をソフトウェア化したものが、「イネーブラ」である。その上に、特定のソリューション業務機能を付加して、ソリューションそのものが生まれる。従って、イネーブラとソリューションが、この面に位置づけられる。本稿ではこの面に位置づけられるイネーブラをテンプレ

レート型イネーブラと呼ぶ。SCM ソリューションのイネーブラに位置づけられる当社製品の Extranet Builder の汎用情報検索テンプレートや発注テンプレート、グループウェアソリューションのイネーブラに位置づけられる IntraBuilder/InfoShare などがこれに属する。

WHAT HOW 面は、「どのような機能がどのような技術によって実現されるか」という機能実現技術（「ファンクションイネープリング技術」）を表す面と言える。このファンクションイネープリング技術を使って定義されるソフトウェアには二つのタイプがある。一つは HOW 軸に近いところに位置づけられるプロダクトセットであり、他は WHAT 軸に近いところに位置づけられるイネーブラである。本稿ではこの面に位置づけられるイネーブラをミドルウェア型イネーブラと呼ぶ。当社製品の DRB、WebASPlus、IntraBuilder の個人情報サービスコンポーネント群がこれに位置づけられるほか、オープン系の基幹系や情報系システムにおいてメインフレームと同じ機能要件で実装形態の異なるミドルウェアのソフトウェアなどがこの範疇のイネーブラである。あるいは、マルチメディアデータの取り扱いに対応して提供される入出力インタフェースなどは、新しい情報技術によって格納、取出し、参照が支援されるミドルウェア型イネーブラの範疇に属する。

ファンクションイネープリング技術の HOW 軸に近いのが「プロダクト利用技術」であり、WHAT 軸に近いのが、「システム化技術」と言える。

HOW WHO 面は、「どのような技術がどのような顧客/ソリューションのキーテクノロジーになるか」という顧客キーテクノロジー面といえる。この面は、あるソリューションを実現するのに必要な革新的な技術、時代を超える技術を生み出す面である。「必要は発明の母」というべき新たな技術を生み出すという意味で、「テクノロジーイネープリング技術」を表す面である。ここで生み出された技術は、WHAT 軸を通して、WHO 軸まで展開されることによって、その技術のビジネス的な価値が評価される。

4.2 イネーブラ創出のアプローチ

イネーブラを生み出すためのアプローチ（手順）について説明する。イネーブラを生み出すアプローチには2通りある。顧客のニーズを起点として生み出す方式のニーズプルアプローチと、テクノロジーを起点として生み出す方式のテクノロジープッシュのアプローチである（プル、プッシュの使い分けはテクノロジーを提供する立場からみている）。

1) ニーズプルアプローチ（反時計回りアプローチ）

ニーズプルアプローチは大きく以下の3段階の連想で生み出していく（図5）。このアプローチは、ソリューションから出発して、それを実現している業務機能、システム機能、さらにそこで使われるテクノロジーやプロダクトまで明らかにし（下向分析の過程）、次に新たなテクノロジーやプロダクトを付加することによって、新たなシステム機能、業務機能、イネーブラやソリューションを生み出していく（上向展開の過程）というものである。この二つの過程の方向が「V」字をしていることから「マーケットV手法」と呼ぶ。

第1段階 (WHO WHAT): この顧客 (業界) が必要としているソリューションや機能はなにか

第2段階 (WHAT HOW): この機能はどのような技術で実現できるか

第3段階 (HOW WHO): この技術はどのような顧客 (業界) に展開できるか

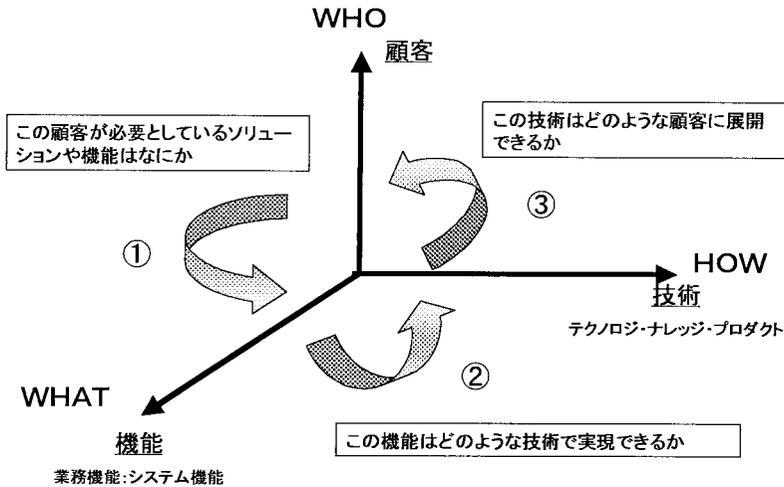


図5 ニーズプルアプローチ (反時計回りのアプローチ)

具体的には以下のステップを進める。

- フェーズ1 業種・業界・顧客ニーズを調査・分析・選定し、業務機能を明確化する
- フェーズ2 イネーブラ候補の業務機能を抽象化し、マッピングできるシステム機能を明らかにする
- フェーズ3 システム機能を実現する基幹技術、実装技術候補を選定し、全体像を明らかにする
- フェーズ4 新技术を付加することによって連想的に新たなシステム機能・業務機能・ソリューションを生み出す

2) テクノロジプッシュアプローチ (時計回りアプローチ)

テクノロジプッシュアプローチは、大きく以下の3段階の連想で生み出していく (図6)。このアプローチは、テクノロジーやプロダクトから出発して、システム機能、業務機能、さらにイネーブラやソリューションを明らかにし (上向展開)、次にそのソリューションを革新的に変える新たなテクノロジーを連想 (下向分析) し、再度上向展開のサイクルにまわす (いわばマーケット「逆V」手法) ものである。

第1段階 (HOW WHAT): この技術はどのような機能に適用できるか

第2段階 (WHAT WHO): この機能はどのようなソリューションや顧客/業界に適用できるか

第3段階 (WHO HOW): この顧客のソリューションはどのような技術で実現できるか

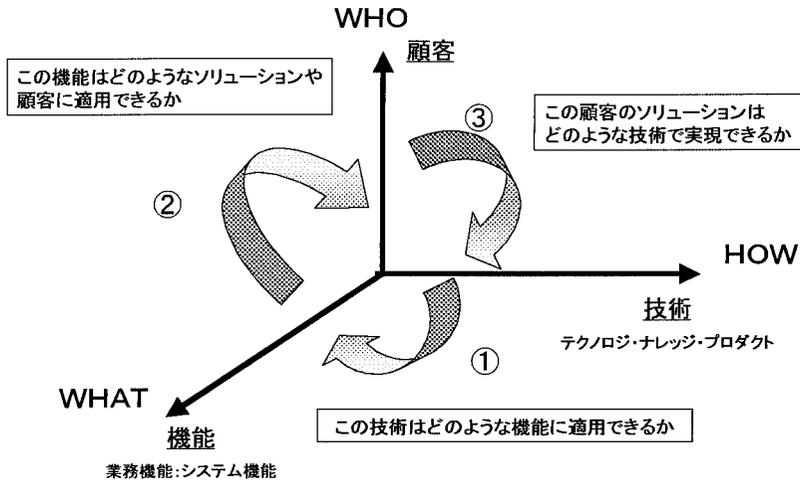


図 6 テクノジブッシュアップアプローチ (時計回りのアプローチ)

具体的には以下のステップで進める。

- フェーズ 1 特定技術とその可能性・有効性を調査・研究・整理する
- フェーズ 2 選定技術で実現できる機能と応用分野を検討しシステム機能を定義する
- フェーズ 3 システム機能をマッピングできる業務機能/ソリューション候補を選出する
- フェーズ 4 選出されたソリューションを実現する新しいテクノロジーや組合せを連想的に検討する

3) ニーズプルアプローチを用いた実践的なイネーブラの創出作業

我々が採ったニーズプルアプローチの実践的な方法について以下に説明する。

① 作業ステップ 1: ソリューション事例の収集と分解, 付加

この作業は、最近の先進的な事例を収集することから始める。この事例を、ニーズプルアプローチでの手順に従って「顧客・ソリューション (WHO)」、それを実現している「業務機能 (WHAT)」、それを実装している「テクノロジー・プロダクト (HOW)」に分解する。これが、「マーケット V 手法」における下向分析過程にあたる。ここで得られるテクノロジーやプロダクトは、極めて不完全なものである。特に、雑誌などからの事例では、この分野を明確に定義することはほとんど出来ないが、おおよその見当はつく。この作業ステップではそれで充分である。

事例を上記のように分解した後、各事例から分解されたテクノロジーや業務機能に対して連想されるものを付加し、新たなソリューション候補を挙げる。これが「マーケット V 手法」における上向展開過程である。ここでは、5人程度のグループから成る集中討議を複数グループで行い、整理していく合宿形式が望ましい。

② 作業ステップ 2: 業務機能の分類・汎用化

個々の事例から作業ステップ 1 で整理された情報に基づいて、業務機能の観

点から類似するものをまとめ、分類するのがこのステップである。この分類作業は、二つのステップから成る。最初のステップは、似たようなソリューションをベースに、業務機能を集めることであり、次のステップは、集められた業務機能の共通化（汎用化）のステップである。

業務機能の共通化・汎用化のステップでは、いわゆる業務アプリケーションと接点を持った「利用技術・システム化技術」が必要になる。すなわち、アーキテクレベルの技術・見識・知識・知恵である。我々が、イネーブラを、「サービスをソフトウェア化したものである」と主張する理由は、ここにある。このステップは、あるレベルの情報を、別の質の情報に飛躍させるステップであるので、アーキテクトと呼ばれるシステムエンジニア（SE）の創造性に依存する。

業務機能の共通化・汎用化は、ソリューションテンプレートを生み出す過程でもある。ある特定のソリューションに依存した業務機能を抽象化するわけであるから、ソリューション固有の業務機能が捨象される。この捨象された業務機能は、新たに生み出されるイネーブラのもとで、ソリューションテンプレートとして再構築（再位置づけ）される。従って、イネーブラ、ソリューションテンプレートが、ユーザのソリューションを、まさに、「イネープリング」することが可能となる。

③ 作業ステップ3：業務機能からシステム機能へのリファインメント

このステップは、共通化・汎用化された業務機能から、ある程度、実装をイメージしながらシステム機能に転換させる（リファインメントする）。ここでのアウトプットは、ラフなシステム実装図である。この実装図が、イネーブラの最小単位となる。どの程度の粒度でイネーブラをマーケットに出すかによって、この実装図を組み合わせることもある。

同時に、このステップで、ソリューションテンプレートを定義する。このソリューションテンプレートは、このステップで定義されたイネーブラから捉え返された、作業ステップ2で捨象した業務機能の再実装である。

④ 作業ステップ4：テクノロジー・プロダクトの定義

システム機能、およびラフな実装図が出来上がれば、すでにテクノロジーやプロダクトの領域に入っているためその後は比較的簡単である。ここでは、システム機能を実装可能なテクノロジー・プロダクトから明らかにする。このステップで充足技術や充足プロダクトおよび不足技術・不足プロダクトが見えてくる。この「不足技術・不足プロダクト」は、「プロダクトやプロダクトセット」での新たな開発項目に結びつく。

4.3 マーケットV手法実践の評価

ニーズプルやテクノロジープッシュのアプローチに対するこれまでの試行の結果、イネーブラを生み出すための幾つかの実践的な教訓を得た。

まず、我々は、基本的にニーズプルアプローチを採用した。何故なら、我々自身が、テクノロジーやプロダクトをベースとした集団であり、テクノロジーからソリューションに上向展開する時に想定するソリューション像を思い描くことを得意としていないか

らである。テクノロジーブッシュアプローチで、テクノロジーをテクノロジーの観点から整理することはできる。しかし、その後のフェーズ2,3のステップには、なかなか進めない。

ソリューションを思い描くことが出来ない中で、ニーズプルアプローチでのフェーズ1のステップ、すなわち、「ニーズを捉え、業務機能を明確にすることがどうすれば可能になるか？」が、次の問題になる。このステップに対して、膨大な数の(比較的最新版で先進的と思われる)事例を検討することで、このステップを遂行した。その結果、先進的なソリューションニーズが洗い出せた。

このニーズプルアプローチを採ったとしても、イネーブラが自然に連想されてくるわけではない。そこでは、「アーキテクト」レベルの知識と知恵が必要となる。一定の方法論と、人間の知識・知恵を組み合わせないとイネーブラは生み出せないということである。

5. イネーブラの適用形態

5.1 Windows 環境におけるシステムの特徴

イネーブラ創出の対象としたプラットフォームは、オープン系のシステムであるが、その中でもサーバシステムとして、これから業務領域で拡がりが見込まれる Windows システムを対象とした。この Windows システムは、Microsoft 社の WindowsNT Server や Windows 2000 Server といったサーバ用の OS 上で稼働するシステムを指す。Windows システムは、メールサーバとか Web サーバといった単機能サーバとして利用したり、従来のメインフレームのような一つのサーバ上で複数の処理をさせるシステム形態で利用したりする。これまでの WindowsNT Server では前者の形態が多いが、Windows 2000 Server では、後者のシステム形態が増加することが予想される。

Microsoft 社は、アプリケーションを利用者とのインタフェースをするプレゼンテーション層、業務処理を行うビジネスロジック層、データを格納するデータストア層に分割して、各層にサーバを割り当て、独立した処理形態で稼働させるアーキテクチャである Windows DNA (Distributed interNet Application architecture) を提唱している。このアーキテクチャは、Microsoft 社のコンポーネントモデルである COM (Component Object Model) を使用することが前提となり、各層に配置された COM コンポーネントの並列性や冗長性を上げることで、システムの可用性や拡張性を向上させることができる。Microsoft 社や他の ISV は、このアーキテクチャに則したシステム構築が容易にできるようにソフトウェア製品の機能充実を図っている。それ故、Windows に特化したシステムの開発であれば、このシステム形態に沿って COM コンポーネントをベースにしたアプリケーション構築を行うことが多い。

Windows のサーバ環境では、メールサーバや Web サーバといった機能サーバのソフトウェアやアプリケーションシステム構築で必要となるミドルウェアなどに ISV が提供するソフトウェアを利用することがほとんどである。ミドルウェアとしては、以下のようなソフトウェアが考えられる。

- ① トランザクション制御システム

- ② データベース管理システム
- ③ サーバ障害やシステム障害に対応するクラスタリングシステム
- ④ ネットワーク負荷分散システム
- ⑤ 運用管理システムを構築するための基盤システム

機能サーバやミドルウェアといった基盤ソフトウェアは、Windows OS に同梱されていたり、ISV ソフトウェアとして提供されていたりするので、利用者はこれらの中から選択してシステム構築することになる。メインフレームであれば、このような基盤ソフトウェアは、ハードウェアベンダーが OS と共に提供してきたものであり、ソフトウェアの利用方法や組合せ方法まで検証済みの状態で提供している。この為、ソフトウェアを利用する場合には、ある程度の利用基準の下で使用することができたし、誤った使用方法を事前を知ることも可能であった。

しかしながら、オープン系の世界では、ソフトウェアの利用方法や検証などは利用者の責任の下で行うことが前提になる。この為、Windows システムで使用する基盤ソフトウェアも同様で、多種多様なソフトウェアが市場に提供されているので、どれを選択するかはシステム開発時に検討して使用する必要がある。

例えば、情報共有システムの構築では、図 7 にあるように Windows OS に同梱されたソフトウェアと Microsoft 社の Back Office 製品を組み合わせたシステム構成が考えられる。この構成では、Web サーバとしての IIS (Internet Information Server)、トランザクション制御用として MTS (Microsoft Transaction Server)、Back office 製品に含まれるデータベースシステムの SQL Server、メールサーバである MS Exchange を組み合わせている。さらに、システム障害に対応する為にクラスタリングとして MSCS (Microsoft Clustering Server) も利用することが考えられる。

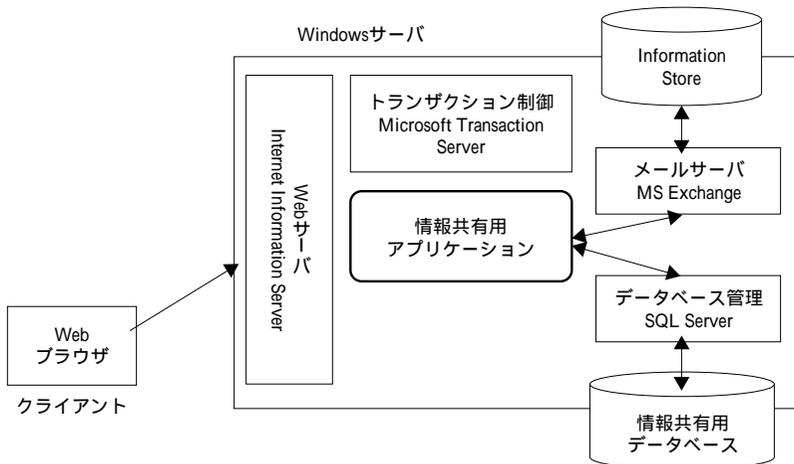


図 7 Windows ソフトウェア構成例

利用者のシステムでは、この例にある Web サーバ (IIS)、トランザクション制御 (MTS)、メールサーバ (MS Exchange)、データベース管理 (SQL Server) のソフトウェアのいずれかを他の ISV ソフトウェアに置き換えたソフトウェア構成であっ

たり、データベース管理やメールサーバを別のサーバに切り出したシステム構成であったりする。このような選択を考えると、利用する基盤ソフトウェアやサーバ構成の組合せは、多種多様に存在し、どの組合せが利用可能かどうかの判断は難しい。この為、基盤ソフトウェア間の相性を確かめた上で利用することになる。ここで云うソフトウェアの相性とは、ソフトウェア間での連携の面だけではなく、ソフトウェアが導入されることで相互の実行に支障を与えることがないか、同時に実行することでの不具合がないかなどがあげられる。このような相互干渉による影響は、実際に導入して確認するしか方法がないので、実績のある組合せを選択することになる。

イネーブラは、ソフトウェア選択の指針として、まず OS 提供ベンダーのソフトウェアを利用することとし、機能不足や未提供の機能についてのみ他のソフトウェアを検討することにしていく。理由としては、多くのシステム開発で同様の選択が取られていて、今後もこのような選択肢をとることが多いと予想され、そのシステムにイネーブラを組み込むことが可能になるからである。

5.2 イネーブラが提供する機能範囲

イネーブラが提供する機能範囲として、ISV ソフトウェアをシステム構築の要件レベルに適用する機能分野と、ISV ソフトウェアを補完する機能分野の二つの範疇に分けられる。

オープン系プラットフォーム上のシステムを構築する場合、対象となる ISV の基盤ソフトウェアに関する適用機能や利用方法とそれを使用した類似システムの事例などの情報収集や情報の検討が必須となる。入手した情報を元にしたプロトタイプを実施し、システム構築上の問題点を除去することも必要となる。このような作業自体を実際に行うかは、システム構築の規模や、システム要員のスキルレベルなどに依存する。いずれにしても、必要な機能テストやソフトウェアの相性レベルの作業は、システム設計時に行うことになる。イネーブラは、この作業を軽減するためにファンクションレベルの機能を提供する部分をシステム要件レベルの機能に纏め上げて提供することを狙っている。

システム要件レベルに合わせるために、イネーブラ製品は、対象とするソリューション分野を限定し、その分野に特化した形態で機能提供を行っている。この為、イネーブラ製品が対象とする基盤ソフトウェアであれば、その機能レベルや組合せの結果などは、イネーブラとして保証しているので、ファンクションレベルでの検証は不要となり、イネーブラ機能に対するシステム要件レベルでの検証を実施すればよい。逆に、イネーブラを利用しない場合には、開発で利用する基盤ソフトウェアを使いこなすスキルを持たなければシステム開発は行えない。

また、ISV ソフトウェアをシステムに適用する場合、同梱されているサンプルと類似した利用形態であれば、対象とする機能の利用方法は理解できるし、システム構築する上での参考にもなる。しかし、利用方法が異なったり、機能要求が高かったりした場合に、どのような利用方法で実現できるか、どこまでが機能として提供されているのかを調べる必要がある。例えば、利用者用のアカウントを生成する API (Application Program Interface) の場合、プログラムで利用者のアカウントを纏めて作成するサンプルが提供されている。しかし、既存アカウントを削除したり、修正したり

するサンプルがないので、これらの機能の有無や利用方法と利用する上での条件などを調査することになる。結果として、システム要件に提供機能が適用できなければ、代替機能を検討しなければならない。

この様に実際のシステム開発では、システム要件に合わせて、対象の基盤ソフトウェアが提供している基本的な機能の適用方法を考えたり、応用動作を推測して適用範囲を広げたりすることになる。このような適用するための技術は、対象分野毎に異なる部分と共通化できる部分がある。共通化できる部分については、そのシステム要件を明確にして、基盤ソフトウェアの上にソフトウェア機能として作成すれば、同様のシステムを構築する場合に利用でき、横展開可能なサービスとして形成することができる。このソフトウェア機能をイネープラとして提供している。

例えば、イネープラ製品には、Web を利用した情報共有システムの構築を支援するものがある。このシステムで末端利用者のセキュリティ機能を考えた場合、個人認証のレベルは Windows の基本機能として利用できるので不要のように考えるが、実際の業務レベルで考えると業務の機密保護は、多くの場合、所属している部署や役職権限といった組織レベルの権限と同期して利用することになる。業務アプリケーションは、どの部署のどの権限レベルで利用できるかといった組織と個人認証を結び付けて管理するソフトウェアや、その権限を判断するコンポーネントなどが必要となる。このような機能を実現するためには、Windows の認証機能を組織管理と結び付けて業務レベルの機密保護を作りこみ、個人管理部分とのインタフェースを提供することになる。これらの処理は、組織と業務処理とを関連させた個人認証という枠の中で一般化することができる。イネープラは、こうした枠を設定して対象となる処理を一般化して再利用できる形態にすることで、業務処理システムへ組み込み可能なインタフェースと管理系のユーティリティを提供できる。

ISV ソフトウェアを補完する機能分野では、以下のような課題を解消している。ISV ソフトウェアは、そのソフトウェアの位置づけや今後の提供機能も含めたシステムの適用形態などを盛り込んだホワイトペーパー（機能白書）と実際に提供されている機能とのギャップがあり、どの機能をいつ出すかのロードマップ的な資料は提供されない。しかしながら、システム構築は、そのソフトウェア機能の充足を待つことはできないので、不足した機能はすべて作りこむ必要があり、その機能が将来提供された時に対応方法を考えることになる。このような不足機能は、対象とする業務処理の中に組み込む場合と業務処理とのインタフェースを保持しながら外付けで組み込む形態が考えられる。この選択は、実際の業務要件によって判断をすることになる。しかし、多くの場合、基盤ソフトウェアが本来持つべき機能であれば、業務処理から切り出して利用できる機能として見ることができ、業務処理内に作りこむ必要がある機能は基盤ソフトウェアとしての機能ではなく業務処理の補完機能と考えられる。

このような不足機能を補完する場合、イネープラは、業務処理から外付けで利用できる機能部分を提供することになる。特に、機能として ISV が提供している部分や今後提供が予定されている機能については、そのまま利用するが必ずしも、利用者プログラムから見れば直接 ISV ソフトウェアとインタフェースするのではなく、薄皮となっているイネープラインタフェースを介して利用する。これが今後のバージョン

非互換に対応できる形態である。この意味は、単に同じ機能を中間的に介在した形態で提供するのではない。利用者は、業務処理的に見た機能として、ISV ミドルウェアの一機能を利用しようとするので、この業務処理的に必要とされる機能をアプリケーションインタフェースとして ISV ソフトウェアとのインタフェースを提供する。

5.3 イネーブラの提供形態

イネーブラとしての提供機能は、図 8 にあるようなアプリケーションインフラ、アプリケーション共通サービス、アプリケーションガイダンスの三つの分野に範疇分けされる。

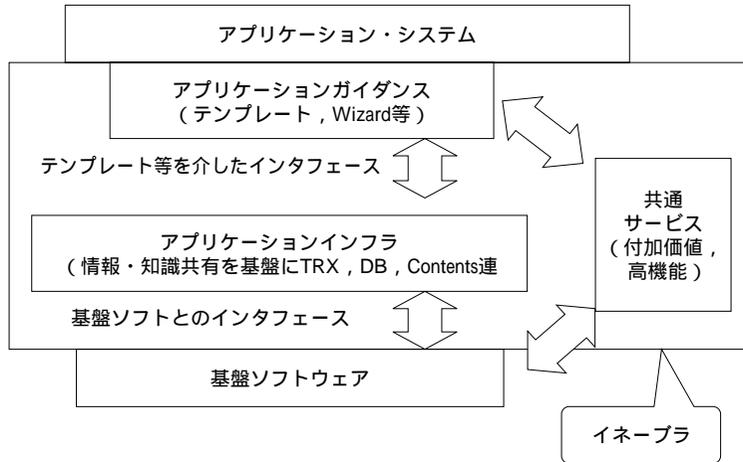


図 8 イネーブラの三つの範疇

アプリケーションインフラは、イネーブラが対象とするシステムで利用する基盤ソフトウェアとインタフェースを行い、システムとして必要となるミドルウェア的な基本機能を提供する。例えば、情報共有や知識共有を行う為に基盤ソフトウェアに MS Exchange や SQL Server を利用し、その上に情報共有基盤としてのアプリケーションインフラを提供している。また、この基盤上にミドルウェア的な機能として、業務処理のオンラインランザクションシステムやデータベースとの連携や帳票出力などの情報として形成されたコンテンツとの連携を可能にするアプリケーションインフラが考えられている。

アプリケーション共通サービスは、基盤ソフトウェアを業務処理で使用する上でインタフェースを簡略化して付加価値機能をつけたり、不足した機能を補完したりするコンポーネントを提供していて、業務プログラムから直接このコンポーネントを利用できるインタフェースを提供している。例えば、分散環境におけるコンポーネントの流量制御とか、利用者レベルでアプリケーションの利用可否を制御する機能といった単機能的なアプリケーションインタフェースを提供している。

アプリケーションガイダンスは、アプリケーションインフラや共通サービスで提供されている機能やコンポーネントを利用してシステム構築を行う為の支援的な役割を持つ。この部分では、適用方法を事例的なサンプルプログラムとして提供したり、対

象システムの処理を形式化したテンプレートを用いて質問形式（Wizard形式）でプログラムやスクリプトを生成する仕組みを提供したりする。

例えば、エクストラネット用イネーブラとして、取引先からの発注データを Web アプリケーションとして実現する受発注テンプレートがある。このテンプレートを利用すると、Wizard形式で Web による商品一覧表示を商品データベースと連動して表示内容の設定を行うことができるし、発注商品を確定するまで一時的に退避させておく発注用カートを使って発注データを収集する発注入力画面の設定を行うことができる。これらの設定を行うことで、イネーブラは、処理を制御する基本ロジックにカスタマイズ用のパラメタを付加して、あらかじめ用意されているロジックを組み込んで Web アプリケーションを生成する。

このようなアプリケーションガイダンスからアプリケーションを生成する仕組みの中では、COM ベースのコンポーネントを多用している。このコンポーネントは、アプリケーション共通サービスの機能コンポーネントとして提供されるが、基盤ソフトウェアが持つ関数的な機能というよりは、業務的な機能をもったコンポーネントとして提供する。関数的な機能であれば、直接基盤ソフトウェアが持つコンポーネントを呼び出すのと変わらないし、そのレベルの機能ではバージョンアップに対応した互換性は保持することは難しい。仮にできるのであれば、基盤ソフトウェアとして提供されるはずである。

この機能コンポーネントは、基盤ソフトウェアが持つ関数的な機能に業務的な意味付けを付加して、業務処理に応じたインタフェースを持つことで、ソフトウェア商品が持つ機能的な面を隠蔽している。

例えば、図9にあるようにデータベースのインタフェースを提供するのであれば Microsoft 社の ADO (Advanced Data Object) があるが、イネーブラは、顧客の取引状態や取り扱う商品を参照したり、更新したりする業務処理に対するインタフェー

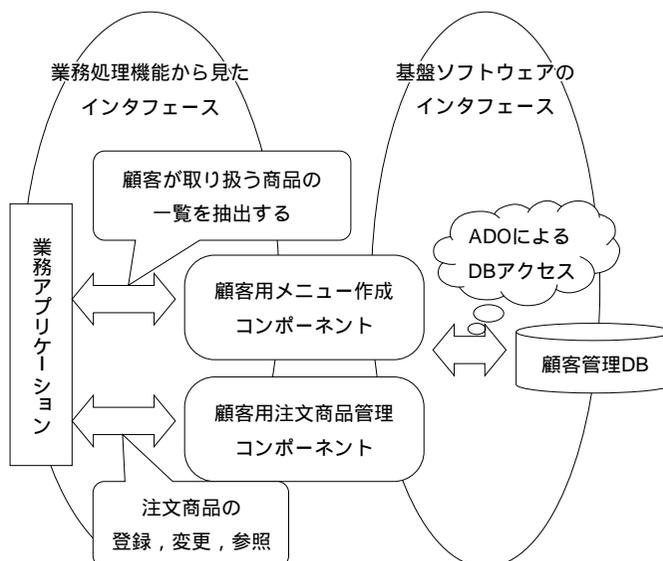


図9 業務要素のインタフェース例

スを提供する。このような場合、ISV ソフトウェアが持つ機能というよりも業務面で必要とする機能が提供できれば、そのコンポーネント内で行う基盤ソフトウェアとのインタフェースに関しては、利用者の関知するところではない。さらに、提供する機能レベルとして、効率性や耐障害性を一定レベル以上に保証できれば、利用者は提供機能にのみ関心を持つことになる。イネーブラは、このレベルでのインタフェースを実現することを目指している。すなわち、基盤ソフトウェアの機能を意識しなくても、イネーブラが提供するコンポーネントの機能を意識していれば、システム構築に支障がなく、基盤ソフトウェアに対する不安要素がシステム構築の負担にならない形態を目指している。

6. イネーブラの課題と今後

6.1 イネーブラの充足度

イネーブラを製品化して市場に投入していく場合、どのような分野でどの程度効果をあげることができるのかといった充足度合いを見る必要がある。イネーブラの充足度を見る場合、顧客アプリケーション構築にどの程度の効果があるか、どの程度のシステム構築分野を対象としているか、テンプレートとして業務処理への広がりやどの程度あるかなどが挙げられる。

顧客アプリケーション構築への効果度合いとは、イネーブラの提供機能とアプリケーション機能要件を対比してイネーブラがどこまでの機能要件を満たしているか、イネーブラを利用することによってシステム構築の作業量がどの程度軽減されるかといった指標があげられる。この指標により、顧客アプリケーションの構築支援の効果度合いが提示できれば、利用者側としても使用可否の判断が付きやすい。しかしながら、顧客アプリケーション機能とイネーブラの提供機能が対比できるような機能比較はできていないし、実際にイネーブラを使用したシステム構築と使用しないシステム構築での工数を比較する場合の指針も検討中である。この効果度合いは、イネーブラが提供できる機能が明らかにシステム構築を省力化できるのであれば、必ずしも計数化する必要はないが、すべての機能が明確に定義できることにならないので、計数化の方法を検討している。計数化できれば、対象イネーブラ製品の機能の有効度合いが測れ、不足機能の題材も明確になり、機能充実を図る上での方向性も出すことができる。

次にイネーブラの対象分野では、現時点は Web アプリケーション分野であり、インターネット、イントラネットにおけるシステム構築を支援する機能の充実を図る方向にある。しかしながら、Web アプリケーションは、利用者インタフェースの部分で Web ブラウザを利用するだけで、サーバ側の処理としては多種多様の処理形態が存在する。サーバ処理では、トランザクション処理、営業支援、データ解析、オフィス支援業務、情報共有など多岐の分野にわたり、それぞれの分野で業務的に処理が分離している。この為、イネーブラが提供するアプリケーションインフラやそれに付随するサービスやガイドンスが、これらの処理を全方的に支援できるように機能を一気に充実させることは困難であり対象を絞って順次機能充実させていく。現時点の対象分野は、情報に関する協調処理という切り口で、後方にある処理を有機的に連携させて情報を交換できる形態を考えている。すなわち、企業情報の入り口としての機能

を提供し、その中から様々な処理を行える仕組みであり、後方処理から提供される情報の操作を容易に行えるアプリケーションインフラを提供していく。このような、アプリケーションインフラであっても、そこで利用される ISV ソフトウェアの絞込みや利用できる機能範囲を明確にしないと、利用者の要件に応じて選択できる機能を定めることはできない。現状のイネーブラは、イネーブラ機能としての提示は可能であるが、業務要件に照らし合わせた機能範囲を明確には提示していないし、利用者要件をどの程度満たすことができるかが明確に定義できていないので、この機能範囲と機能レベルを明確化しよう検討していく必要がある。

イネーブラが提供するテンプレートは、アプリケーションインフラやアプリケーション共通サービスを利用したシステム構築を支援するアプリケーションガイドンスとして提供している。このため、テンプレートは、対象業務を限定して作成することになるので、システム構築対象となる業務テンプレートの品揃えを充実させる必要がある。しかしながら、多種のテンプレートを創出するにしても、対象とする業務が多岐にわたり、共通性を見出せないでテンプレートの作成に時間を要することになり、テンプレートの品揃えに時間を要することになる。その対策としては、実際のシステム構築にイネーブラを適用していく過程で、既存のテンプレートを応用して別のテンプレートを創り出していく形態をとり、対象業務を順次広げていく方法が現実的となる。当然、この方法では対象業務が偏ることになるが、新たな業務分野での適用を一から考えていくよりはテンプレートの充足度は上がるし、改良が加えられていくので適用効果も上がることになる。これは、新たな業務分野を開拓しないのではなく、効果的な流用や市場動向から適用分野を考えていくことを意味する。

実際のシステム構築を考えれば、利用者の業務要件に対して、イネーブラが提供する機能がどの程度適用することができるかがイネーブラとしての利用価値の指標となる。イネーブラ機能がどの程度利用者要件に対応付けて提示できるか、利用者がイネーブラ利用価値を認めるだけの機能提供ができているかなどのイネーブラの機能充足度をどのように明確にしていかが課題として残されている。

6.2 共通化と再利用

イネーブラに付帯するテンプレートや Wizard は、イネーブラ製品として提供機能に合った形態で別々に提供しているのが現状である。単一のイネーブラ製品を利用する場合は、製品特性にあった形態で提供しているので問題ないが、複数のイネーブラ製品を使用する場合は、操作性として設定内容の粒度の違いや画面上の表現方法の違いがあると利用者の混乱を招く可能性を持つ。また、利用者登録のような一度設定すれば他のイネーブラでもその内容が引き継げたり、選択して利用できたりすることを利用者は期待する。この為、テンプレートや Wizard の設定方法を共通化して、イネーブラ間で相互に利用できる仕組みが必要となっている。すなわち、イネーブラ製品の中で共通のコンポーネントを利用したり、同じようなインタフェースを提供したり、例えば Wizard による質問の形式や順序などを統一したりする必要がある。

テンプレートや Wizard の操作面においては、ボタンの配置や質問順序といった画面規定やその中で用いられる用語統一を行うことで順次改善することは可能である。しかし、アプリケーションガイドンスとしての機能を実現するためには、対象機能を

実装したコンポーネントを取り込み、それを組み合わせて利用者向けの環境を作り出す必要がある。このようなコンポーネントの中には、イネーブラ間で共通して利用できるものがあるので、それも共有化していくことになる。

これを推し進めるとイネーブラが提供するコンポーネントのインタフェースを共通化して、利用する機能単位を合わせる事が望まれる。コンポーネントは、基盤ソフトウェアとインタフェースする内部的な部分と利用者へ開放する外部的な部分で作成する機能単位が異なる。利用者へ開放するコンポーネントは、業務的なインタフェースや利用者プログラムのインタフェースを意識し、システムの保守性や障害時の対応に変動を与えないようにする必要がある。逆に、内部的なコンポーネントは、利用者プログラムに影響を与えないのでイネーブラとしての再利用性や基盤ソフトウェアとの効率的なインタフェースに重点を置いた構成になる。

このようなコンポーネントの粒度を考えると、利用者プログラムから呼び出すコンポーネントは、業務処理ロジックの流れの中で利用しやすい単位とか融通性を持つ度合いといった指標でしか粒度を決める要素がないのが現状である。この為、提供できる機能範囲をこのような指標にあわせたコンポーネントを作成することになり、本来のコンポーネントが持つ利点である再利用性が失われる可能性がある。今後、イネーブラの全体系を意識した機能性を考慮し、コンポーネント技術としての業界標準手法などを取り込んで、イネーブラとして再利用性が高められる形態でコンポーネントをどのようにパターン化して作成するかが課題である。

6.3 ISV ソフトウェア製品の機能補完

イネーブラは、ISVの基盤ソフトウェアを使用してシステム構築をすることを前提として、アプリケーションシステムと基盤ソフトウェアとの橋渡しをするソフトウェアとして位置づけている。この為、基盤ソフトウェアを利用する上で機能的に不足している部分を補完して、アプリケーションシステムに機能を提供している。この補完している機能は、本来基盤ソフトウェアが提供すべき機能もあれば、アプリケーションとして取り込まなければならない機能も含まれている。後者については、イネーブラの付加価値機能として提供しているものなので、基盤ソフトウェアが機能改善されても影響を受けることが少ない部分であり、この機能を充実していけばイネーブラとしての価値が上がると考える。しかし、前者については、基盤ソフトウェアの機能改善により提供される機能であり、この部分を機能充実しても将来的にイネーブラとしての価値は変わらない。

基盤ソフトウェアの機能を補完することは、ある意味で大きなリスクを持っている。それは、イネーブラの実装方法と基盤ソフトウェアの新機能の提供方法とのギャップを埋められるかどうかにある。基盤ソフトウェアの機能改善の項目は、実際に提供されない実装形態が見えないので、あらかじめ予測して機能補完することはできない。このことは、イネーブラにより実装した補完機能は、基盤ソフトウェアの新機能が提供されてもイネーブラの機能を継続して提供していくか、新機能に合わせた機能提供に変更するか、いずれかの選択を行うことを意味する。

イネーブラが提供している機能が基盤ソフトウェアの新機能よりも付加価値をもっている場合は、イネーブラとして機能を継続して提供していくことになる。しかし、

イネーブラの機能が劣ってきた場合には、基盤ソフトウェアの機能を統合する必要がでてくる。この場合のイネーブラ提供機能の変更は、利用者プログラムとのインタフェースの変更につながる可能性を秘めている。なぜならば、機能的に同じであっても実装形態が異なれば、利用している機能のインタフェース、例えばパラメタの違いや提供機能範囲の違いが出てくるからである。

システム開発者から見れば、既にシステムができている場合には、そのままイネーブラ機能を利用するが、新規にシステムを構築する場合には、基盤ソフトウェアが提供した機能を利用することがある。このような状況を考えると、イネーブラが提供している機能を基盤ソフトウェアの進化とともに変更したり、機能自体を他の機能と併合したりする必要性がでてくる。これに対応するためには、機能修正を行う場合に機能の互換性を保証する機構や利用者インタフェースを変えないで機能提供ができるように考えていかなければならない。

6.4 今後の計画

これからのオープン系プラットフォームのシステム構築は、ISV が提供しているソフトウェアやコンポーネントを組み合わせて短期間で構築する方向にある。いままでのシステム構築は、基盤ソフトウェアを利用して業務アプリケーションをすべて作成するオーダメイド的な構築形態が多かった。新しい情報技術の適用を必要とする分野、例えば Web アプリケーション分野では、ISV が提供するコンポーネントを組合せ、固有のビジネスロジック部分を組み込んでいくイージオーダ的なシステム構築を行う傾向にある。

今後のイネーブラは、イージオーダ的なシステム作りを可能とするコンポーネントと支援ツールの提供を考えている。特に、Web アプリケーションに関しては、利用者のデータベースに合わせた項目表示や入力画面を制御するプログラムを Wizard 形式で生成したり、メインフレームのデータを情報共有データとして取り込み、利用者提供したりする仕組みなどが予定されている。

また、昨今話題に上がるインターネットを利用した企業間データ交換などは、データに意味付けを行って構造的にデータ表現を可能にする XML (eXtensible Markup Language) を利用して企業間でのデータ交換を実現しようとしている。この分野での基盤ソフトウェアは、すでに市場に出始めているが、実務として利用するにはまだ機能補完や業務的な管理機能が不足している。この分野は、これから機能充実が図られ、データ交換基準が決められていくことになり、多くの企業がこの形態でデータ交換をすることになると予想される。この分野に対応するためのイネーブラとして、ISV が提供するデータ交換の基盤ソフトウェアを選定し、その上で業務処理と連携したり、複数の業務処理データを統一的に収集したり、配布したりすることができる仕組みを考えている。この仕組みに業種別のテンプレートを加えることで Wizard 形式でのデータ交換用のシステムを生成することが可能となる。

イネーブラの方向性としては、インターネット/イントラネットを利用したシステムを対象を絞って、そこで利用される新しい情報技術を取り込みながらイージオーダ的なシステム構築を可能にする機能や仕組みを提供していく予定である。

7. おわりに

イネーブラ候補を創出する上で、ユーザの抱える問題にイネーブラとしてどの程度対応できるかという課題に答えるために、ITの潮流と顧客のニーズを見通し、両者間を結び付けて新しいイネーブラを生み出す方法論を説明した。この方法論という転換機構によって、顧客ニーズとテクノロジーをさまざまな素材から調査・分析し、創造力を駆使して市場性の高い新しいソリューションモデルを考え、最新のITとプロダクトの組合せで開発すべきイネーブラ候補を生み出すことを目標に取り組んでいる。この方法論をベースに現在開発・提供しているイネーブラは、主にWebアプリケーションの分野を対象としたものである。この分野では、現在基盤ソフトウェアとしての機能充実を図っているが、Eビジネスの進展に伴い新しいシステム構築の要求も増加し、対応できるスキルをもった要員の不足も顕在化してきている。

一方、新しい技術やソフトウェアを使用するオープン系プラットフォームでの開発はその習熟度の低さから生産性の低下や品質の不均一を招き、オープン性からくるプロダクト間の非互換という問題を抱えている。「サービスのソフトウェア化」を目指すイネーブラの提供はこの問題に対する一つの解でもある。オープン系プラットフォームでのソフトウェア開発における品質の均質化、高生産性、プロダクト間の非互換吸収を可能にするイネーブラは、オープン市場の多様化とEビジネス進展のスピードに対応したオープンアーキテクチャベースのソリューション提供において新しい位置付けのソフトウェアとしてその重要性は更に高まるものと考えられる。

-
- 参考文献**
- [1] M・E・ポーター「競争の戦略」ダイヤモンド社, 1995.
 - [2] 松本幸男「ハイテク企業のオープン市場戦略」原田保, 清家彰敏編『創造する経営』第三部第四章, 日科技連, 1999
 - [3] 藤本隆広「アーキテクチャーベースの産業論」日本経済新聞, 1998年3月23日.
 - [4] 森澤好臣, 岩田裕道, 外山晴夫「クライアント/サーバシステム構築のためのオープンソリューション・フレームワーク」日本ユニシス技報, Vol. 16, NO. 2 (1996)
 - [5] 今江泰, 丑久保年常「新世代メインフレームの基本思想」日本ユニシス技報, Vol. 16, NO. 1, 1996.

執筆者紹介 松本幸男 (Yukio Matsumoto)

1970年早稲田大学理工学部電気工学科卒業。同年日本ユニシス(株)入社。OS, 基本ソフトウェアの開発・保守と利用技術サービスに従事した後, 製造・流通・社公の客先システム開発・サービスに従事。HMPシリーズの製品企画を経て現在, W2Kテクノロジーセンターに所属。情報処理学会会員。

森 良 行 (Yoshiyuki Mori)

1975 年上智大学工学部数学科卒業．同年日本ユニシス(株)入社．金融機関の客先サービスや A シリーズの利用技術サービスに従事した後，Windows 関連イネーブラの企画及び開発に従事する．現在，W 2 K テクノロジーセンターに所属．