

“Whitehorse”が疾駆する日

特集

2

マイクロソフト株式会社
デベロッパーマーケティング部
アーキテクトエバンジェリスト
成本 正史 NARUMOTO, Masashi

本稿ではマイクロソフトが新たに提供する予定のVisual Studio 2005 Team Systemについて解説する。第1部では、このようなツールが登場したコンセプトから説明する。第2部ではTeam Systemについて駆け足ながら解説を試みる。

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:
Visual Studio 2005
Team System

Level



Samples

第1部

新しい時代に突入するIT業界

昨今、エンタープライズアーキテクチャ、サービス指向アーキテクチャおよびユーティリティコンピューティングなどいろいろなキーワードが登場し業界を賑わせている。これらのキーワードにはすべて共通した背景がある。ITに限らず何事にも歴史があり、ある時点での課題を克服することで次の時代へと継続的に発展してきたわけである。今日までのIT発展の歴史を見るとデータ入出力や統計データ処理などの省力化を目的とした単一アプリケーションにはじまり、それらを組み合わせてかつ複数クライアントから利用できるようにして単一部門内での最適化という段階に到達したのが現状である。ここから先は全社的観点でのIT最適化、そしてビジネスパートナーとのバリューチェーンの構築という方向へ発展していくわけだが、そこで以下のような課題が浮き彫りになってきた。

・大規模化にともなう複雑さの増大

たとえば管理するデータ項目がN倍になると開発工数は N^2 になると言われている。また画面数など単純な技術要素だけの規模ではなくシステム化の対象範囲が広がるにつれて新たな要件が増えてくる。こういっ

た要件を的確に実現するためには、最適な方法論と開発プロセスの採用がキーポイントとなる。またプロジェクト管理の重要性が増すに従って、従来のような属人的な管理ではカバーしきれなくなっている。

・既存資産の有効利用

全社規模でみた場合には新規開発の割合は低く、むしろ既存資産とうまく連携しながらインクリメンタルに開発していくのが常識である。そこで登場してくる課題が複数システム、プラットフォームにわたる連携である。

・保守/運用管理コストの増大

政府を例にあげるまでもなく、保守/運用管理コストに莫大なコストをかけているのが実情である。一般的にはIT投資のうち70%が保守に費やされているといわれており、今後さらに指数関数的に増えていくと予想される。この状態を克服するにはいかに運用管理の自動化、簡素化をしていくかが問われている。

・変化への柔軟かつ迅速な対応

ビジネスをめぐる環境は短期間かつ大幅に変わる時代になっている。あっと驚くようなM&Aやアウトソーシング、新規市場への参入もしくは撤退といったニュースが

相次いで発表されている舞台裏ではITシステムへの莫大な変化要求が発生しているわけである。また小規模な要件変化は日常茶飯事であり、これらに迅速に対応することがITシステムに求められている。

こういった課題に対処するために、エンタープライズアーキテクチャによりシステム全体の可視化を行ない、サービス指向によって柔軟な変更管理を実現し、ユーティリティコンピューティングによる仮想化などの技術を導入することでインフラの有効利用していくというのがこれからの方向性となる。

Dynamic Systems Initiativeとは

上記背景をもとに、マイクロソフトが提唱するのがDynamic Systems Initiative (DSI) である。DSIとはIT業界がこれから克服すべきチャレンジに対してWindowsプラットフォームを最大限に活用/発展させていくためのイニシアチブであり、分散システムのビジネスモデル、設計から運用管理にいたるライフサイクル全般に対して大幅な自動化/簡素化を実現するための取り組みである。DSIはマイクロソフト一社で完成できるものではなくハードウェアベンダーからソフトウェアベンダーにいたるまでIT業界を上げた取り組みであり、すでに複数のパートナー企業との協業が始まっている。ここでは簡単にDSIに関する紹介をしよう。

まず、DSIのアーキテクチャを図1に示す。

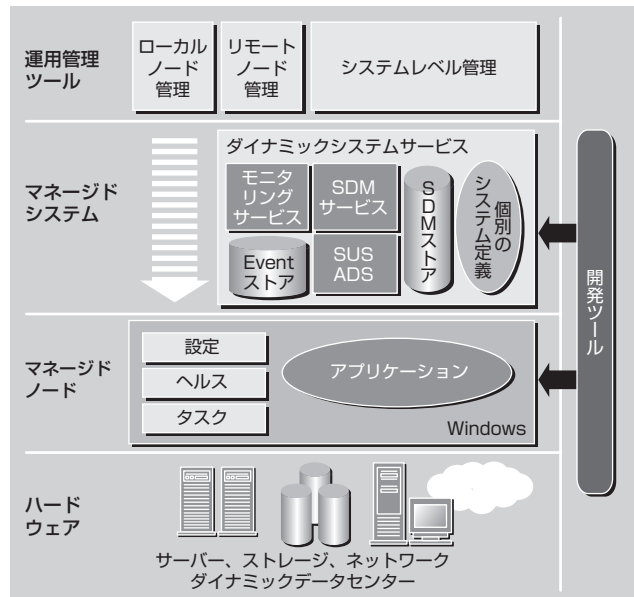
DSIを構成する要素は、大きく4つのレイヤーに分類される。

一番最初はシステムを実行するハードウェアである。図1右下にあるダイナミックデータセンターとは、サーバー、ストレージ、ネットワークなどのセットで構成され、負荷状況に応じて動的に構成可能なものである^{注1)}。次に物理的なハードウェアを抽象化するものとして論理的なノード構成 (Logical Datacenter) があり、トポロジーの構成、ノードにおける稼動状況の公開、実行すべきタスクなどを定義し、それをもとに物理的な展開を実現する。

さらにその上でサービス指向に基づくシステムが動作する。そしてこれらすべてを管理するためのツールが存在する。後述するVisual Studio 2005 Team Systemは図1の中段右にある開発ツールであり、DSIの中では主にシステムと論理ノードの構築および構成定義の役割を担うことになる。

前述したようにDSIの目的とは分散システムのすべての側面に対してできる限りの自動化/簡素化であるわけだが、これを達成するにはまずハードウェアリソースの仮想化によりビジネス要件、負荷状況に応じてリソースの動的なアロケーションを可能とし、サービス指向アーキテクチャによって構築されたシステムがセキュリティ、接続性、信頼性などの運用管理ポリシーに基づいて配置される。そのためにはシステム全体の可視化を行ない、それを上記4つのレイヤ

図1：DSIアーキテクチャ



ーで共有することが重要となる。これを達成するのがSystem Definition Model (SDM) である。

DSIに関する詳細は以下のサイトをご覧ください。

<http://www.microsoft.com/japan/windowsserversystem/dsi/default.aspx>

モデリング言語

DSIの中で中心的な役割を果たすのがSystem Definition Model (SDM) である。SDMとは簡潔にいうと分散システムを構築するためのモデリング言語である。SDMを用いることで分散システムの設計、配置、運用というライフサイクル全般に対するモデルの構築が可能となる。

まず、デザイン時においてシステムが動作するために必要とするリソース、システム構成およびポリシーの定義を実施し、次にツールによって定義された情報をもとにサーバーとの連携によるアプリケーションの自動配布、負荷に応じたハードウェアリソースの自動アロケーション、最後にSDMを解釈可能な運用管理ツールにより大幅に簡素化されたシステムの運用管理が実現される (図2)。

SDMよりシステム開発の初期段階からモデルが構築され、分散システムの物理構成から運用ポリシー、稼動状況の公開、管理にいたるまでシステム全般にわたり適用されることがおわかりいただけるであろう。通常“モデル”という単語からは設計時のある局面において利用されるドキュメントを連想される方が多いと思うが、SDMの場合には一度構築されたモデルが分散システムのライフサイクルと共に存在し続け、またシステムのバージョンアップにも重要な役割を果たすことになる。SDMにより規定されるメタモデルは開発ツ

注1) すでに一部パートナーと協業を開始している。