

ホストマイグレーションやオフショア開発を支援する「N字統合開発ソリューション」

1990年代から担当プロジェクトの9割でXupperを活用し、これまでMDFrame/XやXupper IIのIPOエディターなどXupper自体の機能拡充にも深く関わってきた日本アイ・ビー・エムの森下氏。同氏は今回、ホストシステムのマイグレーションやオフショア開発において多くのユーザー/ベンダーが直面している課題を解決し得るソリューションとして、Xupper IIを核とした「N字統合開発ソリューション」を発表した。



ホスト環境の複雑化とオフショア開発上の課題

ホストシステムの更改は高額な投資を必要とし、業務に与える影響も大きいことから、容易には行えない。そのため、フロントエンドのみWeb化を進め、結果的にシステム環境の複雑化・肥大化を招いているケースや、複数の異なるホスト基盤の運用に苦勞しながらも、なかなか統合に踏み切れないというケースは多い。こうした中で、リスクを考慮した実現性のあるマイグレーションを、より短時間で可能にするソリューションを求める声が増えているという。

加えて、今日多くの企業が直面しており、今後ますます増えていくと思われるのが、オフショア開発上の課題だ。オフショア開発においては言語や文化の壁を乗り越えなければならないのはもちろんだが、課題はそれだけではない。森下氏は、解決すべきポイントとして、設計情報を確実に伝える方法の確立、属人性の排除、オフショア側の作業スコープ検討（コーディングや単体テストだけでなく、基本設計や結合テストも含めたオフショア化でコスト削減効果を高める）などを挙げた。

そして、森下氏はこれらを含めたシステム開発上の様々な課題とその原因を整理。解決すべき内容を、次のように定めた。

あるべきソリューションの狙い・目標

- ① システム業務全体像の見える化と要件を決めるベースラインの確立
- ② システム変更による全体への影響分析と変更実現性の評価を可能とする
- ③ マイグレーションによるベースラインの確立と資産の有効活用
- ④ 業務要求について発注側と請負側の双方で共通理解を持つ方法の確立
- ⑤ 成果物の変更による整合性を保証し、設計内容とテスト仕様の不整合をなくす
- ⑥ オフショアで有効な開発方法の確立
- ⑦ 並行開発における設計情報更新の整合化を図る
- ⑧ 機能要件の上流での実現性評価とテスト段階での早期検証

- ⑨ 設計リポジトリとプロジェクト管理上有効なエンジニアリングの活用
- ⑩ 従来型のウォーターフォールモデル開発から反復型開発方式へのシフト

マイグレーションにおいて必須の現行保証を上流工程で実現

これらを具現化したソリューションとして森下氏が提案するのが、従来のV字モデル（ウォーターフォールモデル）にリバースソリューションによる現行分析工程を組み合わせ、ソフトウェアエンジニアリングの手法を取り入れた「N字統合開発ソリューション」だ。

ほとんどの場合、ホストマイグレーションにおいては現行機能保証が必須となる。通常のコンバージョンツールを使ったマイグレーションでは、コンバージョン実施後にテスト局面にて現行保証が確定するため、テスト工数の増加や手戻り発生リスクが大きい。N字統合開発ソリューションでは、従来のV字モデルにおける設計工程の前段階に、リバースによる現行分析を実施。現行分析情報と設計情報はXupper IIの設計リポジトリにより一元管理され、上流工程での現行保証が可能となる。

分析工程では、まず現行プログラムのソースコードをベースに現行設計書にリバースして設計情報を現状最新化し、現行システムの全体像としての、フォワード開発のベースラインを確立する。リバース作業は、2つのアプローチで行なう。1つはソースコードリバースからプログラム設計レベルのリバースアプローチであり、もう1つは不完全な現行設計情報のベースをトップダウン観点で整理しながらリバース作業を通じて詳細設計、外部設計、要件定義レベルに仕上げしていくアプローチである。2つのアプローチの成果物は、設計リポジトリとしてプロセスモデル、データモデルが整合性の取れた形でXupperリポジトリに実現でき、設計のベースラインを確立することができる。

最初のリバース作業でのポイントは、移行対象のソースはできる限り削除してデッドコードを排除し、規模の最適化を行うこと。特に4GLからの移行では、利用しているマクロ関係を展開すると規模が膨らむため、マクロ変換して畳むなどの開発保守対象規模を抑える工夫が必要である。2番目のアプローチでは、全体像を把握するのに必要なプログラム構造分析、ジョブフロー図、判定条件検出などのリバースツールを活用しながら、現行設計ベース情報を設計リポジトリとしてXupperリポジトリに構築する。1番目のアプローチでソースリバースしたプログラム仕様（IPO情報）を設計リポジトリにインポートすることで、設計情報とソース情報のトレーサビリティを実現する。

現行保証は2段階で実施。まず、移行後のソースレベルで現行比較し、保証されたものをベースに設計に反映する。そして、テストによる現行保証を行う。すなわち、リバースの結果ベースラインとして作成した設計リポジトリにソースとの紐付けを行い、現行ソースの追跡を可能とし、テスト工程での現新比較の精度向上を実現する。

これにより、手戻りリスクを最小限に抑え、かつ現行設計情報・プログラムを最大限活用できるという（図1）。

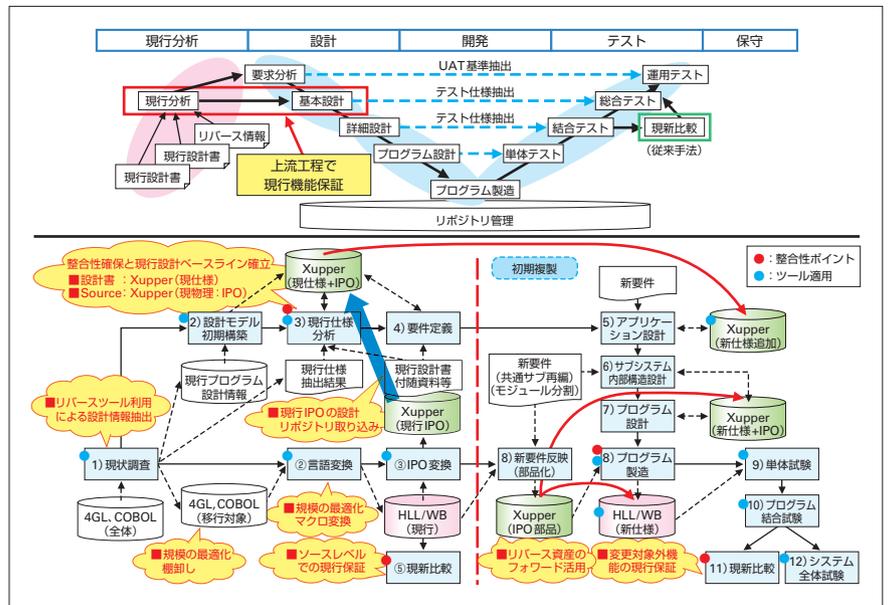


図1：Nモデル 開発工程実装イメージ（Nモデルによる現行保証と最適化開発方式）

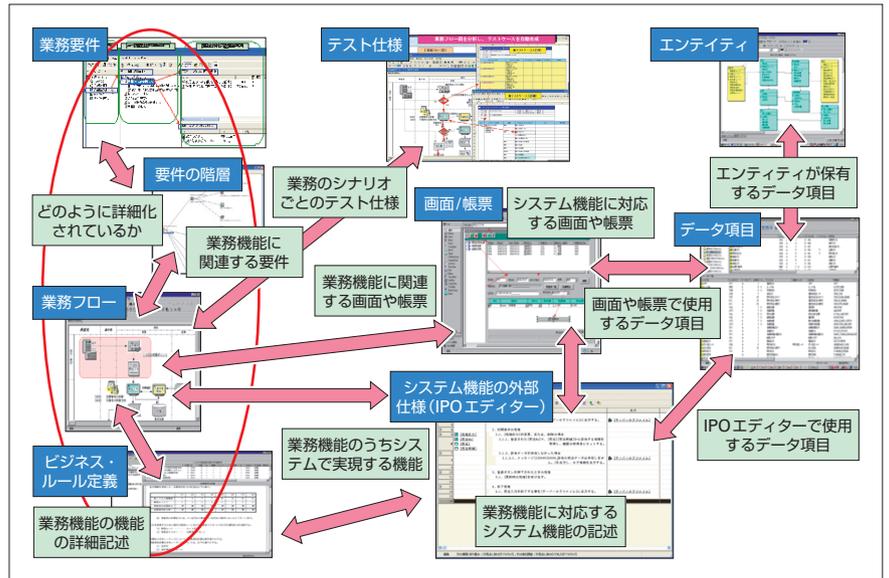


図2：統合開発ソリューションによる全体業務の見える化と成果物関連図

設計リポジトリによる課題解決とオフショア開発への活用

課題解決ソリューションとして、Xupper II 自体の持つメリットは最大限に活かすことができる。要件管理機能では、業務要件とシステム要件の対応、要件と実装との関係などのトレーサビリティを実現し、業務全体の見える化が可能（図2）。処理機能記述をIPO（入力情報、処理、出力情報）形式で定義するIPOエディターにより、リポジトリ情報を参照しながらIPO情報を定義することで整合性を保持できる。要件変更によるシステムへの影響調査も、クロスリファレンス機能で設計・ソースレベルまで一貫した分析が可能だ。リポジトリを活用した設計作業により、膨大な成果物（設計情報）間の整合性の保証ならびに並行開発における設計情報の整合

性も保証できる。

N字統合開発ソリューションでは全工程をXupper IIの設計リポジトリで一元管理することから、リモート開発やオフショア開発との親和性も高い。設計情報からコーディングレスでソースコードを自動生成するツールを取り入れているため、安定した開発品質と生産性を確保できるほか、単体テストの自動化およびテストケース抽出機能とテスト半自動化ソリューションの採用もオフショア開発において有効だ。とりわけ、クラウド環境を活用した設計リポジトリのオフショアとのシェアにより、最新情報の提供や、スキル伝達と同時に人材の流動性のリスク対策が可能である点は大きなメリットといえるだろう。