

第12回 Xupper ユーザ事例紹介セミナーレポート

ケン・システムコンサルティングが主催する「Xupper ユーザ事例紹介セミナー」が、2008年10月に東京都内で開催された。1994年の登場以来、上流分析・設計ツールの定番として多くのユーザに支持され続けている Xupper II。毎年恒例となっている本セミナーも、今回で第12回目を迎えた。

今回、事例を発表したのは日本電気およびキリンビジネスシステムの2社。いずれも単独の開発案件ではなく、開発スタイルの変革や開発方法論を含めた標準化への取り組みといった特色あるテーマであり、いつも増してユーザの注目度が高かった。また、今回は初の試みとしてパネルディスカッションを実施。事例紹介に登壇した両社の講演者ほか2名がパネリストとして参画し、ソフトウェア産業における人材育成について議論を行った。

以下で、各セッションの内容を見ていこう。

事例 1

日本電気株式会社 第2MCシステム事業部 事業部長 大場彰夫氏

Xupper IIと各種ツール連携で自動化促進 知識集約型開発への転換を目指す

オフショア開発がますます進展する中で、従来のような労働集約型の開発スタイルのままでは価格競争に陥り、海外の安価な労働力に太刀打ちできない。こうした状況を見極め、日本電気では知識集約型開発への転換を図るべく、新たな開発手法の確立に取り組んでいる。Xupper IIのリポジトリによる設計情報の一元管理や MDFrame/Xのプログラム自動生成は、その中で不可欠の要素として位置付けられている。



■ NECが目指す 「TCO削減指向AP開発手法」とは?

昨今のアプリケーション開発では、以前に比べて全体の開発期間や開発費が縮小傾向にある一方で、基本設計や機能設計といった上流工程に関しては、より多くの時間・コストが必要とされるようになってきている。また、開発コストに対するユーザの要求も変わりつつあり、従来のように新規開発時だけでなく、追加開発も含めたシステムのライフサイクル全体を通じてのコスト低減が強く求められるようになった。

もう1つ、最近の傾向として挙げられるのが、開発リソースの問題だ。いまやSEは「新3K」と呼ばれるほど人気低迷しており、優秀なSEの確保が非常に難しい。加えて離職率も比較的高いことから、熟練者の育成も困難な状況となっている。

こうした環境の中で、より早く、コストを抑えながら、品質の高いアプリケーションを開発していくために、日本電気(以下 NEC)では開発スタイルの変革に着手。その狙いは、従来の労働集約型から知識集約型開発への転

換だ。具体的なポイントとしては、開発要員の労力を上流工程の設計や評価試験の設計に注力させること。そして、ツールを活用した自動化によって中下流工程を効率化すること。維持フェーズにおける効率化のためには、設計情報を一元管理することでメンテナンス性を向上し、デグレードを抑止することも重要となる。

この知識集約型開発への転換に向けて、NECでは上流設計工程、中流・下流工程、開発リソースのそれぞれにおける課題に対して「あるべき姿」である目標を設定。それらを実現するために、さまざまなアプリケーション開発技術や開発手法、さらにはプロジェクトマネジメントの要素を連携した「TCO削減指向AP開発手法」の確立に取り組んでいる。

■ Xupper IIを用いた ■ アドバンスドDOA上流設計

上流設計工程では、仕様のゆらぎを防ぎ、もし仕様変更が発生しても容易に追加変更できることが目標となる。その実現手段として、TCO削減指向AP開発手法では、アドバンスドDOA上流設計、業務モデル化、リポジトリ

一元管理、イテレーション開発といった要素を採り入れた。

アドバンスドDOAとは、IF～THEN形式のルールで問題解決をするプログラミング技法である「ルールベース」の考え方と、従来のDOA(Data Oriented Approach)を組み合わせたアプローチだ。アドバンスドDOAでは、まずユーザの視点に立ち、ユーザの理解しやすい業務フローを主体に、業務知識(ビジネスルール)をルールベース化。それらをXupper IIのリポジトリで一元管理する。同時にデータ分析を行い、従来のDOAと同様にビジネスルールとエンティティの関係を分析して、アプリケーション構造を確定していく。

プログラムの自動生成は、MDFrame/Xのアクションダイアグラムから行うが、将来的にはルールベースシステムとして、ロジックを解析できるものについてはルールエンジンによりプログラムを自動生成することを目指している。

業務モデル化では、業務フローによって業務をモデル化することで、人や組織、IO、データ項目などの管理対象を明確化し、業務モデルからフレームワークと部品化すべきモデルの

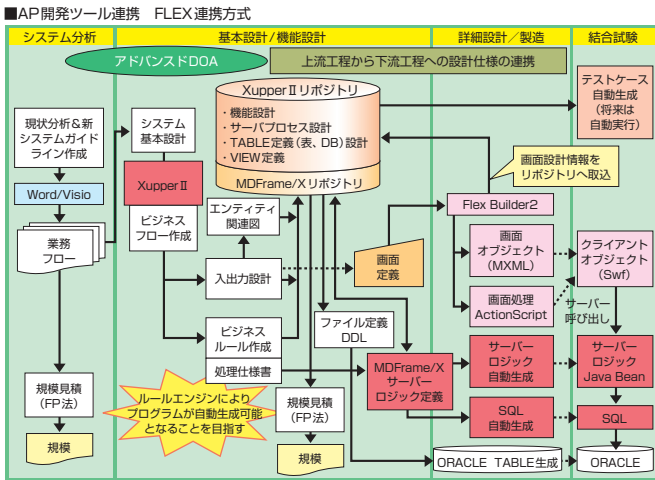


図 1: アドバンスド DOA 上流設計と AP 開発ツール連携

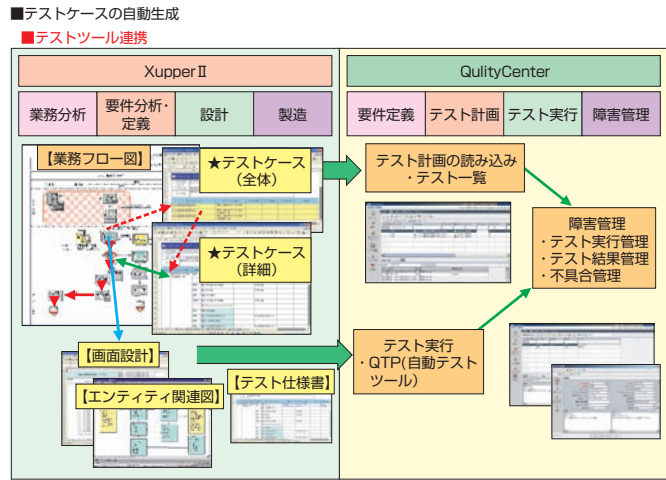


図 2: 自動生成したテストケースを自動実行できるテストツールとの連携も今後予定されている。

独立性を重視しながら抽出。データ分析からオブジェクト候補を洗い出す。こうしてフレームワークや部品を効率的に活用していくことで、作り込む部分を減らしたり、作る場合でも均質なものができるようになる。

こうした上流設計工程を支える基盤となるのが、Xupper IIによる設計情報の一元管理(リポジトリ化)だ。業務フロー図を中心に、さまざまな情報を関連づけて管理できるので、仕様のゆらぎや追加が発生した際も、リポジトリ情報からIOやデータ項目などの影響ポイントを抽出し、影響のあるビジネスルールやエンティティを検索できる。

また、基本設計の確定後にイテレーション開発によって業務要件の成熟度を段階的に上げ、業務フローの要件やアプリケーション構造の確定を実施することで、仕様要件のゆらぎを抑えることが可能となる。

■ 各種ツールの活用で ■ プログラム作成やテストも自動化

中流から下流工程の目標は、上流設計終了後のプログラム開発の効率化と開発期間短縮だ。ここでは、各種ツールの活用による開発の自動化が鍵となる。

画面設計に関しては、コンポーネントエディタやAdobeのFlex Builder 2を利用する。Flex Builder 2は、実際に動く画面(モックアップ)を生成して、仕様を確認することができるツールだが、そのUI定義をMXML化してお

けば、モックアップの画面設計情報をそのままXupper IIのリポジトリへ取り込むことができる。

データベースのアクセス部品(SQL)は、MDFrame/Xの機能によって自動生成。プログラムについても基本的に、業務ロジックをMDFrame/Xでアクションダイアグラムとして登録し、Javaのプログラムを自動生成する。これにより、中流工程以降の開発の効率化および期間短縮が期待できる。このアクションダイアグラムからの自動生成に加え、今後、ルールエンジンによるプログラム自動生成が実現すれば、さらに生産性が向上するはずだ。

また、TCO削減指向AP開発手法には、テストケースの自動生成も含まれている。たとえば、試験漏れ防止のためにXupper IIのリポジトリからテストすべきケースを自動抽出したり、業務フロー図を分析してすべての業務の流れをテストケースとして抽出することが可能だ。テストケース(詳細)として、テストで使用する画面やテーブルなどのデータと確認すべき情報をリポジトリ上で連携し、試験仕様書を容易に作成することもできる。

今後はテスト工程の効率化をさらに推進すべく、テスト管理ツールのHP Quality Centerや自動テスト実行ツールのHP Quick Test Professionalなどと連携させ、テストケースの作成から実行、管理まで自動化するための検証を進めているという。

■ スモールチーム開発で ■ 知識の集約化を図る

開発リソースの課題に対しては、スペシャリスト集団による知識集約型開発という目標が掲げられた。TCO削減指向AP開発手法においては、これを実現するためにスモールチーム開発やディレクタ組織といったプロジェクト体制が組まれる。

知識の集約化を図るには、少数精鋭のスモールチームのほうが望ましい。また、優秀なSEを育成するには、できるだけ多くの場数を踏ませる必要があるが、その点でもスモールチーム開発は有効だ。大規模開発においても開発部隊を10~30人までのスモールチームに分け、すべてのローテーションをチーム単位で行うことで、チームメンバーの連携が強固になり、結果的に生産性を向上させることができるという。

なお、スモールチーム開発を有効に機能させるためには、各チームのリーダーと仕様管理者から構成されるディレクタ組織というものも必要だ。TCP削減指向AP開発手法では、このディレクタ組織が仕様管理を一元的に実施し、業務間相互レビューによって仕様齟齬などを防止する役割を担うことになる。

NECのTCO削減指向AP開発手法は完成形ではなく、まだ進化の途上にある。今後、日本のソフトウェア開発のあり方を変革する新たな指標となることを、大いに期待したい。