

事例 3

株式会社 大和総研 金融ソリューション事業本部 証券ソリューションシステム開発部 次長 室橋和紀氏

金融機関向けシステムの保守開発を Xupper II と MDFrame/X で効率化

金融機関向けに幅広いソリューションサービスを展開する大和総研では、投信窓販システムの保守開発効率化のために Xupper II を採用。上流工程における設計情報の整合性確保、DOA による安定したデータベース構造の設計に取り組み、緊急の変更対応やシステムの根幹にかかわる機能拡張などもスムーズに行える環境を構築した。また、Xupper II の設計情報と連携した MDFrame/X の SQL 自動生成機能も活用し、生産性と保守性を高めている。



Xupper II + MDFrame/X によるプログラム自動生成

ケン・システムコンサルティングでは、Xupper のリポジトリを拡張し、モデル駆動型の開発を実現するツール MDFrame/X も提供している。この MDFrame/X は、リポジトリに登録された設計情報からプログラムを自動生成するので、プログラミングによるバグの発生が抑えられる。

上流工程では Xupper を用いて業務フロー図とビジネスルールをベースに、エンドユーザにも分かりやすい設計情報を作成する。そして、下流工程では MDFrame/X を用いて、詳細モデルを記述し、そこから Java、.NET、Biz/Browser 等のプログラムを自動生成する。

詳細モデルは、業務ロジックのみを記述するだけでよい。工数を最小化できる。さらに、仕様変更が発生した場合でも、モデルの変更のみで対応できるため、設計にかかる負担は大幅に軽減できる。

これによって、上流から下流まで一貫して設計情報の一元管理ができ、「見える化」を実現するとともに、設計プロセス全体の効率化やスピードアップも可能となり、さらには品質の向上や保守性の向上も図ることが可能である。

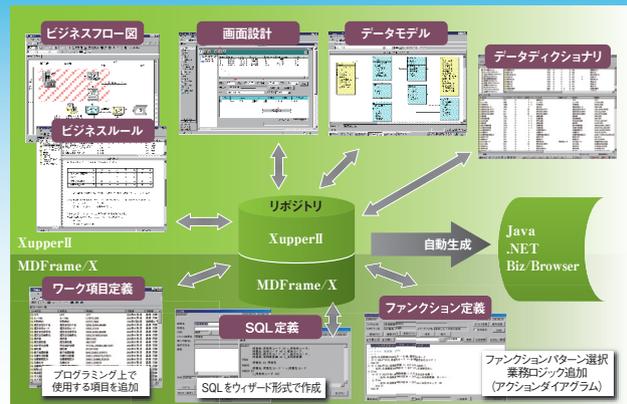


図 Xupper と MDF による自動生成

設計情報の整合性確保と生産性・品質の向上を目指して

大和証券グループのシンクタンクである大和総研では、銀行の投信窓口販売業務を支援するソリューションとして、投信窓販フロント・ミドルシステムの「SONAR-FR」、投信窓販インターネットフロントシステムの「SONAR-IC」などを提供している。

ユーザはこれらのシステムを ASP サービスとして利用する仕組みであり、低コストで導入・維持できるのが特徴だ。複数社でサーバやソフトウェアなどのリソースを共同利用することから、法制度改正への対応なども 1 社あたりの負担が少ないというメリットもある。また、各社各様のコンプライアンスチェック機能や顧客カード情報管理機能、金融商品取引法対応などを柔軟に取り込むことも可能となっている。

大和総研では 2004 年に SONAR-FR のシステム開発に着手し、2005 年 4 月よりサービス提供を開始した。一方、SONAR-IC は

2006 年末に開発をスタートし、2007 年夏にリリースした。これらのシステムにおいては、上記のようなサービス特性からも分かるように、リリース後もシステム全体にかかわる変更やユーザごとのカスタマイズといった保守開発が頻繁に発生する。安定したサービスを提供し、かつ継続的に機能向上を図っていくためには、この保守開発をいかに効率的に行うかが重要な鍵となる。

この SONAR-FR と SONAR-IC のシステム保守開発において、大和総研ではいくつかの課題を認識していた。1 つは、ビジネスフローや ER 図、テーブル定義書といった設計情報が、Word、Excel、Visio など別々のドキュメントフォーマットで作成され、管理されていたことだ。そのため、設計情報を更新する際は各ドキュメントを個別に修正する必要があり、それぞれの関連性は人間が判断するしかない。設計情報をより効率的に、整合性の取れた状態で管理することが求められていた。

また、ユーザごとのカスタマイズ要件をシス

テムに取り込んでいく上では、その影響範囲を正確かつ迅速に把握することも必要だ。さらに、安定したデータベース構造とするために、DOA(Data Oriented Approach) によるデータベース設計も必要と考えていた。

Xupper II と MDFrame/X をそれぞれ最適な範囲で適用

大和総研では、こうした課題を解決するために最適な CASE ツールを検討し、Xupper II の導入を決定。決め手の 1 つは、Xupper II は他製品との連携機能など、CASE ツールとして機能が豊富であったことだ。また、ライセンス体系も重要なポイントとなった。開発ピーク時のユーザ数に合わせたライセンスが必要となるとコスト的な負担も大きい。Xupper II は同時利用ユーザ数のライセンス体系なので、開発要員の増減にも柔軟に対応できる。加えて、社内の別部門でも導入されているという安心感もあった。

Xupper II 導入後には、Xupper II の設計情

報から SQL や Java ソースを自動生成できるという MDFrame/X についても導入を検討。コーディングの属人性を排除することは品質向上にも効率化にも直結すると判断し、採用を決定した。

既存システムに対して Xupper II を適用する上で必要な作業として、まず取り組んだのはディクショナリの整理だ。ネーミングについては Xupper II の制約を満たす必要がある。たとえば、これまでは異なるテーブルでそれぞれ「商品区分」という同じカラム名が存在していた。一方はファンド情報の「商品区分」であり、もう一方は取引情報の「商品区分」だ。もちろん管理している情報は異なり、桁数も異なる。このままでは登録できないため、取引情報のカラム名を「取引商品区分」という名称に変更した。同様に、対象となるすべてのプログラム、テーブル、カラム名を変換し、整理した。

Xupper II の利用方法を確立する上での苦労もあった。SONAR-FR や SONAR-IC は頻繁に機能向上やカスタマイズが発生するシステムであり、あまりに詳細にわたって完璧な管理を行おうとすると、Xupper II の管理自体がオーバーヘッドになってしまう場合もある。開発プロジェクトの性質に応じて利用機能を決定し、管理すべき設計情報の範囲を決めることが必要だ。そこは試行錯誤を重ねながら、最適な利用方法を模索していった。

また、Xupper II では設計情報を変更するとすべて最新の情報に置き換わり、いつどのように変更したのか履歴が分からなくなってしまう。複数のプログラムが同時に変更をかけていくと、どこまでが実装されているのか、あるいは開発中なのかといった管理が難しくなるため、実際の運用上は、特定の担当者が設計情報の登録を集中して行い、管理することでカバーした。

MDFrame/X についても、最適な適用範囲を検討。MDFrame/X でプログラムまで自動生成することも可能だが、大和総研では Struts による Web アプリケーション開発を行っており、開発の自由度や既存資産の有効活用を考慮して、SQL 自動生成のみ適用することとした。既存のプログラムに関しては、SQL 部分の書き直しを実施。一部のプログラムでは複雑な SQL を生成できないこともあり、そこは複数の単純な SQL に分解してプログラムのロジック変更で補う形で対応した。

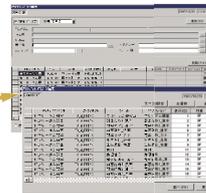
MDFrame/X による SQL 自動生成。Xupper II の設計情報を取り込み、ウィザード形式で SQL を生成できる。



例:SQLの詳細設定



インプットパラメータの設定



アウトプットパラメータの設定



WHERE条件の設定

```
SELECT
  TB_M_FRKYK.FRDM_KYK_CODE AS FRDM_KYK_CODE
  ,TB_M_FRKYK.KYK_NAME_KJ_1 AS KYK_NAME_KJ_1
  ,TB_M_FRKYK.KYK_NAME_KJ_2 AS KYK_NAME_KJ_2
  ,TB_M_FRKYK.KYK_RYAKU_NAME_KJ AS KYK_RYAKU_NAME_KJ
  ,TB_M_FRKYK.KYK_RYAKU_NAME_KN AS KYK_RYAKU_NAME_KN
  ,TB_M_FRKYK.KYK_RYAKU_NAME_EN AS KYK_RYAKU_NAME_EN
  ,TB_M_FRKYK.SEINENGAPPI_GENGO AS SEINENGAPPI_GENGO
  ,TB_M_FRKYK.SEINENGAPPI_WAREKI AS SEINENGAPPI_WAREKI
  ,TB_M_FRKYK.SEIBETSU AS SEIBETSU
  ,TB_M_FRKYK.YUBIN_NO AS YUBIN_NO
  ,TB_M_FRKYK.TODOFUKEN_KJ AS TODOFUKEN_KJ
  ,TB_M_FRKYK.KYK_JUSHO_KJ_1 AS KYK_JUSHO_KJ_1
  ,TB_M_FRKYK.KYK_JUSHO_KJ_2 AS KYK_JUSHO_KJ_2
  ,TB_M_FRKYK.KYK_JUSHO_KJ_3 AS KYK_JUSHO_KJ_3
  ,TB_M_FRKYK.KYK_DENNA_NO AS KYK_DENNA_NO
  ,TB_M_FRKYK.CIF_CODE AS CIF_CODE
FROM
  TB_M_FRKYK TB_M_FRKYK
WHERE
  TB_M_FRKYK.FRDM_KYK_CODE = ?
```

```
SELECT
  TB_M_FRKYK.FRDM_KYK_CODE AS FRDM_KYK_CODE
  ,TB_M_FRKYK.KYK_NAME_KJ_1 AS KYK_NAME_KJ_1
  ,TB_M_FRKYK.KYK_NAME_KJ_2 AS KYK_NAME_KJ_2
  ,TB_M_FRKYK.KYK_RYAKU_NAME_KJ AS KYK_RYAKU_NAME_KJ
  ,TB_M_FRKYK.KYK_RYAKU_NAME_KN AS KYK_RYAKU_NAME_KN
  ,TB_M_FRKYK.KYK_RYAKU_NAME_EN AS KYK_RYAKU_NAME_EN
  ,TB_M_FRKYK.SEINENGAPPI_GENGO AS SEINENGAPPI_GENGO
  ,TB_M_FRKYK.SEINENGAPPI_WAREKI AS SEINENGAPPI_WAREKI
  ,TB_M_FRKYK.SEIBETSU AS SEIBETSU
  ,TB_M_FRKYK.YUBIN_NO AS YUBIN_NO
  ,TB_M_FRKYK.TODOFUKEN_KJ AS TODOFUKEN_KJ
  ,TB_M_FRKYK.KYK_JUSHO_KJ_1 AS KYK_JUSHO_KJ_1
  ,TB_M_FRKYK.KYK_JUSHO_KJ_2 AS KYK_JUSHO_KJ_2
  ,TB_M_FRKYK.KYK_JUSHO_KJ_3 AS KYK_JUSHO_KJ_3
  ,TB_M_FRKYK.KYK_DENNA_NO AS KYK_DENNA_NO
  ,TB_M_FRKYK.CIF_CODE AS CIF_CODE
FROM
  TB_M_FRKYK TB_M_FRKYK
WHERE
  TB_M_FRKYK.FRDM_KYK_CODE = ?
```

```
SELECT
  TB_M_FRKYK.FRDM_KYK_CODE AS FRDM_KYK_CODE
  ,TB_M_FRKYK.KYK_NAME_KJ_1 AS KYK_NAME_KJ_1
  ,TB_M_FRKYK.KYK_NAME_KJ_2 AS KYK_NAME_KJ_2
  ,TB_M_FRKYK.KYK_RYAKU_NAME_KJ AS KYK_RYAKU_NAME_KJ
  ,TB_M_FRKYK.KYK_RYAKU_NAME_KN AS KYK_RYAKU_NAME_KN
  ,TB_M_FRKYK.KYK_RYAKU_NAME_EN AS KYK_RYAKU_NAME_EN
  ,TB_M_FRKYK.SEINENGAPPI_GENGO AS SEINENGAPPI_GENGO
  ,TB_M_FRKYK.SEINENGAPPI_WAREKI AS SEINENGAPPI_WAREKI
  ,TB_M_FRKYK.SEIBETSU AS SEIBETSU
  ,TB_M_FRKYK.YUBIN_NO AS YUBIN_NO
  ,TB_M_FRKYK.TODOFUKEN_KJ AS TODOFUKEN_KJ
  ,TB_M_FRKYK.KYK_JUSHO_KJ_1 AS KYK_JUSHO_KJ_1
  ,TB_M_FRKYK.KYK_JUSHO_KJ_2 AS KYK_JUSHO_KJ_2
  ,TB_M_FRKYK.KYK_JUSHO_KJ_3 AS KYK_JUSHO_KJ_3
  ,TB_M_FRKYK.KYK_DENNA_NO AS KYK_DENNA_NO
  ,TB_M_FRKYK.CIF_CODE AS CIF_CODE
FROM
  TB_M_FRKYK TB_M_FRKYK
WHERE
  TB_M_FRKYK.FRDM_KYK_CODE = ?
```

設計情報の一元管理と SQL自動生成が生む効果

SONAR-FR および SONAR-IC の保守開発に Xupper II を導入したことにより、上流工程における設計情報の一元管理が可能となった。その効果として大和総研が第一に挙げるのが、システムの根幹にかかわる機能拡張もスムーズに行えるようになったことだ。システム変更の影響を分析するために、従来のように複数のドキュメントやソースを解析する必要がなく、Xupper II のリポジトリ情報から迅速に影響範囲を確認できる。

また、以前はテーブルレイアウトなどの設計情報をファイル配布していたが、それを止めて Xupper II の参照を全員に義務付けた。もちろん、プログラム変更が発生した際は、必ず Xupper II の設計情報を修正することもルール化している。これにより、全員が常に最新の情報で開発を進めることができるようになった。頻繁に発生する項目追加やテーブル追加にも、スムーズに対応可能となっている。

なお、Xupper II の設計情報登録完了まで、下流工程では作業着手できないということになるが、これにより従来のような見切り開発がなくなった。その分、手戻りはほとんど発生しないようになったという。

ディクショナリの整理を行ったことで、プロ

グラムの可読性も向上した。同じ項目名で内容の異なる項目が一切なくなったので、プログラム内のソースの検索によって当該項目の使用箇所を確実に抽出することが可能だ。これは、保守性の向上につながっている。

MDFrame/X に関して、現状は SQL 自動生成のみの適用だが十分な効果を生んでいる。それは、自動化による作業時間短縮だけに留まらない。以前は複数のプログラマが SQL を書いていたため、どうしてもプログラマによって書式にクセが出て、読みにくい SQL や効率の悪い SQL も存在していた。それが、MDFrame/X でソフトウェア的に自動生成される SQL なら、誰が操作しても SQL の書式は統一されることになる。もちろん、複雑な SQL や効率の悪い SQL の生成は抑制されるので、生産性と保守性をともに高めることができるのだ。

なお、大和総研では、今回の Xupper II と MDFrame/X を活用した保守開発効率化の取り組みを通じ、Xupper II に対する設計情報の変更履歴管理機能の追加など、いくつかの要望事項をケン・システムコンサルティングに挙げている。こうした現場のニーズを取り入れていくことで、Xupper II や MDFrame/X が今後もさらに開発の効率化や品質向上に貢献するツールとして進化し続けていくことに大いに期待したい。