

ドメイン参照モデルに基づくオブジェクト指向開発

3 Q - 1

-ITS 画像交換システム開発-

早瀬 健夫*1, 安東 孝信*1, 丸尾 秀史*1, 新館 秀和*2, 櫻本 秀明*1

*1 株式会社 東芝 e-ソリューション社 *2 東芝アドバンストシステム 株式会社

1. はじめに

近年、オブジェクト指向技術は、再利用性や拡張性などのメリットから広く普及しこれまでに多くの適用事例が報告されている。特に、設計・実装の再利用により開発工数が短縮されることが確認されている。一方、オブジェクト指向開発においても、ソフトウェアの品質を向上することは重要である。ソフトウェアのライフサイクルという立場で考えてみると、分析から実装で開発作業が終りではなく、その後にテストなどの重要なフェーズが控えている。ライフサイクル全体のコストから見ると、テスト以降のフェーズはむしろ大きな比重を占めていると言われることもある。

本稿では、オブジェクト指向を用いてエラーの少ないソフトウェアを開発するための技術を明らかにする。我々は、再利用性を向上するためのメタモデルとしてドメイン参照モデルいう新たな概念を既に提案しており、適用事例を通じてその有効性を確認している[1]。そこで、このドメイン参照モデルを用いて、開発の早い段階でエラーが発生する原因となるリスクを洗い出すアプローチを提案し、事例を通じてその効果を明らかにする。

2. オブジェクト指向開発の現状

オブジェクト指向開発を進めるための方法論として RUP(Rational Unified Process)[2]がある。現在のオブジェクト指向開発では、UML(Unified Modeling Language)と呼ばれるモデリング言語を用いて、RUPなどの開発方法論を参考にし、必要に応じてカスタマイズしながら進めることが多い。

しかし、RUP ではオブジェクト指向分析と設計の開発フェーズに関する記述が中心になっている。

一方、実際の開発では再利用性の高いオブジェクトモデルを構築することも重要であるが、エラーの少ないソフトウェアを実現する上で、障害となるリスク要因を洗い出しておくことも重要な作業の一つである。例えば、ユーザの要求が少しづつ変わることによる仕様変更、ツールなどの開発環境の変更、開発者の技術不足などに起因するプログラミングエラーなどが挙げられる。これらのリスクがオブジェクトモデルに与える影響は少なくなく、再利用性を向上する上でも軽視できない点と考えられる。RUPなどの従来のオブジェクト指向方法論では、このような観点は充分明らかにされていない。

3. ドメイン参照モデルに基づく品質向上アプローチ -ITS 画像交換システム開発-

我々は、オブジェクト指向を用いてエラーの少ないソフトウェアを開発することを目的として、ドメイン参照モデルに基づくオブジェクト指向開発アプローチについて示す。ソフトウェアのエラーとは、ソフトウェアの中に障害を作り込む人間の動作である。我々は、このエラーを発生させるリスク要因の中で、ソフトウェア構造に関わる要因を対象に品質の作り込みを行う。したがって、開発者の技術不足や開発者のコミュニケーションの問題などプロジェクトに関わる要因については対象としない。以下に、ITS(Intelligent Transport Systems)画像交換システムの通信ソフトウェア開発事例を示す。本システムは、クライアントの指示にしたが

Object-Oriented Development Based on Domain Reference Models: Developing Video-Exchange Systems on ITS

Takeo Hayase*1, Takanobu Ando*1, Hideshi Maruo*1, Hidekazu Niitate*2, and Hideaki Enomoto*1

*1 e-Solutions Company, Toshiba Corporation, *2 Toshiba Advanced Systems Corporation

3-22 Katamachi, Fuchu-shi, Tokyo 183-8512, JAPAN

って、道路に設置されたカメラの動画像をモニタに転送するシステムである。

3.1. ドメイン参照モデル

ドメイン参照モデルとは、対象ドメインをオブジェクト指向でモデル化するためのメタモデルである。図1にITS画像交換システムで用いたドメイン参照モデルを示す。Relayは、アプリケーション側と伝送部側があり、データを相互に通信し、通信プロトコルに依存する部分を隠蔽する。Entityはドメインのデータとそのデータに対する操作を提供する。Serviceはシステムのサービスを提供する。

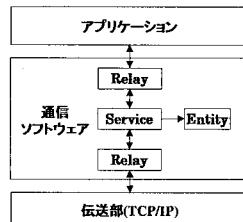


図1 ドメイン参照モデル

3.2. 開発プロセス

システム分析からオブジェクト指向設計までのフェーズで品質を作り込むための作業を示す。

システム分析

システム分析レベルで、ドメインの動向、仕様の流動性や曖昧さなどに影響を受ける部分（ホットスポット）を検討する。具体的には、ユースケースごとに、ユースケースを実行する上で仕様変更などにより影響を受けるホットスポットがドメイン参照モデルの中でどのモデル要素に相当するかを洗い出す。オブジェクト指向分析設計

オブジェクトレベルで、仕様変更などに影響を受けるホットスポットを検討する。具体的には、リスク要因ごとに、ドメイン参照モデルに基づき抽出したオブジェクトモデルでホットスポットを洗い出す。

本システムで提供するユースケースにおいて、リスク要因となる項目を洗い出し、この項目ごとに図1で示したドメイン参照モデルに沿って仕様変更などで影響を受ける部位を分析した。影響の大きさによって0～3までの重み付けを行い、ドメイン参照モデルに基づくオブジェクト群ごとに影響度の値を

合計した結果を図2に示す。

3:影響大、2:影響中、1:影響小、0:影響なしあるいは限りなく小さい

リスク要因	Relay										Service										Entity														
	属性	関数	コマンド	状態	メッセージ	接続	認証	認可	インポート	セーフティ	ハンドル	位置	速度	曲率	加速度	減速	停止	回転	直進	左折	右折	停止	回転	直進	左折	右折	停止	回転	直進	左折	右折	停止			
MS-001の初期化	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
属性の更新	2	1	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ASh機能の追加	0	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
ASh機能の削除	2	1	3	0	0	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
サービス登録	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
サービス登録	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
データ登録	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
データ登録	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
接続コントローラ接続	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	
ソースコードの変更追加	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ソースコードの削除削減	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
変数登録の削除	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
変数登録の追加	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
合計	33	28	19	4	20	3	44	23	28	8	24	11	10	3	3	5	4	4	4	5	0	7	2	7	8	2	3	8	2	4	8	9	0		

図2 ホットスポット分析

3.3. 結果

通信ソフトウェアの単体テストを開始して以降（試験日数20日目以降）は、他社との接続テストも含めてバグ発生件数は非常に少なく抑えることができた。まず、ソースコードレビューにより全部で約350件の指摘事項を洗い出した（試験日数1～20日目）。これらを修正した後に通信ソフトウェアの単体テストを実施した。単体テストで発見した不具合は約30件であった（試験日数21～30日目）。さらに、単体テスト時の不具合を修正した後に、アプリケーションとの接続を本格的に始めた（試験日数31日目以降）。接続テストでは、数件程度の不具合しか発見されなかった。

4. おわりに

本稿では、オブジェクト指向を用いて質の高いソフトウェアを開発するための技術を明らかにした。また、本アプローチによるソフトウェア開発の事例として、ITS画像交換システム開発を紹介し、本アプローチの有効性を明らかにした。オブジェクト指向による開発では、再利用性が注目されがちであるが、実際の開発では品質の問題も同様に重要である。

今後は、同種のシステムを多くのユーザーに展開するリピート開発を通じての本アプローチによる効果を評価していく。

参考文献

- [1]早瀬健夫 他：参照モデルに基づくオブジェクト指向フレームワーク開発手法、電気学会論文誌、Vol.121-C、No.3、pp.642-652、2001.
- [2]フィリップ・クルーシュテン：ラショナル統一プロセス入門、ピアソン・エデュケーション、1999.