オブジェクト指向モデルの頑健性の研究*

4J－10　北村嘉彦　井内雅樹　吉田敦　吉岡貴芳　磯田定宏†
豊橋技術科学大学 知識情報工学系‡

1　はじめに

これまでに数多くのオブジェクト指向方法論が提案されているが、その中でもI. Jacobsonが開発したOOSE[1]は高い評価を受けている。JacobsonはOOSEに基づいて開発したシステムは仕様変更に対して頑健であると主張している。本稿ではこの主張を検証するために仕様変更がモデルに与える影響を分析する。

2　OOSEについて

OOSEはユースケースに基づきオブジェクトと関連を識別する。またソフトウェアの保守性を高めるために図1に示すように3種類のオブジェクトを組み合わせてモデルを構築する。

a. 実体（entity）オブジェクト：システムが持つ属性とそれに対する操作をカプセル化したものである
b. 境界（boundary）オブジェクト[1]：システムとシステム外部に存在するアクタのインタラクションを仲介する
c. 制御（control）オブジェクト：複数のオブジェクト間の協調動作を制御する

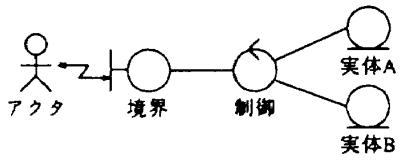

図1: ユースケースを実現する部分的なモデル

3　Jacobsonの仮説

ソフトウェアシステムのライフサイクルにおける各種要素の変更されやすさについて経験的に表1が言われている。最も変更されやすい要素はインタフェースおよび機能である。これに基づきJacobsonは次のことを主張している。

図1に示す構造でシステムを構築することにより、インタフェースの変更が境界オブジェクトに、また機能の変更が制御オブジェクトに局所化され、システムが頑健になる。

表1: 各種要素の変更されやすさ[1]

項目	変更されやすさ
アプリケーションのオブジェクト	低
永続性のある情報構造	低
受動的なオブジェクトの属性	中
振る舞いの順序	中
外界とのインタフェース	高
機能	高

4　実験

4.1　実験の方法

Jacobsonの仮説を検証するために次の実験を行う。実験対象は実用化されている交通情報処理システムである。このシステムは路上に設置されたセンサーで検出したデータをリアルタイムで処理する。このシステムにOOSEを適用してオブジェクト指向モデルを作り、改版にともなう仕様変更を加えモデルに対する影響を分析する。実験の手順を以下に示す。

1. システムの初版をOOSEに基づきモデル化する
2. システムの改版にともなう仕様変更を仕様書から分析し、その仕様変更を一つ前の版のモデルに適用して次の版のモデルを作る
3. 仕様変更を分類し、仕様変更の分類ごとにモデルに与える影響を分析する

4.2　結果

仕様書第2版で実施された仕様変更の種別ごとの発生頻度を表2に示す。インタフェースおよび機能の変更が他の種別の仕様変更と比較して多く、したがって表2は表1を支持する。

次に仕様変更ごとにモデルが受ける変更についての

---

* A Study on Robustness of Object-Oriented Models
† Yoshihiko KITAMURA, Masaki IUCHI, Atsushi YOSHIDA, Takayoshi YOSHIOKA, Sadahiro ISODA
‡ Department of Knowledge-based Information Engineering, Toyohashi University of Technology
[1] 境界オブジェクトはUML[2]で採用した名称である。OOSEではインタフェース（interface）オブジェクトと呼んでいる。

表 2: 各種仕様変更の発生頻度

種別	個数	割合
インタフェースの変更 *1	3	33.3%
機能の変更	4	44.4%
アルゴリズムの変更	1	11.1%
データの変更	1	11.1%

*1 装置間フォーマットの変更

分析結果を表3に示す。「IF の変更1」ではフォーマットの共通部が変更され、関係する境界オブジェクトが変更を受けた。

「機能の変更1」では制御オブジェクトが変更を受けただけでなく属性がオブジェクト化された (図2)。すなわち区間の属性である手動渋滞情報が二つの属性を持つオブジェクトになり、一つの区間が複数の手動渋滞と対応関係を持つことになった。

「機能の変更2」では実体オブジェクトが追加された。その結果、制御と実体の両方のオブジェクトに変更が及んだ。

表 3: 仕様変更の影響を受けたオブジェクトの数

仕様変更	実体	境界	制御
IF の変更 1	−	4	−
IF の変更 2	−	2	−
IF の変更 3	−	1	−
機能の変更 1	2	−	1
機能の変更 2	2	−	1
機能の変更 3	2	−	−
機能の変更 4	1	−	−
アルゴリズムの変更	1	−	−
データの変更	1	−	−

図 2: 機能の変更 1

5 考察

表3の「IF の変更1」と「機能の変更1, 2」以外はおおむね少数のオブジェクトに変更が局所化された。また「IF の変更1」については事前に共通部を汎化しておけば変更を一つの境界オブジェクトに局所化できた。したがってインタフェースの変更については Jacobson の仮説が成り立つ。

「機能の変更1, 2」については制御オブジェクトのみでなく実体オブジェクトも変更を受けるため仕様変更の影響が制御オブジェクトに局所化されなかった。また「機能の変更3, 4」については制御オブジェクトではなく少数の実体オブジェクトに仕様変更の影響が局所化された。すなわち機能の変更については Jacoboson の仮説は成り立っていない。

一般にシステムの機能は実体オブジェクトと制御オブジェクトの操作を組み合わせて実現される。したがって Jacobson の仮説が成り立つためには次の2つの条件が成立する必要がある。

(a) アプリケーション特有の仕様は制御オブジェクトにより、また個々のアプリケーションに依存しない分野共通の仕様は実体オブジェクトにより実現されている
(b) 保守段階に変更される仕様はアプリケーション特有の仕様である

これより Jacobson の仮説が成り立たなかった理由として次のいずれかが考えられる。

1) アプリケーション特有の仕様の変更だけではなく分野共通な仕様の変更が含まれていた
2) 対象としたシステムに制御オブジェクトがカプセル化すべきアプリケーション特有の仕様がなかった、あるいは少なかった
3) アプリケーション特有の仕様が制御オブジェクトだけでなく、実体オブジェクトにも割り当てられていた。

6 おわりに

本実験で扱った機能の変更については Jacobson の仮説は成り立たなかった。今後は仕様書第3版以降について同様な実験を進める。またこれと並行して制御オブジェクトの役割を解明し、それよりどのような機能の変更であれば制御オブジェクトのみの変更で対処できるかを明らかにする。

謝辞 実験に際して仕様書をご提供頂いた松下通信工業株式会社殿には心から感謝致します。

参考文献

[1] Ivar Jacobson, et al., *"Object-Oriented Software Engineering - A Use Case Driven Approach"*, Addison-Wesley, 1992.

[2] Rational Software Corporation, *"Unified Modeling Language"*, http://www.rational.com/.