

オブジェクト指向開発技術の問題点の分析と解決法の提案  
—エレベータ作業計画作成システムの移植・拡張の事例による—

4 C-7

末延 寿朗\* 本間 正喜\*  
\* (株) 日立ビルシステム

増田 英孝\*\* 笠原 宏\*\*  
\*\*東京電機大学工学部

1. はじめに

ソフトウェア保守作業にかかる労力を軽減するために、オブジェクト指向ソフトウェア開発技術[1]（以下オブジェクト指向技術）が有効であると言われているが、その適用は一般に難しく、広く開発現場で使用されている状況にはない。

本稿では、エレベータ作業計画作成のための比較的大規模な実用システムの移植・変更の経験に基づき、オブジェクト指向技術の適用上の問題点を分析し、これを段階的に適用していく方法を示す。

2. 既存システムの保守作業へのオブジェクト指向技術適用上の問題点

2.1 エレベータ作業計画作成システム

エレベータ作業計画作成システムは、(株)日立ビルシステムのエレベータ保全技術員の作業計画を作成するシステムである。本システムは1990年にワークステーション上で稼働するシステムとして構築され、現在はパーソナルコンピュータ上のシステムとして稼働している。この構築において、一部の機能（計画自動作成機能）にオブジェクト指向技術（データアクセス方法の統一・局所化）を導入した。その後、ユーザの要望や業務環境の変化に対応するための修正が継続的になされてきた。

さて、本システムは1996年にパーソナルコンピュータ（以下PC）への移植と機能変更・追加が行われた。この移植過程で発生したバグを分析し、以下の結果を得た。

- ①「データアクセス方法の統一・局所化」を適用した機能においては、バグの発生が他の機能と比較して桁違いに低い（図1）
- ②データの不正な初期化が原因で、不具合が多く発生している（図2）

この結果、本システムの全てにオブジェクト指向ソフトウェア技術を適用しておけば、移植・保守工数を低減できたと思われるが、以下の問題があり実現しなかった。

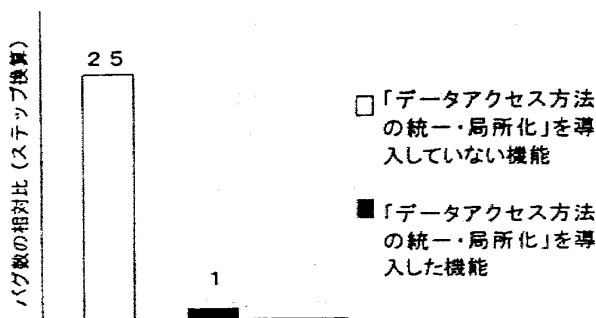


図1 PC移植時のバグ相対比 (ステップ換算)

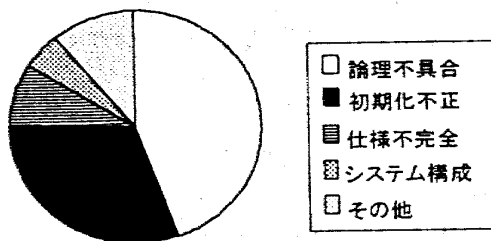


図2 PC移植時のバグの原因別分類

2.2 適用上の問題点

エレベータ作業計画作成システムに、オブジェクト指向技術を適用する際の問題点を下記に示す。

- ①オブジェクト指向言語で従来システムを書き換えるには、大きな工数がかかる。
- ②再利用可能、変更容易などの効果はすぐには上がらない
- ③オブジェクト指向は従来技術と考え方が異なるため、開発担当者の教育が難しい

2.3 上記問題点の解決法の提案

- ①従来手法からオブジェクト指向へ段階的に移行し、担当者を徐々に教育する
- ②各段階で必要な工数は、短期的で可能かつ見合った効果を上げられる程度に制限する

3. オブジェクト指向技術への段階的移行方式

以下①～④のように、段階的にオブジェクト指向技術を適用する具体的方法を提案する。なお、図の「アクセス関数」は、大域変数へのデータアクセスに特化した手続きで、「手続き」はアプリケーションの機能を提供するための手続

きである。以下の各段階に分けると移行も比較的容易で、ツールによる自動化が可能な部分もあるので、移行工数も比較的小さくできると考える。

①データアクセス方法の統一 (図3)

大域変数を隠蔽し、アクセス関数経由でデータを使用するように統一する。データへの不正なアクセスは発生しなくなる。

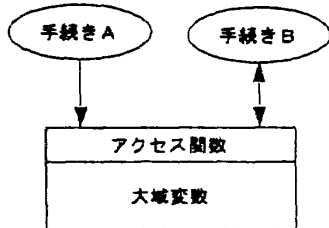


図3 データアクセス方法の統一

②データアクセスの局所化 (図4)

大域変数をグループ化し分割する。データ構造の変更による影響範囲が小さくなる。

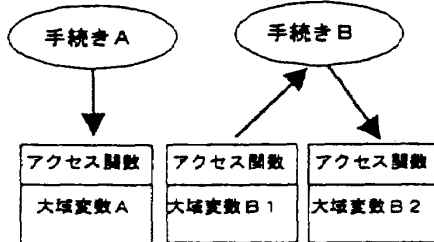


図4 データアクセスの局所化

③実装手続きの分割 (図5)

大域変数のグループ間にまたがる手続きのうち、分離できるものを分離する。手続きの変更による影響範囲が小さくなる。

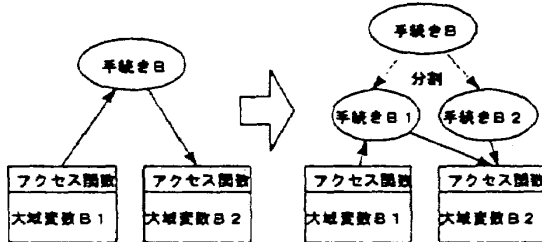


図5 実装手続きの分割

④オブジェクト化 (図6)

従来あるコードから、オブジェクトを構築する。分割した手続きとデータを一つのオブジェクトに実装し、オブジェクト化することが可能であると考えられる。

具体的にはオブジェクト生成時に大域変数からデータをセットし、消滅時に大域変数に書き戻す。手続きは③で分割したものを実装する。

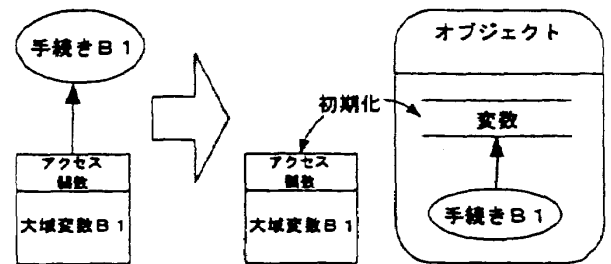


図6 オブジェクト化

4. エレベータ作業計画作成システムへの適用の検討

本解決法①により、バグの約30%を占めていた(図2)、データの初期化におけるバグの発生を回避することができる。実際①②を適用した計画自動作成機能でバグが他の機能の一分以下であった事実(図1)から、これは期待できる。システム全てへの解決法①の適用によって、プログラムの変更や新機能追加で発生するバグの件数を減少させることができる。

また、解決法②③の実施により、プログラム変更の影響範囲を小さくできプログラムの可読性が増すことで、PC移植時に発生したバグの最も大きな部分を占めるプログラム論理のバグ(図2)も減少させることができる。これも実際に解決法①②を導入した計画自動作成機能においてバグが桁違いに少ない事実から、解決法②もPC移植時のバグに対して効果を上げていると考えられる。

解決法④では、従来方式からオブジェクト指向技術へのほぼ完全な移行を可能とする。この過程で、プログラムの段階的な移行のみならず開発者の段階的な教育もほぼ完成すると考える。

5. おわりに

本方式により、従来技術によって開発したソフトウェアにオブジェクト指向技術を段階的に適用することで、スムーズに移行できる。

今後は、データおよび手続きを分割するための判断基準を提供することと、オブジェクト化によって得られる利点の検討を行う予定である。

謝辞

本方式のアイデアとその適用に関し、ご示唆ご指導頂いた、株式会社日立製作所の鶴田節夫主任研究員に感謝致します。

参考文献

[1]James Rumbaugh: オブジェクト指向方法論OMT, トッパン (1992)