

組込制御ソフトウェアへのオブジェクト指向設計手法の適用

井上勝博 赤嶺暁子 斉藤悦生

株式会社 東芝 研究開発センター
システム・ソフトウェア生産技術研究所

我々は、オブジェクト指向分析/設計手法を各種分野の実システムの開発に適用することにより、その有効性を調査し、実用化を進めている。今回、組込制御ソフトウェアの一つであるエアコンの制御ソフトウェアの設計に、オブジェクト指向設計手法として有力視されているBooch法を適用した。

本稿では、この経験を基に、組込制御ソフトウェアの開発におけるBooch法の有効性と課題を明確にし、その課題に対する解決案を提案する。特に、段階的な設計と設計結果の明記に重点をおき、Booch法の手順指針を詳細化し、表記法を拡張する。また、本手法による設計結果の有効性を、仕様変更への対応性と、従来の手法との比較の観点から評価する。

APPLYING THE OBJECT-ORIENTED DESIGN METHODOLOGY
TO EMBEDDED SOFTWARE SYSTEMS

Katsuhiro INOUE Akiko AKAMINE Etsuo SAITO

Systems and Software Engineering Laboratory,
Research and Development Center, TOSHIBA Corporation
70 Yanagi-cho Saiwai-ku Kawasaki, 210 Japan

We have been applying several Object-Oriented Analysis and Design methodologies such as Coad and OMT to the development of practical systems. As new application, we apply the Booch's Object-Oriented Design methodology to the development of air-conditioners which are embedded micro-computer systems.

In this paper, we describe advantages and disadvantages of the Booch's methodology from our experience of developing the software of air-conditioners. We also propose the improvements of disadvantages. They are the extension of the Booch's methodology and includes detailed process and extended notation. In addition, we evaluate the results of our design from the viewpoint of ease of modification and comparison with conventional methodology.

1. まえがき

オブジェクト指向ソフトウェア開発方法論は、現実の世界を計算機の世界に直接モデル化することにより、理解性と保守性に優れたソフトウェアを開発できるといわれている。

我々は、各種分野の実システムの開発にオブジェクト指向分析/設計手法を適用することにより、実問題における有効性と課題を調査し、実用化のためには何が必要かについて検討している。既に、適用対象としてトランザクション処理とユーザ・インタフェースの機能をもつ画像検索システム、画像ファイリングシステムにOMT法[Rumbaugh 91]を適用し、その有効性と課題を確認してきた。[山城 92]

今回、新たな分野として組込制御システム(エアコン)を取り上げ、この開発にオブジェクト指向設計手法(Booch法[Booch 91])を適用し、その有効性と課題を調査した。

本稿では、この適用におけるBooch法の有効性と課題を明示し、また、課題に対する解決策を提案することにより、制御システムの開発におけるオブジェクト指向設計手法について考察する。

次章では、適用の対象となるエアコンのシステムについて紹介する。3章では、今回用いたオブジェクト指向設計手法Booch法について概略を紹介する。4章では、適用におけるBooch法の課題と提案する解

決策を述べる。5章では、実システムの開発において得られた知見を、手法の観点と設計結果の観点からまとめる。

2. 適用対象

開発の対象とするシステムは、一般的なエアコンのシステムである。エアコンのシステム構成を図1に示す。説明のため実際のシステムのものを簡略化している。エアコンは室内機と室外機に分かれるが、ここでは室内機の制御ソフトを対象とする。

基本的な機能は、リモコンによって設定された冷房、暖房、自動等の運転モード、設定温度等を基に、室内の温度や湿度をユーザの希望する状態にすることである。そのために、室温、湿度等をセンサによって測定し、コンプレッサ、ルーパ、ファン等を制御する。

本システムは、要求に時間的な制約が含まれる制御システムである。本システムは、既に稼働中であるが、今回の開発における目的は、今後の仕様変更に対して容易に対応できるソフトウェアを再設計することである。そのために以下の点に着目した。

- 1)仕様変更への対応性の良いソフトウェア・アーキテクチャを設計すること。
- 2)自然言語レベルの仕様とソフトウェアの構造がうまく対応しており、仕様変更箇所の発見が容易であること。
- 3)設計結果を明確に表記できること。

3 適用手法

3.1 分析か設計か

オブジェクト指向に基づく方法論は、既に多く提案されているが、ほとんどの方法論は分析が主体である。ソフトウェアの開発は、分析、設計の順に行う。分析は、実世界のモデル化に重点をおいて、ユーザの曖昧な要求を明確化する。これに対し、設計は、計算機に実装するという観点から、分析モデルをベースにその実現方法を明確にする。分析時に設計の要素が含まれることは、設計の自由度を妨げる

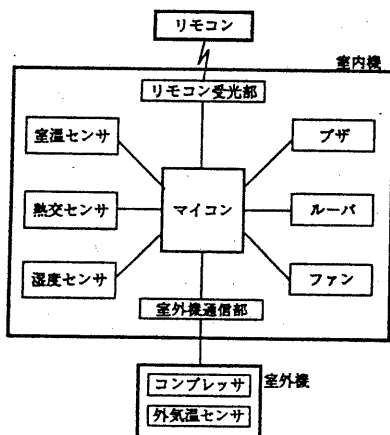


図1. エアコンのシステム構成

ことになり良くないと言われている。

今回の開発対象は、既にシステムが稼働中であり、要求自体はある程度明確である。また、ソフトウェアに対する要求には、実現の仕様や制約が多く含まれており、これを明記することが要求されている。このような観点から、ここでは、分析よりも設計における観点が重要であると考ええる。

設計方法論として、現段階で有効であると目される手法としてBooch法がある。Booch法は、表記法が豊富であること、実時間問題にも対応していること等の点でこの問題に適していると考えられる。ここでは、Booch法をベースとした手法でエアコンのシステムを開発する。

3.2 Booch法

まず、Booch法の概略について述べる。図2に、Booch法の基礎となるモデルと表記法の関係について図示する。括弧内が表記名である。

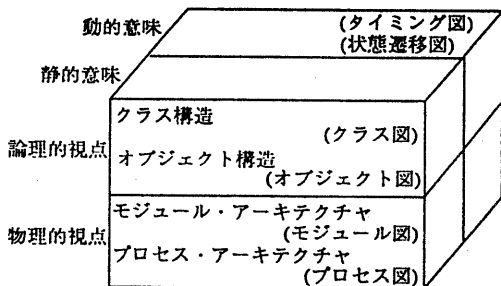


図2. Booch法のモデルと表記法

他の手法と比較した場合の大きな特徴は、多くの分析モデルがデータモデルを発展させたものであるのに対し、Booch法は、オブジェクト指向言語から発展したモデルを持つことである。したがって、言語に密接な設計レベルの詳細な表記法を持つ。ただし、表記の豊富さに比べ、手順、指針が明確ではないことも特徴である。手順は、大まかに以下のように定義されている。

- 1)与えられた抽象レベルでクラスとオブジェクトを識別する。

問題空間における抽象化の鍵を見つける。また、ある機能を実行するために要求されるふるまい

を与える重要なメカニズムを考える。

- 2)それらの意味を識別する。

各オブジェクトが互いに外側からどのように見えるか(インタフェース)を決定し、各オブジェクトのプロトコルを決定する。

- 3)それらの関係を識別する。

クラス構造(利用関係、継承関係等)とそれを簡潔化するパターンの発見、オブジェクト間の可視性の定義を行う。また、モジュール・アーキテクチャを決定する。

- 4)それらを実装する。

クラスとオブジェクトの表現に関する設計の決定、クラスとオブジェクトのモジュールへの割付け、プログラムのプロセッサへの割付け等を行う。

4 適用における問題点と対策

適用は、Booch法の豊富な表記法を有効に活用するように、前節の手順に沿って進めた。ただし、適用において以下の問題があった。

- ・手順に関する4つのステップは曖昧であり、各ステップにおいて、何をどのように行うかが明確ではない。特に、オブジェクトの識別、オブジェクトの意味の識別等が難しく、これらに関する明確な指針が必要である。

- ・Booch法の表記法では記述できない仕様がある。上記問題点への対策として、今回の適用対象に適した手順と指針を提案し、表記法の拡張を行う。その特徴は以下の通りである。

- 1)Booch法の4つのステップを活用し、それを詳細化した手順を与える。
- 2)オブジェクトの識別に関する明確な指針を与える。
- 3)オブジェクトの階層モデルを活用し、抽象的なオブジェクトから具体的なオブジェクトへと段階的に詳細化する手順を与える。
- 4)当初の要求として、自然言語レベルの仕様からプログラムがどのように構成されたかを追跡できることがある。この追跡性の維持に関して対

策を与える。

5) 記述できない仕様を明確化し、表記法を拡張する。

以下では、提案する手順に沿って、その手順における指針、表記法について述べる。

4.1 オブジェクトの識別

●問題の抽象的把握

最初のステップでは、問題の抽象化の鍵を見つけ、これをクラス、オブジェクトの候補とする。その指針として、仕様書より名詞を抜き出しこれを候補にする方法があるが、実問題では、仕様書より名詞を抜き出すことは困難である。実問題の仕様書は非常に複雑であり、仕様書中の名詞の数は膨大である。それを整理すること、その中から重要な鍵を抜き出すことは容易ではない。特に、今回の問題は、既存のシステムであり、仕様書には実現の詳細まで非常に細かく記述されており、問題の抽象的把握が難しい。

問題を抽象的に把握するには、改めて抽象化した仕様書を作成する方法が有効であった。この仕様書は、OMT法では「問題の記述」と呼んでいる。問題の記述の作成には、本質的に何が重要かに留意して、専門家とのインタビューを行うことが有効である。抽象化の指針としては、

- ・ハードウェアに直接依存する仕様を省く。
- ・ハードウェアの駆動方法、通信データフォーマット等がこれに含まれる。
- ・付加的な制御モード（機能）を省略する
- ・保守用の機能や異常処理を省く
- ・機種を限定する

などを挙げることができる。

●オブジェクトの識別の指針

オブジェクトの候補として名詞を挙げることは、指針としてはまだ曖昧である。ここでは、段階的にオブジェクトを識別する方法を提案する。

まず、第1段階のオブジェクトの識別の指針として以下の指針を用いる。

- ・システム構成を直接写像する。

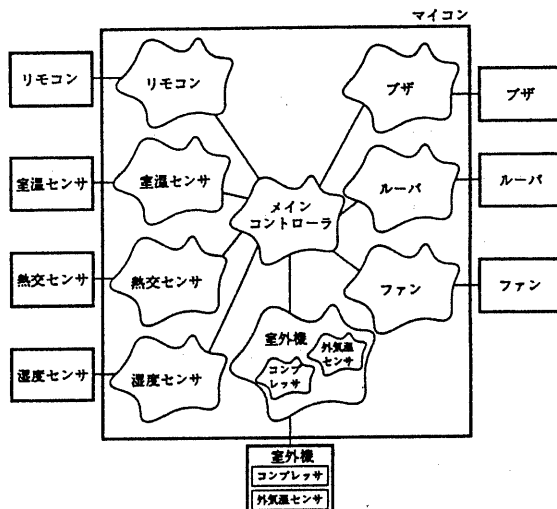


図3. システム構成の写像

図3にエアコンの事例を示す。この指針により、直感的なオブジェクトの識別が可能となる。

更に詳細なオブジェクトは、各オブジェクトの仕様の詳細化と共に識別する。詳細は4.3節に述べる。

4.2 オブジェクトの意味の識別

このステップは、オブジェクトのインターフェースと各オブジェクトのプロトコルを決定することと定義されているが、その明確な定義はない。ステップ3の「関係の識別」で行うべきことが、クラス構造の洗練（継承の導入と構造の見直し）とオブジェクト間の可視性の定義であると解釈すると、一般にオブジェクト指向モデルで定義すべき以下の項目は、全てステップ2で識別しなければならない。

- ・初期のクラス、オブジェクト間の関係の決定
(アソシエーション、メッセージ通信など)
- ・各クラス（オブジェクト）の仕様の決定
(属性、操作、振舞いなどの形式化)

これらを識別するには、多くの作業が必要であり、詳細な指針/手順が必要である。

これらの識別のために以下の手順を用いる。

1) オブジェクトの意味の識別

1-1) システムにおけるオブジェクトの役割を決める。

識別したオブジェクトについて、各オブジェクトがそのシステムにおいてどのような役割をなす

2) クラスの意味の識別

この段階で初めてクラスについて識別する。オブジェクト指向では、属性の決定、振舞いの形式化等のオブジェクトの内部仕様を識別することが必要であるが、Booch法ではこれをどのステップで行うかについての明確な指針がない。我々は、これらをこのステップで行う。

2-1) オブジェクトに対応するクラスを識別する。

オブジェクトの共通性からクラスを識別し、クラスの仕様を明記する。クラス間の関係(構造)はクラス図で、詳細仕様はテンプレートに記述する。

2-2) 各オブジェクトのふるまいの形式化

状態遷移図により形式化する。状態遷移図は、記述能力の点で、OMT法で用いているStatechart [Harel 87]が有効であり、我々はこれを用いた。

4.3 サブオブジェクトの識別

本手法では、オブジェクトの段階的な識別による設計を行う。責任の大きなオブジェクトは、サブオブジェクトへの分割の候補となる。

例えば、エアコンの事例では、「メインコントローラ」がこの候補である。「メインコントローラ」の振舞いの形式化を試みると、冷房、暖房等の運転モードがその振舞いの主要因となっていることがわかる。また、運転モード決定のための処理の流れと、デバイスオブジェクトの制御のための処理の流れの意味の異なる2つの流れがあることがわかる。この2つを異なるオブジェクトに分割することを試みる。

ここでは、[モード]と[基本制御]オブジェクトに分割した。その結果、理解し易く自然なモデル化が行えることが確認できた。

処理の流れを制御スレッドと解釈すれば、以下のことが言える。

- ・オブジェクト内部に存在する複数の制御スレッドは、オブジェクト抽出の指針となる。

したがって、振舞いの形式化は、オブジェクト識別の重要な要因となる。

4.4 オブジェクト間の関係の識別

このステップでは、継承を導入し、クラス構造を洗練する。また、オブジェクト間の可視性について明確化する。

4.5 オブジェクトの実装

このステップでは、以上の結果をプログラム言語で記述する。本ステップでは、以下の問題があった。すなわち、実装言語としてC++を選択したが、Booch法による設計結果をC++で記述しようとした場合、次の点が適合しない。

- 1)メッセージパッシングメカニズムとC++のサブルーチンコールのメカニズム
- 2)アクティブなオブジェクト(並行オブジェクト)の実装

上記2)に関しては、マルチタスクのメカニズムを導入すれば実現できる。我々は、C++にマルチタスクのモニタを組み込むことにより、並行オブジェクト

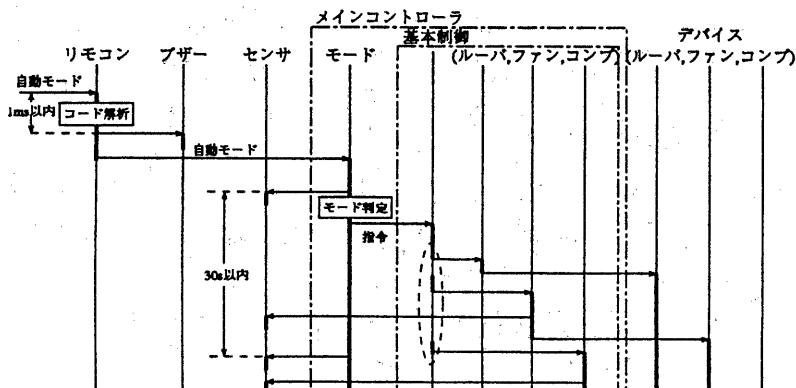


図5 メッセージトレース図

を実現したが、このとき、オブジェクトとタスクをどう対応させるかが問題となった。この対策として、以下の観点を明確に記述できる図を用いた。

- ・クラスとタスクの関係
- ・並行オブジェクト間のメッセージ通信
- ・メッセージ通信のサブルーチンコールによる実現
- ・オブジェクト間のサブルーチン・コーリング・シーケンス

5. 評価

5.1 手法のまとめと考察

Booch法をベースとした設計手法でエアコン制御ソフトの設計を行った。本手法は、Booch法の長所を活かし、短所を補うかたちでBooch法を拡張した。以下の本手法を手順と指針、および、表記法の観点でまとめる。

●手順と指針

Booch法は、手順および指針に関しては、大まかに定義されているが十分ではない。ここでは、以下の点に特徴のある手順と指針を提案した。

- 1) オブジェクトの階層構造によるモデルを有効に活用し、抽象的なオブジェクトから具体的なオブジェクトへと段階的に詳細化する手順を与えた。これは、オブジェクト識別を容易にし、複雑な仕様の理解に有効であった。
- 2) オブジェクトの識別に関する明確な指針を与えた。特に、実世界の写像を重視し、実体をオブジェクトに対応させることにより、直感的なオブジェクト抽出が行えた。
- 3) 抽象的な問題の記述を作成し、それを各オブジェクトに分配することにより、オブジェクトの非形式的な仕様を作成する方法を用いた。これにより、自然言語レベルの仕様からソフトウェアへの追跡性が得られた。また、オブジェクトの形式的な仕様の作成にも有効であった。

図6に詳細化した手順をまとめる。

●表記法

Booch法は、設計の要素を記述するための豊富な表記を持つ。十分に理解し、整理すれば有効に活用で

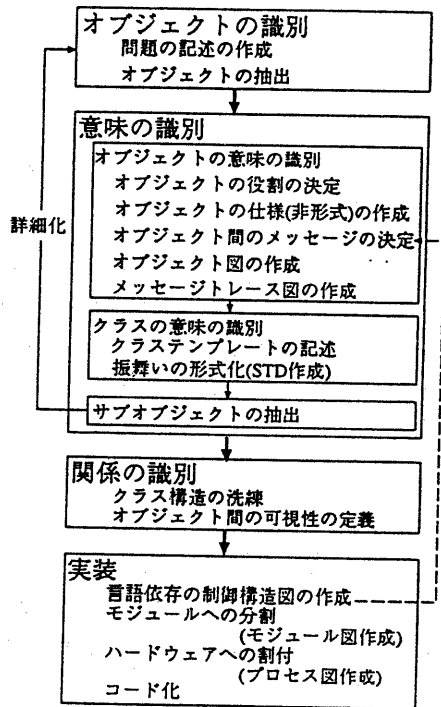


図6. オブジェクト指向設計手順

きる。本適用においては、以下の表記法が有効であった。

- ・メッセージ通信によるオブジェクト間の関係定義
- ・アクティブオブジェクトの明記
- ・メッセージのタイミング図
- ・階層表現の可能なオブジェクト図

ただし、以下の点が記述できず拡張した。

今回の問題では、Booch法の表記で以下のものが記述できず拡張した。

- ・要求に含まれる時間的制約
- ・言語依存の制御構造
(オブジェクトとタスクの関係、メッセージ通信とサブルーチン呼出の関係など)

また、今回の手法は、段階的な詳細化を意識したが、これを明記するために、表記法にも階層的な記述が望まれる。Booch法のオブジェクト図は、オブジェクト間の関係を階層的に表記することに有効であった。その他の図に関しても階層化の検討を行う方針である。

5.2 設計結果の評価

本節では、設計結果として得られたモジュール構成を、仕様変更への対応性から評価する。また、構造化手法により得られたモジュール構成と比較することによりオブジェクト指向設計手法の有効性を評価する。

●仕様変更への対応性

実際の製品における仕様変更の影響が、設計結果として得られたオブジェクト構成にどのように及ぶかについてその影響範囲を調査した。仕様変更は、全40件であり、それを当てはめたところ、次の結果が得られた。

・新規オブジェクトの追加	2件
・オブジェクトの削除	0件

・単一オブジェクトのみの変更で対応	
全変更項目の87.5%	
・複数オブジェクトに影響	
全変更項目の12.5%	

上段がオブジェクトの追加/削除の件数である。今回の仕様変更では、新規オブジェクトの追加が2件であり、削除されたオブジェクトはない。新規オブジェクトの追加は、ハードウェアの変更によるものである。システム構成の写像によりオブジェクトを識別したことにより、ハードウェアの変更に対して自然に対応できる。

また、仕様変更の影響範囲はほとんどが単一オブジェクトの変更で対応できる。その他の変更もほとんどがせいぜい2つのオブジェクトの変更で対応でき、変更の局所化が実現できている。

●構造化手法との比較

以前に構造化手法に基づいた手法で同問題の構造化を行った[井上 91]。この手法は、既に実装済みのプログラムからデータフロー図を作成し、それをグルーピングすることによりタスクの構造を得る。この手法と今回の手法の設計結果を比較した。

オブジェクト指向手法の利点は、「モジュール決定の指針が明確であり、理解性に優れる」ことであり、欠点は、「実装上の資源効率」である。

また、構造化手法の顕著な問題点は、「実装に依

存したモジュールが上位に現れる」ことである。これは、実装方法を変更することにより根本的なモジュール構造が変更されることになる。これに対し、オブジェクト指向手法では、実装に依存しない安定性のあるモジュール構造が得られる。

5 あとがき

オブジェクト指向設計手法であるBooch法をエアコン制御ソフトの開発に適用することにより、実問題における手法の課題をあげ、その対策として、詳細な手順と不足している表記法について提案した。また、得られた設計結果を、仕様変更への対応性と構造化手法との比較の観点から評価した。今後、得られた知見を基に支援環境の検討を行う予定である。

参考文献

- [Booch 91] Grady Booch. "Object-Oriented Design with Applications", Benjamin/Cummings, 1991
- [Chen 76] P.P.S.Chen. "The Entity-Relationship Model - toward a unified View of Data", ACM Transactions on Database Systems 1, 1, 1976
- [Harel 87] David Harel. "Statecharts: a visual formalism for complex systems", Science of Computer Programming 8, pp.231-274, 1987
- [Rumbaugh 91] J.Rumbaugh, M.Blaaha, W.Premierlan, F.Eddy, and W.Lorensen. "Object Oriented Modeling and Design", Prentice Hall, 1991
- 邦訳：羽生田栄一監訳，「オブジェクト指向方法論 OMT モデル化と設計」，トッパン，1992
- [井上 91]井上，西村：マイコン組み込みシステムのためのソフトウェア・アーキテクチャ設計の一手法，情報処理学会研究会報告，Vol.91，No.66，91-SE-80，p.9-16，1991
- [山城 92]山城，手島，井上，前田：画像検索システムへの OMT 手法の適用，情報処理学会研究会報告，Vol.92，No.16，92-SE-84-3，1992