Ａｄａを用いたオブジェクト指向設計技法

4S－4

小林正男　富士通㈱システムラボラトリ

筒井良夫　大興電子通信㈱，長崎総合科学大学講師

１．はじめに

宇宙ステーション「フリーダム」はNASAを中心とした国際協力の下で開発が進められており，我が国はフリーダムに取り付けられる日本実験モジュール（JEM）の開発を担当している．国際共通化要求に従い，JEM に搭載される運用管理ソフトウェアは Adaで開発される．我が国に於いて，Adaを用いたソフトウェア開発の実績は乏しく，Ada によるソフトウェア開発には多くの課題が存在するが，先ず第一に対処すべきは設計技法の確立である．

Ada の言語仕様には，1970年代に提唱されたソフトウェア・エンジニアリング的概念が取り入れられている為これらを効率的に用いることが必要であり，それにはオブジェクト指向設計技法（OOD）が適している．Ada 研究者達の論文等に於いて，彼らはおしなべて OODを推奨しており，我々の調査／検討でも同様の結論を得ている．OOD は Adaとの親和性が高く，ソフトウェア・エンジニアリング的にも優れた技法であるが，オブジェクトの決定方法に困難さがある．そこで，我々はこの問題に対処すべく，大規模開発を念頭に置いた効率の良い理解容易な設計技法であるWOOD（Widely Applicable OOD)を開発した．

２．ＷＯＯＤ

OOD で困難な点は「良い」オブジェクト／クラスの識別だがWOODでは制御フローを含む Ward 流のデータフロー図（拡張DFD)を下敷きに，問題空間の実体との対応に優れ（理解し易い），階層性や複雑性などに配慮した「適切な」オブジェクト群，クラス群の識別を試みる．このように拡張DFD が設計過程に直接取り込まれて，オブジェクトやクラスの識別に重要な役割を果たす．また制御変換を含むDFD を用いる為，制御系の設計では「制御機構」を容易に表現出来る．

オブジェクト抽出に関してWOODは柔軟な立場をとる．実世界の実体からの意味の移入が容易な「優れた」オブジェクトから，システムの構築上の都合に過ぎない「偶然の」ものまで，その抽象性に幅を認める．これは，OOD の利点（抽象性の良いオブジェクトの識別による，理解容易性，再利用性等の向上）を認めた上で，実開発に耐え，広範囲に適用可能な設計法とする為である．

【設計過程】

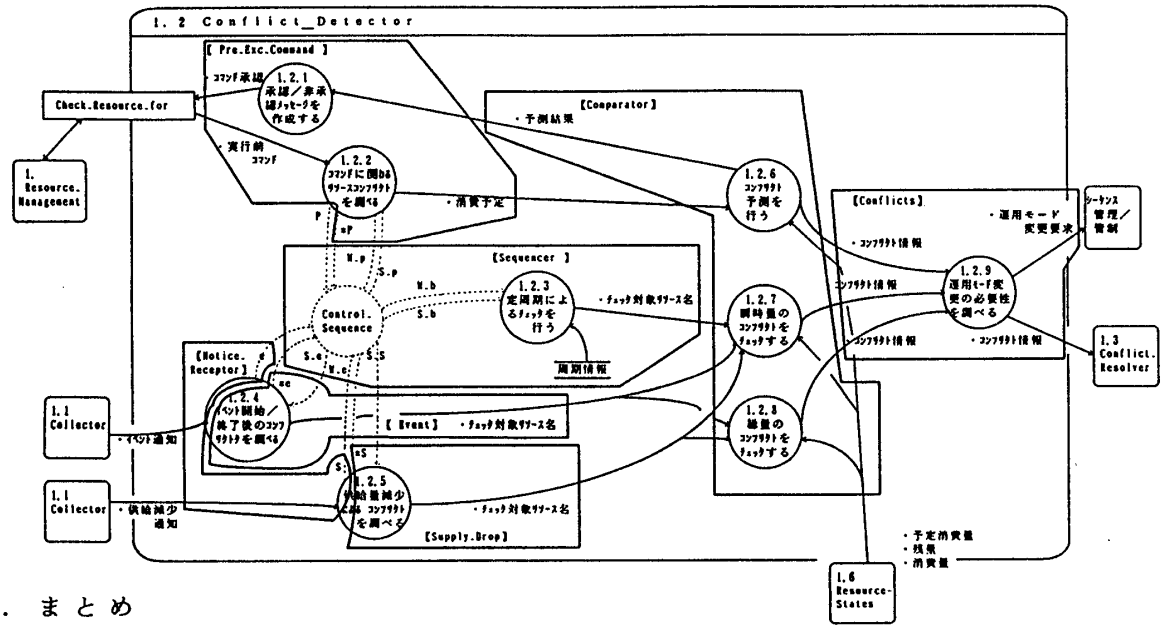
データフローや制御フローを手掛かりに問題の分割とオブジェクトによる解の構成をトップダウンに行い，最終的に階層化されたオブジェクト群によって解を構成する．この過程は，あるオブジェクトから幾つかの子オブジェクトを導く基本的なステップからなる．このステップは親オブジェクト（外部インタフェースと機能が定義された「仕掛品」のオブジェクト）に適用され，以下(1)～(4)に述べるサブステップによって，親自身を「完成」（対応する Adaパッケージのコード

Object Oriented Design Method Using Ada

Masao KOBAYASHI　Yoshio TSUTSUI

FUJITSU LIMITED　DAIKO LIMITED

が定まるという意味で）し，同時に仕掛品の子オブジェクト群を生む．
(1)親オブジェクト内データフロー，制御フローを DFD，制御変換，これを定義する状態遷移図等によって明らかにする．(問題の理解と設計上の決定が混在する)．
(2)拡張DFD の上で，流れるデータ，制御変換，データストア，プロセス記述などを吟味し，親オブジェクト内に存在すべきエンティティ（解空間の実体）を識別する．ここでは実空間のエンティティとの対応の率直さなど，解の分かり易さに力点が置かれる．下図で，プロセスを囲う太線は識別されたエンティティを表す．この囲い込みの善し悪しが，オブジェクトの抽出に大きく影響する．
(3)エンティティ群をもとに設計の複雑性，制御の階層性などに配慮し，抽象性の「良さ」とのトレードオフを経て子オブジェクト群とクラス群を識別する．またこれらの各々について提供手続き，主要な内部ステートのデータ構造を定めるとともに，子オブジェクト間のインタフェースを決定する．
(4)以上(1)～(3)の決定を Adaに変換し，コンパイルを行うことによって子オブジェクト間及び親子間のインタフェースの静的な確認を行う．



3．まとめ

前節で述べた様に，WOODでは，構造化分析の結果の拡張DFD を直接用い，特定の観点でオブジェクトの候補を識別する．その為，拡張DFD の視覚的効果と相俟って，対象ソフトウェア全体の振る舞いを見据えつつ，オブジェクト候補を容易に識別する事が出来る．一方，構造化分析からアーキテクチャ設計に亘る作業上の一貫性を確保し，思考過程を残す事が出来る為，設計者の設計方針／意図の理解が可能となり，コーディング／メンテナンス作業の効率向上が期待出来る．WOODは Adaを用いることを前提としているが，オブジェクトの候補を識別する方法はプログラミング言語に依らない部分である為，Ada 以外の言語でも適用可能である．今後，一般的オブジェクト／クラス識別への応用も検討して行きたい．

〔参考文献〕

Paul,T.Ward., The Transformation Schema: An extention of the Data Flow Diagram to Represent Control and Timing, IEEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL.SE-12,FEB. 1986

General Object Oriented Software Development, GSFC Report, AUG. 1986.