

5 J - 5

Ada 向けプログラム設計支援ツール Xspd*Ada の試作

大瀧 潔 斎藤 正則 山上 博三

日本電気(株)

1 はじめに

Ada は、米国防総省が開発した言語で、1987年にISO標準化、そして1990年にJIS標準化されている。Adaは、強型付け、パッケージによる情報の隠蔽及び抽象化、実行時における例外検査機能といった各種特徴を持ち、優れた生産性・保守性等を提供する。しかし、Adaには、もう一つの重要な特徴がある。それは、米国防総省の要請により、当初から、コンパイラ及びリンカだけでなく、Ada言語を利用したソフトウェア開発を支援する開発環境を提唱し、その実現を要求している点である。

Ada開発支援環境はAPSE(Ada Programming Support Environment)と呼ばれ、支援系の要求条件を示した論文である、いわゆるSTONEMAN[1]で提唱された。APSEは、Adaで記述する大規模ソフトウェア開発での方法論及びその方法論による開発を支援する各種支援系の重要性並びにAPSEの基本的枠組みを提案している。

Adaを利用したソフトウェア開発を行う場合に、各種の方法論とその支援ツールが提案されている。それらのほとんどは各モジュール/ライブラリの分割及びインタフェースの設計に重点がおかれ、各モジュール/ライブラリ内部の制御構造を設計する支援を行っていない。プログラムの制御構造を設計する手法の一つであるSPD[2]は、単純化された記述、簡易な記号の使用を通して、プログラムそのものの制御構造を明らかにし、効率的で読解性にすぐれたプログラム設計を実現する。

本論文は、当社におけるAPSEの一環として試作したAdaによるSPD記法支援ツール(Xspd*Ada)に関し、報告するものである。

2 Adaを指向した開発環境

私達は、Ada開発支援環境を図1のように構築していた。下流工程の各種ツールは、UNIXワークステーション上ですでに実現されているNACS(NEC Ada Compilation System)に、幾つかのツールを補う一方、上流工程CASEツールには、市場に紹介されているAdaを支援するCASEツールを導入した。

APSEを示した図1を参照すると、上流工程のツールと下流工程のツールの間にギャップが存在することが判明する。すなわち、設計とコード化の境界部分、いわゆる詳細設計、の支援がまったくないか、弱いという問題点があった。

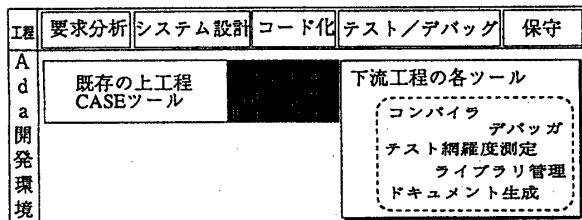


図1: NECのAda開発環境の一例

Xspd*Ada: A prototype system for Ada programming support tool
 Kiyoshi OTAKI, Masanori SAITOH, Hirozoh YAMAGAMI
 NEC Corporation

3 詳細設計の支援

ソフトウェアの設計工程は、一般に次のような段階に分けてとらえられることが多い。[3]

- 方式の設計
- インタフェースの仕様化
- 構成要素の仕様化
- アルゴリズムの設計(詳細設計)

ここで詳細設計とは、最後のアルゴリズムを設計する段階を言う。詳細設計より前の各段階については、Jackson[4]やConstantineとYourdon [5]さらに、オブジェクト指向に関してBooch[6]の著作等に示される様な方法論が確立又は広く認められており、積極的にこれらの方法論を支援する上流工程のCASEツールが多数存在する。しかし、詳細設計の段階においては、設計技法としてSPD・HCP・PAD等のチャート類が知られているにも関わらず、これらの技法を支援する有効なツールが存在しなかった。

詳細設計の支援ツールが市場等に登場しない理由の多くは

1. モジュール分割で、充分小さな大きさになっていればわざわざ小さい単位に対して各種チャートを書くのは無駄である。
2. (特に英語文化圏の人々にとって、)Adaソースコードは、すでに英文の形態に近く、充分わかりやすい。
3. バグ等でソースコードとの整合性をとるための費用がばかにならない。

といったものである。しかし、私達は各種のソフトウェアを開発する現場から、

1. ほとんどのモジュールが実際には当初目標とした見開き2ページに納まり切れない長さになってしまう。
2. 長いソースコードは、制御構造が一見複雑に見える場合があるが、チャート化することによって、見通しを良くすることができる。
3. 日本語文化圏の人間にとっては、ソースコードは特殊な記号列であり、ソースコードが読みやすいとは一概に言えない。
4. ソースコードとの整合性をとるためのツールを開発することによって、容易に整合性を取ることが可能となる。

との声を良く聞く。わが国において、詳細設計の各種チャートが使用され続けているのはこれらの理由にあると考えられる。

一方、詳細設計支援ツールとして、APSEのツールを捕らえた場合、以下に示す様に、Adaであるがゆえに、積極的に適用する必要があると考えられる理由がある。

- 現在のAdaでは、その言語仕様により、(たとえ注釈部分であっても)ASCII7ビット文字以外の文字を使用してはいけないとされている。ソースコードに対応するチャートに日本語を使用することにより、注釈の代替とすることが出来る。
- 一般にAdaコードは、CやFortranといった他の言語より長くなり、かつ構造のネストが深くなり、プログラムの制御構造が見にくいいため、チャートを作成することにより制御構造を明らかにできる。

4 Ada向けのSPD記法

SPD記法は、プログラムの制御構造を示す標準的なチャートである。この記法は、単純化された記述・簡単な記号の使用を特徴としているが、今回Ada向けに幾つかの記号を導入し、記述法を拡張した。

Ada言語の特徴を考慮し、特に以下の方針に基づいてSPD記法を拡張している。

1. 従来のSPD記法についての基本的な枠組みは変更しない。
2. 仕様記述に関しての記法を追加する。
3. 階層化されたブロック(例外条件ハンドラ等を含む)を表現可能とする。
4. タスク関連の記述に対しての記法を追加する。

SPDは、プログラムの制御構造を明らかにするのが主目的であるため、Adaの仕様記述部分等については詳細には記述させていない。これは、各種仕様記述については、上位の設計で決定されるべきものであるためである。

図2にSPD記述例を示す。

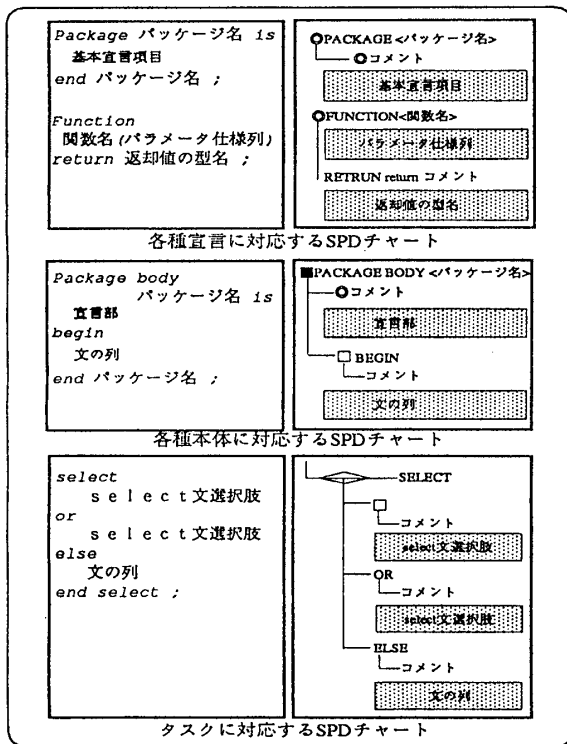


図2: SPD記述例

5 Xspd*Adaの構成

Adaによる詳細設計支援ツールXspd*Adaは、Xウィンドウを利用したUI構築ツール「鼎」上に実現され、以下に示す部品から構成される。

1. *xspde*: Adaを指向したSPDチャートエディタ
2. *xspdtoAda*: SPDからAdaプログラムへの変換ツール
3. *xAdatospd*: AdaプログラムからSPDへの変換ツール
4. *xspdpr*: *xspde*又は*xAdatospd*で作成されたSPDファイルからドキュメントを生成するツール

これら部品の構成を図3に示す。

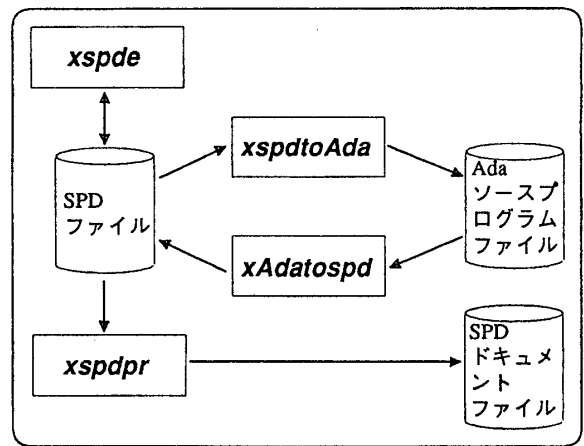


図3: Xspd*Adaの構成

特に重要な機能として、*xspdtoAda*と*xAdatospd*がある。これらは、SPDとプログラムの相互変換を行うもので、この機能により、SPDとソースプログラムの対応を容易にとることができ、設計書とプログラムの不一致を防ぐことを可能としている。

6 適用結果

本ツールをAda関連業務に適用した結果から、以下に示す様な利用を行うと効果があることがわかった。

- 分岐が多い場合や、構造のネストが深い論理を表現する場合、SPDの効果が発揮され、見通しが良くなる。
- *xAdatospd*を用い、SPDが存在しないプログラムをSPD化することにより、プログラムの制御構造を把握しやすい。
- プログラムのレビュー若しくは机上検査を行う場合、チャート形式で見ることができるので、誤りを見つけやすい。

7 おわりに

現在のところ、本ツールは適用評価を行っている状況である。最終評価は、これらの報告を待たなければならないが、現在の時点で、特にレビュー時の効用が指摘されている。また、SPDとソースの対応をとりやすいことから、バグ等に伴う修正の容易さも効果を上げているようである。今後、上流工程CASEツールとの連動を行う予定であり。それによりさらに大きな効果が期待できる。

最後に、本研究にご支援をいただいた方々にお礼を申し上げます。

参考文献

- [1] J.Buxton : Department of Defence Requirements for Ada Programming Support Environments, Stone-man, 1980
- [2] M.Azuma, T.Tabata et. al. : SPD: A Humanized Documentation Technology, IEEE Transactions on Software Engineering Vol. SE-11 NO.9, 1985
- [3] I.Sommerville and R.Morrison : Software Development with Ada, Addison-Weasley, 1987
- [4] M.A.Jackson : System Development, Prentice-Hall, 1982
- [5] L.L.Constantine and E.Yourdon : Structured Design, Prentice-Hall 1979
- [6] G.Booch : Software Engineering with Ada, Addison-Weasley, 1983