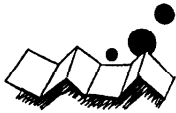


## 解説



## Ada と支援環境†

寛 捷 彦††

## 1. 経 緯

米国防総省 (DoD) の共通高水準言語計画は、1975年に、省内で今後用いられる組込み型計算機システム向けに共通の高水準言語を定めることを目的として始められた。言語に対する要求仕様は、何段階かを経て、1978年6月に STEELMAN としてまとめ上げられた。言語仕様の初版は、1979年5月に完成し Ada という名が定められた。初版の仕様に対してさまざまな意見、使用経験が集められて、1980年7月にその改訂版が作られている。一方、Ada の翻訳系検定機能 (ACVC\*) の開発は、1979年9月に開始されているし、陸軍・空軍は、それぞれ独自に、Ada 翻訳系の作成に着手している。

こうした Ada 言語そのものについての活動と並行して、Ada による作譜過程を支援するための環境整備の活動が始められた。支援環境に関する要求仕様は、1978年に、SANDMAN、すぐ続けて PEBBLEMAN として公表され、翌 1979年1月にはその改訂版が出された。ここまでの作業は、主として DoD 内部で行われていたが、この年から John Buxton\*\* が参画して精力的に要求仕様の統練に当たり、数度の改訂のあと 1980年2月に要求仕様書 STONEMAN が世に公表された。

DoD の各種企画は、要求仕様書の確定・仕様書の確定・具現化\*\*\* という手順を踏むのがふつうであり、しかも、各段階では競争入札が行われている。この意味からすると、Ada 支援環境は、やっとその第1段階が終わったばかりである。そこで、本解説は、もっ

ばら STONEMAN の内容にあてておくことにする。

## 2. 要求仕様書 STONEMAN

STONEMAN では、Ada 言語による作譜過程の支援環境を APSE\* と呼ぶ。STONEMAN は、したがって、APSE が用意しなければならない機能を定めたものである。

その背景としての思想は、つぎのようである。

(1) 支援環境は、長期にわたって譜構\*\*の支援を行うものであり、その信頼性を高めるのに役立つものである。

(2) Ada 言語設定の目的からして、APSE が支援すべき企画の対象は、組込み型計算機系の機構\*\*\* および譜構である。しかしながら、このことはその企画の遂行が対象計算機\*\*\*\* だけを用いて行われることを意味しない。多くの場合、企画の遂行のためには、大型の計算機が用いられるであろう。したがって APSE も、親計算機\*\*\*\*\*——対象計算機という構成を想定する。

(3) APSE は、後立てとなる料基\*\*\*\*\* と、道具の組とから成る。

支援環境というものは、言語に比べて、未開拓の分野である。つまり、学界においても、その思想・構成に対して裏付けのある定説というものできていないといつてよからう。特に、支援環境をより高度なものとするには、作譜の方法論が大きく影響を与えるが、方法論についても諸説が並立しており、そのひとつを選択し固定することは困難である。

一方 DoD 内部の事情もある。空軍は、Ada 計画に積極的であり、支援環境についてもそうである。おそらく裸の計算機上に、新しく APSE を構築するであ

† Ada and its Support Environments by Katsuhiko KAKEHI (Department of Mathematics, Faculty of Science, Rikkyo University).

†† 立教大学理学部

\* Ada Compiler Validation Capability

\*\* John Buxton IFIP の専門委員会 TC2 (Programming) の英国代表委員。また同委員会下の研究団 WG 2.3 (Programming Methodology) の一員でもある。

\*\*\* implementation

\* Ada Programming Support Environment

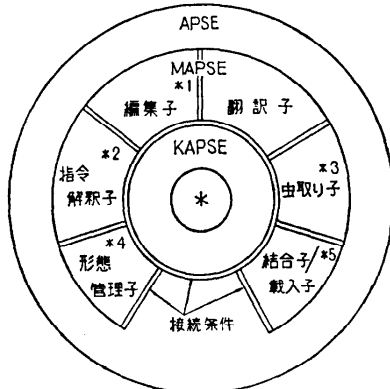
\*\* software

\*\*\* hardware

\*\*\*\* target computer

\*\*\*\*\* host computer

\*\*\*\*\* data base



- \* 親計算機の機構・譜構
- \*1 editor
- \*2 command (JCL) interpreter
- \*3 debugger
- \*4 configuration manager
- \*5 linker/loader

図-1

ろうと予想される。陸軍は、Ada 言語の導入について非常に積極的であるが、支援環境についてはそれほどでない。おそらく、既存の計算機の機構・譜構の上に Ada 翻訳子\*と若干の補助道具を作ることで Ada を実用化していくであろう。これに対して、海軍は Ada 計画そのものに消極的であり、既存の言語と譜構とに満足している。したがって、海軍に対しては Ada 言語・翻訳子から始めて順次技術移入を行うことを考えなければ、DoD の共通言語計画そのものに破綻を生じかねない。

こうした事情を考慮して、STONEMAN の要求仕様は、つぎのように組み立てられている。まず、APSE が満たすすべての条件を掲げる。APSEは、究極的には Ada 言語で記述されることが望ましいが、すべてをそうするように要求はしない。それどころか、親計算機に用意されている操作系\*\*を活用する形の具現を想定する。すなわち、APSE の各機能を Ada で具現するために必要となる基本機能の組を定め、これを具現する方法は問わないことにするのである。この基本機能の組を、核環境 KAPSE\*\*\*と呼ぶ。KAPSE は、したがって、仮想操作系であるといってよく、その接続条件\*\*\*\*のみが与えられる。APSEの各道具は、この KAPSE の接続条件の上に、Ada 言語を用いて作成

- \* compiler
- \*\* operating system
- \*\*\* Kernel APSE
- \*\*\*\* interface

する。こうして、APSE の可搬性\*を保証しようというのである。

つづいて、APSE が含んでいなければならない最小の道具の組を規定する。これを最小環境 MAPSE\*\*と呼ぶ。MAPSE は、したがって、それだけでもひとつの APSE として認められるものである。

以上の構成は、左の図のように表示されている。

Buxton 自身の言によれば、STONEMAN は、DoD 内部に対して APSE の外観を伝え、学界に対して APSE の内観を伝えるに足るよう書かれているという。その章立ては、つぎのようである。

1. 序, 2. 概観, 3. 一般原則, 4. APSE の要求仕様, 5. KAPSE の要求仕様, 6. MAPSE の要求仕様, 7. APSE の要素.

### 3. 支援環境 APSE

支援環境に対しては、高品質・単純性を要求する。それは、対象となる譜構の最初（仕様設定）から最後（保守）までを通して支援し、そうした企画\*\*\*の管理をも支援するものでなければならない。利用者にとって使い勝手の良いもので、各種の約束事は統一的に作られていなければならない。APSE が可搬性を持つことは当然であるが、さらに、その上で実行される各企画もほかの APSE 上へ移せるよう作られていなければならない。APSE は堅牢でなくてはならず、任意に組合せできる単位から十分に集積されたものであり、しかも追加・変更が容易にできるように作られていなければならない。こうした APSE の構築にあたっては、高性能・大記憶の親計算機を利用することになろう。

以上が、一般原則である。

APSE の後立てとなる料基としては、つぎのような要件を満たすものを要求する。料基には、Ada の原譜\*\*\*\*、翻訳後の（中間語・機械語による）譜、備譜\*\*\*\*\*およびその仕様といったものを含め、さまざまな情報を貯えられること。こうした各情報を「物件\*\*\*\*\*」と呼ぶことにすると、各物件には一意名を付けて参照できるほか、引用・変更等の権限制御ができること。いくつかの物件を集めた「集団\*\*\*\*\*」を指定でき、集団が識別できる程度の不完全名を与えたときは、その集

- \* portability
- \*\* Minimum APSE
- \*\*\* project
- \*\*\*\* source program
- \*\*\*\*\* library programs
- \*\*\*\*\* "object"
- \*\*\*\*\* "version group"

団中であらかじめ指定しておいた物件が自動参照できること。ほかの物件の結合形態を示すための物件も、料基に含めることができ、この「形態\*」物件を通じての不完全名による参照を許すこと、あらかじめ指定しておいた物件に関しては、それが生存する限り、それに関連するほかの物件も自動的に保存すること。料基中の物件を、外部の企画に関連して「区分\*\*」に分けることができ、この区分単位に引用・変更の権限制御をすることも可能なこと。料基についての管理情報を APSE の道具へ知らせることができるようになっていたこと。

APSE で用いる種々の接続条件は、親計算機に依存しない形で、明確・簡潔な原則に基づいていることを要求する。特に、接続条件に含まれた機能には、つぎのものを要請する。APSE の任意の道具を起動できること。利用者は使用する端末に依らず利用できること。利用者は起動した道具と会話できること。道具の中から、ほかの道具を起動する等の接続条件の各機能が利用できること。利用者と APSE とのやりとりは、すべて Ada の標準文字集合のみで行えること。利用者が APSE を使い始めるのに親計算機の操作系の助けを借りてもよいこと。APSE から元の操作系へ戻りしかが用意されていること。

APSE の道具については、つぎのことを要求する。道具は、Ada 言語での作譜を支援するものであること。(特に、どの APSE も分割翻訳\*\*\*を支援しなければならない。)道具は、ほかの道具と組合せてより複雑な仕事にあてることもできるよう、簡明な機能を持つこと。道具は Ada で書かれていること。APSE は、道具をいつでも追加ができるよう開かれた構成であること。道具間の通信、道具と利用者との会話は、簡単に統一されたものであること。(特に誤りが生じた場合も同様に扱えること。)道具に対して適切な「援助\*\*\*\*」機能があること。Ada 言語で書かれた算譜を、Ada の表現と概念とだけを用いて検査\*\*\*\*\*し、虫取り\*\*\*\*\*するための道具があること。さらに、対象計算機上での実行についても、検査・虫取りの道具が用意されていること。企画管理のための資料を提供するための道具があること。

\* "configuration object"  
 \*\* "partition"  
 \*\*\* separate compilation  
 \*\*\*\* "help"  
 \*\*\*\*\* test  
 \*\*\*\*\* debug

#### 4. 核環境 KAPSE

核環境について、3 での要件をより具体的に述べている。

KAPSE の料基については、つぎのような諸条件を満たすことを要求する。料基中の物件につける一意名は、Ada での英字名\*の列であり、集団の要素には、集団名による修飾を許すこと。物件として貯えられる情報の種類に制限を置いてはならないこと。各物件には、形態管理を行うに足りる履歴、そこに貯えられた情報の類別、参照の権限関係といった属性を持たせること。すぐには必要とならない物件を2次記憶に貯える等の保存\*\*機能を持つこと。APSE の各道具から、Ada 言語の入出力を用いて、各物件の情報・属性を取り扱うことができ、さらに、各種の関係をたどることができること。

KAPSE には、つぎの各機能が備わっていることを要求する。Ada 算譜に対する基本実行時支援・入出力支援・料基の取扱い支援。APSE の道具が、ほかの道具を、必要なら引数を渡して、起動できる機能。APSE の道具から Ada の入出力を用いて、それを起動した端末装置と会話できる機能。端末から非同期的な制御を行える機能。

KAPSE については、Ada 言語での package 仕様として書かれた接続条件を具現したものでなければならぬと規定する。こうした仕様としては、基本操作(入出力、料基関係等)と、Ada の翻訳の各段階に応じた抽象料\*\*\*とがなければならないとする。特に、後者については、具体的につぎのものを要求する。翻訳単位となる原譜の表現、抽象構文の定義、構文解析・意味解析終了後の中間言語の定義、Ada 算譜の実行形式の定義(少なくとも算体\*\*\*\*と料空間\*\*\*\*\*を含むこと)、名表\*\*\*\*\*の定義(名前に関して、出現行・相互参照・算譜の構造等、解析・検査・虫取りのための道具で必要となる情報を含むこと)、備譜に関する貯蔵形式の定義。

#### 5. 最小環境 MAPSE

APSE が、最低含んでいなければならない諸機能について要求を定義する。

\* identifier  
 \*\* archiving facility  
 \*\*\* abstract data type  
 \*\*\*\* code segment  
 \*\*\*\*\* data space  
 \*\*\*\*\* symbol table

**文面編集子** 仕様書・設計書・その他の文書とともに算譜をも編集することができる編集子があること。その指令としては、検出・変更・挿入・削除・書式整理・入力・出力・移動・転写・置換が用意されていること。文面の位置決めには、行番号を用いても、文面内容を用いてもよい。

**清書子** 各種文書・算譜の書式を整え出力するための道具があること。特に料基内の物件を適切な形式で出力できること。

**翻訳子** Ada 算譜を、親計算機および少なくとも1個の対象計算機用の命令列に翻訳する翻訳子が、1個はあること。翻訳子は、KAPSE との接続条件を守っていること。

**結合子** 親計算機および対象計算機向けに、それぞれ結合子があること。結合子は、言語仕様に沿って各算譜単位を部分的に結合したり、あるいは、これらから実行可能な算譜を生成したりできること。後者については、不要な単位を除去したり、重載\* したりする機能も併せ持つこと。

**載入子** 対象計算機に対して、外部媒体を介したり、あるいは直接通信を用いたりして載入する機能を持つこと。

**静的解析子** 算譜について、各算料の代入・引用関係を解析する解析子、および、制御構造(手続きの呼出し関係、例外処理の関係、並行処理の相互関係)を解析する解析子があること。

**動的解析子** 動的解析子は、会話形式でつぎの諸機能を含むこと。時点表示\*\*、中断\*\*\*、追跡\*\*\*\*、接続条件の模擬、実行管理(文単位)、並列処理の時刻関係解析。一括処理形式でも、また対象計算機に関しても、これらの機能(の一部)が使えること。

**端末操作子** それぞれの種類に応じて、端末とのやりとりをするための操作子\*\*\*\*\*が用意されていること。これらは、端末と KAPSE 内の諸機能・諸料との仲介を行う。

**算帖管理子** 算帖に関して、その内容転送の標準形式を定め、さらにつぎの諸機能を有すること。比較・異常処理・転送・見出しの転送・履歴転送。

**指令解釈子** MAPSE の道具を起動するための指令解釈子があること。端末からの利用者に対して、各指

令を実行する前にそれを編集したり検査できるようにし、また、どんな操作に対しても積極的な応答をすること。また、指令列を料基に貯えておいて後で実行することもできるようにしておくこと。

**形態管理子** 各企画の長期にわたる形態管理を補佐するための道具があること。この道具は、少なくとも、履歴属性を調べたり、料基中の各情報の存続を制御したりするための便宜を与えること。

MAPSE には、さらに、つぎのような備譜が用意されていないなければならない。(1)言語仕様に定める仕様を拡張した、あるいはそれに替る仕様を持つ高水準の入出力用譜包\*。(親計算機用)(2)高水準入出力(対象計算機用)。(3)算帖の物理的操作用譜包および索引系\*\*。(対象計算機毎)

## 6. 展 望

具体的な要求(STONEMAN 中での語法に従えば、“shall” が用いられる項目)は、以上述べてきたものに尽きる。一般的な(概念的な)APSE への要求と、より具体的な KAPSE-MAPSE 方式の具現への要求である。

これらを通読してみると分かるように、具体的な要求は、いずれも、既存の操作系——あるいは計算機会社から備え付けられてくる道具類——で十分こと足りる程度のものである。“Ada 言語”が対象であることと“多対象計算機”を考慮している点とが(当然のことだが)異なるだけである。

もちろん、“APSE”と称されるものが、これらだけで十分であるとは、STONEMAN 自身考えているわけではない。実際、その7章で、つぎのような諸道具類を掲げているのである。

(1) Ada 言語向きの算譜編集子。(2) 文書化支援機能。(3) 企画管理系。(4) 形態管理系。(5) 各種測定機能。(6) 障害報告系。(7) 要求仕様の取扱い支援。(8) 設計支援系。(9) 検証系。(10) 変換系——ほかの道具から利用できる、あるいは、ほかの道具を作る場合の要素となりうる字句解析・構文解析・名表操作・命令生成等の構成単位から成るより高度な変換系、(11) 指令解釈子——Ada 構文に似せた言語構造を持つより高度なもの。

また、各応用分野ごとに、またそこで採用されている方法論ごとに、それぞれ適切な備譜を用意する必要

\* overlay  
\*\* snap shot  
\*\*\* break  
\*\*\*\* trace  
\*\*\*\*\* handler

\* package  
\*\* directory system

もある。特に、数値計算向けのものについては、その要求仕様が設定されるであろう、とも述べている。

DoD が、Ada 計画に踏み切った目的である「譜構に係る（要求・仕様・設計・生産・出荷・保守にわたる長期の）費用の軽減」のためには、APSE の具現は必須のものといえる。単に言語を定めたからといって、上記目的が、それだけで達成できるはずはないからである。その意味で、今後 DoD は APSE 具現を推し進めていくものと予想される。そして、それは直接間接に「支援環境」のあり方に大きな影響を与えるであろう。今後の動向に注目する必要がある。

#### 参 考 文 献

1) "STEELMAN", Requirement for High Order

Computer Programming Languages, p. 20, DoD (1978).

2) Preliminary Report on the Ada Programming Language, SIGPLAN Notices, Vol. 11, No. 6 A (1979).

3) Reference Manual for the Ada Programming Language, DoD (1980).

4) "PEBBLEMAN", Revised, Requirements for the Programming Environment for the Common High Order Language, p. 41, DoD (1979).

5) "STONEMAN", Requirements for Ada Programming Support Environments, p. 44, DoD (1980).

(昭和 55 年 11 月 21 日受付)