N89-16305

S26-61
167353
8 P.

ADA(R) ASSESSMENT : AN IMPORTANT ISSUE

WITHIN THE EUROPEAN COLOMBUS SUPPORT TECHNOLOGY PROGRAMME

P. VIELCANET

INFORMATIQUE INTERNATIONALE
2, rue Jules Védrines
31400 TOULOUSE FRANCE

Tél. (33) 61.34.01.92

1. INTRODUCTION

Software will be more important and more critical for COLUMBUS than for any ESA previous project. As a simple comparison, overall software size has been in the range of 100 K source statements for EXOSAT, 500 K for SPACELAB with IPS, and will presumably reach several millions lines of code for COLUMBUS (all elements together).

Based on past experience, the total development cost of software (facilities, simulation, test items, on-board software...) can account for about 10 to 15 % of the total space project development cost. For COLUMBUS, this share will grow over the entire space system life cycle as maintenance and evolution will be vital within its very long operationnal phase. Considerable savings will be possible by properly managing software and by exploiting fields of commonality.

The Ada technology may support the strong software engineering principles needed for COLUMBUS, provided that technology is sufficiently mature and industry plans are meeting the COLUMBUS project schedule.

Over the past three years, Informatique Internationale has conducted a coherent programme based on Ada technology assessment studies and experiments, for ESA and CNES as indicated in figure 1.

This specific research and development programme benefits from Informatique Internationale fifteen years experience in the field of space software development and is supported by the overall software engineering expertise of the compagny (e.g deep involvement in the european ESPRIT and MAP programmes).

(R) ADA is a registered trademark of the US Department of Defense

2. ADA TECHNOLOGY ASSESSMENT PROGRAMME

The logical construction of the space station oriented Ada technology assessment programme appears in figure 1. Four main layers may be distinguished :

a) Ada development environments procurement policy (Rolm ADE and Verdix VADS), set up of convenient methods and development of new tools :

GET, a tool for automatic production of interactive test environments for Ada packages.

SOPHIA, an advanced syntax-directed editor for Ada designed to operate on advanced work stations and providing features for adding new functionalities (e.g. static or dynamic analysis of programs).

b) Ada space specific experiments for CNES and ESA aiming at a rather broad investigation (e.g. ground and space segments) :

ADEXII, a two years experiment and assessment project undertaken for CNES (100 man-months budget over 83-85) with following main tasks based on careful monitoring of the activity :

. Assessment of the Ada language with respect to training, effective use and degree of applicability

. Assessment of the Ada environment and resulting Ada products

. Production of guidelines for an efficient transition to Ada.

ESA/ADA, one year experiment conducted for ESA in 84-85, aiming at the Ada development of a complete simulation of the GIOTTO spacecraft Attitude and Orbit Control System from an existing Fortran program. The organization of the project based on partial and parallel development by INFORMATIQUE INTERNATIONALE, CESELSA (sub-contractor) and ESA itself successfully demonstrated unique features and suitability of the Ada language for large space projects (significiant guidelines on an Ada development methodology have been established).

CCSDS, six months project conducted for CNES in 85 demonstrating the successful use of Ada as a data description and data handling language for the GALILEO spacecraft telemetry (modelling and processing according to the international CCSDS standards).

c) On-board Data Management System (COLUMBUS class) feasibility studies

- ESA/OBCA, comparative study on distributed microprocessor based computer system architectures

- ESA/HOL, a study of the applicability of High Order Languages for on-board software production (assessment and selection of the best candidate among Ada, Modula 2, C, LTR 3, Pascal and HAL/S).

d) Ada detailed assessment for COLUMBUS on-board distributed Data Management System.

ESA/SSADA, two years project (start end 85) investigating three important issues :

- availability of Ada tools (near and mid-term) for the development of distributed application software
- links between Ada features (language and implementation) and specific requirements of a typical space station mission
- specification and development in Ada of a study case software system (derived from space station requirements analysis) which can produce significiant insights on powerful model of future Ada software production environments.

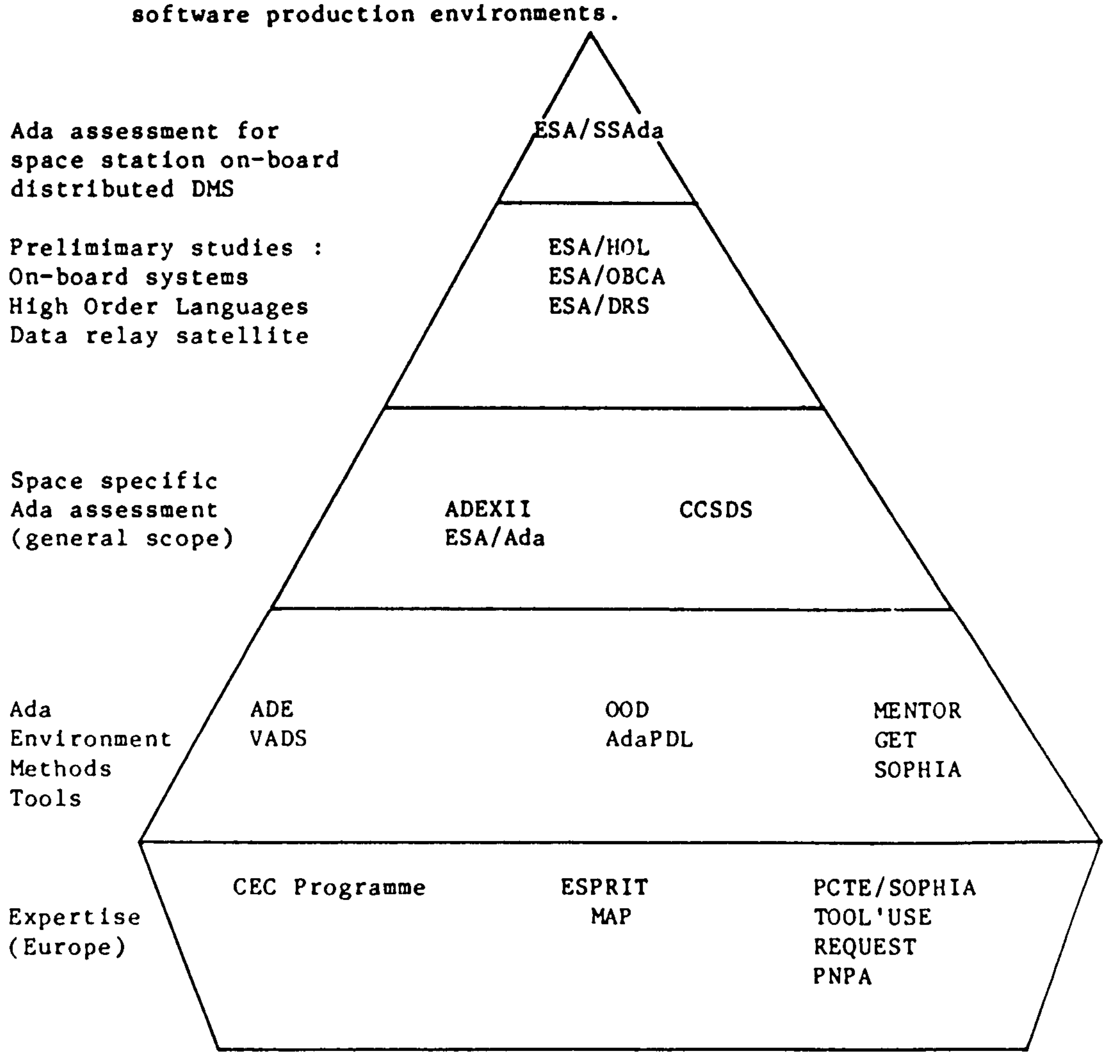

FIG. I. : <u>INVOLVEMENT IN COLUMBUS SUPPORT TECHNOLOGY</u>

3. PRESENTATION OF CNES AND ESA ADA EXPERIMENTS

3.1. CNES ADEXII EXPERIMENT

As previously stated, Informatique Internationale conducted an Ada experiment for the french national space agency (CNES) in Toulouse, France. The experiment main objectives were to provide information on the suitability and effective use of the Ada language for space applications and to locate the potential benefits and possible drawbacks to be expected when introducing Ada into the aerospace industry environment.

As such results and lessons learnt can contribute to a better understanding and management of a space-oriented Ada technology transfer. Education and development methods were especially discussed. The experimental data collected over the project have been extracted from a development effort of six software engineers over two years with a total production of 30 000 Ada source lines (ASL).

The experiment had then to cover two main areas :

- introduction of the language (i.e. how it is used and learned in practice by personnel with different technical backgrounds)

- suitability of the language for applications specific to the aerospace industry, particulary real-time applications.

These topics were futher refined, analyzed and balanced against technical Ada constraints (mainly lack of information and training on Ada software engineering) and three evaluation areas were defined :

- learning and use of the Ada language

- development of Ada software products

- performance and assessment of a validated Ada environment.

To reach these goals within budgetary constraints, it was decided to redesign and redevelop existing Fortran applications, meanwhile monitoring related activities. These applications corresponding to small-scaled projects were preferred to a single large real-time project, due to the high risks implied by such a choice at the time the project started. Previous papers (Labreuille 84 and Papaix 85) give an in-depth discussion of the project tasks and the resources involved.

With respect to the initial objectives, the following conclusions were reached :

Productivity

High productivity ratios have been experienced (up to 1400 ASL per man-month for small Ada developments) but this data should be interpreted with care and balanced against a real industrial context. In this experiment context, the development team was small, motivated, enthusiastic and experiencing the learning process and the use of Ada and programming environment tools.

More than the achievement of good productivity figures over the project, the identification of the main contributors to productivity improvements were pointed out :

- early validation through the use of Ada at the design phase

- automatic recompilation features supported by convenient configuration control system

- reuse of software components

Training

This experiment has proven that acceptable level of proficiency in Ada could be reached rather quickly (in less than a month).
Ada, as a programming language is no more difficult to learn than another language, but making full use of its underlying software engineering principles requires some additional effort. Due to Ada richness, special training is required for "good use" of advanced features, as well as to avoid systematic use of "well experienced" subset.

Environment

The availability of a number of tools is of great help, but one should not forget that learning how to use them effectively is almost as important as learning the language itself and takes time and effort as well.
Evidence was shown that an Ada compiler must be a validated one, tools must be of good quality as well and should be suitable for the development of large Ada programs (more than 10 000 ASL).

Development methodology

Use of Ada impacts heavily on traditional methods through :

- early and continuous use from design

- early validation of design through prototyping and step-wise PDL refinement

- design effort which is increased by up to 50 % while integration is reduced up to 5 times

- effective parallel developement.

3.2. ESA ADA EVALUATION STUDY

As part of its Technical Research Programme, in preparation for using Ada, the European Space Agency has just completed a study to evaluate the use of Ada in a typical space-oriented software project, with particular emphasis on the impacts on METHODOLOGY and the prospects for PORTABILITY, REUSABILITY and developement at multiple sites. The study was based on rewriting in Ada the Attitude and Orbit Control Software and the simulation of the satellite dynamics and operators environment of a recent satellite, which were previously implemented in Assembler and Fortran.

As a result of this study, ESA has now a set of Ada packages which has been used to evaluate many of the existing Ada compilers and Ada supporting toolsets as reported in (Robinson 86). This proved to be a valuable way of identifying some of the key aspects for providing portable software, and for identifying strong and weak features of existing and potential APSES.

The study project was performed by Informatique Internationale (acting as prime constractor) and CESELSA (Spain) under the direction of ESA Technology Centre (ESTEC). The main activity was to rewrite in Ada

a) the Attitude and Orbit Control Equipment (AOCE) software of a recent satellite, from the existing design written in Caine, Farber Gordon PDL and the listings of the RCA1802 Assembler programs,
b) the simulation of the satellite dynamics and operators environment which were previously implemented in Fortran.

The Ada program consists of 6 components as indicated in figure 2. The core of the program is the package P_AOCE containing the satellite software. The RAM is visible to provide access to data for operator display, and part of the RAM (T_RAM1) is available to write telecommands. This package is embedded in a simulation of the real world environment, consisting of telecommand management, hardware interface, dynamics simulation and operator command/display interface.

ESA standards for software life-cycle (ESA 84) were followed to assess their suitability for Ada. These consist of phases for software requirements, architectural design, detailed design and implementation, each phase terminating in a formal review. Full documentation was produced.

The Software Requirements Document was written by Informatique Internationale to pull the requirements together and as a familiarisation task to provide a clear definition of the work to be done.

As an experiment, two Architectural Designs were produced, at both Informatique Internationale and CESELSA. Each consisted of narrative, design diagrams and Ada Specification parts. In addition, the major task structure was prototyped using TEXT_IO to provide a listing of the flow of control, thus demonstrating that the overall architecture is correct, and that the specification parts were consistent and compilable. After the review, the ADD which was based on Object Oriented Design was selected since this provided the more coherent and complete view of the design. It was decided to use OOD on the detailed design of the dynamics part in the next phase to gain more experience of this technique.

The Detailed Design was also repeated by the two contractors, using the same architecture as a baseline for each. The main difference was that Informatique Internationale decided to use SEPARATE compilation extensively in the design of the larger packages. This has the benefit of reducing the time for recompilation due to changes in only one procedure during module testing. It results in more source files and a slightly more complex library structure with therefore more need for Ada Program Library tools to manage the re-compilation and configuration management activities.

To try out the multi-site aspects of the project with a set of independently coded packages, the satellite software was programmed in ESA and the simulation parts were programmed in Spain (CESELSA). These were then integrated at a third site in France (Informatique Internationale), with the help of all parties.

Acceptance was based on 10 test cases from the ESTEC Assembler/Fortran implementation, which produced identical plots in 9 cases and a better result at the 5th significant digit in the 10th case. Differences between computers were therefore insignificant.

The main part of the study produced working software, and the software development lifecycle worked satisfactorily. Module testing at package level lead to easy integration, with good support from the symbolic debugger. There is a clear conclusion that it pays to do module testing, and that the resulting integration effort with Ada is relatively low in that case. A "module" in Ada is defined as package, for which each visible part (data, procedure, functions) is tested.

OOD was found to provide a natural method of producing a clear picture of the design, which leads easily into Ada definition, implementation and integration.

A summary of the statistics of the project is shown below :

Item	Fortran	Ada
Simulator lines	4800	4174
P_AOCE lines	-	2738 = 6912
Lines of test code	-	886 = 7798
Comment lines	1600	3677 = 11475
Compile time	5 min	113 min
Execution time	80 sec.	350 sec.

Phases	Man days	
Requirements	40	
Architectural design	77	
Detailed design	101	
Code, test & integration	152	
TOTAL	370	= 31 lines/day

REFERENCES

[ESA 84] ESA Software Engineering Standards BSSC (84)1

[LABREUILLE 84] B. LABREUILLE, M. HEITZ : "The Introduction of Ada in French Aerospace Industry", ADA-EUROPE, Adatec 1984, Brussels Conference.

[PAPAIX 86] M. PAPAIX, M. HEITZ, B. LABREUILLE : "Two Years of Ada Experiments : Lessons and Results", ADA-EUROPE, 1986, Edimburgh Conference.

[ROBINSON 86] P. ROBINSON : "Ada Evaluation and Transitions Studies", ADA-EUROPE 1986, Edimburgh Conference.

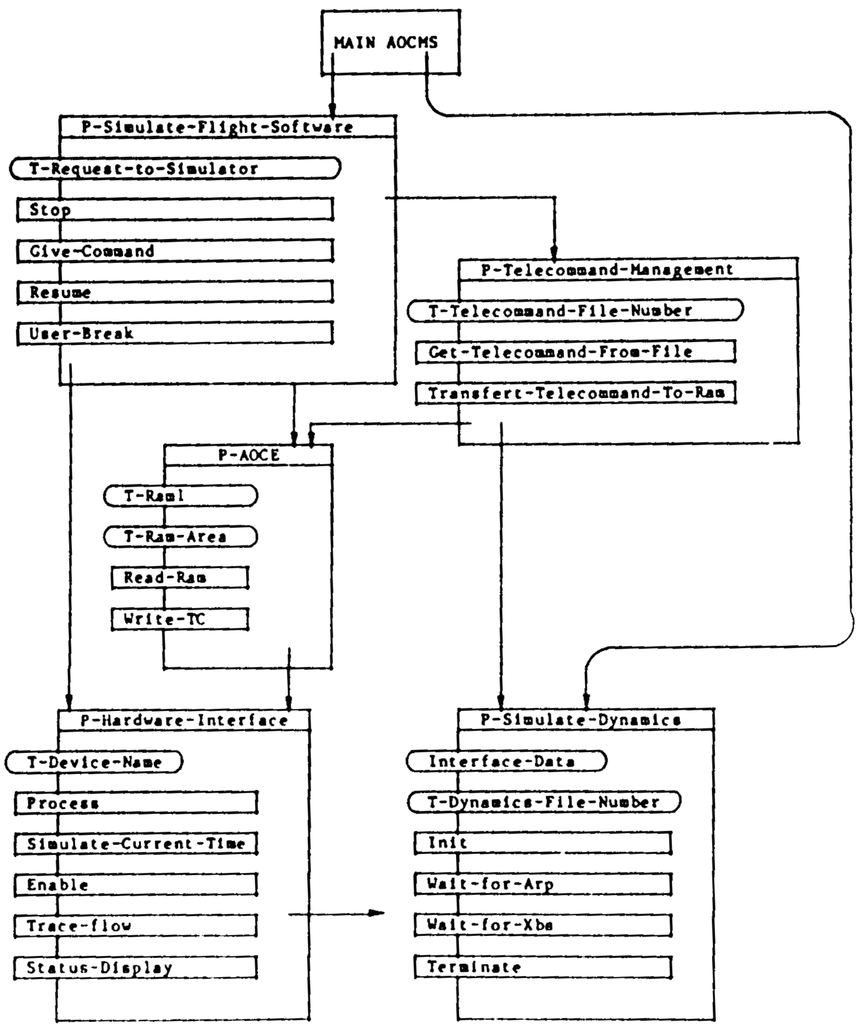

Fig. II : AOCMS Architectural Design