

# Ada Conformity Assessments: A Model for Other Programming Languages?

**Michael Tonndorf**

IABG  
Ada Conformity Assessment Laboratory  
D-85521 Ottobrunn

Tel: +49 89 6088 2477  
Fax: +49 89 6088 3418  
Tonndorf@iabg.de

## *Extended Abstract*

*Submitted as technical article to the SigAda '99 Conference*

**Short Abstract:** This paper presents the actual status of Ada Conformity Assessments after the transition of Ada Conformity Assessments from the Ada Joint Program Office to ISO. The principles of Ada Conformity Assessments according to the ISO/IEE Final Committee Draft 18009 are reflected and the commonalities and differences to the previous practices are discussed. The main part of the work is a comparison of conformity assessments for Ada C, C++, and Java. The paper concludes with an assessment and outlook on the future development on compiler conformity assessments in general.

**Keywords:** Ada Conformity Assessments, Ada Standardisation, Ada Language Maintenance.

## 1. Introduction

After the closing of the AJPO October 1998 the assessment process of Ada compilers had to be re-organized. This is realized by moving Ada validation, or as it is called now "conformity assessment" under the sole roof of ISO. This leads to the question how conformity assessment is practiced for other programming languages in use, e.g. C, C++, and Java.

## 2. Development of the Ada Certification System

ISO with its headquarters in Geneva, Switzerland, is the world's top level organization for standardization. ISO's policy is put into effect by the national member bodies. In the time of the standardization of Ada 83 the role of the US national ISO member was carried out by NIST (the National Institute for Standards and Technology). However NIST transferred the right to implement the Ada standardization policy to DoD's

Ada Joint Program Office for practical reasons. And as an Ada Validation Facility the NIST was always in the loop.

During the prospering period of Ada 83 from 1985 to 1993 the world of validation was well ordered:

- The *Ada Joint Program Office* as sponsor and political instance,
- The *Ada Validation Organization (AVO)* as technical coordinator of the validation process and the Ada Validation Facilities;
- *5 Ada Validation Facilities (AVFs)*; USA: NIST, Wright Patterson Airforce Base; UK: NCC; France: AFNOR; Germany: IABG) as testing laboratories responsible for conducting validations,
- an *ACVC team* contracted for the maintenance of the ACVC testsuite.

However with the upcoming standard Ada 9X this world started to crumble. At first the number of yearly conducted validations decreased significantly. The French AFNOR and the British NCC went out of business due to a lack of validation contracts. Then the AJPO announced to return validation authority to NIST again in October 1997. In September 1997 however NIST announced to give up all direct IT standardization activities in spring 1998. This brought back AJPO resp. IDA's Ada Validation Organization to validation business until September 1998. So after the closing of the Wright Patterson Airforce Base AVF and NIST only EDS and Germany's IABG remained in the Ada validation business. IABG's experiences in the first decade of *Ada validation* is described in more detail in [1].

In parallel the trend accelerated to withdraw the Ada mandate in general. This was finally accomplished in April 1997. This means that even in the case of using an Ada compiler a valid certificate was no longer required.

So in the course of the year 1998 a decision between two alternatives had to be made:

- no structured "Ada certification body"; no coordination of validation laboratories; no coordinated maintenance and development of the Ada testsuite, no mutual recognition of validation certificates,
- establishing of an authority, that ensures a uniform validation process, using the same testsuite with the same procedures and rules. In fall 1998 it turned out that this option was feasible. A technical authority for Ada validations was founded and sponsored by the ARA and DISA (DoD's Defense Information Systems Agency).

### 3. Actual Situation

For all major programming languages an ISO standard exists. As this is true for Ada as well it was obvious to integrate the new Ada certification system under ISO too. With its Working Group 9 (WG9) ISO was already continuously involved in the Ada standardization. So the main goal under direct ISO participation was to develop a uniform terminology, compatible with the other ISO standardized programming languages, and to otherwise continue with the proven and accepted validation process as developed under the AJPO. Since spring 1999 there exist draft procedures containing requirements on the "new" validation process and certification body: *ISO/IEC Final Committee Draft 18009 Information Technology – Programming Languages – Ada: Conformity Assessment of Language Processor* [2]. The next table shows the old and new terms used in the ISO environment:

[2] reflects the basic requirements in the following areas:

- Ada Conformity Assessment Testsuite,
- Ada Conformity Assessment Authority,

- Ada Conformity Assessment Laboratory,
- Operating Procedures for Ada Conformity Assessments.

New Name or Institution	Short Term	Old Name or Institution	Short Term
Ada Conformity Assessment Authority	ACAA	Ada Validation Organization	AVO
<i>ACAA Technical Agent</i>	<i>Randy Brukard</i>	<i>AVO Technical Agent</i>	<i>Dan Lehman</i>
Ada Conformity Assessment Laboratory	ACAL	Ada Validation Facility	AVF
Ada Conformity Assessment Test Report	ACATR	Validation Summary Report	VSR
Certified Processors List	CPL	Validated Compilers List	VCL
Ada Conformity Assessment Test Suite	ACATS	Ada Compiler Validation Capability	ACVC
<i>Ada Resource Association</i>	<i>ARA</i>	<i>Ada Joint Program Office</i>	<i>AJPO (Ada Sponsor)</i>
certified conforming		validated	
conformity assessment		validation	
self testing		prevalidation	
witness testing		validation	

Table 1

#### 4. A Model for Conformity Assessments of Programming Languages

Which conditions must be satisfied in order to provide a “Model for Compiler Conformity Assessments”? For that purpose we start with some basics in software quality and build a framework of terms yielding the unified terminology of ISO.

An obvious requirement coming along with an international standard is to have one uniform worldwide certification system, in this case of course a certification system for the programming language Ada.

Testing object is an identified Ada implementation. Test objective is to verify that the testing object complies with ISO standard ISO/IEC 8652:1995 (Ada). Execution of testing requires a testing method. The testing method for Ada is the Ada Compiler Validation Capability (ACVC) which was developed in parallel with the first standardization of Ada as ANSI/Mil-Std 1815A in 1983. The ACVC is the foundation of a testing procedure that comprises a defined set of test cases. These test cases can be partitioned into two groups: negative and positive test cases. A positive test case is a program sequence that complies with the Ada standard. The tested implementation has to *process* this sequence in accordance with the standard. A negative test case is the intended violation of the standard which has to be detected by the implementation. For efficiency reasons usually a test is made of a series of related test cases which have equal or similar test objectives. Thus a test is the smallest component of the test suite, identified by a name. Furthermore part of a testing procedure are rules guiding the evaluator of a test result by grading the test result as *conforming* or *not conforming*. Obviously it's inefficient for the tester to lookup the language reference manual every time in order to grade the test result. For negative test cases these rules are source code comments at appropriate places, positive test cases report their results

usually automatically, following a self-checking mechanism. This method is described in more detail in [3].

The tight interpretation of the term test procedure – as a systematic way of conducting tests - leads to the broader understanding of a test procedure as a detailed set of rules as how to manage the whole process of a conformity assessment including the rules how to interpret and grade test results.

Although it seems obvious it is not: for a programming language there should be only one generally accepted testsuite in order to be able to compare implementations. Notwithstanding the Plum Hall testsuite (see Chapter 5) praises itself to be winner of the one and only once held contest for C-testsuites. A testsuite itself is like any other large and complex piece of software subject to quality management and configuration management. The ISO procedures define requirements for a sensible use of the testsuite.

Result of a conformity assessment is always a report with a detailed log of all test results. If all test results for the implementation comply with the grading criteria then a *Mark of Conformity* can be issued by the testing laboratory.

Conducting conformity assessments requires a certain qualification. Therefore these should only be executed by recognized testing laboratories. The meaning of *recognized* or *accredited* has to be discussed later. [2] requires that the testing laboratory works according to the accepted principles of ISO. This means it should be embedded in a well defined organization and operate on the basis of an approved quality manual. Finally it is required that all testing laboratories recognize themselves equally in order to provide a uniform conformity assessment process.

In addition to the purely executing role of a testing laboratory a technical instance is needed that performs general tasks. With that the "language policy" should be enforced by taking care of the issues

- quality management and configuration management of the testsuite,
- future maintenance of the testsuite,
- accompanying individual conformity assessments:
  - approve special test modifications,
  - disputes ruling,
  - quality control of the test reports.

For Ada there was always a discussion how general a certain certificate should be, which platforms should be covered by a specific certificate. IN Ada a common understanding emerged, although the opinions never converged completely. Validation is always conducted on a specific base platform. Following up the vendor then has the choice to *extend* the status *certified conforming* to related implementation classes which were obtained by

- maintaining an implementation within certain limits,
- rehashing an implementation.

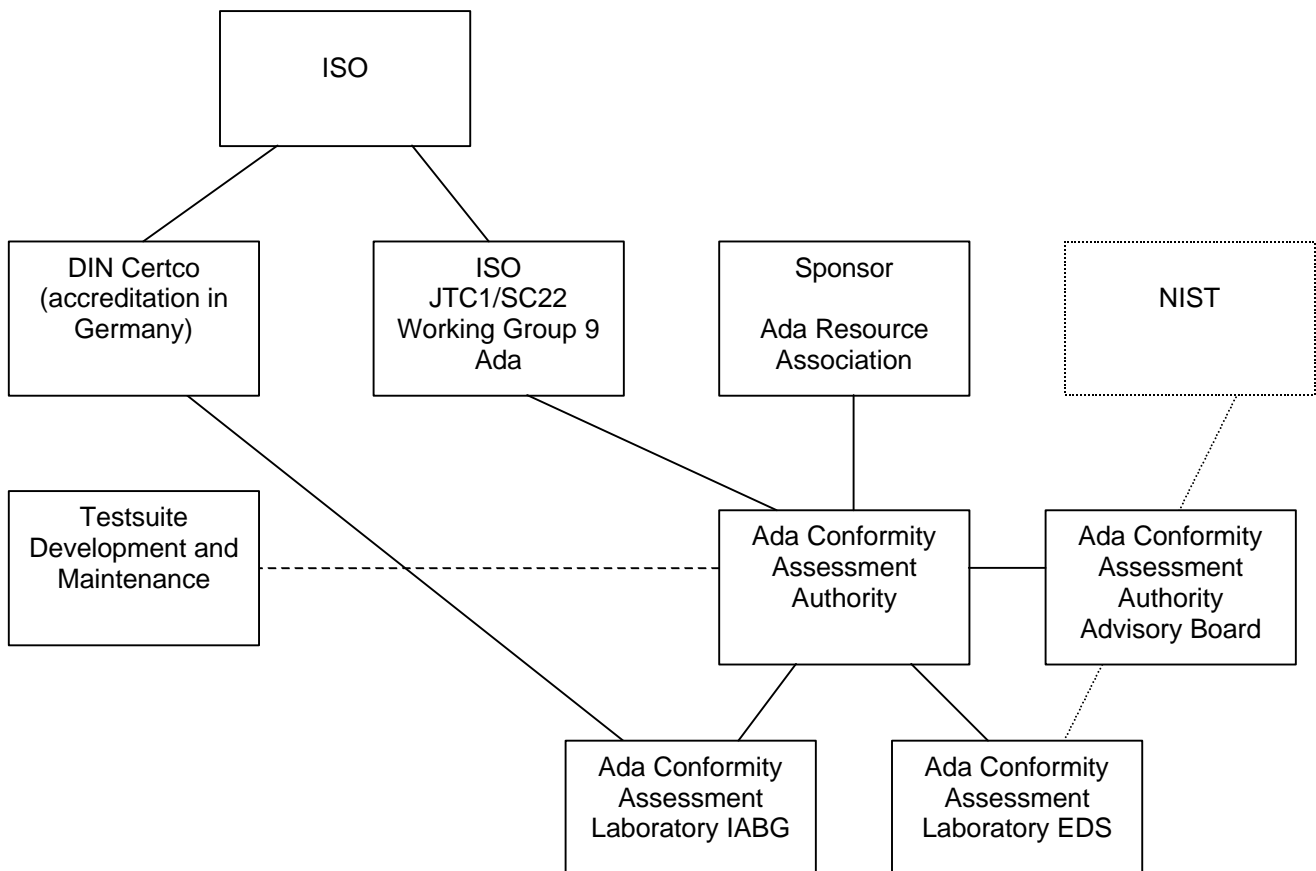
Finally this infrastructure does not operate for free. The costs cannot be paid by the fees for a compiler conformity assessment. An important role for conformity assessments is a sponsor representing the interests of the users. They expect to find a working certification system which is kept alive by the sponsor. However the sponsor should not use his financial position to control the policy of the certification system. Since 1998 the Ada Resource Association (ARA) is sponsor of the Ada certification system.

In summary the components of the new Ada certification system are presented below as required by [2]:

- ISO board controlling the evolution of the language standard (here: WG9),

- Sponsor (here: ARA),
- Conformity Assessment Authority,
- Conformity Assessment Laboratories,
- Testsuite including user- and version-documentation,
- Operating Procedures for Ada Conformity Assessments.

Pic. 1 below shows the essential dependencies of the parties involved in Ada conformity assessments.



Pic.1

## 5. The Situation for Other Programming Languages

A direct comparison of Ada's certification system with that one of other programming languages is difficult. This is due to the unique situation of Ada compared to other languages. Ada is the only language for which a certification system was built up by the sponsor and customer of that language. Therefore Ada is the only language for which information about certified implementations is freely available.

As mentioned already in the first part of this paper the NIST discontinued IT standardization, especially standardization and validation of programming languages. There are many implications of this decision: first of all there is no party entitled to accreditate conformity assessment laboratories. This means that any organization or institution can declare itself as an accreditation facility and thus accreditate new testing

laboratories. Moreover nothing prevents an organization from declaring itself as a testing facility. The only controlling mechanism for that is the market.

Furthermore the information pool role is changing. NIST was the editor of a “validated products list” which was also available in printed form until 1995. Basis of these products are US “Federal Processing Information Standards (FIPS). After all Ada is also a FIPS Standard (FIPS Pub 119-1). From 1997 on however there are no new entries to this list. This means that the length of the list is continuously approaching zero, which is expected for the end of 1999. It is obvious that this list does not reflect the actual status of validated implementations by any means.

For a comparison of conformity assessment candidates for Ada living languages should be selected for which conformity assessments are actually being conducted. This statement is actually true for C, C++, and Java. In addition conformity assessment for the Draft Standard Embedded C++ and for the C++ standard library are emerging. On the market of compiler conformity assessments two main players can be identified: Perennial and Plum Hall. They gathered their experience with conformity assessments during the involvement in the standardization process of C, C++, and Java.

Information about the testsuites was retrieved from the internet. Below two tables with some characteristic figures are given. The metrics of the test suites (size of code, numbers of test cases etc.) are not given in a consistent manner so that a direct comparison is impossible.

Plum Hall		
Language / Library	Testsuite	Metric
C	CVS 9.00 1998	24_956 test cases, 56 kLOC
C++	Suite++ Vers. XVS5	3_200 executable tests, 1_600 test facts
C++ standard library	LibSuite++	2_000 test cases
Java	JVS	6_200 test cases, 2.4 MegLOC, 800_000 executable items

Table 2

Perennial			
Language	Testsuite	Version / Release- Date	Number of test cases
C	ACVS	Vers. 4.5 Jan. 98	8_000
C	CVSA	Vers. 6.7 Oct. 98	43_000
C++	C++VS	Vers. 5.1 Feb. 99	72_000
Embedded C++	EC++VS	Vers. 1.0 Aug. 98	22_000
Java	JETS	Vers. 1.1 Oct. 98	18_000

Table 3

The development of the testsuites for C and C++ at Perennial reflects also the development of the languages. ACVS was contracted by the US government after the standardization of C early 1990. The suite was developed further as commercial product CVSA which contains ACVC as a subset. CVSA eventually was extended from 1992 on to result in C++VS. The suites have an open architecture allowing

a customer to add individual tests to the suite. All testsuites are structured according to the resp. language reference manual.

The testsuites of Perennial and Plum Hall are constructed according to the same principles. Contrary to Ada however they are delivered together with script programs that allow immediate processing on standard operating systems (Unix, WinNT). Thus the user has a tool at hand allowing him to process arbitrary parts of the testsuite and to display the evaluated results in a human readable form. The scripts also support regression testing. In these aspects the commercial testsuites must be clearly attested increased user friendliness. However this also has a price, starting at 10k GB Pounds license fees per year. The philosophy of testing is the same for every language: using positive and negative test cases grouped together in identified test programs. It is remarkable however that Ada puts increased emphasis on negative test cases (the "B-tests" for those who are familiar with).

The goal of using the testsuites is again different for Ada and C/C++/Java. For Ada the real goal is to go through a formal testing procedure and obtain a certificate issued by a neutral third party testing organization. For the other languages the testsuite is primarily the basis for compiler vendors' self tests. Issuing a certificate is an exception, *branding* is offered on demand. Besides IABG there is no other testing laboratory for any of the languages Ada, C, C++, and Java with a valid national accreditation. This means that compiler validation or conformity assessment is again subject to free enterprise. Except for Ada there is no direct comparability of the tested implementations:

- competing testsuites,
- no single list of tested implementations,
- no expiration of validated status,
- conformity assessment by independent testing laboratories not as the regular case.

Along the lines with the new development for Ada there is no national accreditation for any testing laboratory for C, C++, or Java. Moreover after NIST's move out of standardization there will be no national accreditation for IT testing laboratories in the US anymore. The same statement holds true for Europe except IABG's valid DIN accreditation, which was actually renewed for Ada 95 in February 1999 by Certco, DIN's subsidiary for national accreditation.

## 6. Summary and Assessment

It has been shown that conformity assessments for Ada play a special role. A considerable investment by the DoD led to a single testsuite and to a working certification infrastructure. The achievements of the certification system could be preserved after DoD's withdrawal from validation. The testsuite covers the whole language and is freely available, however the development of the ACATS cannot be regarded as completed. Information about test implementations can be always be retrieved from a central internet site ([www.adaic.org](http://www.adaic.org)). For C, C++, and Java there are competing testsuites available as commercial products. The purpose of these testsuites is primarily self-testing for the compiler developers and vendors. Neither conformity assessments of independent testing facilities nor certificates as evidence for successful testing are the regular case.

Criteria	Ada	C, C++, Java (Plum Hall, Perennial)
Certificates	Part of the process	Branding on demand
third-party-testing	Regular	Exceptional (e.g.. "Perennial Conformance Test Center")
Independent testing	Yes	Identical with testsuite vendors

laboratories		
Accreditation	Germany: yes (IABG with DIN CERTCO), USA: no (EDS)	No national accreditation at present
List of tested implementations	Yes, available in the internet	No, competing testsuites
Testsuites price	Free	High: (more than 15.000 US-); testsuites as trademarks
Testsuite maintenance	Independent of the customer contract	Maintenance as part of the customer contract (6/12 months)
Testsuite comfort	Only tests themselves are under configuration management	Testsuite includes execution/evaluation scripts
Regression Testing	Not supported	Supported by the scripts
Technical authority as a separate organization	Yes	No

**Table 4**

Table 4 above summarizes the essential differences between the conformity assessments of Ada and C/C++/Java.

Assuming a realistic viewpoint the importance of compiler conformity assessments shows a downward tendency. Vendors use testsuites as part of their internal quality management cycles. A certificate demonstrating a successful conformity assessment does not really increase the market chances for a compiler. Seen positively compiler Technology is becoming more and more mature while still being away from a satisfactory maturity level. This reduces the need for an expensive third-party-testing procedure. The concentration process on a handful of platforms also contributes to a consolidation in compiler development.

In summary the thesis of this paper can be answered by *yes with restrictions*. With *Ada validation* a model was created which is leading with regard to objectivity, impartiality, and completeness. The development of the last decade has shown that this is not the general need of the compiler business outside of Ada. In this domain primarily a comfortable tool for compiler self-tests is sought. However as Ada is the first choice in the growing safety critical domain Ada conformity assessment will not lose its importance, just the opposite holds true. Moreover the use of certified tools is only one requirement within the software certification cycle for safety critical software. And the need for the maintenance of vendor independent standards in the software industry is more urgent than ever.

## 7. References

- [1] Ten Years of Tool Based Ada Compiler Validations. An Experience Report. Ada-Europe '98 International Conference on Reliable Software Technologies. Conference Proceedings, Lecture Notes in Computer Science 1411, Springer.
- [2] ISO/IEC Final Committee Draft 18009. Information Technology – Programming Languages – Ada: Conformity Assessment of Language Processor, 1999.
- [3] An Efficient Compiler Validation Method for Ada 9X. Michael Tonndorf, Ada-Europe '93 Conference Proceedings, Lecture Notes in Computer Science 688, Springer.