

Ada Compiler Konformitätsprüfungen: Ein Modell für andere Programmiersprachen?

Michael Tonndorf

IABG
Labor für Ada Konformitätsprüfungen
85521 Ottobrunn

Tel: 089 6088 2477
Fax: 089 6088 3418
Tonndorf@iabg.de

Abstract: In diesem Papier wird der aktuelle Stand der Konformitätsprüfungen für Ada nach der Schließung des Ada Joint Program Office beleuchtet. Es werden die Prinzipien der Konformitätsprüfungen nach dem ISO/IEC Final Committee Draft 18009 dargestellt und die Gemeinsamkeiten und Unterschiede zum bisherigen Verfahren diskutiert. Anschließend wird ein summarischer Vergleich mit den Konformitätsprüfungen anderer Programmiersprachen an den Beispielen C, C++ und Java, gezogen. Schließlich wird eine Bewertung von Konformitätsprüfungen für Compiler im Allgemeinen gegeben.

Schlüsselworte: Ada Konformitätsprüfungen, Ada Standardisierung, Ada Sprachpflege.

1. Einleitung

Eine wichtige Forderung bei der Entwicklung und Einführung der Sprache Ada war es, im Gleichschritt mit der Sprachdefinition eine Testsuite für Ada Implementierungen zu entwickeln. Damit sollte ein herstellerunabhängiges, wirkungsvolles und praktikables Verfahren eingeführt werden, mit dem Ada Compiler auf Normkonformität überprüft werden können. Dieses Verfahren wurde aus dem DoD-Umfeld heraus "Ada Compiler *Validation*" genannt, obwohl der Begriff Validierung im Qualitätsmanagement eine andere Bedeutung hat. Wir werden auf die Entwicklung der Begriffe später ausführlicher eingehen.

Die Errungenschaften bis zum Ende des Engagements des US-Verteidigungsministeriums wurden z.B. in [1] beschrieben. Nach der Schließung des Ada Joint Program Office im Oktober 1998 musste nun auch die Prüfung der Ada Compiler auf Normkonformität neu geordnet werden. Der Anspruch, der mit dem Prädikat *konformitätsgeprüft* verbunden ist, kann an dieser Stelle nicht abschließend bewertet werden. Natürlich ist mit dem Prädikat *validiert* oder *konformitätsgeprüft* nicht verbunden, dass die geprüfte Implementierung fehlerfrei oder 100% standardkonform ist. Auch dürfen Konformitätsprüfungen nicht als qualitätsfördernde Maßnahmen für Compiler alleine betrachtet werden. Schließlich ist ein Compiler erst die Voraussetzung für die Erstellung einer Anwendung mit besonderen Zuverlässigkeitsanforderungen.

2. Entwicklung des Ada-Zertifizierungssystems

Die Historie der Validierungen oder Konformitätsprüfungen für Ada kann an dieser Stelle nicht im Detail geschildert werden. Trotzdem sollen die wichtigsten Stationen hier aufgeführt werden.

Weltweite, anerkannte Dachorganisation für Normungen ist die ISO, International Organisation for Standardisation, mit Hauptsitz in Genf. Von ISO verabschiedete Normen werden durch nationale Einrichtungen (in Deutschland z.B. DIN) in die Praxis umgesetzt. So lange die nationalen Einrichtungen (*member bodies*) die Operationalisierung der Standards betreiben, beteiligt sich die ISO nicht selbst aktiv daran. Im Fall von Ada 83 war NIST (das US-amerikanische *National Institute for Standards and Technology*) zuständig für die Umsetzung des Standards Ada 83. Dies umfasste die (Weiter-) Entwicklung der ACVC Testsuite und die Einrichtung einer Infrastruktur zur Durchführung der Ada Compiler Validierung (der so genannte *Certification Body*). Da der Haupt-Sponsor von Ada, das US-DoD, selbst ein starkes Interesse an einer konsequenten Umsetzung des Standards hatte, hatte NIST dem US-DoD das Recht überlassen, die Validierungs-Infrastruktur zu entwickeln. In Form einer eigenen Validierungsstelle war NIST aber immer in den Prozess mit eingebunden. So war das Ada Joint Program Office die ausführende Institution, die mit der Operationalisierung des Standards Ada 83 beauftragt war. So war in der Blütezeit von Ada 83 die Welt der Validierung wohl geordnet:

- das *Ada Joint Program Office* als Sponsor und politisches Gremium,
- die *Ada Validation Organisation* (AVO) als technische Koordination der Validierung und Prüflabors,
- 5 *Ada Validation Facilities* (AVFs; USA: NIST, Wright Patterson Airforce Base; UK: NCC; Frankreich: AFNOR; Deutschland: IABG) als Prüflabors zuständig für die Durchführung der Validierungen,
- ein *ACVC Team*, beauftragt mit der Pflege und Weiterentwicklung der ACVC Testsuite.

Doch mit dem Herannahen des Standards *Ada 9X* begann diese Welt zu bröckeln. Zunächst ging ab 1993 die jährliche Anzahl von durchgeführten Validierungen weltweit signifikant zurück. Ab dieser Zeit gab es für jede bedeutende Hardware-Plattform mindestens einen Ada Compiler. Dies führte dazu, dass sich erst die französische AFNOR und später auch das britische NCC aus dem Geschäft zurückzogen. Doch was in den USA im Zusammenhang mit den Zuständigkeiten für die Validierung passierte, kann getrost als kurios bezeichnet werden. Zunächst kündigte das AJPO im Frühjahr 1997 an, dass es sich aus der Ada-Standardisierung bis zum September 1997 ganz zurückziehen werde. Damit war gemäß der ursprünglichen Kompetenzverteilung automatisch wieder NIST zuständig. Im Februar 1997 kündigte dann seinerseits NIST an, sich bis Ende März 1998 ebenfalls zurückzuziehen, und zwar aus der gesamten operationellen Umsetzung von IT-Standards – also auch der Ada Validierung. So blieb aus pragmatischen Überlegungen nichts anderes übrig, als die Zuständigkeit wieder dem AJPO zu übertragen. Damit stand NIST auch nicht mehr als AVF zur Verfügung. Es war klar, dass dies nur eine Aktion auf Zeit war: schließlich war das *Aus* für das AJPO Anfang 1998 ebenfalls eine beschlossene Sache, wobei sich hier nach einigen Irritationen der 30. September 1998 als Termin für die Schließung festigte. 1997 schloss auch die Wright Patterson Airforce Base in Dayton als AVF ihre Pforten, die personelle Verantwortung ging unter Phil Brashear schließlich auf die private Firma EDS über.

Parallel dazu beschleunigten sich die Bestrebungen in den USA, das Ada Mandat generell aufzugeben. Diese Entwicklung muss allerdings im allgemeinen Trend der staatlichen Standardisierungsorganisationen gesehen werden (egal, ob militärisch oder zivil), sich aus der direkten Standardisierung von Produkten und Verfahren zurückzuziehen. Daraus resultierte im April 1997 die Aufhebung des Ada Mandats, also der prinzipiellen Verpflichtung, für alle Verteidigungsprojekte Ada einzusetzen. Ja selbst in dem Fall, in dem Ada verwendet wird, wird kein gültiges Validierungszertifikat mehr gefordert.

Im Frühjahr 1998 war also klar, dass die Verantwortung für die Ada Validierung ab dem Herbst 1998 erneut vakant sein würde. So reifte in der WG9, der ISO-Arbeitsgruppe für Ada, der Gedanke, ausgehend von WG9 eine tragfähige Infrastruktur für die Ada Validierung aufzubauen. Eine Reihe

von Interessengruppen war schon auf dem Plan, aber es fehlten die prozeduralen und organisatorischen Grundlagen. In allen Prozeduren wurde stets auf die Entscheidungskette Ada Validation Facility – Ada Validation Organisation – Ada Joint Program Office verwiesen. Die zwei letztgenannten Einrichtungen gibt es aber seit September 1998 nicht mehr.

So standen im Wesentlichen zwei Alternativen zur Wahl:

- ersatzlose Streichung der technischen und organisatorischen Dachorganisation für Ada. Dies hätte zur Konsequenz, dass die Arbeit der Validierungsstellen nicht mehr koordiniert wäre. Die Zertifikate wären nicht mehr vergleichbar, jede AVF verfolgte ihre eigene Politik. Insbesondere wäre eine einheitliche Pflege der Testsuite nicht mehr gewährleistet. In der Konsequenz wäre dann ein Validierungszertifikat von EDS nicht mehr vergleichbar mit einem Zertifikat der IABG, da EDS im Gegensatz zur IABG keine staatliche Akkreditierung hat.
- Schaffung einer Koordinierungsstelle, die zumindest auf technischer und prozeduraler Ebene die Einheitlichkeit des Verfahrens gewährleistet. Schnell war klar, dass die Ada-Gemeinde die zweite Alternative befürwortete. Es war nur wieder nicht klar, wer hierfür das Geld aufbringen würde, denn diese Aufgabe lastet im Jahresdurchschnitt eine Person sicher zu mehr als die Hälfte aus. Nach einigen Wochen des Hin und Her zwischen USA und Europa wurde schließlich eine Lösung gefunden: Die Ada Resource Association nahm Bewerbungen für eine Institution “Ada Conformity Assessment Authority (ACAA)” entgegen. Diese soll von einem “ACAA Technical Agent” repräsentiert werden. Das Geld hierfür stellt – und das ist die eigentliche Ironie – die DISA (Defense Information Systems Agency) bereit; die Behörde des US-DoD für Informationssysteme. Damit ist das DoD wieder – oder immer noch – in das System der Ada Konformitätsprüfungen involviert. Allerdings hat diese keinerlei Mitwirkungsrecht.

Kommen wir nach dieser eher epischen Schilderung der Entwicklung der letzten zwei Jahre nun zu der systematischen Darstellung des Ist-Zustandes im April 1999.

3. Aktuelle Situation

Eine ganze Reihe von Sprachen sind bereits durch ISO standardisiert. Da Ada ein ISO-Standard ist, war es nahe liegend, das neue Ada Zertifizierungssystem unter dem Dach der ISO zu integrieren. Auch war ISO ja bisher schon in die Zertifizierung von Ada durch die Arbeitsgruppe 9 (ISO Working Group 9, kurz WG9) maßgeblich eingebunden.

Dieser Prozess wird von zwei Prinzipien geleitet:

- Herbeiführen einer einheitlichen Terminologie, kompatibel zu der der anderen Programmiersprachen, die ISO-Standards sind,
- kein inhaltlicher Bruch mit dem bewährten, bisher praktizierten Verfahren der Ada Compiler Validierung unter dem US-DoD.

Auf dieses Ziel hin wurde bereits ein großer Teil des Wegs zurückgelegt. Es liegt seit Januar 1999 ein Entwurf der ISO-Prozeduren vor, die als Working Draft des deutschen Arbeitskreises AK9 der ISO-Normierung von Ada zur Verabschiedung empfohlen wurden: *ISO/IEC Final Committee Draft 18009 Information Technology – Programming Languages – Ada: Conformity Assessment of Language Processor* [2]. Ausgehend von den Grundbegriffen des Software-Qualitätsmanagement wird im folgenden Kapitel das Modell für Ada Konformitätsprüfungen entwickelt, wie es in den ISO-Prozeduren für Ada zum Ausdruck kommt.

Die Integration unter ISO hat es mit sich gebracht, dass die Begriffe denen der anderen ISO-Standards angepasst werden sollten. Zuerst werden deshalb in Tabelle 1 die alten und neuen Bezeichnungen gegenübergestellt.

Neue Bezeichnung / Institution	Kurzform	Alte Bezeichnung / Institution	Kurzform
Ada Conformity Assessment Authority	ACAA	Ada Validation Organisation	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 (Sponsor von Ada)</i>
konformitätsgeprüft (certified conforming)		Validiert	
Konformitätsprüfung (Conformity Assessment)		Validierung (Gesamtprozess)	
Selbstprüfung (Self Testing)		Prävalidierung	
Konformitätsprüfung (Witness Testing)		Validierung (Prüf- Vorgang)	

Tabelle 1

Die Einflußmöglichkeiten der ARA als Vereinigung der Ada-Hersteller können aber nicht mit denen verglichen werden, über die das einstige AJPO verfügte. Hier fehlt ganz einfach das wirtschaftliche Potential.

Ausgangspunkt der Neu-Orientierung unter ISO ist [2]. Dort werden zu folgenden Themen Mindestanforderungen formuliert:

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

Dieser Standard ist eher eine Sammlung von Anforderungen denn ein Regelwerk. Die Anforderungen werden dann in nachgeordneten Standards in Regeln und Verfahrensanweisungen umgesetzt. Dieses Zertifizierungssystem ist in dieser Form seit Oktober 1998 in Kraft. Als Technical Agent für die ACAA wurde Randy Brukard von R.R. Software beauftragt. Darüber hinaus wurde ein ACAA Advisory Board etabliert, das der ACAA beratend zur Seite steht. Derzeit ist es mit 9 Mitgliedern aus Industrie, Anwendern, Universitäten und ACA Prüflabors besetzt.

Für eine Einordnung in die Standardisierung der Programmiersprachen im Allgemeinen werden unten in Tabelle 2 alle Sprachen verzeichnet, für die eine ISO-Arbeitsgruppe existiert. Diese Gruppen sind eingegliedert in das *Subcommittee (SC) 22 Programming Languages and their environments and system software interfaces* im *ISO/IEC Joint Technical Committee (JTC) 1*.

Darüber hinaus existieren Arbeitsgruppen, die nicht mehr aktiv sind, aber noch nicht aus der Liste der SC22 Arbeitsgruppen gelöscht sind.

ISO-WG	Programmiersprache
WG3	APL
WG4	COBOL
WG5	FORTRAN
WG9	Ada
WG11	Binding Techniques
WG13	Modula-2
WG14	C
WG15	POSIX
WG16	ISLisp
WG17	Prolog
WG19	Formal Specification Languages
WG20	Internationalization
WG21	C++
SC22/JSG	Java Study Group

Tabelle 2

4. Ein Modell für Konformitätsprüfungen von Programmiersprachen

Welche Bedingungen sollten erfüllt sein, um dem Anspruch "Modell" gerecht zu werden? Zur Entwicklung eines Modells gehen wir von den Grundlagen des Software-Qualitätsmanagements aus und entwickeln ein Begriffsgerüst, das in der vereinheitlichten Terminologie der ISO mündet.

Im Sinne eines internationalen Standards ist es zunächst einmal eine nahe liegende Forderung, dass es weltweit nur ein akzeptiertes und einheitliches Zertifizierungssystem gibt, in diesem Fall für die Programmiersprache Ada.

Prüfgegenstand ist eine identifizierte Ada Implementierung. Prüfziel ist, die Konformität des Prüfgegenstands *Ada Implementierung* mit der Norm DIN ISO/IEC 8652:1995 (Ada) festzustellen. Hierzu muss eine Prüfmethode gewählt werden. Diese Prüfmethode wurde mit der ersten Festschreibung von Ada als ANSI/Mil-Std 1815A als die Ada Compiler Validation Capability (ACVC) Testsuite entwickelt. Die ACVC Testsuite ist die Grundlage für eine Prüfprozedur, die eine definierte Menge von Prüffällen umfasst. Die Prüffälle lassen sich in zwei Kategorien einteilen: positive und negative Prüffälle (Testfälle). Ein positiver Testfall ist eine Programmsequenz, die dem Standard entspricht. Die Implementierung muss die Sequenz also dem Standard entsprechend "verarbeiten". Ein negativer Testfall enthält mindestens an einer Stelle eine bewusste Verletzung des Standards. Die Implementierung muss also diesen Testfall als nicht Standardkonform zurückweisen. Prüffälle werden aus Effizienzgründen zu einzelnen Tests ("Prüfungen") zusammengefasst, die jeweils ein gemeinsames Ziel haben. Ein Test ist also die kleinste über einen Namen identifizierbare Einheit der Testsuite. Zur Prüfprozedur gehören schließlich noch die Regeln, die die Interpretation eines Prüfungsergebnisses als *konform* oder als *nicht konform* gestatten. Schließlich ist es ineffizient, wenn der Prüfer bei jedem Prüffall den Sprachstandard wälzen muss um festzustellen, ob das Prüfergebnis die Konformität oder Nicht-Konformität der Implementierung bestätigt. Diese Regeln finden sich bei negativen Prüffällen als Kommentare im Quelltext, positive Prüffälle werten sich in der ACVC nach einem *self-checking* Mechanismus im Normalfall selbst aus und geben dem Prüfer eine standardisierte Meldung aus (*passed, failed, not applicable*). Diese Methode ist in [3] detailliert beschrieben.

Von der engen Auslegung des Begriffs *Prüfprozedur* als systematische Vorgehensweise bei der Durchführung einer Prüfung kommt man von alleine zu dem Begriffsinhalt von Prüfprozedur, wie er als Bestandteil des Ada Zertifizierungssystems verstanden wird: als detaillierte Verfahrensanweisung, die den gesamten Prozess einer Konformitätsprüfung beschreibt und eine Leitlinie darstellt, wie Testergebnisse zu interpretieren sind.

Obwohl man es für nahe liegend halten sollte, ist es keineswegs selbstverständlich, dass es wie bei Ada nur eine allgemein akzeptierte Testsuite gibt. Ja die Testsuite von Plum Hall (siehe Kap. 5) rühmt sich sogar, Sieger des einzigen jemals durchgeführten Wettbewerbs der C-Testsuiten zu sein. Eine Testsuite unterliegt wie jede andere komplexe Software selbst dem Qualitätsmanagement (Wartung, Weiterentwicklung) und Konfigurationsmanagement (Versionen, Änderungen, Dokumentation). Auch hierzu werden in den ISO-Prozeduren Anforderungen formuliert, die eine angemessene Nutzbarkeit der Testsuite zum Inhalt haben.

Schließlich gehören zur Prüfprozedur Regeln, wie die Summe der Ergebnisse aller Tests zu werten ist. Ergebnis einer Konformitätsprüfung ist immer ein Bericht mit einem detaillierten Nachweis über die Prüfungsergebnisse (*Prüfprotokoll*). Falls die Implementierung die Regeln erfüllt, dann kann als Beleg ein *Mark of Conformity* ausgestellt werden.

Zur Durchführung von Konformitätsprüfungen muss eine gewisse Qualifikation vorausgesetzt werden. Deshalb sollen diese nur von anerkannten Prüflabors durchgeführt werden. Was in diesem Zusammenhang unter einer *Anerkennung* oder *Akkreditierung* zu verstehen ist, muss später diskutiert werden. [2] fordert, dass ein Prüflabor nach den allgemeinen Grundsätzen der ISO betrieben wird, das heißt, Teil einer wohldefinierten Organisation ist und es auf Grundlage eines akzeptierten Qualitätshandbuchs arbeitet. Schließlich wird gefordert, dass die Prüflabors sich selbst untereinander anerkennen und damit Voraussetzungen für die Einheitlichkeit des Verfahrens gegeben sind.

Weiterhin wird neben der rein ausführenden Rolle eines Prüflabors eine technische Instanz benötigt, die übergreifende Aufgaben erfüllt. Diese soll an erster Stelle die Sprachenpolitik in Gestalt der Themen

- Qualitätsmanagement und Konfigurationsmanagement der Testsuite (z.B. Korrigieren oder Zurückziehen von Tests),
- Fortschreibung der Testsuite,
- Begleiten von einzelnen Konformitätsprüfungen,
 - erlaubte Modifikationen von Tests,
 - Entscheiden von *disputes*,
 - Qualitätskontrolle der Test Reports

formulieren und durchsetzen.

Bei Ada spielte auch immer die Frage eine Rolle, wie "allgemein" ein Zertifikat ist, d.h. welche Hardware-Plattformen durch ein Zertifikat abgedeckt sein sollten. Dabei kristallisierte sich eine gemeinsame Sichtweise heraus. Die Validierung selbst wird auf einer *Basis-Plattform* durchgeführt. Der Hersteller hat daraufhin die Möglichkeit, den Status *validiert* auf "ähnliche" Plattformen auszudehnen. Dies umfasst

- allgemein akzeptierte Maßnahmen der Software-Wartung (*perfective, corrective, adaptive*),
- Übertragung auf Prozessorfamilien,
- Re-Hosting der Implementierung.

Schließlich kostet diese Infrastruktur Geld. Diese Kosten können nicht vollständig durch die Gebühren abgedeckt werden, die von den Kunden einer Konformitätsprüfung aufzubringen sind. Zu den Rollen bei Konformitätsprüfungen gehört also auch ein Sponsor. Dieser verkörpert die Interessen der Anwender, die ein funktionierendes Zertifizierungssystem erwarten. Der Sponsor sollte seine finanzielle Stellung allerdings nicht für eine Einflussnahme auf die Politik des Zertifizierungssystems ausnutzen. Sponsor bei Ada ist seit 1998 die ARA mit Rückendeckung durch die DISA.

Abschließend werden hier noch einmal die Komponenten des Ada Zertifizierungssystems zusammengefasst, die von [2] gefordert und in den Grundzügen spezifiziert werden.

- ISO-Gremium, das die Evolution des Sprachstandards begleitet (hier: WG9),
- Sponsor,
- Conformity Assessment Authority,
- Conformity Assessment Laboratories,
- Testsuite mit Nutzer- und Versions-Dokumentation,
- Conformity Assessment Procedure (einschließlich Regeln zur Gültigkeit und Ausstellung von Zertifikaten).

Bild 1 unten gibt die wesentlichen Abhängigkeiten der bei den Ada Konformitätsprüfungen beteiligten Institutionen wieder.

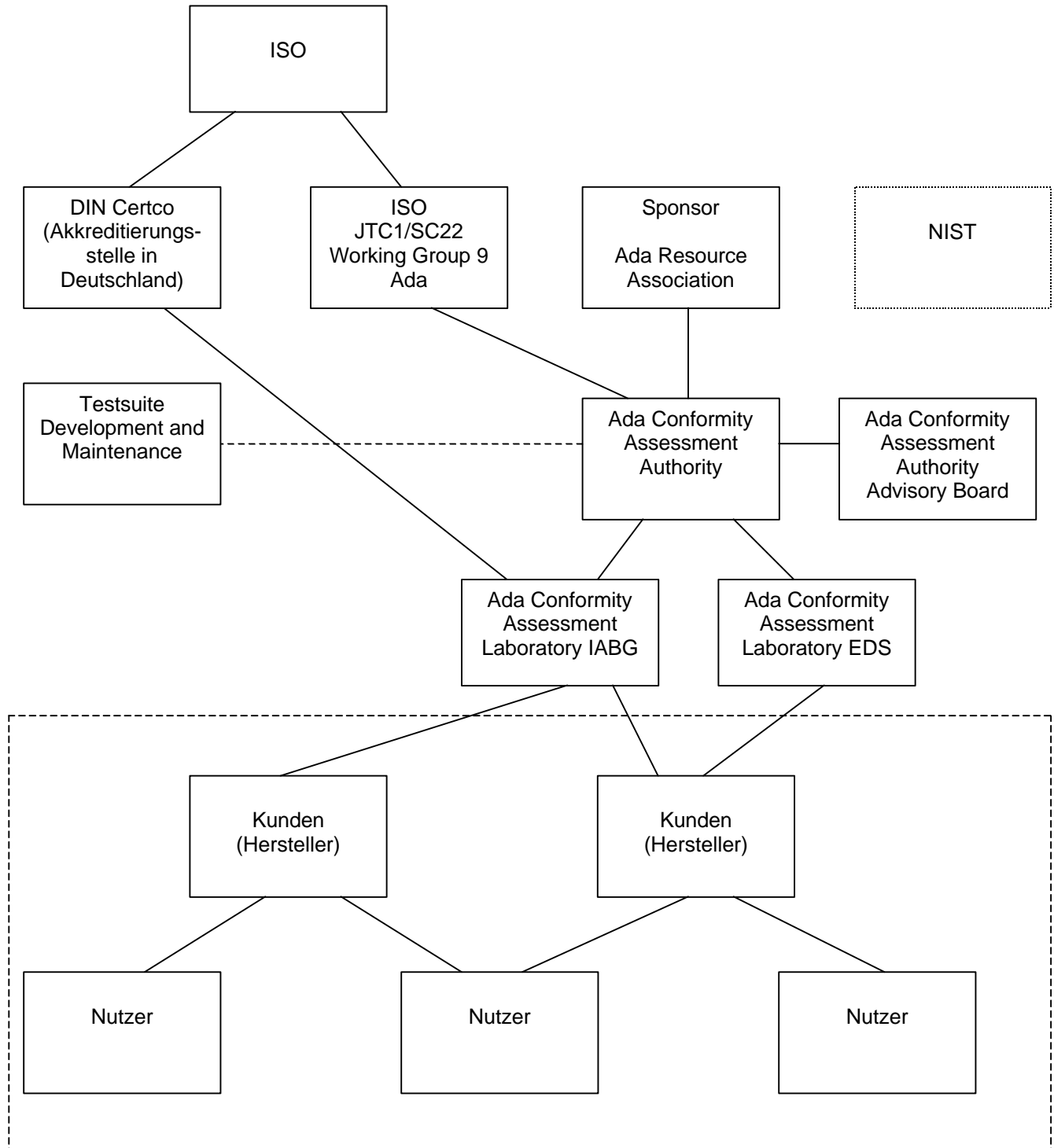


Bild 1

5. Die Situation bei anderen Programmiersprachen

Ein direkter Vergleich des Zertifizierungssystems von Ada mit anderen Programmiersprachen ist schwierig. Die Gründe hierfür liegen wieder in den programmiersprachlich-kulturellen Besonderheiten von Ada. Während bei Ada die Konformitätsprüfungen "zentralistisch" vom DoD aufgebaut wurden und wohl auch in diesem Sinne über die ISO weitergeführt werden, gibt es für keine andere Sprache ein derart einheitliches und weltweit akzeptiertes und praktiziertes Zertifizierungssystem. Deshalb sind auch diesbezügliche Informationen nicht bei einer zentralen Instanz wie bei Ada abrufbar. Insofern kann die Frage aus dem Titel dieser Ausarbeitung vorweg mit einem "ja, Ada ist Modell" beantwortet werden. Allerdings gibt es bei der Benutzerfreundlichkeit auch Einschränkungen bei Ada. Der Rest dieses Papiers soll diese These mit charakteristischen Beispielen belegen.

Wie im ersten Teil schon erwähnt, hat sich das US-amerikanische NIST 1997 aus der IT-Standardisierung, also auch der Standardisierung der Programmiersprachen, zurückgezogen. Die Konsequenzen sind weit reichend: so fällt in den USA zunächst einmal die Funktion des Akkreditiers für Prüflabors weg. Das bedeutet, dass ohne staatliche Autorisierung jede Organisation oder Institution mit genügend Geld und Marktmacht sich selbst zur Akkreditierungsstelle ernennen und damit Prüflabors zulassen, oder besser noch sich selbst zum Prüflabor deklarieren kann.

Weiterhin fällt auch die Rolle als Informationspool weg. Von NIST wird im Auftrag des Department of Commerce eine "Validated Products List" geführt, bis 1995 auch in gedruckter Form. Gemeinsame Grundlage aller dieser Produkte sind "Federal Information Processing Standards" (FIPS), nationale IT-Standards in den USA. So ist Ada neben ISO/IEC- eben auch ein FIPS-Standard (FIPS Pub 119). Seit 1997 werden jedoch keine neuen Einträge in die Liste mehr aufgenommen, so daß die Länge dieser Liste langsam aber sicher gegen Null geht. Beispielsweise hatte die Liste im Januar 1999 noch einen Eintrag für Fortran, vier Einträge für COBOL und dreizehn Einträge für C. Alle Einträge verlieren im Laufe des Jahres 1999 ihre Gültigkeit. Es ist damit klar, dass die NIST-Liste nicht mehr einen repräsentativen Stand an konformitätsgeprüften Produkten widerspiegelt.

Für einen Vergleich der Konformitätsprüfungen bei Ada sollten also Sprachen herangezogen werden, für die Konformitätsprüfungen auch aktuell praktiziert werden. Diese Aussage trifft derzeit auf C, C++ und Java zu. Desweiteren sind Konformitätsprüfungen für den Draft Standard Embedded C++ und für die C++ Standard-Bibliothek im Kommen. Auf dem "Markt" der Compiler Konformitätsprüfungen für C, C++ und Java lassen sich derzeit zwei Anbieter ausmachen: Plum Hall und Perennial. Die Erfahrungen mit den Zertifizierungssystemen zu C, C++ und Java stammen aus Auswertungen der Web-Seiten von Plum Hall und Perennial. Beide Firmen wurden von Experten gegründet, die am Standardisierungsprozess von C, C++ und Java direkt beteiligt sind.

Tabellen mit den wichtigsten Kenndaten der Testsuiten finden sich unten stehend. Die Metriken der Testsuiten (Codegröße, Anzahl der Testfälle, usw.) werden nicht konsistent angegeben und sind deshalb nicht direkt vergleichbar.

Plum Hall		
Sprache / Anwendung	Testsuite	Metrik
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

Tabelle 3

Perennial			
Sprache	Testsuite	Version / Release-Datum	Anzahl Testfälle
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

Tabelle 4

Zur Abrundung sei hier erwähnt, dass das National Technical Information System (NTIS) eine COBOL 85 Testsuite (CCVS Vers. 4.2 März 1993) vertreibt. Nach den vorliegenden Informationen ist ein Export nach Japan oder Europa jedoch verboten! Weiterhin sind Testsuiten für Fortran 77/90 im Einsatz.

Beide Anbieter, Plum Hall und Perennial, bieten weitere Softwareprodukte zur Qualitätssicherung von Anwendungssoftware an (z.B. guideline checkers, coverage analyzers, metric tools).

Die Entwicklung bei Perennial der Suites für C und C++ spiegelt sehr anschaulich auch die Evolution dieser Sprachen wider. Die Suite ACVS wurde 1990 von der US-Regierung in Auftrag gegeben, nachdem C als ISO-Standard akzeptiert wurde. Als kommerzielles Produkt entwickelte Perennial ACVS zur Testsuite CVSA weiter, die jetzt eine geschützte Marke von Perennial ist und ACVS als Teilmenge enthält. CVSA wurde schließlich zu C++VS ab 1992 als Testsuite für C++ weiterentwickelt. Die Suites bieten darüber hinaus auch die Möglichkeit, proprietäre Spracherweiterungen in das Testen mit einzubeziehen. In ihrer Struktur orientieren sich alle Testsuiten am Aufbau des jeweiligen Sprachmanuals.

Die Testsuiten von Perennial und Plum Hall sind nach den gleichen Grundsätzen aufgebaut. Im Gegensatz zu Ada werden sie jedoch zusammen mit Skript-Programmen ausgeliefert, mit denen die Tests auf Standard-Betriebssystemen (Unix, WinNT) direkt zum Ablauf gebracht werden können. Damit hat der Anwender der Testsuiten ein fertig einsetzbares Werkzeug in der Hand, mit dem nach beliebigen Selektionskriterien Teile der jeweiligen Testsuite ausgeführt und die Ergebnisse fertig aufbereitet präsentiert werden können. Auch werden Regressionstests von den Ablaufskripts unterstützt. In diesen Punkten muss man den kommerziellen Testsuiten ganz klar mehr Anwenderfreundlichkeit bescheinigen. Allerdings hat diese auch ihren Preis (> 30 TDM pro Jahreslizenz). Die Philosophie des Testens mit positiven und negativen Prüffällen ist bei allen Testsuiten gleich. Lediglich die Gruppierung von Prüffällen zu Tests variiert von Sprache zu Sprache. Dabei wird bei Ada auffallend viel Wert auf die negativen Prüffälle (*B-Tests*) gelegt.

Bei der Art der Nutzung der Testsuiten kann wiederum ein deutlicher Unterschied festgestellt werden. Während bei Ada das eigentliche Ziel ein förmlicher Akt des *third-party-testing* durch ein unabhängiges Prüflabor mit anschließender Erteilung eines Zertifikats ist, dient bei C, C++ und Java die Testsuite in erster Linie als Basis für Selbsttests der Compiler-Hersteller. Ein Zertifikat (*branding*) ist dabei die Ausnahme. Bis auf die Ada-Zertifikate der IABG ist aber keines der sonstigen Zertifikate durch ein staatlich akkreditiertes Prüflabor gedeckt. In dieser Hinsicht herrscht also wieder voll die freie Marktwirtschaft. Eine Vergleichbarkeit der jeweiligen konformitätsgeprüften Implementierungen ist also nicht gegeben:

- konkurrierende Testsuiten,
- keine zentrale Liste der konformitätsgeprüften Implementierungen,
- kein Verfallsdatum für den *validiert*-Status,
- in der Regel keine Prüfungen durch unabhängige Prüflabors.

In Übereinstimmung mit der neuen Entwicklung bei Ada sind auch die Prüflabors für Konformitätsprüfungen bei C, C++ und Java nicht mit einer aktuellen staatlichen Akkreditierung ausgestattet. In den USA wird es nach dem Rückzug von NIST aus der Standardisierung gar keine staatlich anerkannten Prüflabors mehr geben, alles ist dem freien Spiel der Kräfte ausgeliefert. Und in Europa ist außer der IABG für Ada überhaupt kein aktives Prüflabor für irgend eine der Sprachen Ada, C, C++ und Java sichtbar. Das früher aktive NCC in Großbritannien und AFNOR in Frankreich sind dem Beispiel von NIST gefolgt und sind als Prüflabors oder Standardisierungseinrichtungen nicht mehr unmittelbar aktiv.

6. Zusammenfassung und Bewertung

Es wurde gezeigt, dass die Konformitätsprüfungen für Ada eine Sonderrolle spielen. Hier wurde mit großem Aufwand, zentral gesteuert, eine Testsuite entwickelt und ein Zertifizierungssystem aufgebaut. Die Errungenschaften des Zertifizierungssystems konnten auch nach dem Rückzug des AJPO erhalten werden. Die Testsuite deckt den gesamten Sprachumfang ab und ist frei erhältlich. Allerdings kann die Entwicklung der ACATS nicht als abgeschlossen bezeichnet werden. Informationen über zertifizierte Implementierungen sind an einer zentralen Stelle erhältlich. Für die Vergleichskandidaten C, C++ und Java gibt es konkurrierende Testsuiten als kommerzielle Produkte. Die Testsuite dient dort in erster Linie als Grundlage für Selbsttests von Compiler-Herstellern. Prüfungen von unabhängigen Prüflabors sind nicht die Regel, das heißt, dass auch Zertifikate als Belege für erfolgreiche Prüfungen nicht die Regel sind.

Die unten stehende Tabelle 5 fasst noch einmal in einen Vergleich die wesentlichen Unterschiede zwischen den Konformitätsprüfungen für Ada und C/C++/Java zusammen.

Vergleichskriterium	Ada	C, C++, Java (Plum Hall, Perennial)
Zertifikate	Teil des Prozesses	Branding auf Anfrage
<i>third-party-testing</i>	regelmäßig	Ausnahme (z.B. "Perennial Conformance Test Center")
Unabhängige Prüflabors	ja	identisch mit Anbieter Testsuite
Akkreditierung	D: ja (IABG bei DIN CERTCO akkreditiert), USA: nein (EDS)	keine aktuelle staatl. Akkreditierung
globale Liste der geprüften Implementierungen	ja, im Internet abrufbar	nein, konkurrierende Testsuiten
Testsuite Kosten	frei	hoch (>30 TDM; Testsuiten als geschützte Trademarks der Anbieter)
Testsuite Pflege	unabhängig vom Vertrag mit Kunden	Maintenance im Rahmen des Kundenvertrags (6/12 Monate nach Kauf eingeschlossen)
Testsuite Komfort	nur Tests selbst unter KM	Ausführungs-/Auswerteskripts Teil der Testsuite
Regression Testing	nicht unterstützt	von den Skripts unterstützt
Technische Autorität als eigene Organisationseinheit	Ja	nein

Tabelle 5

In aller Nüchternheit muss also festgestellt werden, dass die Bedeutung der Konformitätsprüfungen für Compiler derzeit ganz allgemein rückläufig ist. Testsuiten werden als Teil der Hersteller-internen QM-Maßnahmen regelmäßig genutzt. Ein Zertifikat als Beleg für eine erfolgreiche Konformitätsprüfung hat jedoch keinen großen Stellenwert für die Vermarktung eines Produkts. Positiv ausgedrückt wird die Einschätzung vertreten, dass die Compilertechnik inzwischen einen hohen Reifegrad erreicht hat, wodurch der Ruf nach einem aufwendigen Verfahren eines Third-Party-Testing leiser geworden ist. Die Konzentration auf wenige marktgängige Hardware- / Betriebssystemplattformen ist ebenfalls ein Beitrag zur Konsolidierung der Compilerentwicklung.

So kann die Frage aus der Titelzeile mit einem eingeschränkten *Ja* beantwortet werden. Durch die Ada Validierung wurde ein Modell geschaffen, das in Bezug auf die Vollständigkeit und Unparteilichkeit des Verfahrens vorbildlich ist. Allerdings wird damit nicht (mehr) das Haupt-Bedürfnis des Markts getroffen, der in erster Linie ein wirkungsvolles Instrument zum Selbsttest von Implementierungen verlangt. Begleitend zur Weiterentwicklung der Programmiersprachen wird das Instrument der Konformitätsprüfungen allerdings nichts von seiner Bedeutung einbüßen, ganz im Gegenteil. Schließlich ist das Bedürfnis nach Durchsetzung von herstellerunabhängigen Standards aktueller denn je.

7. Literatur

- [1] Am Ende der Übergangs von Ada 83 zu Ada 95: brauchen wir noch Ada Compiler Validierungen? Software-Technik-Trends 18.4, Gesellschaft für Informatik 1998.
- [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-Verlag.