Am Ende des Übergangs von Ada 83 zu Ada 95: Brauchen wir noch Ada Compiler Validierungen?[1]

Michael Tonndorf
IABG
Ada Validation Facility
85521 Ottobrunn
Tonndorf@iabg.de

Abstract: Nach den Regelungen der *Ada Validation Procedures* endete die Gültigkeit aller Zertifikate, die unter ACVC 1.11, 2.0 und 2.0.1 ausgestellt wurden, am 31. März 1998. In diesem Papier wird eine Bilanz gezogen über die Entwicklung der ACVC Testsuite und die Entwicklung des werkzeuggestützten Validierungsverfahrens bei der Validierungsstelle IABG. Anschließend wird ein Ausblick auf die zu erwartende weitere Entwicklung der Ada Validierung gegeben.

Schlüsselworte: Ada Compiler Validierung, Ada Standardisierung, Ada Sprachpflege.

1 Einleitung

Die Programmiersprache Ada wurde zuerst von ANSI im Jahr 1983 standardisiert. Von ISO wurde die Standardisierung 1987 nachvollzogen. Zur Nachweisführung, daß ein Compiler die Sprache gemäß der Sprachdefinition implementiert, wurde die Ada Compiler Validation Capability (ACVC) Testsuite entwickelt. Das US-DoD, Sponsor der Sprache und größter Anwender von Ada-Produkten, verlangte, daß in von ihm beauftragten Projekten nur validierte Compiler zum Einsatz kommen.

Im September 1984 wurde die IABG als Ada Validierungsstelle in Deutschland akkreditiert. Ihre erste Validierung führte die IABG im November 1984 durch. Nach drei Jahren manueller Validierungen wurde begonnen, Unterstützungswerkzeuge für die Validierung zu entwickeln. Die Entwicklung der Werkzeuge steht in Verbindung mit der Verbreitung von portablen Computern und dem Internet. Seitdem hat die IABG 145 Validierungen durchgeführt und ist eine der zwei aktiven Validierungsstellen weltweit.

Dieses Papier ist wie folgt aufgebaut: In Kapitel 2 findet sich eine Zusammenfassung der Prinzipien der Ada Compiler Validierung. Kapitel 3 enthält eine Darstellung der Erfahrungen der IABG und der Entwicklung der Validierungswerkzeuge. In Kapitel 4 wird die Entwicklung zusammenfassend bewertet und ein Ausblick auf die Zeit nach der Schließung des Ada Joint Program Office gegeben.

2 Prinzipien der Ada Compiler Validierung

2.1 Allgemeines

Mit der Validierung eines Ada Compilers soll demonstriert werden, daß ein Compiler die Sprache genau so implementiert, wie sie im Sprachmanual definiert ist. Es ist offenkundig, daß dieses Ziel auch nicht annähernd erreicht werden kann. Die Komplexität der Sprache macht es zu einem aussichtslosen Unterfangen, einen Compiler auf hundertprozentige Normkonformität prüfen zu wollen. Insbesondere die Kombination von zwei oder mehreren Sprachelementen, wie sie in realen Programmen alltäglich vorkommt, kann nicht mit akzeptablem Aufwand auf korrekte Implementierung getestet werden. Um trotzdem zu einem aussagekräftigen Verfahren zum Prüfen der Normkonformität zu kommen, wurde vom US-DoD das Ada Validierungssystem beauftragt und in Kraft gesetzt. Die Elemente des Systems sind

- die Organisationen, die bei der Validierung mitwirken,
- die Testprogramme einschließlich der Dokumentation (ACVC),
- das Regelwerk, das den Validierungsprozeß definiert und beschreibt. Dazu gehören insbesondere auch die Akzeptanzkriterien. Die Ada Validation Procedures [1] und das AVF Operation Manual sind die wichtigsten Dokumente in diesem Zusammenhang.

[1] Dieser Beitrag stellt die schriftliche Fassung des Vortrags auf dem Ada-Deutschland Workshop Bremen 1998 dar, welcher vom 22.4. - 23.4.1998 in Bremen bei der DASA-RI stattfand. Die Beiträge des Workshops sind im Bericht FZKA 6177des Forschungszentrums Karlsruhe erschienen. Anfragen bitte an Dr. Hubert B. Keller, stellvertr. Sprecher von Ada-Deutschland.

Begriff	Bedeutung
Testergebnis	Das Ergebnis eines ACVC Tests (im allgemeinen eine Datei mit einem Compiler/Binder-Listing oder die Ausgabe eines ausführbaren Tests)
Validierungs-ergebnis	Alle Testergebnisse einer Validierung zusammengenommen
Wertungs-kategorie	Eine der Kategorien *PASSED, FAILED, UNSUPPORTED, NOT APPLICABLE* (siehe unten)
Ergebnis-einstufung	Die Zuordnung eines Testergebnisses zu genau einer der Wertungskategorien
Auswertungs-ergebnis (Validierungs-ergebnis)	Alle Ergebniseinstufungen einer Validierung zusammengenommen

Tabelle 1

2.2 Überblick über den Validierungsprozeß

Die Ada Compiler Validation Capability (ACVC) ist die einzige und mit Einschränkungen fachlich akzeptierte Methode zum Prüfen der Normkonformität von Ada Compilern. Von den ACVC Tests werden aber keine Leistungswerte der Implementierung ermittelt. Es werden auch keine Optimierungsverfahren oder besondere Charakteristiken der Implementierung abgeprüft. Die ACVC Testsuite [3] ist eine Sammlung von Testprogrammen, die sich aus sechs Klassen zusammensetzt: A, B, C, D, E und L.

Der erste Buchstabe eines Testnamens identifiziert die Klasse, zu der der Test gehört. Die Klassen A, C, D und E enthalten ausführbare Tests, die unter Zuhilfenahme von Report-Funktionen selbstablaufend sind und das Ergebnis in einem standardisierten Format ausgeben. B-Tests enthalten bewußte Sprachverletzungen, die vom Compiler erkannt und zurückgewiesen werden müssen. L-Tests enthalten intendierte Fehler, die beim Binden erkannt und zurückgewiesen werden müssen.

In Tabelle 1 werden die Begriffe definiert, die im weiteren Verlauf des Papiers verwendet werden.

Die Anzahl der Test in den sechs Klassen A, B, C, D, E, L variierte im Laufe der Zeit. Die größte Anzahl von Tests wurde bei ACVC 1.11 mit 4.071 Tests erreicht. Die nachfolgend angegebenen Zahlen geben für ACVC 1.X größenordnungsmäßig die Verteilung der Tests auf die Testklassen an:

- 2500 ausführbare Tests (Klasse A, C, D, E),
- 100 Binder Tests (Klasse L),
- 1500 Compiler Tests (Klasse B). Insgesamt gibt es in allen B-Tests zusammengenommen über 12.000 konstruierte Fehler, die vom Compiler erkannt werden müssen.

Für die aktuelle Version ACVC 2.1, die seit Juli 1997 in Kraft ist, gibt die nachfolgende Tabelle 2 die wichtigsten Kennzahlen wieder:

Die Technik der Ada-Validierung kann an dieser Stelle nicht im Detail beschrieben werden. Die grundlegenden Ideen wurden 1993 bei der Ada-Europe-Tagung präsentiert [4].

Tabelle 3 gibt einen Überblick über die sechs Schritte eine Validierung. Diese Schritte sind ausführlich in den Ada Validation Procedures [1] beschrieben. Das AVF-Handbuch macht hierzu weitere detaillierte Ausführungen. Einer AVF ist die Methode der Durchführung einer Validierung freigestellt. Sie muß aber auch stets darauf vorbereitet sein, zu einem Audit zur Verfügung zu stehen.

Die Motivation für den Schritt *Prävalidierung* (siehe Tabelle 3) stammt aus den Zeiten, als ein Validierungslauf zwei oder mehr Wochen dauerte. Eine Prävalidierung wird vom Kunden in eigener Zuständigkeit durchgeführt. Damit sollte das Risiko für eine Validierung beim Kunden minimiert werden („Test Readiness Review“). Jedoch ist es immer noch gute Praxis, zumindest eine Prävalidierung für eine zusammengehörige Serie von Validierungen durchzuführen. Mit den Ergebnissen der Prävalidierung kann die AVF eine Datenbasis für werkzeuggestützte Vergleiche mit den Validierungsergebnissen aufbauen und somit den Arbeitsaufwand während der Validierung beim Kunden minimieren.

ACVC 2.1	Total	Core Tests	Specialized Needs Annexes	Foundation	Docu-ments	Sup-port
Anzahl Dateien	4242	3919	245	44	11	23
Anzahl Tests	3662	3474	188	-	-	-

Tabelle 2

Schritt	Aktivität	Bemerkung
1	Vertrag über Validierungs-Dienstleistungen	Etabliert eine Kundenbeziehung mit der AVF
2	Prävalidierung	• erster Selbsttest in Eigenverantwortung des Kunden • Klärung offener Punkte • Wertung des Validierungsergebnisses durch die AVF • Aufbereitung der Ergebnisse als Vergleichsbasis für die Validierung
3	Validierung	Ablauf der kundenspezifisch angepaßten Testsuite pro zu testender Implementierung unter Aufsicht der AVF
4	Declaration of Conformance	Unterschriebene Erklärung des Kunden in der er bestätigt, den Ada Standard nicht absichtlich zu verletzen und keine Rechte dritter Parteien zu beeinträchtigen
5	Validation Summary Report	Ein Bericht in einer einheitlichen Struktur, in dem der Validierungsprozeß beschrieben wird und alle Validierungsergebnisse, die nicht PASSED sind, erläutert werden
6	Validation Certificate	Ein vom AJPO ausgegebenes Zertifikat, das die Implementierung, die ACVC Version, die Zertifikatsempfänger und das Verfallsdatum des Zertifikats verzeichnet

Tabelle 3

3 Beschreibung der Werkzeugumgebung CANDY

IABG begann 1988, eine Werkzeugunterstützung für Ada Compiler Validierungen zu implementieren. Die Entwicklung ist ausführlicher in [6] dargestellt. An dieser Stelle soll nur die heute im Einsatz befindliche Konfiguration des Werkzeugs CANDY vorgestellt werden.

3.1 Anforderungen an Werkzeuge und Prinzipien der Werkzeugentwicklung

Die folgenden Anforderungen lagen der Entwicklung der Validierungswerkzeuge zu Grunde:

- Geringer Lernaufwand und Benutzerfreundlichkeit,
- Wartbarkeit und Offenheit für Weiterentwicklung,
- Anpaßbarkeit an unterschiedliche Layouts von Listings und Compilerdiagnosen,
- Erhöhtes Vertrauen in die Korrektheit des Auswertungsergebnisses im Vergleich zur manuellen Methode, Nachvollziehbarkeit des Ergebnisses,
- Keine Verlagerung von Entscheidungen über Bestehen oder Nichtbestehen eines Tests in das Werkzeug, mit Ausnahme der normalen ausführbaren Tests,
- Unterstützung der Ergebnisauswertung, Berichtsfunktion/Statistik und Buchführung.

Zum Nachweis und zur Rekonstruktion des Testverlaufs werden alle Protokolldateien („logfiles“) aufbewahrt. Diese Protokolldateien werden von CANDY jedoch nicht ausgewertet.

Im Gegensatz zu möglichen anderen Ansätzen gibt es bei der Methode der IABG keine „automatische“ Wertung von B- oder L-Tests. Diese Idee wurde ausführlich diskutiert, aber dann doch verworfen. Die Tatsache, daß ein Werkzeug entscheidet, ob der Compiler einen B-Test „richtig“ behandelt, wird nicht als akzeptabel angesehen. So steht am Anfang jeder Kette von Dateivergleichen durch CANDY zuerst eine manuelle Bewertung des Testergebnisses.

Eine Validierung läuft immer in zwei Phasen ab, die untereinander verzahnt sein können:

- Ablauf der Testprogramme unter der zu validierenden Implementierung,
- Auswertung der Testergebisse.

Es liegt in der Zuständigkeit des Kunden, die Testergebnisse bereitzustellen. Er bringt die Testprozeduren zum Ablauf, wobei die AVF den Prozeß überwacht. Die Auswertung der Testergebnisse kann in separaten Schritten durchgeführt werden, sobald diese verfügbar sind.

3.2 Manuelle Ergebniseinstufung

Testergebnisse müssen immer dann manuell ausgewertet werden, wenn keine Werkzeugunterstützung verfügbar ist. Dies war insbesondere zu Beginn der Validierungsaktivitäten der IABG so. Zu diesem Zweck muß für jedes Testergebnis festgestellt werden, ob es in Übereinstimmung mit den Sprachregeln ist. Dazu muß für

- jeden ausführbaren Test der Ergebnisausdruck,
- jeden B-Test das Compiler Listing,
- jeden L-Test das Binderprotokoll

ausgewertet und einer der Wertungskategorien zugeordnet werden.

Jede Ergebnisbewertung und Auffälligkeit ist zur Dokumentation in ein Logbuch einzutragen. Es ist offenkundig, daß die Auswertung eines Validierungsergebnisses eine zeitaufwendige Angelegenheit ist. Die vollständige Auswertung einer Validierung benötigt mindestens fünf Personentage. Dazu kommt noch einmal der Aufwand für die Buchhaltung von ungefähr drei Tagen. Falls auch die Prävalidierung zu berücksichtigen ist, dann verdoppelt sich der Aufwand.

Der beabsichtigte Gewinn durch CANDY liegt nicht primär in der automatischen Ergebniseinstufung. Dies geschieht nur für die ausführbaren Tests. Der wirkliche Vorteil liegt im Vergleichen von B-Test Ergebnissen mit bereits ausgewerteten Testergebnissen. Dadurch wird der manuelle Aufwand bei einer Validierung auf die Einstufung der Tests reduziert, deren Vergleichsergebnis nicht übereinstimmt. CANDY benötigt immer zwei Validierungsergebnisse, die es miteinander vergleichen kann: dies können das Prävalidierungsergebnis und das Validierungsergebnis für dieselbe Plattform sein, oder aber zwei Validierungsergebnisse von verschiedenen Plattformen. Erfahrungsgemäß war der automatische Vergleich zwischen den Validierungsergebnissen verschiedener ACVC-Versionen schwierig: nicht nur daß Tests geändert und hinzugefügt wurden. Oft verbesserten die Hersteller anläßlich des Versionenwechsels auch Phasen oder Pässe des Compilers, so daß die Ergebnisprofile erheblich voneinander abwichen.

Innerhalb der ersten drei Jahre als Validierungsstelle wurden alle Validierungen komplett manuell ausgewertet. Teil der ACVC Distribution war das *ACVC Logbook*, in das alle Ergebniseinstufungen der Tests eingetragen wurden. Das Logbook war der Nachweis des Fortschritts und des Status der Validierung. Für die vollständige manuelle Auswertung einer Validierung werden ungefähr acht Personentage benötigt, einschließlich der Buchführung. Wenn wie im Normalfall eine Prävalidierung dazukommt, dann verdoppelt sich der Aufwand.

Eine vollständige Validierung umfaßt jedoch noch mehr:

- kundenspezifische Aufbereitung und Auslieferung der Testsuite,
- Sonderbehandlung der modifizierten Tests,
- Bearbeitung aller implementierungsspezifischen Sonderfälle, die in der Regel in Test Disputes unter Einschaltung des AVO gelöst werden müssen,
- Überwachung und Begleitung von Installation und Ablauf der Testsuite,
- Erstellung des *Validation Summary Report*.

3.3 Werkzeuggestützte Ergebniseinstufung

Die manuelle Verwendung des *ACVC logbook* legte eine werkzeuggestützte Implementierung dieses Verfahrens nahe, zumal die bekannten Unix-Werkzeuge nicht den benötigten Leistungsumfang boten. Die Entwicklung des Werkzeugs CANDY (der Name wurde lautmalerisch nach dem Vergleich von *base* und *candidate results* kreiert) erfolgte in 4 Versionen, vom monolithischen Hostcomputer-Programm über ein portables Tool auf Unix-Basis bis zum Notebook-Tool mit TCP/IP-Anschluß. Die aktuelle Funktionalität ist in [6] ausführlicher dokumentiert. In [5] wurde erstmals ein Gesamtkonzept vorgestellt.

Die wesentliche Leistung von CANDY besteht darin, die Vergleichsresultate von Paaren von Testergebnissen zu ermitteln. Für jeden Test in der Testsuite sind die folgenden Vergleichsergebnisse möglich (siehe Tabelle 4):

CANDY Status	CANDY Report
Basis- und Kandidat-Ergebnis existieren und stimmen überein nach Maßgabe der Vergleichsmuster	match
Basis- und Kandidat-Ergebnis unterscheiden sich in Aspekten, die nicht durch die Vergleichsmuster abgedeckt sind	mismatch
Mindestens ein Ergebnis (Datei) fehlt oder ist unvollständig	no base, no candidate, none, incomplete

Tabelle 4

Diese Reports werden von CANDY für alle Arten von Ergebnissen erstellt. Für ausführbare Tests wird zusätzlich die Ergebniseinstufung ausgegeben. Das bedeutet, für alle Testergebnisse, die übereinstimmen (match), wird die Wertung des Basis-Ergebnisses auf das Kandidat-Ergebnis übertragen. Wenn die

Ergebnisse übereinstimmen, dann ist also kein manueller Aufwand zur Ermittlung der Ergebniseinstufung notwendig. Die Erfahrungen von mehr als 80 Validierungen nach Beginn der Entwicklung der Werkzeuge haben gezeigt, daß 85% bis 99% aller Testergebnisse einer Validierung übereinstimmen, in Abhängigkeit von der Ähnlichkeit der Testergebnisse.

Eine typische Sitzung mit CANDY läuft folgendermaßen ab:

1. Bereite die erforderlichen Daten für einen CANDY-Lauf vor (Basis und Kandidat-Ergebnisse in Verzeichnisstruktur einordnen);
2. Starte das CANDY-Programm;
3. Setze die Optionen für einen CANDY-Lauf:
   - Pfadangabe zu den Basis- und Kandidat-Ergebnissen,
   - ggf. Einschränkung auf einen Ausschnitt aus der Testsuite,
   - Error- und Ignore-Patterns zur Steuerung der erlaubten Toleranzen;
4. Starte die Vergleichsausführung zum Vergleichen aller ausgewählten Ergebnispaare und zum Erzeugen des resultierenden *validation book*.
5. Bereite durch CANDY Übersichten über die Vergleichsergebnisse und die noch manuell zu bearbeitenden Testergebnisse auf.
6. Sichere und archiviere die Validierungsergebnisse, CANDY's Listen und Protokolldateien und das *validation book*.

Die Grafik in Bild 1 zeigt die grundsätzliche Arbeitsweise von CANDY: für jeden Test aus dem Logbuch wird das entsprechende Basis- und Kandidat-Ergebnis auf die *match*-Bedingung geprüft, in Abhängigkeit von den gesetzten Vergleichsmustern.

An manuellem Aufwand verbleibt die Überprüfung aller Ergebnisse, die nicht übereinstimmten. Darüberhinaus gibt es eine Anzahl von Tests, die immer individuell behandelt werden müssen, da über das reguläre Testergebnis hinaus zusätzliche Kriterien abgeprüft werden müssen (z.B. Informationen aus dem Ablaufprotokoll des Betriebssystems oder spezielle Informationen von Tests zu einem *Specialized Needs Annex*).

4 Zusammenfassung und Ausblick

4.1 Bewertung des erreichten Stands

Ada wurde u.a. als Sprache für sicherheitskritische Anwendungsbereiche entworfen. Deswegen werden an den generierten Code besondere Qualitätsanforderungen gestellt, wodurch sich wiederum hohe Anforderungen an den Compiler ergeben. Ein Ada-Compiler ist ein komplexes Produkt, für das parallel zur Standardisierung der Sprache mit großem Aufwand ein Validierungssystem entwickelt wurde. Wenn eine Validierung manuell durchgeführt wird, dann besteht die Gefahr, daß sich Fehler bei der Auswertung einschleichen. Außerdem ist das Verfahren personalintensiv und damit teuer. Die Erfahrungen der letzten zehn Jahre mit Ada Validierungen haben gezeigt, daß der Validierungsprozeß unter Zuhilfenahme von Werkzeugen wesentlich verbessert werden kann. Es verbleibt die Implementierung von Verwaltungsfunktionen (z.B. für Kundeninformationen), die aber den Validierungsprozeß selbst nicht beeinflussen.

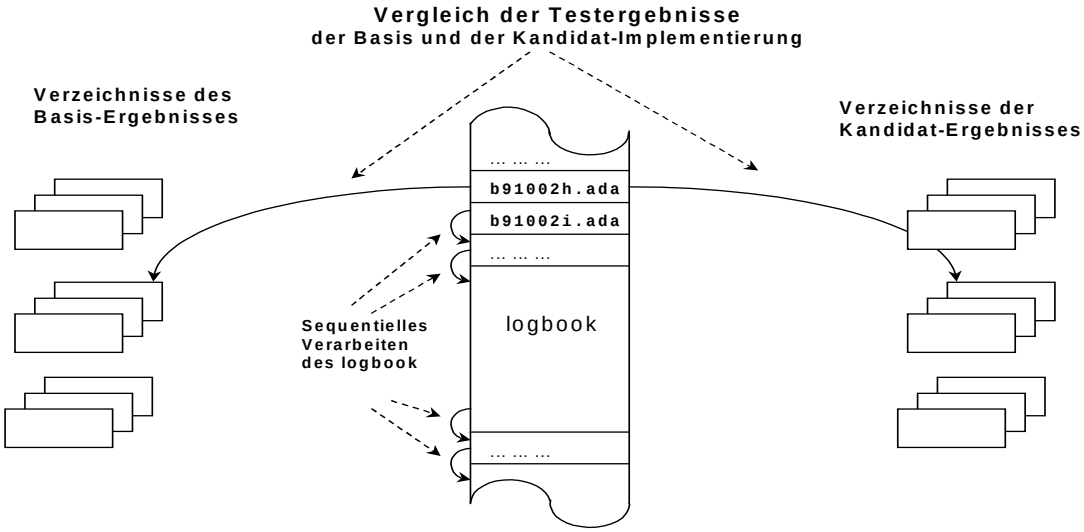

Bild 1: Arbeitsweise von CANDY

Die Erfahrungen der letzten zehn Jahre belegen die Fortschritte: es konnten zuletzt im Juni 1997 zehn Validierungen in sechs Arbeitstagen erledigt werden. Vorausgegangen war eine reguläre Prävalidierung. Der Bearbeiteraufwand belief sich auf acht Tage für die Prävalidierung, sechs Tage für die Prävalidierung zuzüglich drei für Verwaltungsaufgaben. Dies ergibt eine Summe von 17 Tagen. Ohne Werkzeugunterstützung hätte man für die zehn Validierungen zuzüglich der Prävalidierung 11*8, das heißt 88 Bearbeitertage benötigt. Dies bedeutet in der Summe eine Reduzierung des Aufwands um ca. 80% gegenüber dem rein manuellen Verfahren. Diese Kalkulation berücksichtigt nur Zeitaufwände, noch überzeugender sind natürlich die Kosteneinsparungen. Die durchschnittlichen Gebühren für eine Validierung liegen inzwischen unter 10% der Gebühren zu Beginn der Validierungstätigkeiten. Dies ist abhängig von der Anzahl der Validierungen, die parallel durchgeführt werden können.

Erwähnenswert ist auch, daß die Aktivitäten der IABG niemals in irgend einer Form subventioniert wurden. Alle Arbeiten an den Werkzeugen trugen sich intern aus den Gebühren für die Validierung. IABG würdigt hiermit das ausgezeichnete Engagement der Studenten von Staffordshire University, G.B., aus den letzten zehn Jahren. Sie haben einen wesentlichen Beitrag zur Fortentwicklung des Werkzeugs CANDY und des Validierungsverfahrens geleistet.

Unsere Erfahrungen haben gezeigt, daß die Validierung von Ada Compilern optimiert und ohne Reibungsverluste in den Entwicklungszyklus der Hersteller integriert werden kann.

4.2 Ausblick

Seit 1995 ist der fortgeschriebene Standard Ada 95 [2] in Kraft. Für die Gestaltung des Übergangs von Ada 83 zu Ada 95 legte man sich im AJPO auf einen Zwei-Jahres-Zeitraum von März 1995 bis März 1997 fest. Dieser Zeitraum wurde noch einmal bis zum 30. Juni 1997 verlängert. In dieser Zeit waren die ACVC-Versionen 2.0 und 2.0.1 in Kraft, wobei zum Erlangen eines Zertifikats nicht die Implementierung des vollen Sprachumfangs nachgewiesen werden mußte. Zum Ende März 1998 schließlich verloren alle Zertifikate nach ACVC 1.11 (Ada 83) und nach ACVC 2.0.(1) (Ada 95 eingeschränkt) ihre Gültigkeit. Damit existierten am Ende der Übergangszeit nach Ada 95 am 31. März 1998 genau drei Implementierungen, die gemäß ACVC 2.1 validiert sind (Green Hills Software).

Im November 1997 kündigte das US-Verteidigungsministerium an, daß es das Ada Joint Program Office spätestens im September 1998 schließen werde. Dieser Termin wurde mittlerweile auf den 30. Juni 1998 vorverlegt. Die Konsequenzen der Umsetzung dieser Ankündigung sind weitreichend:

- keine direkte Finanzierung einer Ada Infrastruktur,
- keine zentrale Beeinflussung der Entwicklung von Ada,
- keine Kontrolle der Validierung, wie bisher durch die “Ada Validation Organisation” (AVO) gegeben,
- Unsicherheit über die Zukunft der Informationspolitik, wie sie bisher durch das Ada Information Clearinghouse (AdaIC) sehr erfolgreich ausgeübt wurde.

Wohin sich die Ada Validierung bis zum Ende des Jahres 1998 entwickelt haben wird, ist derzeit nur schwer abschätzbar. Die vorherrschende Meinung ist, daß die Validierung ein wesentliches Element der Sprache Ada bleiben sollte. Und dies sehen sogar die meisten der Ada Hersteller so.

Die Fragen, die für die Zeit nach der Aufgabe der Verantwortung der Ada Validierung durch das US-DoD gelöst werden müssen, sind vielfältig:

- Wer bestimmt die “Sprachenpolitik”? (Die Frage “Wer” soll sich allgemein auf Insititutionen, Organisationen oder Gremien beziehen, nicht auf Einzelpersonen.)
- Wer überwacht die Weiterentwicklung der ACVC Testsuite im Sinne eines “Test Method Control Board”?
- Wer pflegt die ACVC Testsuite im Sinne eines “ACVC Team” weiter?
- Wer koordiniert die Arbeit der Ada Validierungsstellen (AVFs) – falls es auch in Zukunft mehr als eine geben wird?
- Wer hat die Autorität, im Zusammenhang mit Validierungen Entscheidungen zu treffen bezügl. Sprachinterpretation, Grenzfällen etc. (Umsetzung von Resolutionen der Ada Rapporteur Group)?
- Wer braucht ein Zertifikat zur Bestätigung einer erfolgreichen Validierung?
- Was sind die Kosten der einzelnen Aktivitäten?
- Wer kommt für die Kosten auf?
- Wie funktioniert die Ada Öffentlichkeitsarbeit in Zukunft, was kann man vom DoD und von DISA erwarten?
- Was ist die Rolle der Ada Resource Association (ARA) und was kann man von ihr erwarten?

Eine Diskussion über diese Fragen wurde auf der TRI-Ada Tagung 1997 in den USA initiiert. Dabei hat sich zwar das Augenmerk auf die ARA konzentriert. Auch wurden erste Dokumente im Sinne der Fortschreibung *der Ada Validation Procedures* mit ARA-Focus erarbeitet. Ganz offenbar sind die Ada-Hersteller in den USA aber nicht bereit, für die zu erwartenden Kosten aufzukommen. Auch kann von der ARA nicht eine solche Distanz zu den Herstellern erwartet werden, wie diese durch das DoD gegeben war.

Nun blickt man aus den USA mit gewissen Erwartungen nach Europa. Da in mehreren europäischen Ländern und in der NATO Ada weiterhin Standard ist, argumentiert man dort mit einigem Recht, daß Europa/NATO sich dann auch mehr um die Validierungs-Infrastruktur kümmern sollten. Schließlich habe man fast 15 Jahre lang in den USA das Validierungssystem aufgebaut und die Kosten dafür aufgebracht.

Die US-Validierungsstelle bei EDS mit Phil Brashear wird vermutlich auch weiterhin für Validierungen zur Verfügung stehen. Es wird jedoch ab dem Zeitpunkt der Schließung des AJPO in den USA kein von einer Regierungsstelle autorisiertes Validierungszertifikat mehr geben.

Bis Juni 1998 haben sich weitere Punkte in der Diskussion um die Zukunft der Ada Validierung herauskristallisiert:

- Die ARA wird zwar Anlaufstelle für Ada Validierungen sein. Es wird jedoch eine unabhängige Autorität geben (AVA; *Ada Validation Authority*), die nur ISO-Gremien (ARG, WG9) verpflichtet ist. Die Ada Hersteller werden somit keinen direkten Einfluß auf die Regeln der Ada Validierung haben. Die AVA fungiert somit als Nachfolgerin der Ada Validation Authority.
- Für die Finanzierung der AVA sind Übergangsregelungen unter Einbeziehung des US-DoD im Gespräch.
- Die ACVC Testsuite wird bis auf Substanzerhaltung in absehbarer Zeit nicht weiterentwickelt.

Egal, welche Konstellation sich nach der Schließung des Ada Joint Program Office abzeichnet: Am Ende dieses Prozesses wird eine Ada Validierungsstelle eine größere Kompetenz haben und autonom bei der Ausstellung von Validierungszertifikaten sein.

Was Deutschland angeht, so ist man hier ohne Organisationsänderungen in der Lage, den Betrieb der Validierungsstelle bei der IABG fortzusetzen und vom DIN bestätigte Validierungszertifikate für ACVC 2.1 auszustellen. Darüber hinausgehende europäische Initiativen zur Unterstützung der Ada Validierung sind derzeit noch nicht bekannt.

5 Literatur

Zur Beachtung: Die URLs der Dokumente [1] und [3] stammen vom Februar 1998 und können sich im Laufe der Zeit ändern.

[1] Ada Compiler Validation Procedures, Version 5.0, 18 November 1997, Ada Joint Program Office , Center for Computer Systems Engineering, Defense Information Systems Agency (*http://www.sw-eng.falls-church.va.us /AdaIC/compilers/val-proc/val5_0fin.shtml*).

[2] Reference Manual for the Ada Programming Language, ANSI/ISO/IEC 8652:1995, FIPS PUB 119-1.

[3] The Ada Compiler Validation Capability (ACVC) Version 2.1 User's Guide, CTA Incorporated, Dayton, OH 45431, March 1997 (*http://www.sw-eng.falls-church.va.us/AdaIC/compilers/acvc/95acvc/acvc2_1/index*).

[4] An Efficient Compiler Validation Method for Ada 9X, Michael Tonndorf, Proceedings of the Ada Europe '93 Conference, Springer Verlag LNCS 688, Berlin Heidelberg, 1993.

[5] An Integrated Tool Environment for Ada Compiler Validations, Michael Tonndorf, Proceedings of the TRI-Ada '93 Conference, ACM SIGAda, New York, 1993.

[6] Ten Years of Tool Based Ada Compiler Validations: An Experience Report. Michael Tonndorf, Proceedings of the Ada Europe '98 Conference on Reliable Software Technologies, Springer Verlag LNCS 1411, Berlin Heidelberg, 1998.