Do-While Jones began his career in analog circuit design but switched to
digital circuit design when he discovered digital circuits were easier to
design. He switched to software design when he discovered programming
computers was easier than building them, then transferred to an organiza­tion
that was planning a large software project because thinking about
programming is easier than programming. Now he lectures on Ada because
talking is even easier than thinking. Jones received a B.S. in electrical
engineering from a midwestern university better known for its football team
than its engineering school and since graduation has been employed in the
defense industry of a well-known free-world nation.

This is the continuing story of the efforts to
write a revised specification for the Ada
programming language. Publication deadlines
forced this paper to be written just before the
July 1991 Ada 9X Distinguished Reviewers meeting.
The oral presentation will include additional up­
to-the-minute information.

BACKGROUND AND HISTORY

The Ada programming language is defined by ANSI/MIL­
STD-1815A, approved 22 January 1983. Every ANSI standard
must be reviewed after five years. Then a decision must be
made either to abandon the standard, renew it without any
changes for another five years, or begin a revision process.
The Ada Board decided that "some omissions, limitations, and
minor errors have been discovered . . . since the ANSI
standard was approved. Although a major revision does not
appear to be necessary to correct these problems, some
revision is warranted."

The Ada Joint Program Office (AJPO) established the Ada
9X Project Office in October, 1988, to revise the Ada
language standard. The designator "9X" comes from the habit
of referring to the Ada standard approved in 1983 as "Ada
83," and the expectation that the revision will be approved
in '199X," where X is unknown. (The current schedule is X =
3.)

During its first year, the Ada 9X project encouraged
the Ada community to submit revision requests. There were
774 responses to this invitation. These revision requests
were combined with 850 Ada language issues and questions
that had previously been identified in the period of time
Do-While Jones began his career in analog circuit design but switched to digital circuit design when he discovered digital circuits were easier to design. He switched to software design when he discovered programming computers was easier than building them, then transferred to an organization that was planning a large software project because thinking about programming is easier than programming. Now he lectures on Ada because talking is even easier than thinking. Jones received a B.S. in electrical engineering from a midwestern university better known for its football team than its engineering school and since graduation has been employed in the defense industry of a well-known free-world nation.

This is the continuing story of the efforts to write a revised specification for the Ada programming language. Publication deadlines forced this paper to be written just before the July 1991 Ada 9X Distinguished Reviewers meeting. The oral presentation will include additional up-to-the-minute information.

BACKGROUND AND HISTORY

The Ada programming language is defined by ANSI/MIL-STD-1815A, approved 22 January 1983. Every ANSI standard must be reviewed after five years. Then a decision must be made either to abandon the standard, renew it without any changes for another five years, or begin a revision process. The Ada Board decided that "some omissions, limitations, and minor errors have been discovered... since the ANSI standard was approved. Although a major revision does not appear to be necessary to correct these problems, some revision is warranted."

The Ada Joint Program Office (AJPO) established the Ada 9X Project Office in October, 1988, to revise the Ada language standard. The designator "9X" comes from the habit of referring to the Ada standard approved in 1983 as "Ada 83," and the expectation that the revision will be approved in '199X,' where X is unknown. (The current schedule is X = 3.)

During its first year, the Ada 9X project encouraged the Ada community to submit revision requests. There were 774 responses to this invitation. These revision requests were combined with 850 Ada language issues and questions that had previously been identified in the period of time
from 1983 through 1987. (These questions came from compiler vendors who weren't sure exactly what the specification meant in certain places, and users who were surprised at the ways compiler vendors had implemented the language.) Furthermore, some DOD projects had requested waivers permitting them to use programming languages other than Ada. These waiver requests were examined to see if there were any technical deficiencies in Ada that made it unsatisfactory for military use. Finally, there were two Requirements Workshops and five special interest meetings which gave invited groups of Ada experts an opportunity to brainstorm about selected issues.

The primary activity of the Ada 9X project from October 89 through October 90 was to process all this data and identify all the possible revision requirements. Then they had to separate genuine requirements from misguided or unimportant suggestions.

During the requirements analysis, several difficult questions surfaced. These complex issues required special, concentrated study. The Software Engineering Institute (SEI) assisted the Ada 9X project by coordinating the studies on these issues and publishing the results. The complex issues are: Exceptions and Optimization; Shared Variables; Fixed Point; Implementation-Dependent Pragmas and Attributes; Input/Output; Reuse; Extended Character Set; Real-Time Issues; and a standard Math Library.

Following the precedent set by the American legal system ("Better that 99 guilty people should go free than 1 innocent person be unjustly punished"), the Ada 9X project took the attitude, "Better that 99 stupid ideas be cataloged and considered than that 1 good idea be overlooked." By the end of 1990 the hundreds of suggested revisions had been consolidated into a slightly smaller number of Revision Issues. (The numbering scheme makes it difficult to say exactly how many revision issues there were.) The revision issues were presented at a Requirements Workshop in Soderfors, Sweden in the spring of 1990.

It was never the intention of the Ada 9X Project Office that all these revision issues become revisions to the language. Only a few of the revision issues had sufficient merit to be worthy of consideration, but the Ada 9X Project Office wanted the Ada community to see that every suggestion was being processed, and that none were rejected without due process.

That approach may have backfired, because the participants at the Soderfors workshop were overwhelmed by the number of revision issues. The September/October 1990 issue of Ada Letters published reports written by some of the attendees of that meeting. John McHugh began his report by saying, "I came away from the workshop with a growing sense of depression." Geoff Mendal's first sentence was, "Unfortunately, I do not have much good news to report from the Sweden workshop." Other correspondents weren't quite so blunt so early in their reports, but there was general agreement that a few ideas were diamonds, but most were just stones. There was fear that Ada 9X would incorporate many of these bad ideas, and wouldn't bear much resemblance to Ada 83.

By the time that issue of Ada Letters was published, it was old news. The Ada 9X Project had already eliminated all but a few revision issues. The remaining revision issues were contained in a short Requirements Document that was presented at the December 1990 Tri-Ada meeting in Baltimore.

The Requirements Document divides the revision issues into Requirements and Study Topics. Requirements are really the design goals for Ada 9X. Study topics describe issues where "it is unclear whether the benefits of making a change are worth the potential costs."

One might assume that all the requirements and none of the study topics will be included in the language revision, but it isn't that simple. The proposed changes will be evaluated by User/Implementer Teams before the revised standard goes out for review. If the implementers discover that a new feature is too costly to implement, or if the users discover the feature is awkward or counter-productive, then it will be discarded. If a by-product of a required revision makes a study topic easy to implement, then it will be implemented.

A BRIEF SUMMARY OF THE REQUIREMENTS DOCUMENT

The Requirements Document begins with some general requirements. These will remove some minor annoyances. For example, the for j in -1..10 loop shortcut will become legal. The distinction between a basic declarative item and later declarative item will be dropped. The sentences that had to be clarified by approved Ada Commentaries will be rewritten to avoid the confusion. Changes will be made to improve efficiency, simplicity, consistency, and error detection.
from 1983 through 1987. (These questions came from compiler vendors who weren't sure exactly what the specification meant in certain places, and users who were surprised at the ways compiler vendors had implemented the language.) Furthermore, some DOD projects had requested waivers permitting them to use programming languages other than Ada. These waiver requests were examined to see if there were any technical deficiencies in Ada that made its unsatisfactory for military use. Finally, there were two Requirements Workshops and five special interest meetings which gave invited groups of Ada experts an opportunity to brainstorm about selected issues.

The primary activity of the Ada 9X project from October 89 through October 90 was to process all this data and identify all the possible revision requirements. Then they had to separate genuine requirements from misguided or unimportant suggestions.

During the requirements analysis, several difficult questions surfaced. These complex issues required special, concentrated study. The Software Engineering Institute (SEI) assisted the Ada 9X project by coordinating the studies on these issues and publishing the results. The complex issues are: Exceptions and Optimization; Shared Variables; Fixed Point; Implementation-Dependent Pragmas and Attributes; Input/Output; Reuse; Extended Character Set; Real-Time Issues; and a standard Math Library.

Following the precedent set by the American legal system ("Better that 99 guilty people should go free than 1 innocent person be unjustly punished"), the Ada 9X project took the attitude, "Better that 99 stupid ideas be cataloged and considered than that 1 good idea be overlooked." By spring of 1990 the hundreds of suggested revisions had been consolidated into a slightly smaller number of Revision Issues. (The numbering scheme makes it difficult to say exactly how many revision issues there were.) The revision issues were presented at a Requirements Workshop in Soderfors, Sweden in the spring of 1990.

It was never the intention of the Ada 9X Project Office that all these revisions become revisions to the language. Only a few of the revision issues had sufficient merit to be worthy of consideration, but the Ada 9X Project Office wanted the Ada community to see that every suggestion was being processed, and that none were rejected without due process.

That approach may have backfired, because the participants at the Soderfors workshop were overwhelmed by the number of revision issues. The September/October 1990 issue of Ada Letters published reports written by some of the attendees of that meeting. John McHugh began his report by saying, "I came away from the workshop with a growing sense of depression." Geoff Mendal's first sentence was, "Unfortunately, I do not have much good news to report from the Sweden workshop." Other correspondents weren't quite so blunt so early in their reports, but there was general agreement that a few ideas were diamonds, but most were just stones. There was fear that Ada 9X would incorporate many of these bad ideas, and wouldn't bear much resemblance to Ada 83.

By the time that issue of Ada Letters was published, it was old news. The Ada 9X Project had already eliminated all but a few revision issues. The remaining revision issues were contained in a short Requirements Document that was presented at the December 1990 Tri-Ada meeting in Baltimore.

The Requirements Document divides the revision issues into Requirements and Study Topics. Requirements are really the design goals for Ada 9X. Study topics describe issues where "it is unclear whether the benefits of making a change are worth the potential costs."

One might assume that all the requirements and none of the study topics will be included in the language revision, but it isn't that simple. The proposed changes will be evaluated by User/Implementer Teams before the revised standard goes out for review. If the implementers discover that a new feature is too costly to implement, or if the users discover the feature is awkward or counter-productive, then that requirement will be discarded. If a by-product of a required revision makes a study topic easy to implement, then it will be implemented.

A BRIEF SUMMARY OF THE REQUIREMENTS DOCUMENT

The Requirements Document begins with some general requirements. These will remove some minor annoyances. For example, the for j in -1..10 loop shortcut will become legal. The distinction between a basic declarative item and later declarative item will be dropped. The sentences that had to be clarified by approved Ada Commentaries will be rewritten to avoid the confusion. Changes will be made to improve efficiency, simplicity, consistency, and error detection.
One of the most significant changes will be the addition of some optional annexes. There are some things that are undefined in Ada 83 and are currently implemented in vendor-specific ways. The new annexes will specify the manner in which certain optional things must be done, if the vendor chooses to implement them. This will improve portability by limiting the implementation-dependent differences.

The popularity of Ada outside the United States has created some requirements for extending Ada's character set to include foreign characters. There are requirements for use of extended characters in strings and comments, and I/O capability that supports strings of extended characters. The use of extended characters in identifiers is not a requirement, but is a study topic.

There is a set of requirements and study topics that support various programming paradigms. These requirements deal with control of storage leakage and minimizing recompilation costs. Enhanced library support, the use of subprograms as objects, inheritance, and polymorphism are study topics.

System programming needs have lead to some requirements for unsigned integers and pointers, and improved interrupt handling. There is also a study topic dealing with the layout of data to improve data interoperability.

Information systems need decimal-based data types and must interface with data bases and COBOL data files containing EBCDIC characters. These needs are addressed by requirements and study topics that will probably wind up in an Information Systems annex to the specification. (That is, if the target computer has special instructions that operate on packed decimal data, the annex will specify how packed decimal arithmetic must be done in Ada 9X. Compilers targeted to small embedded computers that don't have special packed decimal instructions and aren't typically used in information systems, won't have to implement this annex.)

A set of real-time requirements address issues of time measurement, task scheduling, pre-elaboration, asynchronous transfer of control, and asynchronous communication. Since these issues are only important to a particular class of users, these requirements will probably be in a real-time annex.

Parallel and Distributed Processing, and Safety-Critical and Trusted Applications have some specialized needs. There are several requirements and study topics addressing these areas that aren't easily explained in a few words. These special features will probably be in annexes.

Finally, there are some study topics related to the finer points of scientific and mathematical applications.

THE GREAT SCOPE DEBATE

The magnitude of changes to Ada has been a hot topic. The extremely conservative position is that Ada is 98% perfect already. Only a few minor annoying features need to be cleaned up slightly. Major changes would cause too much disruption to the Ada community. It is not acceptable if existing Ada programs have to be extensively modified to compile and run correctly under Ada 9X. Programmer retraining should be kept to a minimum. Major changes may result in penalties in compile-time and run-time performance.

The extremely liberal position is that Ada must change with the times or else she will soon become obsolete. People who aren't forced to use Ada will use C++, so the number of Ada users will be small. It is important to have a large user base so competition will encourage better tools, better performance, and lower cost. Powerful new features will make it easier for programmers to solve difficult problems. Order-of-magnitude improvements can't be made by sticking bandages here and there-- major surgery is required.

The most visible participants in this debate are the Mapping/Revision Team (MRT) and the Distinguished Reviewers (DRs). The MRT has a contract to map the needs expressed in the requirements document to specific language features. The DRs are volunteer Ada experts selected by the Ada 9X project to review the MRT's proposals. There is a lively interchange on the network (about 20 e-mail messages a day), and face-to-face meetings about four times a year.

The MRT tends toward the liberal position. The DRs tend to be conservative. The MRT proposed many changes that the DRs felt went beyond the requirements, would be too disruptive to the Ada community, and could not possibly be implemented in the time allowed by the schedule.

The MRT and DRs got into a continuing scope-of-changes debate. It was suggested that the DRs were biased and did
One of the most significant changes will be the addition of some optional annexes. There are some things that are undefined in Ada 83 and are currently implemented in vendor-specific ways. The new annexes will specify the manner in which certain optional things must be done, if the vendor chooses to implement them. This will improve portability by limiting the implementation-dependent differences.

The popularity of Ada outside the United States has created some requirements for extending Ada's character set to include foreign characters. There are requirements for use of extended characters in strings and comments, and I/O capability that supports strings of extended characters. The use of extended characters in identifiers is not a requirement, but is a study topic.

There is a set of requirements and study topics that support various programming paradigms. These requirements deal with control of storage leakage and minimizing reccompilation costs. Enhanced library support, the use of subprograms as objects, inheritance, and polymorphism are study topics.

System programming needs have lead to some requirements for unsigned integers and pointers, and improved interrupt handling. There is also a study topic dealing with the layout of data to improve data interoperability.

Information systems need decimal-based data types and must interface with data bases and COBOL data files containing EBCDIC characters. These needs are addressed by requirements and study topics that will probably wind up in an Information Systems annex to the specification. (That is, if the target computer has special instructions that operate on packed decimal data, the annex will specify how packed decimal arithmetic must be done in Ada 9X. Compilers targeted to small embedded computers that don't have special packed decimal instructions and aren't typically used in information systems, won't have to implement this annex.)

A set of real-time requirements address issues of time measurement, task scheduling, pre-elaboration, asynchronous transfer of control, and asynchronous communication. Since these issues are only important to a particular class of users, these requirements will probably be in a real-time annex.

Parallel and Distributed Processing, and Safety-Critical and Trusted Applications have some specialized needs. There are several requirements and study topics addressing these areas that aren't easily explained in a few words. These special features will probably be in annexes.

Finally, there are some study topics related to the finer points of scientific and mathematical applications.

**THE GREAT SCOPE DEBATE**

The magnitude of changes to Ada has been a hot topic. The extremely conservative position is that Ada is 98% perfect already. Only a few minor annoying features need to be cleaned up slightly. Major changes would cause too much disruption to the Ada community. It is not acceptable if existing Ada programs have to be extensively modified to compile and run correctly under Ada 9X. Programmer retraining should be kept to a minimum. Major changes may result in penalties in compile-time and run-time performance.

The extremely liberal position is that Ada must change with the times or else she will soon become obsolete. People who aren't forced to use Ada will use C++, so the number of Ada users will be small. It is important to have a large user base so competition will encourage better tools, better performance, and lower cost. Powerful new features will make it easier for programmers to solve difficult problems. Order-of-magnitude improvements can't be made by sticking bandages here and there-- major surgery is required.

The most visible participants in this debate are the Mapping/Revision Team (MRT) and the Distinguished Reviewers (DRs). The MRT has a contract to map the needs expressed in the requirements document to specific language features. The DRs are volunteer Ada experts selected by the Ada 9X project to review the MRT's proposals. There is a lively interchange on the network (about 20 e-mail messages a day), and face-to-face meetings about four times a year.

The MRT tends toward the liberal position. The DRs tend to be conservative. The MRT proposed many changes that the DRs felt went beyond the requirements, would be too disruptive to the Ada community, and could not possibly be implemented in the time allowed by the schedule.

The MRT and DRs got into a continuing scope-of-changes debate. It was suggested that the DRs were biased and did
not represent the feelings of the Ada community. So, an Ada 9X workshop was held in Orlando, FL, in March 1991. Some "outsiders" were invited to attend and hear the MRT's proposed changes. These outsiders were proficient Ada professionals who had not previously been involved in the Ada 9X project. The DRs were careful not to prejudice the audience by criticizing the MRT proposals. The outsiders' reactions were similar to the reaction at the Swedish workshop the previous year. They were alarmed and disturbed by the magnitude of the changes.

The last day of the Orlando workshop was the Ides of March, so the DRs seized the opportunity to demand some brutal cuts. The MRT retreated and worked on scaling back the changes. They presented their less ambitious proposal at a meeting with the DRs in May. The DRs felt that the reductions were a step in the right direction, but still weren't satisfied. Further reductions were requested for the DR meeting in July, 1991. (That meeting had not yet taken place when this paper was written.)

THE TWO BIGGIES

The two big, controversial proposals are protected records and object-oriented features.

The motivation behind protected records is a desire for lightweight asynchronous communication and immediate asynchronous transfer of control. Real-time programmers don't want to incur the cost of a full task switch every time they read or write a shared variable. They want fast response to interrupts and mode switches. Protected records are proposed as a cheap alternative to full Ada tasking. There is general agreement that this is a good thing to have. There is not agreement as to whether the proposed solution is technically sound and will satisfy the need. I can't tell you exactly what the proposed solution is because it changes slightly from time to time in response to criticism. It is a technical problem whose solution has not been firmly decided.

The object-oriented debate is murkier. Both sides admit that there is no real need for object-oriented extensions. (The Requirements Document classified inheritance and polymorphism as Study Topics, not Requirements.) Both sides also admit there is a real user perception that object-oriented extensions are needed. The questions are, "Will Ada 9X be accepted by users if she doesn't have unnecessary features that users think they need?" and "How much will it cost to put these unnecessary (but desired) features in?"

The difficulty with putting object-oriented features into Ada comes from a basic incompatibility. Ada is a strongly-typed language with heavy emphasis on compile-time checking. Object-oriented features have to be loosely-typed. (Operations must work for a class of similar types, not just one specific type.) Object-oriented features tend to depend on run-time dispatching that can't be checked at compile time. Introduction of tagged types and classes into Ada causes some significant changes. The conservative view is that we can't afford to put these changes in. The liberal view is that we can't afford not to put them in.

THE OTHER PLAYERS

It may appear that the only people involved in the Ada 9X project are the Mapping/Revision Team (from Intermetrics) and the 30 Distinguished reviewers (from various places in North America and Europe). That's because they are making the most noise and attracting the most attention. There are other people behind the scenes who are making major contributions.

There are three User/Implementer Teams who are following the work of the MRT closely. The Telesoft and Unisys team is making experimental modifications to their 680X0 target compiler and testing the proposed changes with special emphasis on C3 application programs. Tartan and TRW are modifying an 80960 compiler to see the effects on Avionics and Space programs. The Aetech/RR/AdaSoft team is doing similar things to their 80386 compiler and testing the effects on Information Systems programs. These three teams provide valuable input based on real experiences with the proposed changes.

At NYU, the Implementation Analysis Team is scrutinizing each proposal to see how difficult it would be to implement, and how performance would be affected. The Language Precision Team at Odyssey Research Associates is watching the MRT proposals with a watchful eye toward validation and safety-critical considerations.
not represent the feelings of the Ada community. So, an Ada 9X workshop was held in Orlando, FL, in March 1991. Some "outsiders" were invited to attend and hear the MRT's proposed changes. These outsiders were proficient Ada professionals who had not previously been involved in the Ada 9X project. The DRs were careful not to prejudice the audience by criticizing the MRT proposals. The outsiders' reactions were similar to the reaction at the Sweden workshop the previous year. They were alarmed and disturbed by the magnitude of the changes.

The last day of the Orlando workshop was the Ides of March, so the DRs seized the opportunity to demand some brutal cuts. The MRT retreated and worked on scaling back the changes. They presented their less ambitious proposal at a meeting with the DRs in May. The DRs felt that the reductions were a step in the right direction, but still weren't satisfied. Further reductions were requested for the DR meeting in July, 1991. (That meeting had not yet taken place when this paper was written.)

THE TWO BIGGIES

The two big, controversial proposals are protected records and object-oriented features.

The motivation behind protected records is a desire for lightweight asynchronous communication and immediate asynchronous transfer of control. Real-time programmers don't want to incur the cost of a full task switch every time they read or write a shared variable. They want fast response to interrupts and mode switches. Protected records are proposed as a cheap alternative to full Ada tasking. There is general agreement that this is a good thing to have. There is not agreement as to whether the proposed solution is technically sound and will satisfy the need. I can't tell you exactly what the proposed solution is because it changes slightly from time to time in response to criticism. It is a technical problem whose solution has not been firmly decided.

The object-oriented debate is murkier. Both sides admit that there is no real need for object-oriented extensions. (The Requirements Document classified inheritance and polymorphism as Study Topics, not Requirements.) Both sides also admit there is a real user perception that object-oriented extensions are needed. The questions are, "Will Ada 9X be accepted by users if she doesn't have unnecessary features that users think they need?" and "How much will it cost to put these unnecessary (but desired) features in?"

The difficulty with putting object-oriented features into Ada comes from a basic incompatibility. Ada is a strongly-typed language with heavy emphasis on compile-time checking. Object-oriented features have to be loosely-typed. (Operations must work for a class of similar types, not just one specific type.) Object-oriented features tend to depend on run-time dispatching that can't be checked at compile time. Introduction of tagged types and classes into Ada causes some significant changes. The conservative view is that we can't afford to put these changes in. The liberal view is that we can't afford not to put them in.

THE OTHER PLAYERS

It may appear that the only people involved in the Ada 9X project are the Mapping/Revision Team (from Intermetrics) and the 30 Distinguished reviewers (from various places in North America and Europe). That's because they are making the most noise and attracting the most attention. There are other people behind the scenes who are making major contributions.

There are three User/Implementer Teams who are following the work of the MRT closely. The Telesoft and Unisys team is making experimental modifications to their 680X0 target compiler and testing the proposed changes with special emphasis on C3 application programs. Tartan and TRW are modifying an 80960 compiler to see the effects on Avionics and Space programs. The Aetech/RR/AdaSoft team is doing similar things to their 80386 compiler and testing the effects on Information Systems programs. These three teams provide valuable input based on real experiences with the proposed changes.

At NYU, the Implementation Analysis Team is scrutinizing each proposal to see how difficult it would be to implement, and how performance would be affected. The Language Precision Team at Odyssey Research Associates is watching the MRT proposals with a watchful eye toward validation and safety-critical considerations.
Of course, all this work is being directed by the Ada 9X Project office. Christine Anderson (the Ada 9X Project Manager) has the ultimate responsibility for the final draft of the revised standard.

FINAL APPROVAL

Ultimately, though, the really important players in the Ada 9X process are the people on the ANSI and ISO canvass lists. It doesn't matter what the MRT proposes and DRs like or don't like if the ANSI/ISO canvass process fails. The ANSI and ISO representatives won't pick and choose from the new features offered. Their vote will either be YES or NO for the whole standard. If there isn't an overwhelming YES consensus from the ANSI/ISO canvas lists, then the Ada 9X project will suffer a serious blow. They won't have to go all the way back to Square One, but close to it.

That's one of the reasons why the Ada 9X project has been taking every opportunity to tell the world what is going on. If they are making any serious mistakes, they want to know now, before it is too late. The Ada 9X project maintains several electronic bulletin boards. In the United States, call 1-800-ADA9X25 or use the ARPANET to connect to the AJPO host machine and look in the ADA9X directory. In Canada, call 1-301-459-8939. In Europe, use EUROKOM. The telephone number in Dublin is 353-1-697890 or HELP_DESK@EUROKOM.IE.

If you prefer to talk to a real person, call the Ada Information Clearinghouse at (703) 685-1477. They can tell you how to obtain all the Ada 9X documents released to the public. This includes the SEI studies and the Requirements Document.

Chris Anderson, the Ada 9X Project Manager, occasionally visits her office at Eglin AFB. If you are lucky, you can reach her at (904) 882-8264. If she isn't there, her assistant Fran Jenkins will either answer your question or leave a message for Chris. If you are on the ARPANET, you can send e-mail to ANDERSON@UV4.EGLIN.AF.MIL.
Of course, all this work is being directed by the Ada 9X Project office. Christine Anderson (the Ada 9X Project Manager) has the ultimate responsibility for the final draft of the revised standard.

FINAL APPROVAL

Ultimately, though, the really important players in the Ada 9X process are the people on the ANSI and ISO canvass lists. It doesn't matter what the MRT proposes and DRs like or don't like if the ANSI/ISO canvass process fails. The ANSI and ISO representatives won't pick and choose from the new features offered. Their vote will either be YES or NO for the whole standard. If there isn't an overwhelming YES consensus from the ANSI/ISO canvas lists, then the Ada 9X project will suffer a serious blow. They won't have to go all the way back to Square One, but close to it.

That's one of the reasons why the Ada 9X project has been taking every opportunity to tell the world what is going on. If they are making any serious mistakes, they want to know now, before it is too late. The Ada 9X project maintains several electronic bulletin boards. In the United States, call 1-800-ADA9X25 or use the ARPANET to connect to the AJPO host machine and look in the ADA9X directory. In Canada, call 1-301-459-8939. In Europe, use EUROKOM. The telephone number in Dublin is 353-1-697890 or HELP_DESK@EUROKOM.IE.

If you prefer to talk to a real person, call the Ada Information Clearinghouse at (703) 685-1477. They can tell you how to obtain all the Ada 9X documents released to the public. This includes the SEI studies and the Requirements Document.

Chris Anderson, the Ada 9X Project Manager, occasionally visits her office at Eglin AFB. If you are lucky, you can reach her at (904) 882-8264. If she isn't there, her assistant Fran Jenkins will either answer your question or leave a message for Chris. If you are on the ARPANET, you can send e-mail to ANDERSON@UV4.EGLIN.AF.MIL.

SCHEDULE

The Ada 9X project is shooting for a draft standard in the summer of 1992. This draft standard will be distributed to the ANSI canvass list then, and ANSI approval could be as early as March 1993. It is a tight schedule, but the Ada 9X project is trying to hold to it.