Panel: Ada 9X Implementation

Chair

Joyce L. Tokar
Tartan, Inc.

Panelists

Randall Brukardt, *RR Software, Inc.*
Jean-Claude Heliard, *Alsys, Inc.*
Edmond Schonberg, *New York University*
Bevin Brett, *Digital Equipment Corporation*
Gary J. Dismukes, *TeleSoft*
Stephen Zeigler, *Verdix Corporation*
Joyce L. Tokar, *Tartan, Inc.*

Panel: The Implementation of Ada 9X

Moderator
Joyce L. Tokar, Tartan, Inc.

Panelists

Bevin Brett, Digital Equipment Corp.
Randall Brukardt, RR Software
Gary Dismukes, TeleSoft
Jean-Claude Heliard, Alsys, Inc.
Edmond Schonberg, New York University
Joyce L. Tokar, Tartan, Inc.
Stephen Zeigler, Verdix Corporation

The process of developing a standard for Ada 9X is nearing its completion. Once a 9X standard is approved it will be the responsibility of the Ada compiler vendors to bring this standard to the users via compilation systems. The objective of this panel session is to allow representatives from several compiler vendors to present their strategies for developing Ada 9X compilers. The vendors are encouraged to discuss the features of 9X that they believe will be implementable in the near future as well as those features that represent some difficulty.

The Ada 9X transition strategy has been designed to ease the impact of the transition on vendors. In particular, the ACVC test suite and process have been adjusted to support an incremental transition to Ada 9X. ACVC 2.0 will be modularly constructed so that the vendors may choose the set of tests that best satisfies their customer base. There will be a minimum number of modules that all vendors must pass. A vendor may also pass modules for specific functional areas of 9X which address their customers' needs. ACVC test suite 2.1 will be more robust, requiring vendors to pass all core language tests. This modular approach to the validation of Ada 9X compilation systems will allow a vendor to focus their preliminary systems on those areas of 9X which are most critical to their customers leaving the remaining features for a later release.

In addition to the discussion of vendors' plans for making the transition to Ada 9X, some of the presentations may focus on the implementation experiences of the vendors that participated in the 9X User/Implementer Demonstration Project. New York University has been asked to participate in this panel because of their experiences with implementing features of Ada 9X as part of the Ada 9X Language Analysis Team.

The primary objective of this panel session is to provide the audience with the assurance that the vendors understand the issues that they must address to transition to Ada 9X and that this understanding will decrease the time to delivery of mature 9X compilation systems.

Position Paper for Ada 9X Implementation Panel

Mr. Randall Brukardt
RR Software, Inc.

Ada 9X will have significant costs to the vendors of Ada compilers. That is undoubted. Our experience with implementing some of the Ada 9X features for the AETECH/RR User/Implementer team shows that even the easiest features to implement can affect major portions of a compiler. Our current estimate is that 6-9 person-years will be required to update an Ada 83 compiler to Ada 9X.

However, the implementation effort could have been much worse. We have been happy with the responsiveness of the Ada 9X Mapping/Revision Team to eliminating features that are hard to implement without a corresponding user benefit. Many dubious ideas and expensive constructs have been scrapped.

Ada 9X also simplifies Ada for both users and implementers. For instance, subprogram conformance is now much more logical and much closer to the idea of semantic equivalence. Subunit names now need only have a unique fully expanded name, which is closer to user expectations. Changes like these will allow existing Ada compilers to eliminate some junk code. The changes are much more useful to major redesigns and new implementations of Ada compilers, since they will not have to design around the old misfeatures as existing Ada compilers had to.

However, I think the cost to vendors is the wrong question to ask. The key question is whether Ada 9X can give users what they want, in order to keep Ada on the path of a growing and viable language. Certainly, making no changes at all to Ada at this point would result in its becoming a footnote in the history of computers. That is because Ada is losing its commercial customers to other languages, and Ada cannot survive as an important language without commercial customers. Ada 9X alone will not solve this problem, but it will eliminate many of the technical barriers to the adoption of Ada. (The problems of inertia and antipathy are every Ada user's problem to help solve). Once this question has been answered satisfactorily, only then can we worry significantly about implementation costs.

For example, many users have switched to C++, in part because of the supposed benefits of OOP. This has lead to many user requests for OOP in Ada. C++ has upped the ante further by adopting many of Ada's best features (exceptions and generics come to mind). Clearly, Ada needs to do the same by adopting OOP. It is important for vendors to not only consider the costs of implementing OOP, but also to consider the costs of NOT implementing OOP (in lost customers).

In summary, Ada is now nearing a crossroads. It can either go forward (by adopting new features like those in Ada 9X), or it can be pushed backwards into oblivion by other languages which meet user needs better.

Alsys Panel Position

Mr. Jean-Claude Heliard
Alsys, Inc.

Alsys is completely committed to supporting Ada 9X functionality. Ada 9X will be implemented on all Alsys strategic products and the timing of implementation will be time phased, the market demand determining platforms and priorities.

Alsys Tools for Ada 9X will be available in the same general timeframe as the compilation systems.

The availability of Ada 9X functionality in Alsys products will be as soon as possible after adoption of the standard, i.e., a few month after standardization. In order to achieve this goal, implementation will start in advance as stabilization of the proposed standard progresses.

As ACVC 2.0 permits the staged implementation of Ada 9X functionality, market demand will determine which domains and annexes of the standard Alsys will support first.

Our current position is to address in priority the Object Oriented features, the Generic Units extensions, the Real Time features and the Internationalization features (multiple character sets).

Among the annexes, we plan to focus primarily on the implementation of the Real Time annex, the System Programming annex and the Safety and Security annex as well as on the Interfacing capabilities.

Although in the long term we will support the full language, the previous policy based on market needs is subject to change depending on market evolution and technology trends.

Ada 9X Implementation Panel
Position Statement

Dr. Edmond Schonberg
New York University

Ada83 proved to be a difficult language to implement, because it pushed the state of the art in compilation techniques in several domains: tasking, generic units, and semantic checks in general. As a result, good implementations took years to appear, potential users were frustrated and eventually discouraged, and the language acquired a reputation for buggy implementations. Now Ada 9X is a larger language than Ada83; among major extensions, it includes protected types for data synchronization, and type extensions for object-oriented programming. It is vital to the success of Ada 9X that the painful implementation experience of a decade ago not be repeated. There are several reasons for optimism:

a) Compiler technology has matured substantially, and current Ada implementers have superb experience in the design of run-time environments, code generators, and semantic analyzers.

b) The major new features of Ada 9X have well-understood models of implementation in other languages. For example, there is ample experience with the implementation of dynamic dispatching in object-oriented languages.

c) Several implementer teams have participated in the design and revision phases of the Ada 9X process, and have prototyped the most important new features of the language. They have uncovered implementation difficulties in early phases of the design, and their experiences have led to a number of simplifications in the language. All other implementers will benefit from the experience of the Ada 9X U/I teams.

Nevertheless, it would be a mistake to assume that full implementations will appear before 1994: the sheer number of details involved in a language of this size insures that more than a year will be needed to sort out the semantic interactions between seemingly independent features. In order to satisfy users that are eager to move to Ada 9X, it might be advantageous to release partial implementations before validation, to convince the Ada community that Ada 9X is real, and that its advantages over other contemporary languages are significant. Whether the Ada community, trained to expect validated compilers that conform exactly to the LRM, will be willing to accept temporarily the looser implementation standards of other languages, remains to be seen.

DIGITAL's Position, Summary

Mr. Bevin Brett
Digital Equipment Corp.

DIGITAL supports the general outline of the proposed Ada 9X language, and most of the details within that outline, as being a sensible and useful path forward for the Ada language. It sees the proposed enhancements as necessary to keep Ada a viable language, updating it to incorporate the lessons learned in language design both from the Ada '83 experience and also with the development of such techniques of object oriented programming.

Bevin will describe DIGITAL's past and current involvement and role in the Ada 9X process, both in terms of what it is and also what it is attempting to achieve.

While it is against DIGITAL policy to discuss possible future products, Bevin will also describe DIGITAL's Ada strategy and possible ways that Ada 9X will be incorporated in that strategy.

Lastly, Bevin will categorize some of the Ada 9X features, and describe some of the issues around their implementation within the DIGITAL Ada product.

Position Statement for the Ada 9X Implementers Panel

Mr. Gary Dismukes
TeleSoft

Since early 1991 TeleSoft has participated as one of the three Ada 9X User/Implementer Teams involved with prototyping and giving feedback on the proposed Ada 9X features. The U/I process has been useful in providing early feedback to the Mapping/Revision Team on aspects of the features that are hard to understand, inconsistently or incompletely specified, or difficult to implement. It was also highly beneficial to have the language changes prototyped in three different compilers because it has helped to provide a broader-based proof of concept that lends credence to implementability across a range of compiler architectures.

The principal goal of our implementation approach was to minimize and localize the impacts on the existing compiler and run-time system and to avoid redesign by building on existing compiler data structures and algorithms. In particular, we hoped to minimize or avoid effects on the target-dependent areas of our software since those are the most costly to maintain and adapt for new targets. In our Ada 9X implementation work to date this has been achieved.

Of the areas we worked on, the largest efforts were applied to the tagged type and protected object features. These are also two of the areas of greatest potential benefit to users. Although these features require significant investments of developer time to implement, we are encouraged by the degree to which we were able to extend and adapt our compiler's existing support for Ada 83 features such as derived, record, and task types.

Other areas proved to be relatively easy for us to implement. Support for the child library unit feature had a fairly localized impact on our Front End and did not require any retooling of our library management system. Of course, what is easy to support for one compiler may be difficult for another, which is why it was useful to have more than one Implementer Team prototyping the 9X features -- the fact that all three of the teams were able to implement child units with reasonable cost is a positive sign for other vendors.

Even though some features might have been implemented somewhat more cleanly by starting over from scratch, we feel that the general strategy we adopted is consistent with the goal for Ada 9X to be a revision, not a language redesign. Also, the degree to which we have succeeded in extending our existing technology without having to rework fundamental structures of the design is probably a good indicator of the likely success that other vendors will have in transitioning to Ada 9X. Of course, the prototyping done by the U/I Teams can only be a partial measure of the expense of adapting to the revised language, and many other cost factors such as impacts on tools and education of programmers are yet to be assessed.

Position Paper: Ada 9X Implementation

Dr. Stephen Zeigler
Verdix Corporation

Verdix is monitoring the progress of Ada 9X. Several Verdix engineers have a vote on the resulting standard canvas. We are represented on ARTEWG, which has been quite active in reviewing the Ada 9X runtime issues. We try to keep up with the VR mailing list. We are active in NUMWG, POSIX and especially the Distributed Annex areas.

Our current plan is to postpone most internally funded changes for Ada 9X until nearer the initial release of the ACVC 2.0 test suite. The ACVC is the operational definition of the language and gives us a steady target to shoot at. The 9X schedule allows us plenty of time to work on 9X after the initial release of the test suite, before validations require Ada 9X. Our OEMs will be getting Ada 9X prototypes so that their customers as well as direct Verdix customers should expect no interruption in validated compiler availability from Verdix.

Verdix would be happy to discuss an earlier implementation of Ada 9X features with customers interested in funding our efforts. Some customers have already expressed their interest in early prototypes for specific features.

Ada 9X Implementation Panel
Tartan's Position Statement

Dr. Joyce L. Tokar
Tartan, Inc.

We endorse the revision of the Ada language to reflect both past experience with Ada and the progress in language design for building large software systems. Languages need to evolve over time to match users interests and needs.

Our principle market is in providing Ada for the development of real-time, embedded systems. This market will determine our approach to implementing the new features proposed in Ada 9X. We will listen to our customers' requests, then prioritize our implementation strategy based upon these requests. Our goal in developing an Ada 9X compilation system is to meet our customers' needs as soon as possible.

As part of the User/Implementer Project we have implemented the following: the protected type, the requeue operation, the new interrupt handler model, and the priority queuing pragma. The preliminary performance results for these features are within the required boundaries specified by our customers.

These 9X features were integrated into the existing runtime system with minimal impact on its data structures and algorithms. In particular, the protected type and requeue operation were incorporated into the runtime system as a separate subsystem. This subsystem interacts with the runtime system; primarily the tasking subsystem and the exceptions subsystem. Once the protected type prototype was complete, the new interrupt model was incorporated into the runtime system as a component of the execution environment with an interface to the protected type subsystem.

The Ada 9X Transition Plan for validation supports our approach to the development of our Ada 9X compilation system. We will be able to validate the initial version of our 9X system using the ACVC test suite 2.0. Then, as additional language features are integrated into our compilation system, we can incrementally validate the system using the additional ACVC tests available to validate specific areas of Ada 9X.

Our customers are concerned with real-time, systems programming and distribution. Hence, our implementation strategy will be to complete those facets of Ada 9X. We will then proceed with the remaining 9X proposals in an order that reflects our customers' needs.