Using Ada in Non-CS Majors

A.T. Chamillard
Department of Computer Science
U.S. Air Force Academy, CO
80840
719-333-7101
ChamillardAT.dfcs@usafa.af.mil

Ronald J. Lisowski
Department of Astronautics
U.S. Air Force Academy, CO
80840
719-333-4110
LisowskiRJ.dfas@usafa.af.mil

Richard R. Young
Department of Computer Science
U.S. Air Force Academy, CO
80840
719-333-7101
YoungRR.dfcs@usafa.af.mil

1. ABSTRACT

Ada has been the core language of the computer science curriculum at the U.S. Air Force Academy for several years. Recently, other majors have decided to use Ada as the primary language in their curricula. In this paper we discuss the rationale for making this transition to Ada and discuss the results of using Ada in majors other than computer science.

2. INTRODUCTION

All students attending the U.S. Air Force Academy (called USAFA hereafter) are required to take an introductory course in computer science. For the past two years, Ada 95 has been the programming language taught in that course [2]. In addition, Ada has been the "language of choice" in the computer science curriculum for several years; in some computer science courses, Ada has been used for 6 years. It appears to be fairly common for computer science professors to consider and adopt Ada as their desired programming language [3]. We have been unable to discover in the literature, however, indications that other academic disciplines are also selecting Ada for use in their curricula.

At USAFA, two academic majors have decided to use Ada in their curricula. The Department of Astronautics is using Ada in one of its core courses and one of its major's courses, with plans to incorporate Ada into two more major's courses. The Operations Research major, a major that is jointly administered by the Departments of Computer Science, Economics, Management, and Mathematical Sciences, has also decided to use Ada in its programming courses. Finally, the Department of Engineering Mechanics is considering use of Ada in its core course.

The primary motivation for using Ada in other majors is that Ada is becoming the de facto "standard" language at USAFA, since all students attending USAFA learn Ada in the introductory course in computer science. Instructors in other majors are thus assured that students have been exposed to programming in the same language being used in their courses. This previous exposure lets the instructors focus more on the actual algorithm development for the problem solutions and/or more advanced language features.

The next section discusses Ada use in the Department of Astronautics, and Section 4. presents the use of Ada in the Operations Research major. The final section provides our conclusions.

3. DEPARTMENT OF ASTRONAUTICS

The Department of Astronautics has been using Ada in two courses for the past year, with plans to transition to Ada in two other major's courses in the next several years. This section provides a historical perspective of the department's choice of programming language, discusses the two courses in which Ada is currently used, and presents future plans for other courses.

3.1 Historical Perspective

The Department of Astronautics has a long history of requiring students to program solutions to astrodynamics problems, extending back to the early 60's. From the early 60's to the late 70's, the department used ALGOL as its programming language. This was the language taught to freshmen entering USAFA and was also the language taught to Computer Science majors, so it was natural for the department to use ALGOL for its programming language as well.

ACM 1-58113-033-3/98/0011

In the late 70's, however, the department began receiving feedback from recent graduates that the most common language in its field was FORTRAN. To better prepare its students for the jobs they would hold after leaving USAFA, the Department of Astronautics decided to transition to FORTRAN 77 for its programming language. FORTRAN 77 was the programming language used by the department until Fall 1996.

Several developments in the late 80's and early 90's led the Department of Astronautics to consider transitioning to Ada as its programming language. The adoption of Ada 83 as the DOD programming language had some impact, and the Department of Computer Science started using Ada in some of its courses in 1992. In addition, support for the FORTRAN 77 environment used by the Department of Astronautics was becoming more and more difficult to obtain, and transitioning to FORTRAN 90 would have been a non-trivial effort for the department. The department makes extensive use of code libraries for its student projects, and the effort required to change these libraries to FORTRAN 90 would have been significant.

In 1993, the Departments of Astronautics and Computer Science discussed potential future use of Ada across the USAFA curriculum, and it was clear that Ada would become the language of choice for computer science courses. As a result, in 1994 the Department of Astronautics awarded a contract for the conversion of all its code libraries from FORTRAN 77 to Ada, with the intent to use Ada as the programming language for Astronautics courses. In Fall 1996, the Department of Computer Science transitioned to Ada in its core course taught to all students, further solidifying the use of Ada across USAFA.

Although one of the original motivations for the transition to FORTRAN 77 was to better prepare graduates for the language they were likely to see on the job after graduation, the Department of Astronautics decided that it was more important to use the standard language at USAFA. In addition, they felt that students would develop a more thorough understanding of high-level programming by using a single programming language throughout the curriculum at USAFA. They could then transfer this understanding to other high-level languages they encountered after graduation.

In the Fall semester of 1996 and the Spring semester of 1997, the Department of Astronautics taught its Introduction to Astronautics for the Engineer and Scientist course both in FORTRAN 77 and Ada; starting in the Fall semester of 1997, the course has been taught solely in Ada. Last semester, the department taught its Astrodynamics course in both FORTRAN 77 and Ada, and this course will be modified next semester to use Ada exclusively. Finally, the department plans to transition its Advanced Astrodynamics and Space Mission Design courses to Ada over the next two years.

3.2 Introduction to Astronautics for the Engineer and Scientist

Introduction to Astronautics for the Engineer and Scientist is a required 3 semester hour course at USAFA for all students majoring in science or engineering (Biology, Chemistry, Computer Science, Math Science, Physics, Aeronautical Engineering, Astronautical Engineering, Civil Engineering, Electrical Engineering, Engineering Science, Engineering Mechanics, Mechanical Engineering, and Space Operations). The course work includes two programming assignments (called Part I and Part II), for which the students read from a data file provided by the department, process the data appropriately, and output the processed data. Points are allocated for algorithms and programming style, but the bulk of the points are allocated to correct output.

All students enrolled in this course have either completed the Introduction to Computer Science course at USAFA or have validated that course through transfer credit or Advanced Placement Test performance. Students in the course have therefore had a minimum of one course in programming (or equivalent experience). The students using FORTRAN learned Pascal in the Introduction to Computer Science course, while the students using Ada learned Ada in that course.

For each of the programming assignments, students are provided with a list of the procedures they should write, the parameters for those procedures, and a description of the required functionality for each of the procedures. In addition, the department provides a library of functions and procedures that are available for students to use if they choose.

Providing the students with the decomposition into procedures certainly absolves the students from performing that decomposition on their own. While this may imply a focus on Ada programming rather than problem solving, we note that students are still required to generate detailed algorithms for each of these procedures and the main program before generating code. Students therefore still have a significant amount of design work to perform before implementing their Ada programs.

To quantify the results of the transition from FORTRAN 77 to Ada in this course, we can compare the grades on these programming assignments. The grade data is provided in Figure 1, where the FORTRAN 77 classes were those offered in the Fall 1996 and Spring 1997 semesters and the Ada classes were those offered in the

		Part I		Part II	
	Count	Mean	Standard Div	Mean	Standard Div
FORTRAN	242	88.3	18.6	88.5	20.4
Ada	315	89.6	18.3	87.9	21.9

Figure 1. Grade Data for FORTRAN 77 and Ada Programs in Introduction to Astronautics for the Engineer and Scientist

Fall 1997 and Spring 1998 semesters. Means and Standard Deviations are in percentages.

The Ada grades were higher than the FORTRAN grades on the first assignment, but slightly lower than the FORTRAN grades on the second assignment. The means are fairly close for both assignments, however, leading us to conclude that the students using Ada are generating programs of comparable quality to those generated by students using FORTRAN.

To more thoroughly compare student performance, we looked closely at the grading policies for these assignments. We found that because a large portion of the points are allocated to correct output, students who get this output incorrect the first time get a zero on that portion of the assignment, but can resubmit their programs (within a few days). If the output is correct in the resubmission, students can get half of the correct output points added back into their grade. We provide data about these cases in Figure 2. In the figure, the Number column indicates how many students had incorrect output on their first submission and the Percent column indicates what percentage of the course enrollment this number represents.

therefore be assumed to be caused by the choice of programming language.

The data indicates that, for the first programming assignment, a slightly larger percentage of the FORTRAN students generated incorrect output for their first submission. For the second programming assignment, however, slightly more of the Ada students (as a percentage) generated incorrect output with their first submission. Given the similarity of the numbers and the different results for the two assignments, we do not believe any general conclusions can be drawn from this data.

There are, of course, other factors that could affect the results of our comparisons. For example, the instructors for the courses were different across semesters, so instructors that graded particularly hard or easy could skew the results. The projects themselves did not change, so we can eliminate project changes as a factor, but even ease of use for each programming environment could affect the results.

Anecdotal evidence from instructors teaching the course indicates that the "extra help" sessions with the students using Ada seem to focus more on algorithm development, while the sessions with the students using FORTRAN tend

		Part I		Part II	
	Count	Number	Percent	Number	Percent
FORTRAN	242	34	14%	44	18%
Ada	315	37	13%	59	20%

Figure 2. Incorrect Answer Data for FORTRAN 77 and Ada Programs in Introduction to Astronautics for the Engineer and Scientist

We note that there are numerous potential causes of incorrect output. For example, incorrect output may be a result of erroneous algorithms, or it could be caused by incorrect implementation of correct algorithms. Although we have not explicitly separated these potential causes, we note that the algorithm development is language independent, so we would expect similar student performance in this area no matter what language is used to implement the algorithms. Barring other effects, differences in incorrect output in the first submission can

to focus more on language issues (Number, Order, Type errors in procedure calls, for example). Additionally, instructors prefer the AdaGIDE environment [1] to the FORTRAN environment used. Finally, instructors note that Ada allows easier use of abstract data types, so students using Ada are required to generate a package for an abstract data type (vectors).

Given the comparisons and instructor feedback discussed above, we view the transition to Ada in this course as a success. Students are generating programs of comparable

quality to those generated by students using FORTRAN, using an abstract data type, and are also using the "standard" programming language at USAFA.

3.3 Astrodynamics

Astrodynamics is a required 3 semester hour course for all students majoring in Astronautical Engineering or Space Operations; the course includes three programming assignments. As in Introduction to Astronautics for the Engineer and Scientist, there are points allocated to algorithms, programming style, and correct output, with the bulk of the points allocated to correct output.

For the first two programming assignments, students are provided with a list of the procedures they should write, the parameters for those procedures, and a description of the required functionality for each of the procedures. In addition, the department provides a library of functions and procedures that are available for students to use if they choose. For the final programming assignment, students are required to perform their own decomposition to identify the necessary procedures, though they still have access to the library of functions and procedures.

Two sections of the course were taught last semester, one in FORTRAN 77 and the other in Ada. We compared the grade data for these sections as we did for the Introduction to Astronautics for the Engineer and Scientist. The data is provided in Figure 3, with Means and Standard Deviations again provided as percentages.

assignment (that student had decided to retake the course; he received a 0 on the final examination as well). If we exclude this student's grade from the Ada grades on the third assignment, we calculate a mean of 83.4 and a standard deviation of 10.3 for the students using Ada on this assignment. This result is much closer to what we see on the first two assignments (students using FORTRAN have slightly higher grades and a slightly smaller standard deviation). A portion of these differences may be due to instructor grading policies, since the FORTRAN students had a different instructor from the Ada students, though it is not clear how strong this effect might be.

Astrodynamics uses the same course policy for incorrect output as that described for Introduction to Astronautics for the Engineer and Scientist. That is, students who get this output incorrect the first time can resubmit their programs (within a few days). If the output is correct in the resubmission, students can get half of the correct output points added back into their grade. We provide data about these cases in Figure 4. In the figure, the Number column indicates how many students had incorrect output on their first submission and the Percent column indicates what percentage of the section enrollment this number represents. We have excluded the Ada student who did not submit the third assignment from the data for the third assignment.

The percentages of incorrect output data on the first submission of all three assignments appear virtually the same for the FORTRAN and Ada students. Although the percentage of incorrect output on the first submission of the second assignment is significantly higher than for the other two assignments, the percentages are still approximately the same for the FORTRAN and Ada students.

		Program 1		Program 2		Program 3	
	Count	Mean	Standard Dev	Mean	Standard Dev	Mean	Standard Dev
FORTRAN	16	91.9	12.5	62.9	28.9	85.9	8.0
Ada	12	88.8	13.5	61.1	29.5	76.5	26.0

Figure 3. Grade Data for FORTRAN 77 and Ada Programs in Astrodynamics

		Program 1		Program 2		Program 3	
	Count	Number	Percent	Number	Percent	Number	Percent
FORTRAN	16	3	19%	11	69%	1	6%
Ada	12	2	17%	8	67%	1	9%

Figure 4. Incorrect Answer Data for FORTRAN 77 and Ada Programs in Astrodynamics

We can see from the data that the students using FORTRAN have slightly higher grades on the first two assignments (with slightly smaller standard deviations), while the students using FORTRAN had significantly higher grades for the third assignment. We note, however, that one of the students using Ada received a 0 on the third

		Program 1		Program 2		Program 3	
	Count	Mean	Standard Dev	Mean	Standard Dev	Mean	Standard Dev
FORTRAN	16	11.5	4.8	29.4	8.8	7.4	4.7
Ada	12	11.0	5.3	19.8	6.8	6.7	3.4

Figure 5. Programming Time Data for FORTRAN 77 and Ada Programs in Astrodynamics

Because there were two sections of the course being taught last semester, one in FORTRAN 77 and one in Ada, we also had an opportunity to collect programming time data for each of these programming assignments. Given potential differences in instructor grading policies, we believe programming time may be a better measure of the success of the transition, at least from the students' perspective. The programming time data is provided in Figure 5. Note that the time is measured in hours, and it includes the time to develop the appropriate algorithms and computer programs for each of the programming assignments. Again, the Ada student receiving a 0 on the third assignment is excluded from the data.

The mean programming time for the first assignment is approximately the same for the two programming languages. We do point out, however, that one of the Ada students actually had to learn Ada for the first assignment (he validated the core course), and he took 25 hours for the first assignment. If we exclude this student from the Ada times for the first assignment, the mean time becomes 9.6 hours with a standard deviation of 2.8 hours. For the second assignment, the mean programming time for the students using Ada was almost 10 hours less than for those using FORTRAN. For the third assignment, the mean programming time for students using Ada was slightly less than that for students using FORTRAN. It seems clear that, in Astrodynamics, students programming in Ada can take significantly less time than those programming in FORTRAN.

We view the programming time savings for the students as the most significant result of using Ada in the Astrodynamics course. Although an improvement in the quality of the submitted programs would also have been desirable, it appears that students in this course require significantly less time to generate Ada programs of comparable quality to those generated by students using FORTRAN. Given the extensive time constraints on the students at USAFA, we perceive a reduction in programming time for the students as a large benefit of the transition to Ada.

We believe the transition of the Astrodynamics course to Ada has been a success based on grade and incorrect first submission data; the students using FORTRAN have slightly higher programming grades than those using Ada, but also generally have slightly higher percentages of incorrect first submissions, so the programs generated by students using Ada appear to be of comparable quality to those generated by students using FORTRAN. We also note that the students using Ada took significantly less programming time than those using FORTRAN to generate comparable results.

3.4 Future Plans

Based on its apparently successful transition to Ada in the Introduction to Astronautics for the Engineer and Scientist and Astrodynamics courses, the Department of Astronautics also plans to use Ada in Advanced Astrodynamics and Space Mission Design, two more major's courses. The plan is to teach these courses in both FORTRAN 77 and Ada next year (there are still students using FORTRAN 77 in the pipeline), with Ada as the sole language in these courses the following year.

Other courses in the major are using Matlab, an interactive software environment for matrix computation, which employs a high-level matrix/array manipulation syntax. Although there are no current plans to transition these courses to Ada, student exposure to the data types and control structures in Ada may also help them successfully write the Matlab code required in some of the assignments in these courses.

4. ADA IN THE OPERATIONS RESEARCH MAJOR

The Operations Research (called OR hereafter) major is jointly administered by the Departments of Computer Science, Economics, Management, and Mathematical Sciences. Starting this semester, all OR majors are required to use Ada in their major's courses. Prior to this semester, OR majors were allowed to pick their programming language; their choices included Pascal, FORTRAN, Ada, and C. The primary motivations for transitioning to Ada were to use the "standard" programming language at USAFA, to avoid "relearning" of basic constructs, which in turn allows an introduction to more advanced features, and to avoid the administrative and pedagogical difficulties associated with using multiple programming languages in each course.

4.1 Programming in Ada, Pascal, FORTRAN, or C

The first programming course required of OR majors (after the core course required of all USAFA students) is Programming in Ada (a 1½ semester hour course). In the past, OR majors could take (instead of Programming in Ada) one of the following: Programming in Pascal; Programming in FORTRAN; or Programming in C (all 1½ semester hour courses). Programming in Pascal is no longer offered (because of lack of interest), and with the OR transition to Ada, Programming in FORTRAN will also be cancelled.

Instructors noticed that students enrolled in a programming course that was in a different programming language from the one they learned in the core course required significant help learning even the basic constructs of the new language. The programming courses were intended to further develop the students' programming skills, including developing an understanding of more advanced data and control structures, but they were clearly not meeting this goal when the students were learning a new programming language. Now that Programming in Ada is the required course for all OR majors (all of whom learned Ada in the core course), instructors believe that the quality of the programs is better and that the students are able to handle more advanced language features with greater ease.

4.2 Numerical Methods

Numerical Methods, a 1½ semester hour course, is the second programming course taken by OR majors. In the Fall 1997 semester, the course was taught in Ada, FORTRAN, and C. The administrative details associated with the multiple languages (assigning instructors to sections, keeping class discussions language-independent, etc.) were manageable but awkward. Because the OR major has decided to transition to Ada, the course used only Ada last semester. Using a single programming language permitted the instructors to focus entirely on numerical methods rather than covering programming language topics as well.

4.3 Computer Simulation

Computer Simulation, a 3 semester hour course, is the final programming course required by the OR major. Although most of the programming assignments in the course are accomplished using the simulation language ProModel, the first programming assignment requires development of a discrete-event simulation program using a general-purpose programming language. In Fall 1997, each student completed that assignment using Ada, FORTRAN, or C. Because of the administrative issues discussed earlier and the fact that the OR major is transitioning to Ada, the course will be taught in Fall 1998 using Ada only. The first assignment has historically been very difficult for the students to accomplish successfully, so we are eager to evaluate next year's results.

5. CONCLUSIONS

While computer science professors seem open to incorporating Ada into their curricula, it appears to be much less common for other majors to adopt Ada as their programming language. It is clear from the results above, however, that other academic majors can also successfully adopt Ada in their curricula.

The Department of Astronautics at USAFA has recently transitioned to Ada in two of its courses; based on the apparent success of these transitions, they plan to transition two additional courses to Ada. A comparison of grade data and incorrect output for FORTRAN and Ada does not provide clear indications of a major difference in student performance based on programming language. On the other hand, programming time data indicates that the students using Ada in the Astrodynamics course spent much less time programming than those using FORTRAN, a significant result given the extensive time constraints on students at USAFA. Anecdotal evidence from the instructors teaching the course indicates that students using Ada focus more on algorithm development than language issues during extra help sessions and that using Ada lets the instructors more easily introduce an abstract data type for vectors.

The Operations Research major has also recently transitioned to Ada for its major's courses for various administrative and pedagogical reasons. Although we have no hard data to support a claim of success in this endeavor (the transition has been too recent for us to collect such data), anecdotal evidence from the instructors teaching the course indicates that the transition has resulted in an improvement in student programming proficiency.

For both the Department of Astronautics and the Operations Research major, using the "standard" language at USAFA has also been beneficial. Instructors teaching the courses are assured that students have been exposed to programming in the same language being used in their courses. This previous exposure lets the instructors focus more on the actual algorithm development for the problem solutions and/or more advanced language features.

Given the apparent success of the transitions to Ada in the above two majors, it seems clear that other majors may also benefit from including Ada in their curricula. We are currently discussing this potential with the Department of Engineering Mechanics, which is considering using Ada in their core Engineering Mechanics course.

6. ACKNOWLEDGMENTS

Many thanks are due to Pam Magee and Mike Bettner for gathering the time and grade data for the Astro 321 programming assignments.

7. REFERENCES

[1] Martin C. Carlisle and A.T. Chamillard, AdaGIDE: A Friendly Introductory Programming Environment for a Freshman Computer Science Course. *Ada Letters*, 18(2):42-52, March/April 1998.

[2] A.T. Chamillard and William C. Hobart, Jr., Transitioning to Ada in an Introductory Course for Non-Majors, In *Proceedings of Tri-Ada '97*, St Louis, Missouri, November 1997.

[3] Michael B. Feldman, Ada as a Foundation Programming Language, Online, Internet, 15 Feb 1997. Available: http://www.seas.gwu.edu/faculty/mfeldman/CS1-2.html