Overcoming Limited Resources: An Academia-Government Partnership on Software Engineering and Capstone Projects

Adrian Rusu [1], Amalia Rusu [2], Mike Paglione [3], Fred Snyder [4], and Confesor Santiago [5]

Abstract - **Partnerships can be effective tools for engaging universities with their communities. In this paper we present an academia-government partnership between the Software Engineering, Graphics, and Visualization research group established in the Department of Computer Science at Rowan University and the Federal Aviation Administration's Simulation and Analysis Group at William J. Hughes Technical Center in Atlantic City, New Jersey. This partnership was established with the primary goal of increasing Rowan University's Computer Science undergraduate students exposure to real-world projects in software engineering and other upper-level courses, while allowing the FAA to receive quality professional software products by spending only limited resources. As a result, students acquire skills necessary to enter professional science and engineering occupations immediately after graduation. We present the major outcomes of this partnership, which include quality software products and publications in national conferences, and discuss the set up which allowed us to overcome both academia and government difficulties.**

Index Terms – Academia – Government Partnership, Capstone Projects, Real-World Projects, Software Engineering Education.

INTRODUCTION

Industry – academia and academia – government collaborations and partnerships can result in multiple benefits for all parties participating in joint projects. Therefore, a plethora of such science, engineering, and technology partnerships have been established (see for example [1]-[5]), either as capstone projects or as internships. Benefits for industry or government include reduced cost of training, better prepared local workforce, and useful products at a lower budget. Rewards for academia include additional equipment and funds for research and development, increased student enrollment, and new opportunities and application ideas.

Software engineering and upper-level project-based computer science courses provide students with the opportunity to use their problem solving skills and the technical knowledge they gain throughout their college experience to develop software products that meet specific requirements.

Exposure to real-world development or applied research projects would greatly improve a student's performance in the industry. However, in the absence of any incentives, the extra difficulties associated with real-world projects often deter the faculty in pursuing them. Such difficulties include:

- *Locating appropriate projects*: prior to the beginning of the semester the instructor needs to spend considerable effort in locating partners with appropriate projects available and willing to spend time and resources interacting with the students.
- *Excessive faculty workload*: it is more time-consuming for the instructor of the course to manage real-world projects, especially when there are no additional instructors or teaching assistants available to contribute.
- *Projects may not be finished by semester's end*: the time needed for completing a project is fixed (usually one semester), thus there is little time for deployment, maintenance, and transition from one team to another.

In addition to the difficulties on the academia side, the government encounters difficulties on its own. They include dealing with many restrictions, as well as with limited available funding resources. In general, the government does not have a viable instrument ready to place funding into new partnerships.

In this paper we present an academia-government partnership established with the primary goal of increasing undergraduate students' exposure to real-world projects in software engineering and upper-level computer science courses, while allowing government entities to receive quality professional software products and research expertise by spending only limited resources.

Our partnership has a strong focus on students' incentives and incorporates viable solutions for the above mentioned difficulties. As a result, the students are

[1] Adrian Rusu, Department of Computer Science, Rowan University, 201 Mullica Hill Road, Glassboro, NJ 08028, USA, rusu@rowan.edu
[2] Amalia Rusu, Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14260, USA, air2@cedar.buffalo.edu
[3] Mike Paglione, Simulation and Analysis Group, Federal Aviation Administration William J Hughes Technical Center, mike.paglione@faa.gov
[4] Fred Snyder, University Relations, Federal Aviation Administration William J Hughes Technical Center, fred.snyder@faa.gov
[5] Confesor Santiago, Simulation and Analysis Group, Federal Aviation Administration William J Hughes Technical Center, confesor.santiago@faa.gov

knowledgeable in specific areas of software development and technology, and have skills necessary to enter professional engineering occupations immediately after graduation.

FAA/ROWAN PARTNERSHIP

At the end of the Fall 2004 semester, Federal Aviation Administration's Conflict Probe Assessment Team (CPAT), part of the Simulation and Analysis Group at William J. Hughes Technical Center in Atlantic City, New Jersey, teamed with the Software Engineering, Graphics, and Visualization (SEGV) Research Group established by the Department of Computer Science at Rowan University. The FAA and Rowan University collaboration started through a Cooperative Research and Development Agreement (CRDA), which allows the FAA to share facilities, equipment, services, intellectual property, personnel resources and other cooperation with Rowan University. In general, a CRDA is used to develop an idea, prototype, process, or product for direct application to the civil aviation community and/or indirect application for commercial exploitation. Under this specific CRDA, the FAA provided, as Government Furnished Equipment (GFE), four Desktop Personal Computers and one Linux Server. With this GFE, Rowan University established the FAA/Rowan Air Transportation Research (FRATR) Laboratory in the Department of Computer Science at Rowan University. The objective was to leverage upon SEGV Research Group's capabilities and student talent. The GFE matches the characteristics of the computers on which the software would finally be installed, in order to accommodate for easier deployment.

CPAT has also provided annual co-op positions (full-time during breaks and part-time during the semesters) for two graduate students, senior members of SEGV. These positions have partially supported their graduate studies and have allowed them to become familiar with the FAA working environment and projects' details. Figure 1 displays the overall structure of the partnership. This figure also displays the coordination details. The SEGV director negotiates with the FAA the details of the project(s) prior to their introduction into the classroom and coordinates the roles of the graduate and undergraduate students based on the scope and goals of each project. Being the liaison between the FAA and SEGV, the graduate students play roles on both sides: on the FAA side they schedule meetings and elicit new requirements from the users based on the FAA needs, and on the SEGV side they contribute by sharing the management of the projects and the mentoring of the undergraduate students participating in the projects, thus easing the workload of faculty. When they become very well accustomed with the projects' details, the graduate students could even play the role of the customer, thus relieving part the workload for the FAA managers. In order to keep deliverables on track with the deadlines, weekly meetings between SEGV director and students are usually being scheduled. The graduate students perform installations and maintenance and assure the smooth transition of projects' versions from one undergraduate team to another.

Since its inception, seventeen undergraduate members of the SEGV research group have taken advantage of the partnership with the FAA and have participated on projects in the FRATR lab. Some of them have even contributed to more than one project. They have developed quality software products that are currently in use by the FAA. The students have earned school credit by enrolling in courses provided in the Department of Computer Science at Rowan University. These courses include: Principles of Software Engineering (the semester-long introductory software engineering course takes the students through all four majors phases of software lifecycle development, and is structured based on the approach introduced in [7]), Introduction to Information Visualization (introduces students to graphics programming and to techniques for transforming raw data in visual forms which take advantage of human visual perception), Senior Project (the capstone course), and Independent Study.

The three major software products developed so far are being summarized in the next section, to allow for an easier understanding of their complexity and usefulness for the aeronautical industry.

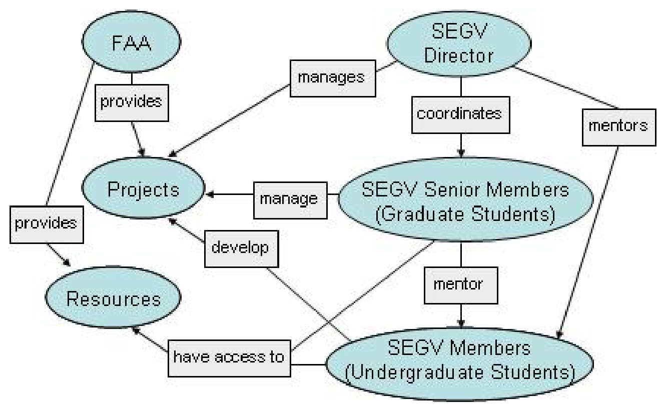

FIGURE 1
OVERALL PARTNERSHIP STRUCTURE

PROJECTS

CPAT analysts use software applications and mathematics to test Decision Support Tools (DSTs). One function of DSTs is to help air traffic controllers safely and efficiently control the National Airspace System. Hence, air traffic controllers are beneficiaries of thorough testing of DSTs. All three software products have scalable object-oriented designs, with their main capabilities implemented in Java, and their graphics implemented in JOGL (OpenGL bindings for Java). The implementation language (Java) was imposed by the FAA because of their workforce training in this programming language, which allows for easy future updates. The JOGL graphics was chosen by Rowan students because of its simple use and powerful 2D and 3D capabilities, which allows for future extensions.

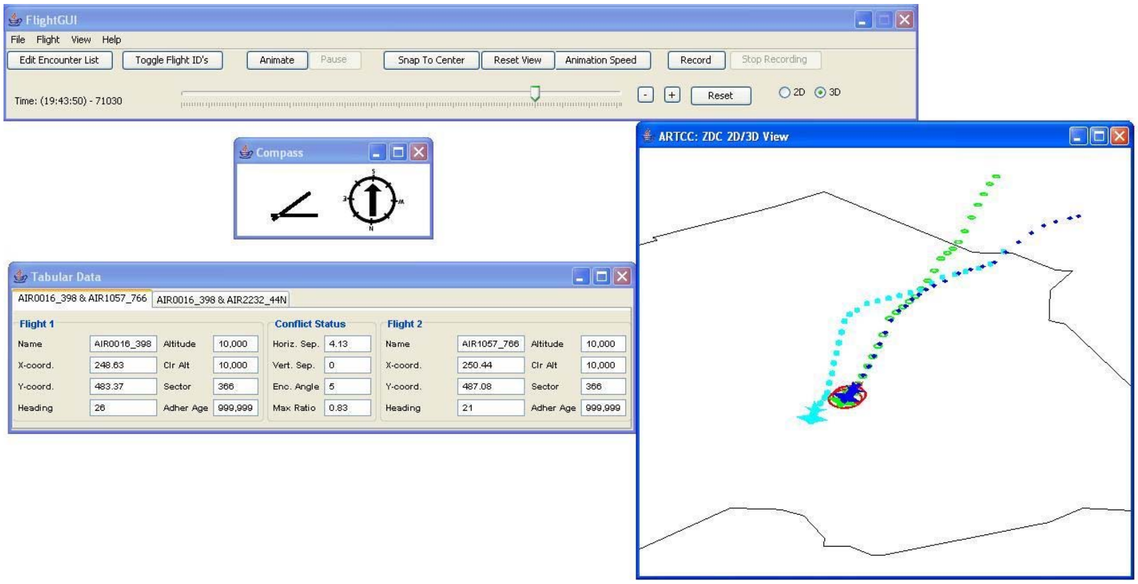

FIGURE 2
SCREENSHOT OF FLIGHT GRAPHICAL USER INTERFACE

I. Flight Graphical User Interface

Conflict probes provide air traffic controllers with predictions of conflicts (i.e., loss of minimum separation between aircraft) within a parametric time (e.g., 20 minutes) into the future. Accuracy testing requires detailed analysis of the conflict probe's predictions. The ultimate goal is to develop algorithms to improve the accuracy of these predictions.

Flight Graphical User Interface (FlightGUI) [8] is a visualization tool developed in the FRATR lab, currently used by the CPAT to test the accuracy of conflict probes. FlightGUI animates the flight paths of aircraft by displaying spatial/time relationship to each other during an encounter of a flight pair. It indicates when a conflict occurs, by encircling the conflicting flights (i.e., non-adherence to minimum separation standards - see Figure 2 for an example), which aids the analyst in noticing the conflict and further study its characteristics by analyzing the information about the aircrafts in tabular form. This allows the analyst to study whether the conflict has been accurately predicted. FlightGUI's main interface includes four separate free-floating, independently positioned, synchronized windows. These windows are the *FlightGUI Toolbar*, the *Animation Window*, the *Compass Window*, and the *Tabular Data Window*:

- The analyst uses the *FlightGUI Toolbar* (located at the top of Figure 2) to control the *Animation Window* that presents 3-D plots of the aircraft track points and Air Route Traffic Control Center (ARTCC) boundaries. During a run, the analyst can click Animate to start the animation, which allows the analyst to visualize the flight path of aircraft under review, and can click Pause to stop the animation, which allows the analyst to process what has occurred, and formulate a report.
- The *Animation Window* (located on the right side of Figure 2) presents plots of the flights' track data along with a plot of the ARTCC boundary. Within the realm of the *Animation Window*, the analyst can rotate the stereographic plane about the center of the ARTCC to get a better view of the flight pair. In addition, the analyst can obtain a clearer view of the flight pairs by "zooming" in and out. Furthermore, the analyst can tilt the stereographic plane, which in effect changes the analyst's view of the horizon.
- The *Compass Window* (located next to the *Animation Window*) shows the incline or tilt of the stereographic plane and the actual compass to the right advertises North, South, East, and West.
- The *Tabular Data Window* (located on the left side of Figure 2) is animation time dependent and contains a tabbed interface, one tab per flight pair selected. Its purpose is to provide detailed information about each flight. During the animation state both the time label, slider bar, and tabular data maintain synchronization with the time state of the animation.

FlightGUI was developed during the duration of the 2005-2006 calendar year, by a total of eight undergraduate students, as follows:

- Software Requirements Specification, Detailed Design Document, working prototype, Testing Document, and User Manual were developed during the Fall 2005 semester, as part of the Principles of Software Engineering course (four students who contributed mostly on the documentation), and as part of the Introduction to Information Visualization course (two students who contributed mostly on the implementation of the prototype).
- Full implementation/design matching, thorough testing and deployment, as part of the Senior Project (two students).

FlightGUI has received interest from other groups at the FAA, such as the Human Factors Group, which intends to adapt it to meet their specifics needs.

II. Trajectory Graphical User Interface

As previously mentioned, DSTs are used to assist air traffic controllers to separate air traffic. DSTs predict aircraft flight paths (trajectories) and foretell potential conflicts. The accuracy of the predicted trajectories generated by the DSTs determines their overall performance.

A trajectory is a four dimensional (latitude, longitude, altitude, and time) prediction of an aircraft's flight path. Various algorithms and relational databases have been developed to investigate and calculate the performance accuracy of trajectory prediction. The large amount of information led to the need for a specialized visualization tool. Trajectory Graphical User Interface (TrajectoryGUI) [9] is a visualization tool developed in the FRATR lab, currently used by the CPAT to evaluate the accuracy of predicted trajectories (see Figure 3 for an example). An analyst can use TrajectoryGUI to package the trajectory accuracy results and illustrate reasons for inaccuracies.

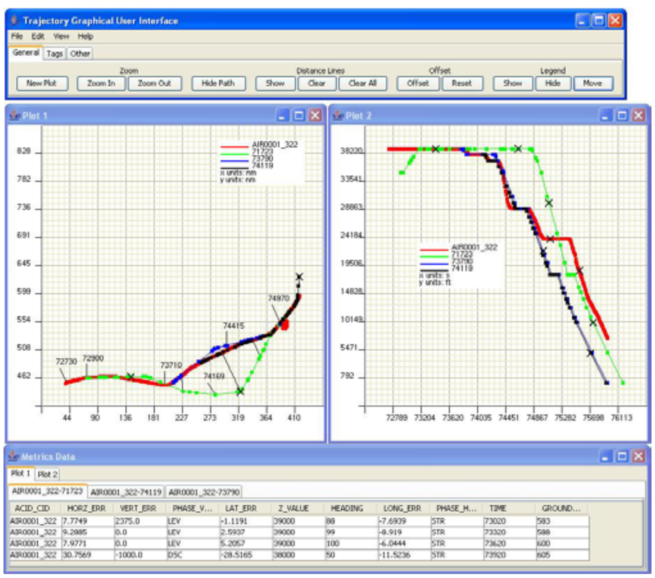

FIGURE 3
SCREENSHOT OF TRAJECTORY GRAPHICAL USER INTERFACE

TrajectoryGUI allows the analyst to create 2-D plots of flight track and trajectory error data fields, such as X-Y (latitude versus longitude) and T-Z (time versus altitude). Each plot can contain multiple actual flight paths, and each flight path can contain multiple trajectory paths. Over the duration of a flight, trajectory modelers generate new predictions. Each new trajectory should be more accurate, because as time progresses more information (i.e. weather forecast, deviated routes) should be available to generate more accurate trajectories.

Its interface includes *the TrajectoryGUI Toolbar*, an *X-Y Plot*, a *T-Z Plot*, and the *Metrics Table Window*:

- The *TrajectoryGUI Toolbar* (located at the top of Figure 3) contains the menu and buttons to use TrajectoryGUI's functionality. During an analysis, the analyst can zoom in and out to get a better view of the data. Also, the analyst can add distance lines to calculate the exact distance between the same time coincident position in the actual flight path and predicted trajectory.
- An *X-Y Plot* (located on the left side of Figure 3) contains the latitude and longitude positions projected on a stereographic plane of both the actual flight path and their trajectory paths. In the example of Figure 3, four different paths are plotted in the *X-Y Plot*. The red plot represents the actual flight path and is identified in the legend by its aircraft identification. The green, blue, and black plots are different trajectories for the flight, identified based on their build time, which is time in seconds of the day. In the Figure, the green trajectory is the earliest trajectory prediction, the blue is in the middle, and the black is the latest. Using TrajectoryGUI, the analyst is able to easily gauge the accuracy of trajectories, and indeed as shown in Figure 3, the earliest (green) trajectory has the most error, and the latest (black) trajectory is quite accurate.
- The *T-Z Plot* is located on the right side of Figure 3, and contains time versus altitude plots. Within a *T-Z Plot*, the analyst can place position tags to retrieve specific altitudes at certain times.
- The *Metrics Table Window* is located at the bottom of Figure 3, and is filled with the accuracy measurement data selected from the database for a more in-depth investigation, if needed.

TrajectoryGUI was developed during the Spring 2005 and Fall 2005 semesters, by a total of nine undergraduate students, as follows:

- Prototype during Spring 2005 as part of Senior Project (four students).
- Software Requirements Specification, Detailed Design Document, fully functional implementation, Testing Document, and User Manual were developed during the Fall 2005 semester, as part of the Principles of Software Engineering course (five students). The transition from the existing prototype to a fully functional system included a complete redesign.

TrajectoryGUI has received interest from other groups at the FAA, such as the Target Generation Facility Group, which intends to adapt it for use on their trajectory predictions.

During the Spring 2007 semester three students enrolled in the Independent Study course developed new functionality for TrajectoryGUI, as well as accordingly adjusted the existing documentation (requirements, design, testing documents and the user manual), and undergone thorough testing. One of the major new features currently in development stage is Trajectory Galaxy Visualization (TrajectoryGalaxyViz). The visualization system will allow an analyst to obtain various sets of flight data from a remote database system and map them into a galaxy visualization

[10]. The galaxy visualization clusters graphical objects, commonly referred to as planets, and each planet represents a certain factor of the trajectory prediction error. Examples of factors currently under investigation are: start and end time of flight, flight type, flight's average ground speed, and flight destination. The spatial location of each planet represents each factor's effect on a certain measurement error of trajectories; hence closer planets, signify similar affecting factors. Thus, TrajGalaxyViz will provide a visualization that could aid FAA analysts in discovering different factors of trajectory prediction inaccuracies, as well as their significance, which could open a new avenue of research in algorithms that build trajectory predictions.

III. Radar Simulator

The Radar Simulator (JPlotASR) is a visualization tool to plot and view stored aircraft position data received from ASR-9 Surveillance Radars. JPlotASR is an update to the original radar simulator software, which was written in the 1980s in an obsolete language that could only run in DOS, and its functionality is now very limited. JPlotASR does not display real-time radar data, but rather displays radar data that has been stored for analytical purposes. It processes authentic flight information stored in a HEX encoded file and displays the flight data in a radar-like view (see Figure 4). The main benefit of the software is it allows for reviewing of received radar data as it was originally displayed on the real-time radar screen, at any time.

Three students developed the Radar Simulator implementation during the duration of the 2005-2006 academic year. They were enrolled in Introduction to Information Visualization course in Fall 2005 and in Independent study in Spring 2006.

PARTNERSHIP SUCCESS

All undergraduate students have been recognized with certificates of appreciation signed by the Director of the FAA William J. Hughes Technical Center, during specially arranged presentation sessions at the end of the corresponding semesters. These certificates have been allocated based on the quality of the products and were subject to the successful completion of the projects. The students have also presented their work and received recognition on campus, during Rowan's Science, Technology, Education, and Mathematics (STEM) Student Research Symposium.

In general, most entry-level software engineers are assigned to the tail end activities of the software engineering lifecycle, such as testing and maintenance, in order to first get familiar with the product domain, and gain experience with a company's operations. The experience accumulated during the development of the products and the certificates of appreciation the undergraduate students received, helped the majority of them secure jobs in the aviation industry and industries alike. In fact, eleven of the seventeen undergraduate students who participated in the collaboration received offers of employment from FAA contractors. In addition, the experiences gained by working on the projects propelled some students into graduate school.

Computer Science and Engineering students who participate in cooperative education programs can benefit greatly from their work experience and domain knowledge [11]. Together with the mentoring experience they

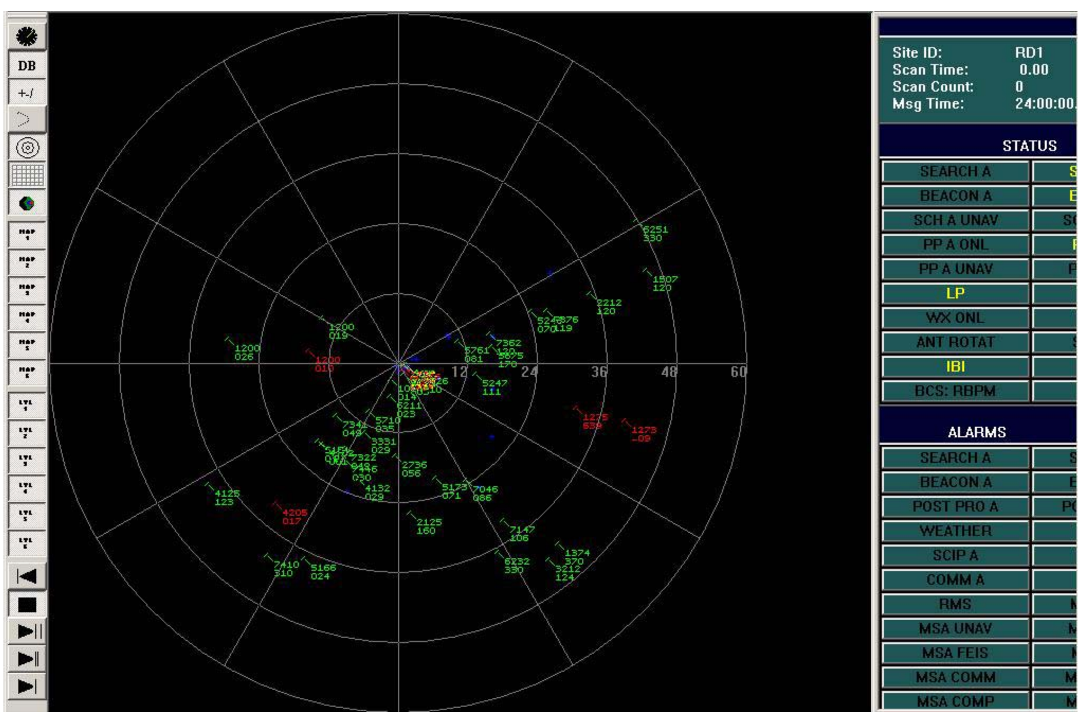

FIGURE 4
SCREENSHOT OF RADAR SIMULATOR

accumulated, the graduate students who benefited from the co-op positions have positioned themselves as prime candidates for quality jobs as government employees, within the FAA. Graduate students were also given the opportunity to present the results of the work of the partnership at national aviation conferences.

In addition to the quality software products which eased their everyday work, the two publications ([8], [9]) produced during the collaboration have also improved CPAT's standing within the FAA. CPAT has also already used TrajectoryGUI in two of their studies. The study of [12] details the impact turns have on radar tracking. CPAT analysts analyzed turning flight paths and generated statistics on how radar tracking is less accurate while an aircraft is in a turn. This defines the need to improve the algorithms of radar tracking systems while aircraft are within a turn. In addition, the study of [13] compared the radar track positions to tracks positions produced by Global Positioning Satellite (GPS). CPAT visualized the radar track and GPS paths, and generated reports on the differences. This study gives insight on the performance of GPS, as the FAA moves to utilize GPS more in their systems.

CONCLUSION

Over the past several years, the trend has been that academia and government or industry develop relationships that expand the continuum in science, engineering, and technology education through government or corporate related University Relations programs. When academic and government or industry have common goals and mutual interests, successful partnerships can generate excellent results.

Through the partnership between the FAA and SEGV research group, the FAA has been able to overcome bureaucracy in order to leverage on research group's expertise and receive valuable software products by spending only limited resources.

The partnership has provided challenging projects to undergraduate students and allowed them to gain valuable real-world experience. Because of the exposure to government partners, students have a competitive edge over their peers, when competing for quality jobs. The majority of students who participated in the projects have received offers of employment from FAA contractors or even from the FAA itself.

Overall, the partnership has directly helped overcoming the difficulties associated with bringing real-world projects into the classroom by providing a near-infinite source of available projects of variable size and level of difficulty. The hiring of the two graduate students as co-ops with the FAA has eased faculty workload by allowing them to participate in the mentoring of the undergraduate students and in setting up details of meetings when needed. We found that, if the undergraduate students workload is set properly, the teams are able to have the projects complete by the end of the semester, therefore limiting the risk of project failure. Because of their in-depth familiarity with the projects' details, the graduate students could perform installations and any corrective and perfective maintenance that may arise, and thus allow the software to be used without major interruptions. The one-to-one correspondence between the projects' documentation and their implementation allows for smoother transition of projects' versions from one undergraduate team to another.

ACKNOWLEDGMENT

We thank FAA employee Mark Henlon for contributing to the collaboration by providing the Radar Simulator project. We also thank instructor John Robinson for his contribution to the Radar Simulator project.

REFERENCES

[1] Sterkenburg, R., "The Unique Challenges and Advantages of Developing an Interacting Partnership Between an Aeronautical Technology Program and Major Air Carriers", *Proceedings 4th International Conference on Engineering Education,* 2001.

[2] Morell, L., Velez-Arocho, J. I., Buxeda, R., Vasquez, R., Johnson, W.C., Irizarry, M. M., Davis, I. T., "Strategic Alliances with Industry to Enhance the Undergraduate Science, Mathematics, and Engineering Education", *Proceedings 4th International Conference on Engineering Education,* 2001.

[3] Radcliffe, D., Brown, A., "Fostering Professional Development Through Integrating Work and Formal Learning", *Proceedings 33rd ASEE/IEEE Frontiers in Education Conference,* 2003.

[4] Dabipi, I. K., Arumala, O. J., Burrows-McElwain, J. B., "Advancing Aviation Sciences Education through Industry Partnerships", *Proceedings 9th International Conference on Engineering Education,* 2006.

[5] Kornecki, A. J., Khajenoori, S., Gluch, D., "On a Partnership between Software Industry and Academia", *Proceedings 16th Conference on Software Engineering Education and Training,* 2003.

[6] Rusu, A., Webb, R., Shanline, D., Santiago, C., Luna, C., Kulak, M., "A Multiple-Projects-Multiple-Winners Approach for Teaching Software Engineering", *Proceedings 9th IASTED International Conference on Software Engineering and Applications*, ACTA Press, 2005.

[7] Santiago, C., Oaks, R. D., Paglione, M. M., Rusu, A., "Flight Graphical User Interface: A Visualization Application for Analyzing Flight Conflict Probe Tools", *Proceedings 51st Annual Air Traffic Control Association Conference*, 2006.

[8] Santiago, C., Paglione, M. M., Rusu, A., Liu, S., Oaks, R. D., "Using Graphical Software for Evaluating Aircraft Trajectory Predictions", *Proceedings 50th Annual Air Traffic Control Association Conference*, 2005.

[9] Crow, V., Wise J.A., Thomas, J.J., Lantrip, D., Pottier, "Multidimensional visualization and browsing for intelligence analysis", *Proceedings of GVIZ '94*, 1994.

[10] Mathews, C. P., Mamaloukas, J., Parrott, S., "The Positive Influence of Cooperative Work Experience on the Capstone Design Experience of Engineering Students", *Proceedings 33rd ASEE/IEEE Frontiers in Education Conference,* 2003.

[11] Paglione, M. M., Santiago, C., Ryan, H., Gramatges, G., "The Impact of Turns on Host Radar Tracking", *Proceedings 25th Digital Avionics Systems Conference*, 2006.

[12] Paglione, M. M., Ryan, H., "Comparison of Host Radar Positions to Global Positioning Satellite Positions", *Proceedings 24th Digital Avionics Systems Conference*, 2005.