Software as a Component in Safety-Critical Systems
Professional Development Leave Report

Dr. James Vallino
Associate Professor
Department of Software Engineering

1. Introduction

The theme for my professional development leave was to consider software usage within safety-critical systems. Computer processors across a range of sizes are commonly embedded within devices and systems. When these are safety-critical devices, the software running on those embedded processors often performs safety-related functions. The safety requirements of these systems necessitate design considerations for the overall system and the software embedded within it that are beyond what the software engineer must consider for standard desktop applications. In the real-time and embedded systems courses that I teach, I discuss safety-critical systems and the constraints imposed by the safety requirements. I used my professional development leave to gain experience working with groups dealing with software in safety-critical systems. With this experience, I became better prepared for the classroom discussions of safety-critical systems. Working in this area also provided contacts for future collaboration on scholarship activities.

The activities during my full academic year leave were different than what I had originally described in my leave proposal (http://www.se.rit.edu/~jrv/publications/Vallino-SoftwareSafety-LeaveProposal.pdf). The work was still directly related to the theme I proposed, but, for a variety of reasons, the three organizations I visited were different than the ones I detailed in my proposal. During my leave, I visited a regulator of safety-critical medical devices (US Food and Drug Administration), a government agency that designs safety-critical space systems (NASA Goddard Space Flight Center), and a manufacturer of commercial and military jet engines (Pratt & Whitney). I also wrote a short description of my year on leave for the spring 2006 issue of the library's Scholarship @ RIT newsletter. The newsletter is available at http://wally.rit.edu/userservices/pubschol/newsletters/ScholarshipNewsletterSpring2006.pdf.

This report describes the work I had done during each of my three visits in more detail than presented in the Scholarship @ RIT article. I also identify the areas where there is potential for continued collaboration with the organizations that I visited.

2. US Food and Drug Administration

In fall 2005, I spent three months, from the middle of August until the middle of November, at the US Food and Drug Administration (FDA), Center for Devices and Radiologic Health (CDRH) in Rockville, MD. The Center is primarily responsible for the regulation of medical devices in the marketplace. This regulation includes review of pre-market applications through the Office of Device Evaluation (ODE) and post-market compliance oversight through the Office of Compliance (OC). I had a position of Staff Fellow in the Office of Science and Engineering Laboratories (OSEL) working with the Division of Electrical and Software Engineering (DESE). OSEL provides engineering and scientific support for the regulatory activities of the other two offices within CDRH. The DESE group is the focal point within the FDA for expertise in electrical engineering and software engineering as it relates to use in medical devices.

Most safety-critical systems are under some form of government regulation. This is certainly the case for medical devices which are under FDA regulation. During neither my previous industrial experience nor my time at RIT have I had any experience working with regulated products. The opportunity to work as a Staff Fellow within a regulatory agency allowed me to gain detailed knowledge of the regulatory process and to learn what software engineering aspects were of importance for the developers of medical device software. My first step was to wade through the FDA bureaucracy to understand how the regulatory process flowed through the agency. The FDA has traditionally had a strong separation between pre-market and post-market regulation. The former is handled by the Office of Device Evaluation and involves reviewing the information that the device manufacturer provides in their pre-market approval application. The latter is under the control of the Office of Compliance which includes the FDA's enforcement group which has the badges and guns, and can shut down a manufacturer's operation. More often, the result of an enforcement action to a medical device recall.

The pre-market application must demonstrate that the device is "safe and effective" and that the manufacturer has a quality system in place for controlling and tracking manufacturing, service, and event reporting. DESE paid for me to attend a four-day Software Validation course with one of their new permanent employees. This was a standard industrial training course being offered only to FDA employees. From the software engineering perspective, the software validation requirements imposed by the regulatory requirement to have a quality system in place are no more than the standard best practices recommended for most software projects. These practices are what we teach throughout our software engineering curriculum. Non-regulated industries all too often may ignore many of these practices. Unfortunately, for medical device manufacturers the result may be "bad things happening to good people."

The main focus for ODE during pre-market approvals is the medical safety and effectiveness of the device. Their people are mostly physicians, statisticians and public health experts. They have a limited amount of engineering expertise. A large function of the DESE group is to review the software aspects of pre-market applications. ODE will request a software consult when a medical device has a large software component. The DESE consultant will concentrate on the quality system associated with the software design and development. As an exercise to see if I understood the process, I asked to review two pre-market applications. I reviewed two devices that a full-time FDA person had already reviewed. After my review, I sat with the original reviewer and compared notes. In neither case was there anything that I missed. With one of the devices, I identified a small number of issues not noted by the original reviewer that he thought could have been mentioned.

The OC will track after-market compliance with regulations. They do unannounced and announced visits to manufacturers to check if they are maintaining the required design and manufacturing quality control system. OC is also responsible for following up with investigations of reported device incidents. Again, DESE engineers are involved with after-market compliance activities in a consulting role. DESE will review the software and electrical engineering aspects of a manufacturer's response to an FDA compliance action. While I was working there, the national news had several stories about failures of implanted cardiac devices similar to the one shown in the x-ray image in Photo 1 at the right. There were multiple device failures that resulted in at least one well-publicized death. The manufacturer had already published a notice about the problems and suggested remedies. When the marketplace reported other failures and the manufacturer rescinded some of the original remedies and issued new instructions, the FDA took a closer look. Enforcement officers found what they believed to be a number of violations of quality system regulations at the manufacturer's site. At the request of DESE, the FDA requested that the manufacturer supply software source code for review. This is an option available under the FDA's regulatory authority that the agency rarely uses. I participated in discussions about concerns found in the software. This resulted in continued actions against the company.

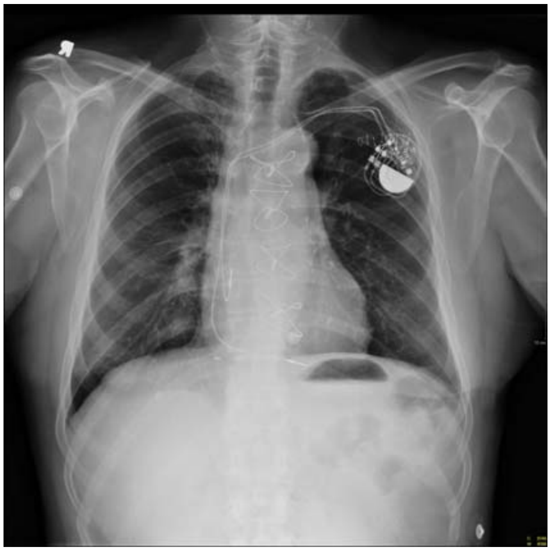

Photo 1 - Implantable cardiac device
Image courtesy of Dr. C Varnis

OSEL is also responsible for forward-looking scientific and engineering investigations. Within DESE most of this activity relates to software. During my visit, I worked on two forward looking projects for the group. The first investigation was to explore the potential to develop a tool that would allow a medical device designer to specify software behavior via a timing diagram. Once the behavior was formally defined, the tool would auto-generate source code to meet that specification. I made suggestions for an approach to develop this tool within the Eclipse framework. A second forward-looking area of interest to the group is static checking of source code. The group is working with several university research groups who are working in that area. My investigation centered on static checking of C source code to the MISRA-C standard. An automotive software reliability consortium developed this standard for software embedded in processors on-board automobiles. Other industrial segments have also adopted the MISRA-C standard. Since a large amount of the software in medical devices is coded in C this appears to be a good standard for device manufacturers to adopt. The FDA would like an open-source tool that would validate MISRA-C compliance. They would make this tool available to any device manufacturers who wanted to move to this level of standard practice. I described several approaches that could be taken to develop such a tool. A final task for me related to DESE's involvement with national and international standards work. One group member is on the committee developing a standard for medical devices that do closed-loop physiologic control. I provided substantial comments on the draft standard that he requested that I review.

I made two presentations during my visit. The first was an OSEL Staff College presentation in which I discussed the work being done to allow developers to use Java in a real-time context. This is important to medical devices since much of the software has real-time constraints. My second presentation addressed the DESE group's division chief's concern that they see a wide variability in the quality of medical device software developers. He believes that some of this could be addressed at the undergraduate education level. While I was there, he asked me to make a presentation to the group that described the guidelines for computer science, computer engineering and software engineering curriculum development and accreditation, and how that aligned with what I saw as the needs for a regulated industry. I also made suggestions for ways in which DESE could potentially have an impact including: membership on industrial advisory boards; participation in senior projects; and contribution of case studies for class assignments.

3. National Aeronautics and Space Administration

From December 2005 through February 2006 I visited the Software Engineering Laboratory at the NASA Goddard Space Flight Center in Greenbelt, MD. Dr. Michael Hinchey is in charge of the laboratory. During this period I had the opportunity to review many of NASA's software development practices including their Software System Safety Guidelines. This review provided several ideas to me for how to make use of safety standards within course assignments. Recently, the Software Engineering Lab developed a research tool called Requirements to Design to Code (R2D2C). This tool starts with the specification of system behavior defined in terms of scenarios of concurrent actions. From these scenarios, a theorem prover, which understands the semantics of concurrency as defined in a particular design methodology, infers a formal design for the overall system. R2D2C is not tied to a particular design methodology. The tool is currently implemented using Communicating Sequential Processes (CSP) as the design method. Once R2D2C has inferred a formal design, it is a straightforward step to autocode source code to implement that design. To translate the CSP system design into compilable source code, R2D2C uses the JCSP framework.

Dr. Hinchey had asked me to investigate incorporating safety concerns into the R2D2C framework. After gaining an understanding of the R2D2C approach, I suggested a way to use scenarios of safe operating conditions to add a safety monitor to the system. These are handled in a manner similar to the scenarios defining all of the system behaviors. After the theorem prover infers a system design, it can model check the system for violations of the safe operating scenarios. If any are found, there are errors in either the requirements scenarios or the safe operating scenarios. These errors must be cleared before proceeding with the design. At this point, if the theorem prover has shown mathematically that there are no safety violations you might see no need for implementing the design inferred from the safe operating scenarios. In safety-critical systems you still need to safeguard against failures that are outside of the formal mathematical proofs. An implementation of the inferred safety monitor could do this. The monitor will track the execution of system actions. As long as the actions occur in an order defined by the safe operating scenarios, the safety monitor quietly continues tracking system operation. Only if an action occurs out of order would the safety monitor generate an alert and possibly initiate failsafe actions. The R2D2C framework will autocode an implementation of the safety monitor from the design inferred from the safe operating requirements scenarios. I suggested this approach toward the end of my visit to NASA. Unfortunately, there was not enough time to implement it before I moved on to the third part of my professional development leave. We do plan to continue work on this idea.

4. Pratt & Whitney

Photo 2 - PW4000 112 inch fan jet engine

The final visit of my professional development leave was with the Real-Time and Embedded Software (RTES) group of Pratt & Whitney in East Hartford, CT. I worked with this group, which is headed by Nannette Savage, from March through early June 2006. East Hartford designs engines for both commercial and military applications. Photo 2, on the right, is of me standing in front of the 112 inch fan on a PW4000 jet engine. The RTES group is primarily on the commercial side of the house, responsible for the PW4000 and PW6000, but the Embedded Systems section or domain cuts across both application areas. There were several tasks with which I was involved during my stay here. First, I needed to gain an understanding of how jet engines work and how they are controlled so that I had a basic grasp of discussion content amongst the engineers. In some ways, this brought me back to my engineering roots of as an undergraduate mechanical engineering major. I enjoyed refreshing my understanding of basic thermodynamic cycles as applied in a modern jet engine. A jet engine is a very simple thermodynamic system but to make it efficient and practical to use requires significant complexity. One of the most surprising things that I learned is that only 15% of the thrust from a modern fan jet engine comes from burning jet fuel. The engine burns the jet fuel primarily to spin a very large fan at the inlet of the engine. The mass of air propelled rearward by that fan provides 85% of the engine’s thrust.

I was asked to participate on a task force charged with defining Pratt & Whitney internal standard practices for performing Fault Tree Analysis, and Failure Modes and Effects Analysis (FMEA). These are two standard techniques for system safety analysis. I took part in discussions of the developing standards, reviewed the documents, and did the final edits incorporating all comments. The Fault Tree Analysis standard practice was approved. Other program schedule pressures caused a delay in work on the FMEA standard and it was not completed before the end of my visit.

The RTES group has responsibility for assembling all of the software that runs on the Full-Authority Digital Electronic Controller (FADEC) that has almost complete control of the Pratt & Whitney jet engines. The controller, shown mounted on the inside of a PW4000 nacelle in Photo 3, is fully redundant with cross-link checking of operations allowing either channel to take over full control of the engine if one channel fails. During my visit I had the opportunity to look in great detail at the organization of the software in the FADEC and how the safety mechanisms are implemented. There are a large number of power-on system tests performed to check that the hardware and software are fully-functional. This attention to self-testing continues with background checks running during system operation. Through both hardware and software mechanisms the redundant channels are kept synchronized to less than 100 µsec. The dual channels run almost identical software. The core software is a multi-level cyclic executive operating system. Laying out the task schedule for the cyclic executive is a tedious process. Both the control and logic designers, and the RTES group are responsible for the executive schedule. It falls on the RTES engineers to ensure that the final schedule does not violate any system timing constraints. Each channel performs run-time checks on meeting critical timing constraints and a channel may be reset mid-flight if a timing constraint is violated. I spent time discussing with the RTES engineers how an executive schedule is created. The process exists in the heads of a few engineers and is not documented in any standard practice document. It is a manual process that is rather tedious. The current approach uses a manual check that all timing constraints have been met which is error-prone. Once a working schedule is made changes are done as infrequently as possible for fear of the ripple effects in other parts of the schedule.

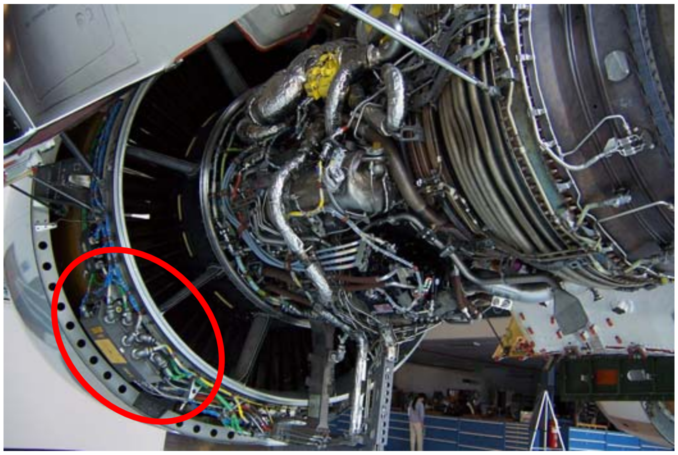

Photo 3 - PW4000 engine highlighting the FADEC

The last project I was given involved working with the F135 military engine for the F-35 Joint Strike Fighter, shown in Photo 4. Working on this project gave me a comparison between the commercial and military operations. The commercial side has better defined processes and is more restrictive in what can be done with the designs. This is due to the requirements of FAA certification that the commercial engines must meet. The F135 engine has a system for automated reporting of performance data. Multiple lists of variable names define the elements of data to report and the location in the data stream where each data item should be placed. Errors in these lists are often found late in the development cycle which requires change requests to be issued. The F135 team wants an editor that will assist the list creator by validating entries against multiple data sources. Generating error-free lists will obviously speed-up the development process. I wrote a set of detailed requirements for the editor. My requirements carefully detailed the exact validations that could be performed on the list from each source of information about the engine performance variables.

Photo 4 - F-35 Joint Strike Fighter Prototype

The F135 team was also interested in moving from the current format for the data reporting lists to an XML-based format that would combine the multiple list files into a single file. I defined an XML schema that incorporated all the data reporting information into a single file. So that they could migrate to an XML-based operation, I developed a conversion script that converted a set of files in their current format to the newly defined XML format. This script is immediately useful to the team even if they do not move to an XML-based list format. As part of processing the set of files, the conversion script performs all the validation checks that are possible based on the data within the files. This covers approximately 80% of the validation checks defined in my software requirements document. Unfortunately, some of the most troublesome errors remain in the other 20%. If the team decides to switch to XML definitions of the reporting data there still is a problem in that the downstream tools are expecting lists in the original format. To handle this problem, I wrote an XSLT stylesheet for translating the XML-based list into multiple lists in the current format. My demonstration of the conversion script uncovered errors in the sets of files that I was given as input data. No one was surprised by these findings. When I started with error-free lists in the current format, I demonstrated that conversion to the XML format followed by stylesheet translation back to the original format yielded lists that were identical to the original. The team was quite happy with the conversion tools that I wrote.

5. Continuing Collaborations

There are potential areas where I may be able to have further collaborations with the groups I visited during my one year of professional development leave. The SE department is interested in expanding the size of its Industrial Advisory Board. The faculty decided that we should invite Al Taylor from the FDA and Nanette Savage from Pratt & Whitney to join our IAB. I extended the invitations and am waiting for their responses.

The FDA is still quite interested in having a MISRA C validation tool. There is an SE Honors student who began work on this project as his SE Honors Independent Study project. I doubt that he will be able to develop a full validation suite in the time he has for Honors Independent Study. I will suggest to the FDA that they fund a summer co-op for one student and some summer support for me to finish the project during summer 2007. While I was there, we had discussed potential projects to fund. The MISRA C validation tool was one of them. Another project I discussed was to perform a set of tests on commercial static checking tools. The FDA is looking at several static checking tools and thought that an academic institution could provide an independent test and comparison of them.

There is certainly the potential to get senior project proposals. The NASA R2D2C implementation needs to be re-engineered from the ground up. Dr. Hinchey thought that this would be an appropriate senior project. I will pursue that with him when we start soliciting projects for the upcoming batch of seniors. Pratt & Whitney was interested in the potential of senior projects but between the military requirements and the export controls on many of their commercial projects finding something that can be given to a group of "uncontrolled" students may prove to be rather difficult.

I am working with my NASA collaborators on a paper that describes incorporating safety monitors into the R2D2C framework. When we finish the paper, we will submit it to an appropriate conference for presentation and publication.

There are several people that I worked with during my leave who I would like to invite to RIT to give a colloquium talk. A talk on regulation of medical devices containing software would be an interesting topic. There is also someone in the DESE group at the FDA who is an expert in general product safety engineering. This topic would have broad interest across GCCIS, KGCOE and CAST. Dr. Hinchey expressed a willingness to come to RIT to describe NASA's R2D2C system and other uses of formal methods within NASA. Nanette Savage, from Pratt & Whitney, is very interested in encouraging young women to pursue careers in technical fields. I would like to invite her to speak to either the Society of Women Engineers or the Women in Computing group on campus or both.

Finally, at all three locations I now have contacts for students for potential co-op and full-time positions.

6. Closing comments

I am very happy that I was given the opportunity to spend the past academic year on professional development leave. It allowed me to explore the domain of safety-critical systems in a way that I would not have been able to do with a normal course load. Being out of the Rochester area for the entire year was difficult but that was the best way to immerse myself in the activities of each of the organizations I visited. I have come back with many ideas for discussions and assignments in my classes along with possibilities for ongoing collaborative scholarship activities. Both my teaching and scholarship have benefited from my year on professional development leave.