Course development: Software Engineering in Robotics

Henrik I Christensen, RIM@GT, Georgia Tech, Atlanta, Ga 30332-0280

Introduction

The present project was setup to provide course material for teaching a one-semester course on software engineering as applied to robotics. The target audience was senior undergraduate students or junior graduate students. The assumption was that the students already had significant programming experience. The course was developed over the summer 2010 and ran for the first time during the fall term 2010. In this report we briefly summarize the course, the implementation and some of the issues related to the course.

Course outline/objective

The objective of the course was to provide the students with a general introduction to software engineering and particular issues related to robotics. The course includes 16 lectures and a major project to be implemented by groups of students to get hands-on experience with the design of a system. The experimental platform used for the programming was the Microsoft Robotics Developer Studio (RDS v3). The course covered topics related to

- What is software engineering? Standard processes?
- Introduction to standard tools for software engineering
- Introduction to Microsoft Robotics Developer Studio
- Design Methodologies
- Use of simulation as part of the design process
- Standard Robot Architecture and Models for Services/Modules
- Evaluation and Testing of Systems

The topics are covered in 15 lectures that cover the topics in some depth. The lectures included a mixture of basic theory, examples of use with a live session on a laptop and discussions to engage the students.

The final set of lectures included

1. Introduction / Welcome
   In this lecture an overview of the course is provided and the administrative details are handled.

2. Internet to Robotics – Motivation/Background
   A broad introduction to robotics and a motivation for why this is an area of growth is provided. The presentation is a broad brush coverage of the entire field.

3. Software Engineering Tools
   This lecture covers standard tools such as development models, UML, process diagrams, etc to have standard models for specification, design, analysis and documentation of systems. The presentation is generic and covers standard tools in software engineering

4. Introduction to Robotics Developer Studio
   A brief introduction to RDS is provided to give the student a sense for the different tools available within the package. The lecture provides a little background, covers, DSS, CCR, VPL, VSE, … to comprehensive but with little depth. Uses a few examples to show to the students how comprehensive the package is.

5. Discussion of projects for the course
   Students, in groups of 3, are required to design a small navigation system with a hardware interface, basic control, planning and localization using the Surveyor SRV-1 robot platform. The basic design for the project is defined and discussed.

6. Use of simulation for system development
   The value of using simulation as part of system development is discussed. The models used in the Visual Simulation Environment are also presented.

7. Process / System Models
   Standard process models are presented. Both reference models as state transition models, process abstractions, and reference models for process implementations. The support of such models within RDS is then presented.

8. Inter Process Communications
   Standard inter-process communication mechanisms are presented. The UNIX V IPC mechanisms are used as an introduction, the different uses of IPC is discussed. Finally, the implementation of the direct IPC mechanisms within CCR is presented.

9. Data Flow Models
   In this lecture the different mechanisms for controlling dataflow in an application are presented. The appropriate use of each of the different design patterns if discussed and it is presented how the design patterns are implemented in DSS/CCR.

10. Common Control Structures
    The lecture covers a range of different models for design and integration of services from pure data driven control over multi-stream integration to fully hybrid deliberative control of the priorities within a service. The implementation of different control mechanisms within DSS is also covered.

11. Hardware Integration Issues
    In this lecture the design and implementation of a hardware abstraction layer is presented. The integration of concrete drivers within RDS is also presented. This is probably one of the most challenging lectures in the course

12. Reference Architectures
    Standard architectures used in robotics are presented. This includes sense-plan-act, subsumption, behavior fusion, hybrid deliberative, and arbitration mechanisms. The lecture primarily attempts to provide broad coverage of the different solutions

13. Project status presentations
    The students present their status on implementation of a system

14. Packaging and Deployment of Systems
    In this lecture tools and methods for source code revision management, distribution (packaging), internet repositories (sourceforge and codeplex), and automated mechanisms to simplify deployment such as DSSDeploy, Autoconf, etc. are presented to give a sense of the diverse set of tools available and to point out some of the typical pitfalls.

15. Integration of legacy code and use of libraries
    In most cases it does not make sense to build complete applications from scratch and there is a need to integrate / interface with legacy code. In some cases these libraries are not written in the same language and there is a need to consider handling of cross language issues such as the distinction between safe / unsafe code or managed / unmanaged code in the MS Common Language Runtime as used by most Microsoft Languages today and an integral part of .NET.

16. Systems Evaluation and Benchmarking
    Methods for benchmarking and evaluation are presented as a basis for having a thorough test methodology in place design of large software systems.

17. Outlook and Summary
    The course provides a comprehensive coverage of software engineering in robotics, but there are clearly many more issues to be considered. The lecture provides a brief recapitulation and discusses some of the open issues.

In addition there is a reflection on the style of the course to see what ought to be revised in future versions of the course.

18. Presentation of project results
    Each group is asked to write up a paper about their design and provide a presentation of the results from the project.

Course implementation

The course was offered as regular advanced course at Georgia Institute of Technology during the fall of 2010. The course included lectures Tuesday and Thursdays through out the term. The course material was primarily the developed powerpoint slides. The slides were made available to the students through the codeplex site – swrobotics.codeplex.com. In addition the students were encouraged to consult the book - "Professional Microsoft Robotics Developer Studio (ProMRDS)" by Kyle Johns and Trevor Taylor, Wrox Publishers. A few of the student chose to acquire the book, but most students relied on slides and on-line tutorial material from Microsoft.

For the exercises the students were required to build a navigation that can navigate within a closed space to detect its own position and drive to a prior defined location. The students are provided with an XML file that specifies the location of colored landmarks in the environment. For implementation of the system the students were provided access to a Surveyor SRV-1 robot as shown below

Figure 1. The Surveyor SRV-1 used for the course exercises.

The robot was not supported in the current version of RDS. A new driver was designed for the robot, which uses a standard differential drive robot abstraction and the camera is modeled as a standard webcam service. This enable easy re-use of existing software.

At the end of the course the students were required to document their design in a report and to provide evaluation data on the efficiency of using different types of data flow models for control across push, pull and subscription. The push model was by far the most effective for this relatively simple design. The pull model is about 10-15% slower. The surprise was that the subscription model has a really poor (300-500% slower) in comparison, and this was true across the groups, which was surprising.

The Surveyor Start Code and an example solution set are available on the swrobotics codeplex site. In addition the slides have been updated after an initial revision.

Lessons

At the end of the class there was a general discussion about the course in terms of structure, exercises, projects, … Some issues that will be considered for the next version of the class include:

1. Many of the students did not have C# experience and there is a need to spend a little more time covering the basics, even for students with a Java or C++ background
2. Rather than jumping directly to the project it would be valuable to have a series of lab exercises a long the lines
    a. Build a service that writes a simple text to the log
    b. Build two services that exchange simple messages
    c. Build a new simulation model to show its use prior to the project

    The learning curve was a challenge and even with a bit of demonstrations at the end of each lecture session there is a need to explore the complexity first hand using simple examples

3. The class textbook was optional. While there is a significant amount of information available on the internet and quite a few tutorials it is difficult for the students to directly explore this. It would be beneficial to have have a textbook as required readings as a basis for the course. The book “Professional Microsoft Robotics Developer Studio (ProMRDS)” is not exactly a book on software engineering, but it might be the best possible compromise as a book that covers many of the challenges experienced by the students during the course.

4. The student were pleased with the coverage of the topics and the exposure to the material has enabled three of the participating M.Sc. students to land jobs with Amazon, MathWorks and Bloomberg, so they were very pleased with the skillset acquired during the course. While none of them will be working directly with robotics their main takeaway was the need to think about systems, clearly defined interfaces, and methods to address heterogeneous systems.
5. The platform used in the course – Surveyor SRV-1 has a number of challenges
    a. The wireless network can be VERY slow to generate a connection
    b. The chargers for the robot literally burn-out after a few months of use
    c. The batteries on the platform all had poor performance at the end of the course. The friction on a carpeted surface is too high and that ruins the batteries over time.

    So fairly standard hardware challenges to be expected in a robotics class!

Summary

The course ran for the first time and a significant investment has been made in the design of slides for the full course. The slides were on purpose designed to have a generic design to allow easy adoption by others. So far the number of downloads from the codeplex is at best modest. A few tweets will be generated to make people aware of the material. Associated software for use of the Surveyor SRV-1 as a platform is also on the web site and an example solution to the navigation problem is also presented.

The course will run again during fall 2011. The last time there was little time to get the course setup as a regular course and it was “numbered” / designated a “first off” course which some students shy away from. It is anticipated that 30 students will sign up the next time the course is given.