

Project #2: Term Project

This project is due on Friday, May 5. It will count for 25% of your overall grade.

Working in teams of three people each, you will extend, create or apply one or more tools for the Semantic Web. You may choose your own teams, as long as the teams are not identical to the teams from your ontology design project (at least one person must be different). Your team will get to propose what kind of project they want to do, but keep in mind that since you have three people working for a month, it is expected that the project will be something substantial. A rough guideline is that the project should result in between 2000 and 4000 lines of your own code, although quality is more important than quantity. You are encouraged to choose projects that dovetail with your own interests and abilities.

Three kinds of projects are acceptable. The first is to design a general-purpose tool that could be used to support a major capability or need of the Semantic Web. Good examples include a tool to extract information from the Web, a more user-friendly tool to annotate pages with Semantic Web information, a reasoner, an ontology library system, or an information integration tool. The second kind of project is to extend an existing tool with an important new functionality. For example, you could create new plugins for Protégé or improve upon the HAWK toolkit created by Lehigh's Semantic Web and Agent Technology lab. However, keep in mind that the new functionality must result in significant new code (2000 to 4000 lines). The third kind of project is to take existing tools and use them to develop an interesting application. In such cases the software development could involve creating a means to convert large amounts of real-world data into Semantic Web format and/or customized query interfaces. There may also be some amount of code that ties various tools together in a novel way. One example is to continue with the Semantic Web-based education theme of Project #1. Such a project could create tools to make it easier to define courses, tools for students to automatically construct trails to learn about specific topics, and tools that would generate HTML pages from the Semantic Web markup, thus eliminating the need to duplicate effort.

Important Dates:

Project proposal	Tue., Apr. 4
Progress report	Fri., Apr. 21
Project due	Fri., May 5

Project Proposal:

Your project proposal will serve as your project description. Therefore, it should describe what you intend to do in as much detail as possible. You should clearly describe the functionality of your system, the major components and rough estimates of the level of effort required to develop them, the programming language(s) you will use, and any supporting software you will use (e.g., Protégé, Jena, etc.). Note, you are encouraged to program in Java, however you may use other languages as long as you provide binaries for either the MS Windows or Solaris operating systems. You should also include a schedule that describes what you intend to accomplish each week leading up to the due date. Your proposal should be at least two pages in length, but no more than five pages (single-spaced). You will be graded on the clarity of your proposal and the quality of the idea you are proposing. After receiving your proposal, I will meet with your group

to negotiate its scope. I will scale down projects that are too big, and scale up projects that are too small. Feel free to discuss potential ideas with me before the formal proposal.

Progress Report:

You are expected to work on the project every week from the time you turn in your proposal to the day it is due. As such, you are expected to have made significant progress by Friday, April 21. Your progress report will consist of a scheduled meeting between your team and me on that date. You should bring a short written description of your progress (in particular, you should compare your progress to the original schedule from your proposal). You will be expected to give a demo that shows the current state of your system (of course, it is expected that this demo will only have limited functionality). We will also discuss any challenges that your team has run into and potential solutions. You will be graded on the extent of your progress, although unexpected issues may serve as mitigating factors.

Final Project:

Your project should be submitted electronically in either zip or tar.gz compressed format. You will submit this using Blackboard. The submission should include:

- source code
- compiled version of the code
- basic setup instructions
- basic user instructions

You should also turn in hardcopies of the source code and instructions. Note, the setup instructions should describe the required platform (e.g., Windows XP, Sun Solaris, etc.), programming language (e.g., Java J2SE 5.0), and any special configuration steps that are needed to run your software. You should also include URLs for any external archive/library files that are needed. The user instructions need only describe how to start your software and use its major features. If your software is reasonably intuitive, then this can be very short, hopefully one page or less.

You are also required to give a **demonstration** of your software to me on Friday, May 5. The demo should be about half an hour long and can either be in my office or in one of the computer labs in Packard.

You will be graded on the extent to which your project meets the expectations set by your proposal, the functionality of the resulting system, and its usability. The readability of your code will also play a factor in your grade.

Grading:

Your grade on the project will be based on the following components:

Proposal	20%
Progress report	20%
Final project	50%
Individual grade	10%

The individual grade will be based on team evaluation forms where you will be asked to evaluate the contribution of each member of your team, including yourself.