COS 470/570

Project Description

Spring 2014

Assigned: 1/28/2014 Due: last class

Proposal due: 2/13/2014

COS 470/570 includes a semester project that will allow you to tie together much of what you learn during the semester. You may work on this project by yourself or in a group with a partner. The group members must both be COS 470 (undergraduate) students or both be COS 570 (graduate) students.

Selecting a Project

You may choose from one of the suggested projects at the end of this document, or you may pick a topic of your own. If you want to do a project on a topic different from one of the suggested ones, you *must* check with me before writing your project proposal so I can give you some guidance.

Keep in mind the following when you select your project:

- You will be stuck with this project for a semester, so pick something that you are interested in.
- The best way to start is to decide on a particular area of AI or a particular domain in which you are interested. Example areas are things like constraint satisfaction, planning, and natural language. A *domain* is what the AI techniques you are working on are to be applied to—the area from which the problem you are solving is drawn. Example domains include controlling autonomous underwater vehicles, controlling a Mars rover, medical diagnosis, or factory scheduling. The two are related. If you are interested in a particular AI area, then the next thing you would do is to decide on the domain in which you'd implement the AI technique(s). If you are interested in a domain, then the next thing you would do would be to decide which technique(s) you would use for your problem.
- Think big! Think big! If you are interested in cooperative systems or interface agents, you might come up with a project to create agents that can be used to schedule meetings for co-workers. You probably won't have a complete agent at the end of the semester, but you may have quite a few of the modules completed. It will become clear to you what should be left out as the semester progresses. The only thing that will affect your grade is the amount and quality of your effort, not the percentage of the proposed project that got completed. (Agreements for project components, however, *will* be used to grade that component. So, you should make sure that the work is done or that the agreement (as defined in your proposal or elsewhere) has been changed.)
- "Double-dipping" is not allowed: that is, you can't get paid or get other class credit for work you are doing in this class. However, students are encouraged to work on projects that are related to other projects of interest (especially senior projects or thesis projects). There is nothing wrong with working on two aspects of a larger project, one as your project in this class, another for your thesis or another project, or working on a project in this class that might form the nucleus for a current or future project.

- If you are considering a group project, form the group either because of common interest in a topic or because you get along with the other person. Keep in mind that you will be spending a large amount of time with that person!
- As you work on your project, make sure that *you are actually doing AI.* Do not get bogged down in interfaces, etc., that have nothing to do with the AI in your project. This is a common problem, and if you are at all unsure, talk to me as early as possible. I have seen some truly impressive projects that got poor grades because, impressive as they were, there was little or no AI in them.

You are free to look for ways your project relates to other assigned work in the class. For example, if your project is on theorem proving, you might be able to use the resolution theorem proving assignment as part of your overall project, thus saving you some work.

Project Components

Project proposal. This should be a very short (1 or 2 pages) description of the project you are planning to do. You should include whether or not the project is a design or an implementation project (see below). If this is a group project, please include the names of the people who are working on the project. The project proposal is due electronically 2/13.

Written portion. You will provide a write up, similar to those for the programming assignments, describing your project. An ideal format for this is that of a scientific paper: introduction, background, methods, results, discussion, and bibliography. (And yes, you *should* have a bibliography for most projects.) This will be turned in electronically by midnight on the day of the last class.

Presentation/demonstration. I will give you additional information about this as the time gets closer. I would like to have at least a poster session, scheduled outside of class time, if not formal presentations.

Design Project or Implementation?

There are two kinds of projects you can do, design and implementation. You will do some design, of course, for implementation projects, and you mayq do some implementation in the design projects. The difference is one of emphasis.

Design projects are free to be larger that you could actually implement in a semester. For example, you might choose to design a Web interface agent that learns your interests and looks for Web pages for you. You probably wouldn't be able to create such an agent, at least not a good one, within a semester, but you could certainly do this as a design project. You might include for the reasoning component of the project something to do with search, planning, or theorem proving; you would implement a small piece of the bigger project. However, you might not be able to implement much more. An implementation project, on the other hand, should be small enough to completely implement in a semester.

I encourage you to make your projects as ambitious as possible. What you will be graded on is not so much whether you were able to solve the problem you selected, but rather your design, your implementation components, and the quality of your work overall. If you find that you have selected an implementation project that is too much to get done, don't worry: talk to me about it, and we can probably switch it to a design project, even relatively late in the semester.

Grading

You will be graded on the design and, if applicable, the implementation of your project, on your write-up, and on your presentation (again, if applicable). Writing quality counts! I reserve the right to refuse to accept projects that are poorly written, and/or to send you to the Writing Center for help with revisions.

Note to COS 570 students: You will be expected to take on a **substantially** larger project than COS 470 students, and you will be held to a higher standard during grading.

Possible Projects

You should pick a project topic that is of interest to you. Below, I provide some possible project ideas, but I encourage you to come up with one of your own.

Planning and Control

Robot Control You have access to our little Scribbler/IPRE robots, and you will have occasion to use them during the semester. You may also use them for your final project. Example projects using the robots include, but are not limited to:

- implementing a high-level planner to control the robot
- implementing a behavior-based controller (e.g., the subsumption architecture) for the robot
- using the vision system for robot navigation or for other vision applications

In addition to the Scribbler/IPRE robots, we have a number of larger, more capable robots in our lab, MaineSAIL.[1] These Videre Erratic robots have stereo color vision and an on-board Linux system so that they are truly autonomous; they communicate with the world via Wi-Fi. Software is included for some behavior-based control.

Simulators for these, together called Player/Stage[2], are available. You may do a project focused on controlling these robots, test it out in the simulator, and then possibly get to use the actual robots.

Possibilities for projects related to the robots include those focused on sensors (computer vision), on low-level control (e.g., behaviors and path planning), and on high-level (mission) control (e.g., planning and acting).

If you are interested in pursuing something along thee lines, see me for more information and to see the robots.

Underwater Robot (AUV) Control One of the research areas our lab, MaineSAIL, is intelligent control of autonomous systems. Our primary domain is autonomous underwater vehicles (AUVs), which are small autonomous submarines of use to industry, ocean science, and the military.[3] We do not have AUVs locally, but we do have two small land robots, and we have access to several AUV simulation testbeds.

One project possibility is to extend our AUV mission controller, Orca. A *mission controller* is an AI program that accepts goals and other information from the robot's user, then decides how best to

[1] The Maine Software Agents and Artificial Intelligence Laboratory

[2] `http://playerstage.sourceforge.net`

[3] See `http://MaineSAIL.umcs.maine.edu` for general information about our projects and research group, and `http://MaineSAIL.umcs.maine.edu/AUV` for information about AUVs in particular.

carry out those goals. An ocean scientist might tell Orca: "Take temperature measurements at A, B, and C, take a photograph of the coral colony at D, and if you see anything unusual, photograph it, too."[4] Orca would have to decide things such as: in what order shall I take the temperature measurements, and should they be interleaved with my other tasks? how will I recognize the coral colony, and what would be the best location from which to photograph it? how to illuminate it? what does "unusual" mean in this context?

Several extensions to Orca are possible. You could work on extending the program's knowledge representation, for example, including adding to its plans (called *procedural schemas*) its knowledge about contexts the AUV might find itself in. Or you could work on designing a coherent knowledge representation scheme for Orca (it needs one!) that includes knowledge about the world, knowledge about the vehicle, uncertainty, fuzzy logic, etc. Or you could work on adding temporal reasoning to Orca, or allowing it to make better predictions about the results of taking actions. You could work on handling unanticipated events, or reasoning about context. You could add case-based reasoning to Orca to make it "learn" from its past experiences. The reasoning components of such a project might overlap with search, theorem proving, and planning.

Another possibility is to try to develop your own AUV mission controller. The reasoning components of such a project could overlap with the search assignment (e.g., plan paths to get from one assignment to another), theorem proving, and planning (to sequence the tasks and to determine how to carry them out).

For either of these projects, you could develop and test your work in a version of the simulator that is part of the basic agent assignment, in one of the AUV simulators we have available, or in the simulator for our land robots.

Automated House A staple of science fiction is a house that is intelligent and can take care of its occupants. For this project, you would design such an agent, possibly implemented some portions and testing them in simulation. You would have to decide what you want the house to be able to do—e.g., observe the occupants to decide when to turn lights on/off, turn the heat up, fix breakfast, etc.—and then decide how you would get it to do it.

Planning Technology There has been a spate of interest in the last 5–10 years on planning research. For this project, you could implement and/or extend a state-of-the-art planner and test it in some domain. If your design is good enough, it might even warrant entering it into the annual international planning competition.

Diagnosis

For this project, you would design and implement a diagnosis system for some domain. For example, if you know something about medicine (or are willing to fake it), you could do a medical diagnosis program, perhaps patterned on an existing one, such as MYCIN or INTERNIST/CADUCEUS. If not, you might pick something else you know about as the domain, such as diagnosing problems with cars, etc.

[4]This is just an illustrative example. Natural language cannot be used except in very restricted cases at the moment, and Orca currently cannot handle a mission as complex as this.

Scientific Discovery

For this project, you will design and/or implement a system that can discover scientific laws, formulate and/or test theories, design experiments, or interpret experimental data. Work has been done in AI on this, including programs that have "discovered" Kepler's laws and several theorems of number theory, designed molecular biology experiments, and interpreted protein folding and spectroscopic data.

Natural Language Processing and Interagent Communication

Interface for a Robot Mission Controller For this project, you would design and/or implement a natural language interface for a robot mission controller, such as for an AUV or Mars rover. You would have to decide what kinds of things would be said, how to represent them, how to control the conversation, etc.

Conversational System For this project, you would try to build a system that can carry on a natural-seeming conversation with a human. To give you an idea of how difficult this is to do completely, if you were to build one that could fool a human into believing it was human, you could win $100,000 (the Loebner prize[5]).

Software Agents

Intelligent Spam Filter The best current spam filters are based on Bayesian (probabilistic) AI techniques. But can we do better? Perhaps by using additional knowledge about the world, or by using natural language processing to actually *read* the e-mail, an agent could tell the difference between, e.g., spam and your boss sending e-mail about spam. For this project, you would design and/or implement such an agent.

Softbot: Buying Agent For this project, you would design and/or build a "software robot" ("softbot") that would function as your agent to buy something on the Web. For example, you might get the softbot to buy a new computer for you. You would need to determine what issues you'd have to address (security and trust come immediately to mind—and I wouldn't expect you to actually field such an agent and test it buy giving it your American Express number!).

Chatterbot You could do the conversational system, above, but embed it in an agent that is on the Web or that participates in an on-line chatroom or game.

Avatar You could build an agent that plays in an interactive game, or that participates in an on-line virtual community or game, such as Second Life or World of Warcraft. (Note: you first have to make sure that such an automated agent is allowed in the system you are planning to work with.)

Constraint Satisfaction

Task assignment For this project, you would use constraint satisfaction programming (CSP) techniques to assign tasks to agents in a multiagent system (MAS). The MAS in question would

[5] `http://www.loebner.net/Prizef/loebner-prize.html`

be an autonomous oceanographic sampling network (AOSN) being controlled by the techniques developed in our CoDA project[6].

Scheduling classes For this project, you would "solve" the graduation problem. Your program, given knowledge about classes, prerequisites, class times, etc., would plan out a multi-year schedule for a student that would allow him or her to graduate.

Creativity

There are many AI programs that have been written to create poetry, novels, paintings, and so forth. You could attempt something in one of these areas as well.

Game Playing

For this project, you would pick a non-trivial game and write a program to play it. Chess, Go, backgammon, poker, Risk, Monopoly—the list goes on.

You could also write a program to play an action game, too. Writing an AI program to play a video game may be tough and beyond your capabilities for a semester-long course, but text-based adventure games (or hybrids, like Nethack) are reasonable to consider.

You could also design and possibly implement an intelligent agent that "lives" in a video game or in an on-line game/world such as Second Life or World of Warcraft, as described above (and as allowed by the rules of the gaming platform). For example, you could write an agent that resides in a Unity or Unreal engine game world.

Security

Cybersecurity Possible projects in this area include:

- detecting attempted or successful attacks on a system or network
- automated response to attacks
- detection and neutralization of unknown viruses
- detection of unauthorized users (e.g., by their behavior)
- intelligent authentication of users (e.g., beyond simple question-asking)

Physical Security Possible projects in this area include:

- intelligent authentication for access to a location
- threat detection via surveillance and sensor networks
- threat (e.g., terrorist activity) detection through analysis of patterns of activity (e.g., cell phone networks of contacts, etc.)
- controlling robot guards and mobile surveillance platforms

[6] `http://MaineSAIL.umcs.maine.edu/CoDA`