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. Each group must contain either all COS 470 students or all COS 570 students.

Selecting a Project

You can choose from one of the suggested projects at the end of this document, or you can 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 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.
- Consider forming groups either because of common interest in a topic or because you get along with the people. Keep in mind that you will be spending a large amount of time with these people.
Knowledge representation is important for most projects, as discussed below. However, although most projects have obvious knowledge representation and reasoning components, and many others can be crafted to include them, not all possible projects do. For example, it would not be obvious how to come up with knowledge representation requirements or a design for a project on social ramifications of artificial intelligence. For such projects, talk to me before writing the project proposal!

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.

When you select a project and write your proposal (after consulting with me, if you have not selected one of the areas below), then I will critique your proposal as fast as I can and let you know if any of the programming assignments overlap sufficiently to warrant working on the two together. You should be thinking along these lines in your proposal.

Project Components

The project will have 5 components:

1. **Proposal.** This will consist of a general description of the project, including an outline of the goals and the project’s scope. This will be due a few weeks into the semester (10/27). The proposal is required, but will not contribute to the grade.

2. **Knowledge representation component.** Select the type of knowledge representation you will use, decide on the basics of the representation, represent a set of knowledge from your project’s domain in the representation scheme, and implement a mechanism for storing/retrieving the knowledge. The knowledge representation homework/programming assignment will tie into this. If you feel your project does not involve knowledge representation, talk to me about it.

3. **Reasoning component(s).** Except for the Lisp practice and basic agent assignments, the assignments have parts that can overlap with your project. For each assignment, there will be some parts that have to be done as written and other parts that you can, if appropriate, adapt to your project. This will be covered more thoroughly in the assignments.

   For example, if your project is to control a robot moving around in the world carrying out tasks, then the part of the problem solving assignment that asks you to choose another domain for the search algorithm might be done as a path planning problem for the robot. You can overlap any of the assignments with your project, but you must do at least one unless: (1) your project simply does not lend itself to any of the techniques, and (2) you have cleared it with me well in advance—ideally, when you turn in the proposal.

4. **Extension.** Extend the project beyond the programming assignments in class in some significant way.

5. **Presentation.** There will be a written and possibly an oral presentations (including, if applicable, a demonstration). If we do oral presentations, they will be scheduled during the last week of classes. A room will be reserved at a time outside of class in which the students can present their projects. The written portion of the presentation will be due the last day of classes. The written portion of the presentation must
include a section evaluating the work done in the project, either empirically (i.e., via experiments) or by some other means.

**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 will 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 could do 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 you 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, on your presentation (again, if applicable), and on how well you did on the programming assignments that were related to your project. 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**

Unless you come up with a project idea of your own—which I encourage—you should pick from among the following projects. If you do pick one of your own, then you must discuss it with me prior to writing your proposal.

**Planning and Control**

**Underwater Robot (AUV) Control** One of our research areas our lab, MaineSAIL\(^1\) 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.\(^2\) We do not have AUVs locally, but we do have two small land robots, and we have access to several AUV simulation testbeds.

---

1. The Maine Software Agents and Artificial Intelligence Laboratory

2. See [http://MaineSAIL.umcs.maine.edu](http://MaineSAIL.umcs.maine.edu) for general information about our projects and research group, and [http://MaineSAIL.umcs.maine.edu/AUV](http://MaineSAIL.umcs.maine.edu/AUV) for information about AUVs in particular.
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 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.” 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 robot 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.

You can also design and/or implement a controller for another kind of vehicle, for example, a land robot or a Mars rover, etc.

As an added incentive, the best mission controller project team will, after the work is tested in the robot simulator, be allowed to demonstrate their work on an actual land robot, if possible.

**Land Robot Control** MaineSAIL has available a number of land robots. Simulators for these, together called Player/Stage[^1], are available. You can 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 these lines, see me for more information and to view the 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.

[^1]: 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.

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.

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!).

\(^5\)http://www.loebner.net/Prizef/loebner-prize.html
**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 would 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 be an autonomous oceanographic sampling network (AOSN) being controlled by the techniques developed in our CoDA project.

**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.

**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. Video games are tough and almost certainly beyond your capabilities for a semester-long course, but text-based adventure games (or hybrids, like Nethack) are reasonable to consider.

---

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