Abstract

The Teaching Artificial Intelligence as a Laboratory Science (TAILS) project aims to teach Artificial Intelligence through an experimental approach modeled after lab sciences such as chemistry, biology, and physics. Rather than using traditional teaching mechanisms, the TAILS project presents coursework through a series of lab experiments which balance both theory and practice, while also promoting sound software engineering development practices such as defining and managing components for re-use.

Advantages

• Instills sound software engineering practices
• Encourages cooperation and teamwork among students
• Reinforces the social aspect of learning and scientific work
• Reinforces material learned in other courses
• Relates course material to real-world instances
• Appeals to various types of learners
• Challenges students at a variety of skill levels
• Aids in retaining women to the discipline

Assessment Tools

| Application Cards | Ability to identify real-world applications for the concepts encountered |
| Elevator Statements | Ability to communicate concept clearly without technical jargon |
| Lab Reports | Understanding of lab |
| Pre/Post Tests | Mastery of course material |

Motivation

Knowledge of existing algorithms is essential for the advancement of computer science. Students gain a deeper understanding of these algorithms when they are viewed from multiple perspectives and in the context of real-world applications. While many software code repositories are available to students, these are simply collections of programs that run. They provide little or no documentation to offer insight into how the code was designed, how a particular program works, or where such code would be used in an actual application.

A TAILS Lab

The structure of a TAILS lab is intended to model the artifacts of the software engineering paradigm. Each lab consists of nine components. The Idea describes the concept taught in the lab. A Sample Input/Process/Output provides the student with a black box view of the concept in action. The Implementation-Independent Design Description gives the student an abstract view of one implementation of the concept. In a closed lab setting, the student works with a partial implementation of the concept and a set of exercises or HINTs that outline the steps needed to complete the code. Each lab is accompanied by a Test Suite and Driver to test the success of the implementation. A set of Experiments accompanies each lab and suggests activities that demonstrate and/or extend the implementation and build upon the basic concept. Another component of the lab describes real-world Applications of the concept and is intended to support intuitive learners. Complexity Analysis reinforces lessons learned in data structures or algorithms courses. By providing the Source Code for the completed implementation of the concept the student is able to check their own work.

Proposed TAILS Architecture

[Diagram showing the client using a web browser to interact with a TAILS server through a web interface and a TAILS database]