

CSE293 Computer Science & Engineering Design Laboratory

Marklin Computer Controllable Model Trains Project

I have a very exciting project planned regarding real-time and computer control of an automated system, with both direct control out and feedback (via sensors) to allow you to query the "state" of the system. The real time control is for, of all things, model railroading trains. If you had them as a kid, you may not be aware that in the past 5 years or so, there has been a digital alternative to model trains, where there is digital control for every train (each train has an onboard computer), for switches, for lights, for feedback sensors, etc. You can find out more on digital trains by visiting the Marklin train site at: <http://www.marklin.com/> and <http://www.marklin.com/digital/>

During Spring 2001, I offered CSE293 with projects related to computer controllable model trains, specifically, Marklin computer controllable model train equipment. This is state of the art equipment, where all of the various components, trains, turnouts, switches, signals, etc., are computerized, and more important, computer controllable from a laptop or a PC. In addition, it is possible to place sensors on the track to obtain real-time feedback on the location and status of the different trains. I am interested in having up to three groups of three students each to work on projects this semester for CSE 293. The computer controllable train equipment is essentially a "real-time" system, where they can both control and get feedback on the status of all computer-accessible components (trains, switches, sensors, signals, etc.) in the system. The equipment comes with an interface hardware component that will take commands through an RS232 interface from a software application, and also feed back sensor data through this same link to a software application. Previous projects included a system with the following capabilities:

- Development of a graphical user interface, written in Java, for controlling multiple trains operating at different speeds on the same track at the same time. In addition, this will also control switches and light and sound. The current location of each train is graphically displayed on a PC. This program runs on a PC that is connected to the computer controllable train equipment.
- Development of a graphical user interface, written in Java, for monitoring sensors and signals placed throughout the track. The sensors give the position of the trains. The signals control when a train must stop and/or go. The current location of each train is graphically displayed on a PC.
- Design and development of a track planning and management tool that can be utilized to lay out track, switches, signals, sensors, etc. As an example of track layout and planning, visit the Atlas railroad site at <http://www.atlasrr.com/> where you can then link and visit their site <http://www.atlasrr.com/software/welcome.asp> to download software for layout and planning.
- Design and development of actual computer hardware for a UPC scanner that can actually scan a UPC code under a train and know which train and which cars are passing. This will require the students to both design a hardware device and to interface that device via the model train controller equipment to the PC. Students for this project will need to have a strong background in hardware.
- Other topics as proposed by students are also possible. For example, one possible project would be a collision detection algorithm and control program. This would

CSE293 Computer Science & Engineering Design Laboratory

Marklin Computer Controllable Model Trains Project

utilize the initial positions of the trains, their direction, and speed, along with real-time input from the sensors to determine if a collision is imminent. Then, it could be avoided by either commands to the trains or commands to the signal(s) to stop the trains.

The intent of the major design experience is to have you work on a real world project. Thus, as part of this course, you will be required to consider the following realistic constraints and their impact and effect on your project: economic, environmental, sustainability, manufacturability, ethical, health & safety, social, and political. Note that not all constraints will apply to all projects. As an example of constraints, for the UPC project, the issues may involve if it can be built at a reasonable cost (economic), and once built, does its manufacture allow it to work under the track and provide correct data at a high rate (99+ percent) - which is sustainability. In a software project, economic issues are related to the cost of the software, which you can try to estimate during the semester, and at the end of the semester, you can give an estimate of the retail cost.

Remember, all of the projects require you to utilize UML for design and documentation. In addition to our use of this equipment in the course, we also intend to utilize the projects as part of demonstrations at the various open houses that are held at UConn to recruit high-school juniors/seniors as college freshman. Note that rather than starting from scratch, it is possible to start with the code of the Spring 2001 offering of CSE293, hence, this offering of CSE293 will also be an exercise in software reuse and extensibility. Note that we have a dedicated lab set up and available for use on the first floor UTEB.

As a major design experience, you will be asked to help assess the way that this course meets the objectives within our CSE programs. Specifically, at periodic times during the semester you will be queried for input. This will include which courses were most useful and critical for this course, and for each of those courses, which topics were most relevant. At the end of the semester, you will be solicited for suggestions for improvement.