Teaching mechatronics using SCADA

Z. Nedić & A. Nafalski

University of South Australia
Adelaide, Australia

ABSTRACT: The University of South Australia (UniSA) has a long tradition of producing work-ready graduates. This requires keeping up to date with the technology used in industry. Because of the current mining boom in Australia, the electrical power industry is a major employer of graduates from the UniSA Bachelor of Engineering (Electrical and Mechatronic) programme. The electrical power industry relies heavily on remote monitoring and control of their transmission and distribution networks, and the associated equipment. There is a rapid introduction of modern technology in the form of Ethernet for integrating remote terminals and other remote units into modern Supervisory Control and Data Acquisition (SCADA) systems. Although SCADA traditionally is utilised for the monitoring and control of industrial processes, it is quickly finding its way into managing other scientific applications and even office processes. This prompted the UniSA to introduce SCADA into their curricula. In particular, this is so for the course Mechatronic System Integration, where students also are exposed to other modern industry-related topics.

INTRODUCTION

The University of South Australia regularly revises its engineering programmes in order to keep up with the newest technology developments relevant to electrical and mechanical industrial applications. This ensures the engineering graduates are *industry-ready* and, as such, have a better opportunity to succeed in gaining their first employment. From last year (2012), the UniSA introduced a course, Mechatronic System Integration, which is common for electrical and mechanical students specialising in mechatronics. Actually, this course is a common course for three UniSA programmes: Bachelor of Engineering (Electrical and Mechatronic), Bachelor of Engineering (Mechanical and Mechatronic), Bachelor of Engineering (Mechatronic) and also for their double degree versions.

The course focuses on teaching students fundamental components of mechatronic systems, such as sensors, actuators, programmable logic controllers (PLCs), and their integration into a Supervisory Control and Data Acquisition (SCADA) system. The delivery of the course is structured around a small lecture component (6 x 2 hours) and much larger practical component (13 x 3 hours). Students also spend additional time in a specialised mechatronic laboratory working on individual and group assignment tasks. The main component of the laboratory work assessment is programming a PLC to control a model of a production line developed by a company - Festo - and includes three main stations: distribution, testing and sorting [1].

WHY TEACH SCADA?

Nowadays, most production processes are computerised and use applications for monitoring and controlling these processes based on SCADA software available in the marketplace. What is becoming even more important is to collect data from these processes, which, then, can be used to optimise and improve the production process to reduce costs and maximise profit. Modern SCADA software supports the creation and management of databases capable of storing large amount of data, such as detailed logs of alarms, number of units produced, number of units rejected by quality inspections and types of fault.

Traditionally, SCADA applications were developed by SCADA specialists contracted by the software distributors and were commissioned as part of a system that typically would include hardware, as well as software. In most cases the companies who were the customers for these systems would have trained operators, but not developers. This would leave them very dependent on the contractors, who also had limited skills. The contractors were inclined to use different SCADA software for different brands of PLCs, e.g. RSView for Allen Bradley PLCs; WinCC for Siemens PLCs. As a consequence, in some cases one company would have more than one SCADA system to support several isolated processes. This would complicate maintenance and make upgrades very difficult.

Proprietors of SCADA software compete for customers and increase the capabilities of their software to make them applicable to wider markets. Nowadays, most SCADA software provides an interface to remote software and hardware via the Ethernet and/or Internet. It also supports a number of different communication protocols at different communication levels. Thus, as the technology develops, there is a growing need for engineering graduates capable of developing SCADA applications.

Research shows a number of universities teach SCADA as part of their engineering curricula, although this number appears to be very small. The authors believe the reason may be a traditional association of SCADA training with professionals at a trade level rather than at an engineering level. However, it needs to be clarified that paraprofessionals usually are trained to work as operators and not application developers.

As SCADA systems are becoming widely used - not only in industrial processes, but also for monitoring and control of scientific and office processes - it is predicted that in the very near future there will be a shortage of engineers with these skills. To address this problem, the UniSA introduced the course, Mechatronics System Integration, where students learn how to integrate industrial process components into a SCADA system.

THE PRACTICAL SCADA COMPONENT

During the 13-week delivery of the course, students have two SCADA computer experiments each of three hours duration. In the first session, students familiarise themselves with the Wonderware InTouch 10.1 software [2] by company Invensys [3], the market leader in real-time operations management software. Students learn how to use the Application Manager, WindowMaker, WindowViewer, and how to create objects and wizards, and tags for objects and variables. They also grasp the concepts of smart symbols and animations, and how to communicate with the external world, through InTouch. At the end of this first session, students create a new application with a graphical model of a simple plant, as shown in Figure 1.

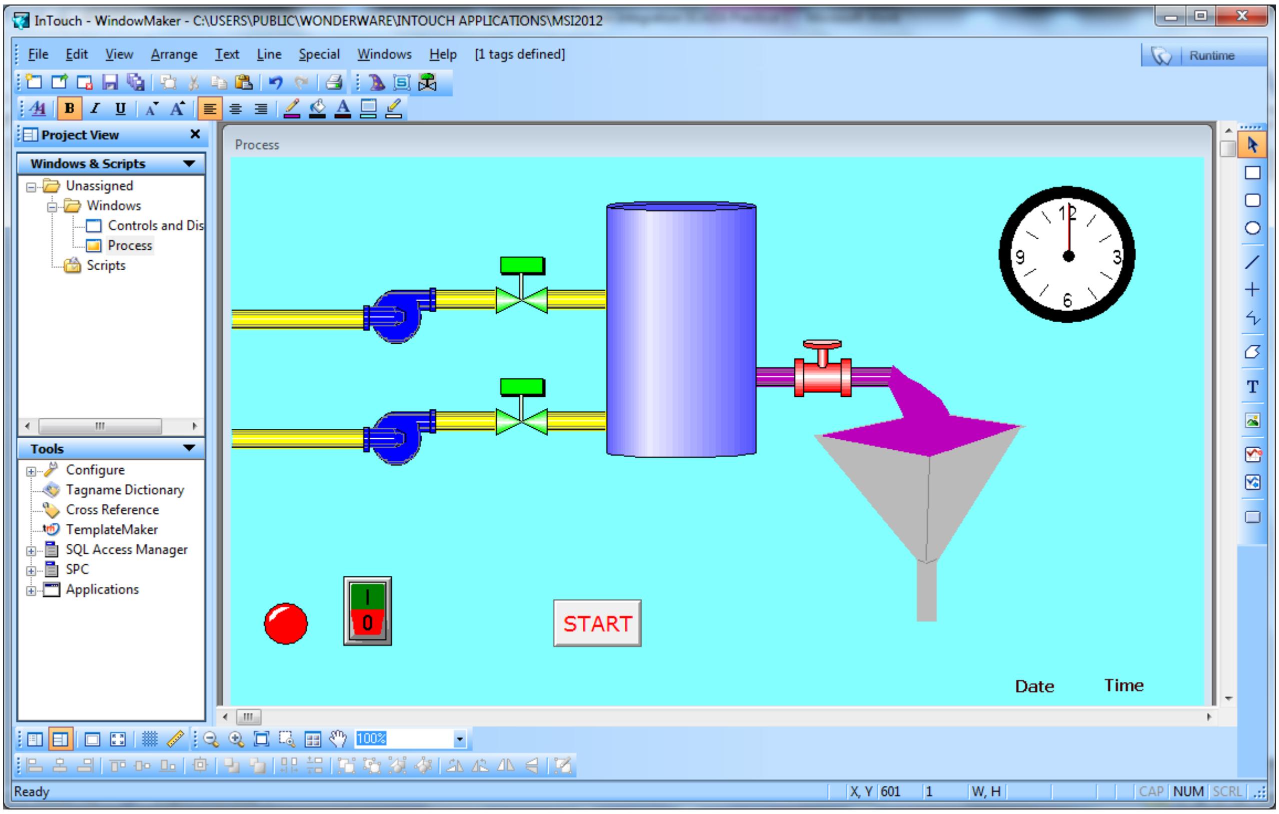


Figure 1: Graphical model of a simple plant.

Students also learn how to create animations. One such animation is shown in Figure 2, where an animated slider wizard is used to vary the volume of fluid in a tank, which, then, can be monitored on the vertical volume indicator.

In the second exercise, the students learn how to create scripts using the InTouch scripting programming language. For this, students need first to create new tags - equivalent to variables, but defined differently - for accessing data stored in internal memory as opposed to I/O tags, where information is accessed from external devices. Figure 3 shows how students create a new tag and also lists the different type of tag. Then, using scripts, such as the ones shown in Figure 4, students create programmes that run when they switch their applications to viewer mode. In this session, students also learn how to exchange data with other MS Windows applications and to create formal documentation with the required information.

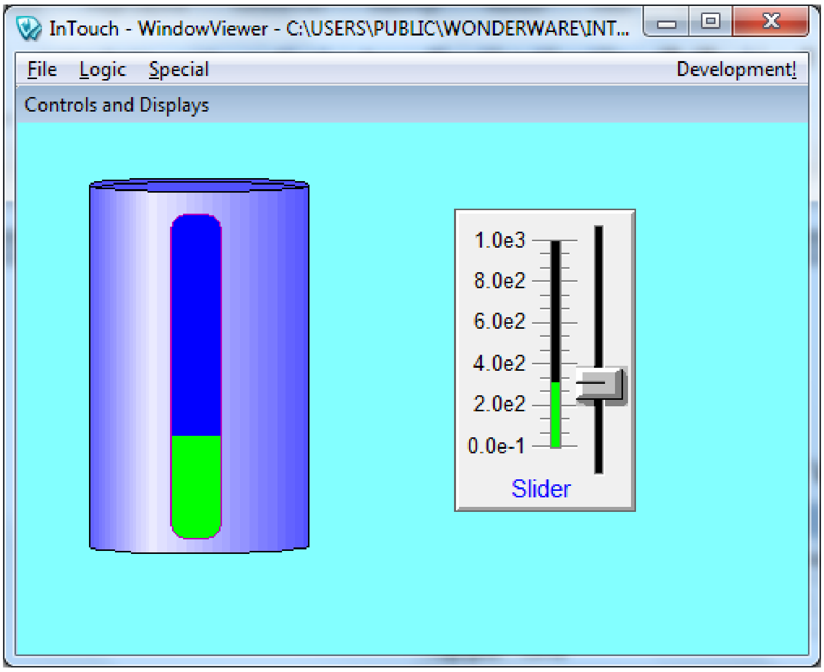
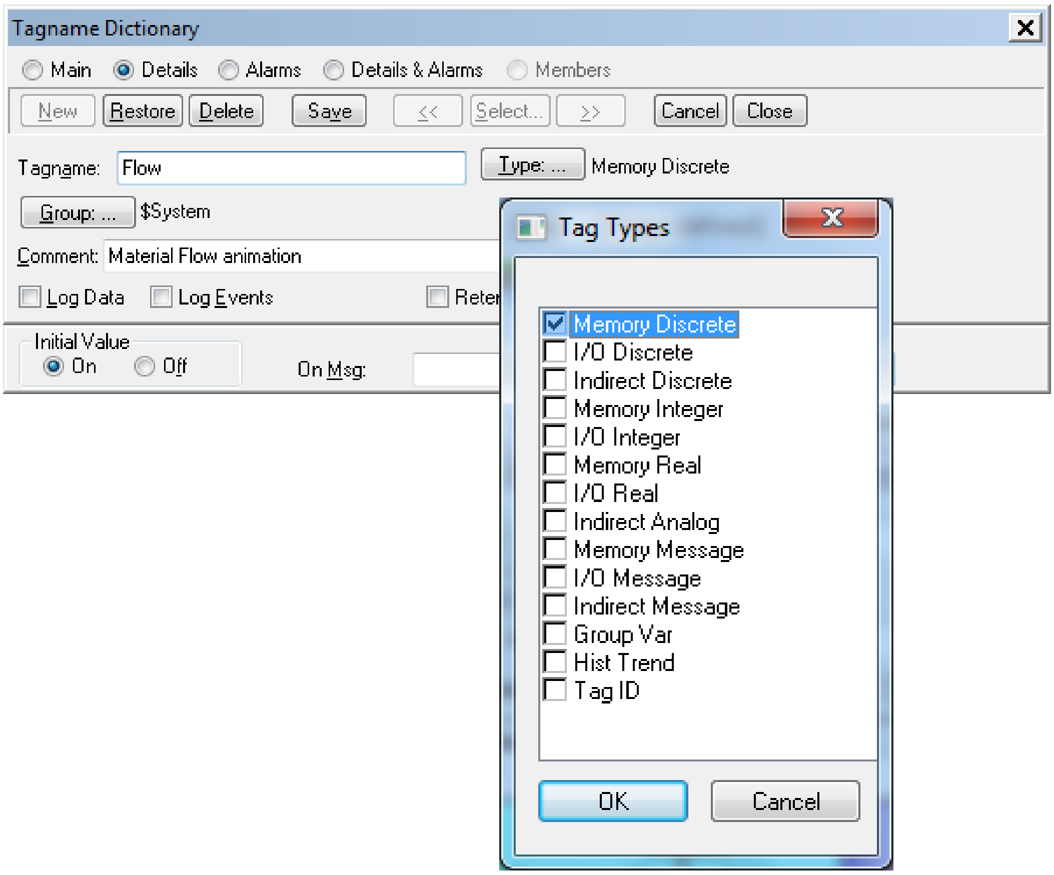


Figure 2: Vertical tank fill animation.



○ Main ● Details ○ Alarms ○ Details & Alarms ○ Members

Tagname: Flow

Type: Memory Discrete

Group: $System

Comment: Material Flow animation

□ Log Data □ Log Events □ Reten

Initial Value ● On ○ Off

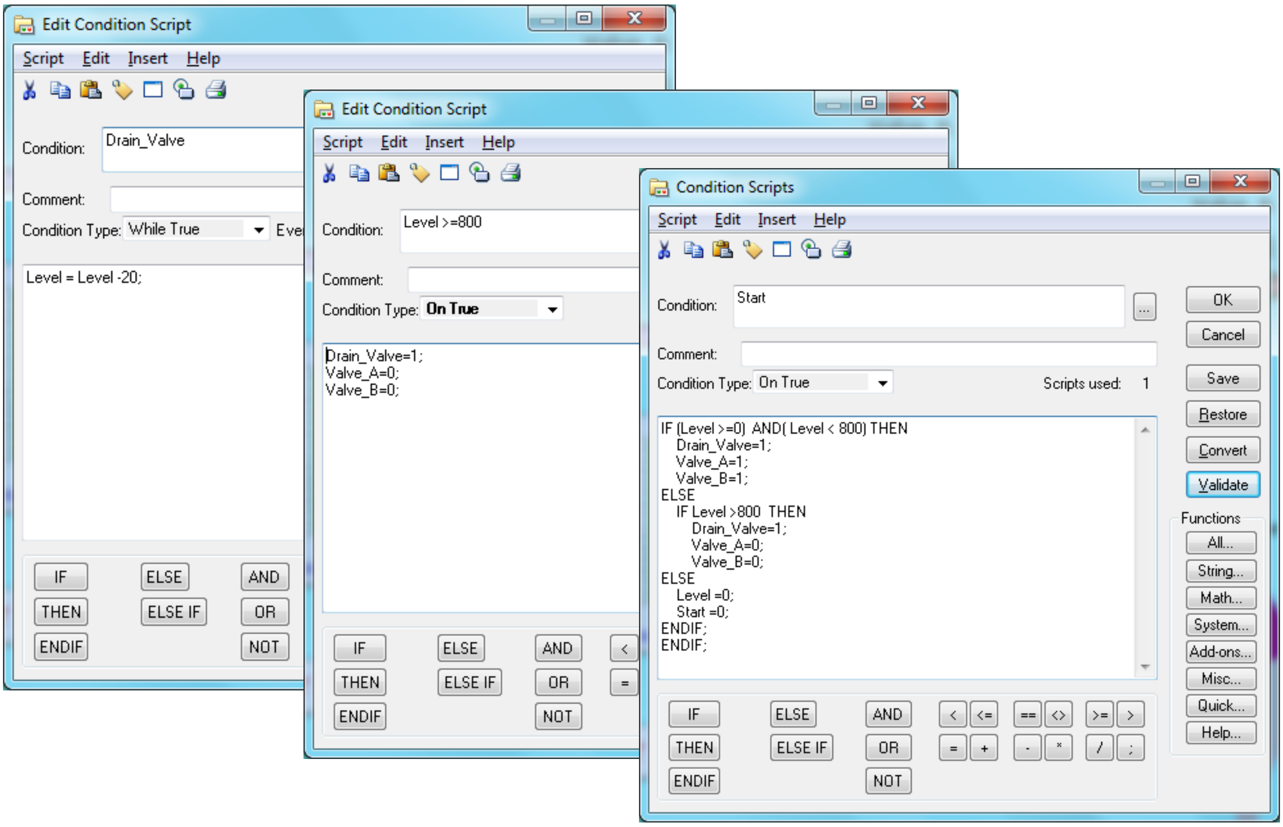
On Msg:

☑ Memory Discrete
□ I/O Discrete
□ Indirect Discrete
□ Memory Integer
□ I/O Integer
□ Memory Real
□ I/O Real
□ Indirect Analog
□ Memory Message
□ I/O Message
□ Indirect Message
□ Group Var
□ Hist Trend
□ Tag ID

Figure 3: Defining a new tag.

After this second session, students are required to complete an assignment, which contributes 20% to their total course mark. The assignment is an open-ended task in which they create an animated model of one or more Festo stations. Students are allowed to work on the assignment individually or in a group. For the group work, they are required to produce a model of a more complex plant, e.g. a model of all three Festo stations.

Students are required to demonstrate their work to the supervisor and to produce a short report that includes complete documentation produced from InTouch, including all windows, tags, scripts, etc. The assessment takes into account the complexity of the plant, the effectiveness of the demonstration and the completeness of the report, as well as the general appeal of the human machine interface (HMI).



Condition: Drain_Valve

Comment:

Condition Type: While True

```
Level = Level -20;
```

Condition: Level >=800

Comment:

Condition Type: On True

```
Drain_Valve=1;
Valve_A=0;
Valve_B=0;
```

Condition: Start

Comment:

Condition Type: On True

Scripts used: 1

```
IF (Level >=0)  AND( Level < 800) THEN
   Drain_Valve=1;
   Valve_A=1;
   Valve_B=1;
ELSE
   IF Level >800  THEN
      Drain_Valve=1;
      Valve_A=0;
      Valve_B=0;
ELSE
   Level =0;
   Start =0;
ENDIF;
ENDIF;
```

Figure 4: Samples of Wonderware InTouch scripts.

IN-CLASS OBSERVATIONS

Students were eager to learn more about SCADA than could be covered in two practical sessions. They did find the software to be unusual, especially the scripting language, compared to other programming languages covered in their studies, e.g. C, Java, MATLAB and LabVIEW. However, they found it much easier than PLC programming, which was covered in the same course. In general, after getting over initial obstacles during the software familiarisation phase, as with learning every other new piece of software, students enjoyed learning SCADA. Some students took the opportunity to be creative by including objects beyond what was necessary, but which made the work fun! Most of the students were able to grasp the new concepts and produce an original piece of work for their assignments.

CONCLUSIONS

The introduction of SCADA to undergraduate engineering programmes at UniSA was described in this paper. The first experiences from teaching SCADA in 2012 were very encouraging as students received the new material well and even expressed a wish to learn more. The decision already has been made to increase the percentage of the course allocated to teaching SCADA to cover more advanced topics.

At this stage the main challenge is the cost of the software licence, which limits the number of development stations and increases the group sizes in class. It also limits the amount of time students can spend working on their assignments. One solution to this is the creation of a remote laboratory. Interestingly, already there are requests from students to create a remote laboratory, similar to NetLab, the UniSA remote laboratory, which will allow students access at any time from anywhere [4].

REFERENCES

1. Festo. Festo - Start Page (2012), 17 December 2012, http://www.festo.com/net/startpage/
2. Invensys. Wonderware - Software Solutions for Real-Time Success^{TM} (2012), 17 December 2012, http://global.wonderware.com/EN/Pages/AboutWonderware.aspx
3. Invensys. Corporate Profile (2012), 17 December 2012, http://www.invensys.com/
4. Nedic, Z. and Machotka, J., Remote Laboratory NetLab for effective teaching of 1st year engineering students. *i-JOE Inter. J. of Online Engng.*, Kassel University Pres, 3, **3**, 6 (2007).