LEARN, PROGRAM AND TEST SEQUENTIAL FUNCTION CHARTS IN AN INTERACTIVE E-LEARNING COURSE

J.P. Keller

Institute of Automation, University of Applied Sciences, FHNW,
Riggenbachstr. 16, 4600 Olten, Switzerland
juerg.keller1@fhnw.ch

Abstract: Sequential function charts (SFC) according to IEC61131-3 are often not very popular in industrial automation. This is mainly due to the fact that it was not learned systematically or its capabilities were never experienced. The high-level interactive e-learning tool described in this contribution offers a didactically well structured step by step e-learning course, disclosing all the necessary details. A graphical SFC-editor allows the user to freely program his own SFC. The user is only restricted to predefined actions and transitions, but with the benefit that he or she can test acquired SFC with real-life plant simulations. Analysis of SFC based on Petri-Net theory and recipe control of batch plants complete the course.

Keywords: Sequential Function Chart; Supervisory Control; Interactive e-learning; User interfaces; Educational aids;

1. INTRODUCTION

Student acceptance of e-learning courses cannot be conjectured in advance. There are several indications (see also Schulmeister 2004) that the majority of talented students are not happy with the e-learning approach. Possible reasons are lack of high level interactivity and the absence of options to solve open and real-life problems. The word “Interactivity” is (ab)-used for simple user-interface navigation at one end and for virtual reality on the other end. Scales to classify interactivity were proposed by Guerra and Heffernan (2004) and Schulmeister (2005). They propose many levels to characterize the possibilities to manipulate the user interface and only the top-level includes all solutions, where the user can constructively contribute to the course and receive feedback on his work.

A top-level interactive system has to deal with problems whose solutions can be described in a formal language. Based on such a description the e-learning-tool can produce a feedback. Evidently, the richness of this language determines the quality of problems to be solved. A sequential function chart is a well defined formal representation of an automaton. If a SFC-editor is available in the e-learning course, the student can constructively contribute to the course, e.g. drawing his own SFC. To be able to provide a feedback, one solution is evident. The plant to be controlled can be simulated within the e-learning course. If the simulated plant is displayed on an user interface, the student has a sensible feedback for his work. The interface to the simulation plant defines the alphabet of the SFC, i.e. the names of the transitions and actions. This also completes the definition of the formal language.

Several contributions were published on the topic of how to design an efficient e-learning course (for example, Satow 2002). These publications indicate that a good instructional design is mandatory in creating a useful e-learning course. The course presented in this paper is designed to accomplish these theoretical requirements.

The e-learning tool is based on LabVIEW, National Instruments, which is particularly well suited for this purpose. Due to the availability of hardware interfaces, it offers the capabilities to easily replace the simulated plant with a real plant. The SFC-editor was presented in Keller (2001).

An on-line simulation of Petri-Net with plant simulation was proposed by Flochova et al (2003). In Donzellini et al (2003) state automata simulation is included in an integrated learning environment for electronics. Both are not e-learning courses design to teach application of automata theory.

In the first section of this paper the design of the e-learning course, called SFC-Trainer, is described. In the next section the learning steps are explained. Afterwards, the SFC-Editor and the simulation plants are presented. Practical experience and student feedback are summarised in the last section.

2. DESIGN OF THE SFC-TRAINER

The SFC-Trainer is a complete e-learning tool consisting of a presentation of theory as text and slides, tests and exercises. The main concepts are that the complete learning task is split into small learning steps and that higher cognitive levels (Taxonomies according to Bloom, 1976) are attained with illustrative and as much as possible real-life exercises. As a consequence the course is organised into lessons. For each lesson, the following information is presented in the top level user interface (see Figure 6):

- Topic: brief description of the subject
- Goal: the operational skills to be attained
- Lesson time: indicates required time to work through the lesson
- Mandatory: flag, indicating the importance of the lesson. This allows the student to adaptively select lessons according to his skills.
- Exercise: Description of the exercise including a picture of the simulation plant

For every lesson with theoretical input, theory is presented in two forms: figures with text unveil the information in a detailed form suitable for a more systematic learning style, whereas a slide show with figures and keywords provides the information in a ‘crash’-course manner, useful for skilled students or students with a priori knowledge. A collection of questions in the test section gives the student the possibility to check his acquired knowledge at a low cognitive level, i.e. knowledge and comprehension.

The key component of each lesson is the exercise. If the student was able to solve the exercise, he should have grasped the important aspects at a higher cognitive level, i.e. the level ‘application’ or even ‘synthesis’. For each exercise a solution is available. The solution can be explored in detail within the SFC-Editor or in conjunction with the simulation plant. Special aspects are elucidated in an explanation, available for each solution.

The user interface is shown in Figure 6. It is very compact and for every button, an online help can be displayed. At present the lessons are available in English and German.

3. LEARNING STEPS

The e-learning course contains 4 sections: Introduction, creating SFC, analysing SFC and recipe control.

3.1 Introduction

The goal of the introduction is activation of the student. Arguments as to why to use SFC are given and an outline of the course including figures of the real-life simulation plants is presented.

3.2 Programming an SFC

The learning task is split into nine lessons. It starts with the basic elements, e.g. steps, actions and transitions. As an exercise, a counter is controlled with a simple linear SFC. In addition to the simple step/transition sequence the appropriate action qualifiers have to be specified. In contrast to Petri-Nets, execution of actions is explicitly specified by means of qualifiers. Missing knowledge of its proper use is one the reasons why SFC cannot successfully be applied in industrial automation.

In the following lesson switching rules and execution of actions is explored more in detail. If two successive transitions are true, it is unclear whether the actions associated with the step between the two transitions are executed. The IEC61131-3 norm failed to specify this explicitly and as a result, execution of actions in SFC is dependent on the engineering tool. The two options are called ‘search for stability’ or ‘no search for stability’ (details see L’Her 1998). With the ‘search for stability’-option, the SFC determines iteratively new markings until no transition can be fired. No actions of intermediately marked steps are executed. With ‘no search for stability’ the SFC is executed in a flowchart like manner. Actions of intermediately marked steps are executed at least once. In this lesson the difference between the two execution options are explained and trained with the modified counter control example. The value of the counter clearly indicates whether a 'count down'-action was executed or not.

After practising simple SFC programming on a simple elevator example, the student has to learn how a sequence can be branched. First the ‘either or’ branch is introduced. This is practised with 3 exercises in which the simple elevator example is successively upgraded with additional functionality. More demanding are the lessons introducing parallel processes. Splitting and synchronisation of processes are learned by solving a simple mixing problem. SFC-programming culminates with two real-life problems: the first is a transport system (Figure 3) and the second is traffic light control. (Figure 4)

3.3 Analysing SFC

SFC can inherit all the theory developed for Petri-Net. In this course only those parts useful for industrial plant control are the topics. Mapping a SFC into a matrix, the incidence matrix, is an instructive example, how a graph can be mapped into another representation. In industrial applications 'reachability' is an important property of a SFC. The SFC-Editor provides information about markings with limited reachability, based on a reachability graph analysis. For a simple example, the student has to draw a reachability graph and compare the results with the automatically created information of the SFC-Editor. As a consequence, the student should be able to modify the example SFC into a completely reachable, i.e. live SFC.

Several IEC61131-3 engineering tools might bother the user with an error message that the SFC is not safe. Analysis of safeness is the last topic of the analysis section. Again the student has to reproduce the error information produced by the SFC-editor by manually drawing a reachability graph for a simple SFC. The example SFC has to be ameliorated into a safe SFC.

3.4 Introduction into Recipe Control

Recipe control is very common in batch industry. The SFC-specification according to IEC61131-3 is extended, so that actions can be parameterised in each step. The theory covers the Batch-models according to the ISA SP88.1 standard. For a pharmaceutical plant with two reactors a control recipe has to be configured.

4. THE SFC-EDITOR

It is not the scope of this contribution to describe the SFC-editor in detail. Therefore, only a short overview of the editor is presented in the following: The basic SFC-Editor, described in Keller (2001), was extended with analysis and recipe functionality. The editor enforces formally correct SFC. Depending on the position and movements of the cursor only admissible SFC-objects are proposed. Colour of rubber bands indicates the viability of a desired connection. With double click, the proposed object is selected. This is the fastest way to draw an SFC! This is illustrated in the following screen shot. Figure 1 shows the cursor below a transition and a step as next object is proposed.

Figure 7 shows how actions can be configured for each step. The box on the right lists the available actions. If selected, a functional description is displayed below. Actions added to the selected step are listed with their qualifiers in the box on the left. Similarly, transitions can be associated with functions to determine its process dependent logical value.

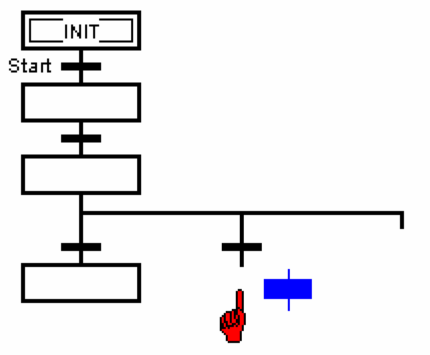

Figure 1: Cursor below Transition

A complete documentation of the SFC can easily be created in html-format. The SFC-specification is stored in a file including an encrypted section of the user name history. As a consequence the SFC-Editor can disclose all user names under which the SFC-specification was saved.

5. SIMULATION PLANTS

Attractive and challenging simulation plants are highly motivating. Real-life automation problems are more attractive than synthetic teaching exercises. In the following 3 of the 6 simulation plants will be presented.

5.1 Transport System

For a soldering machine, a carrier handling system had to be equipped with a PLC-Control. This leads to the simulation plant as shown in Figure 2. Carriers arrive on the conveyor belt C0. An intermediate storage is possible on conveyor C1. An elevator moves carriers down, where it is unloaded to conveyor C2. The SFC-solution is shown in Figure 3. The conveyors C1 and C2 and the elevator can run in parallel but are synchronised, when the carriers are transferred. During simulation an animated representation of the SFC is shown in a separate window. Active steps are marked with a blue background. The screen shot in Figure 3 shows the situation in which a carrier is moved onto conveyor C1 and simultaneously the elevator is moving down. With this example, the student can experience the power of the synchronisation capabilities of SFC.

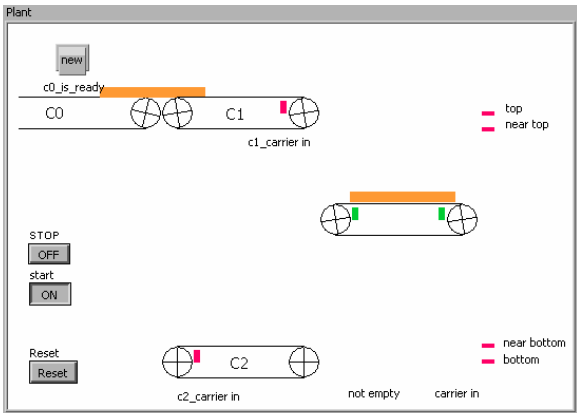

Figure 2: pallet handling system

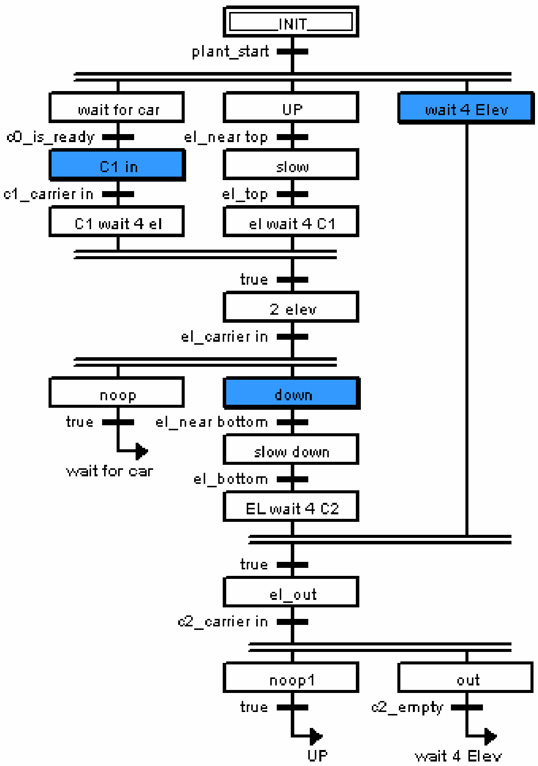

Figure 3: SFC to control handling system

5.2 Traffic Light control

Interaction with traffic light control is a universal experience. As a consequence no lengthy problem descriptions are necessary. In order to make traffic light control tractable within a SFC, a hierarchical control structure is necessary. In the bottom layer, control of an individual traffic light is implemented, whereas in the top layer the track logic is realised. The interface between the two layers consists of two Boolean variables as follows:

Top to Bottom
Traffic light cleared: The traffic light can switch to green, if a car is on contact switch.

Bottom to Top:
Traffic light is red: If no car is on a contact switch and minimal green-time is over, the light changes to red. If there are permanently cars arriving, the light changes to red after maximal green-time is over.

The control logic for all individual traffic lights is already implemented in actions. The student has to solve the problem of the top layer, i.e. the supervisory control of the traffic light system. The simulation display is shown in Figure 4. Arrival of cars is determined with a random number generator. The probabilities on each track can be varied with the controls on the level of the control panel. As a benefit the efficiency of the control can be studied for different traffic loads.

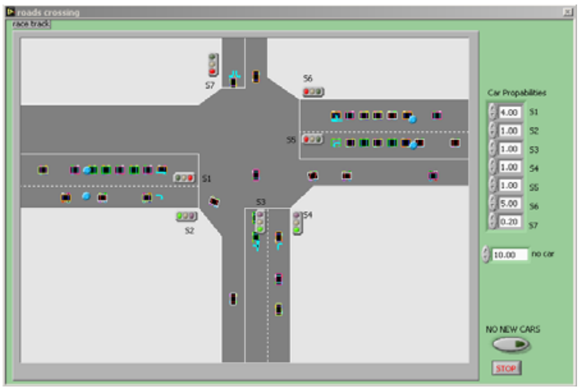

Figure 4: Traffic light control

5.3 Pharmaceutical Plant control

Batch control is a field where sequential function chart are widely accepted. Most of the well known PLC manufacturers provide SFC-Editors to define the control recipe. In Johnsson and Arzen 1996 application of SFC is studied for the different recipe levels defined in the SP88.1 norm. Recipe control of a batch plant requires well structured control logic. Equipment modules control a functional group of equipment that can carry out a finite number of specific processing activities. In the SFC-terminology, they are the actions defining the interface for recipe control. Figure 5 shows a simple multipurpose batch plant with its equipment modules.

In the theory section the SP88.1 batch models are instructed. In addition to these architectural aspects, also practical aspects of recipe control covered. The following topics are investigated: parameterisation, security and documentation of equipment modules for recipe control, i.e. user configured plant control. The student's task is to write a recipe for a simple production cycle. A short description of a typical exercise is: a substance has to be filled into reactor 1, heated up and a second substance has to be added. In a reaction phase, the mixture has to be kept at a constant temperature. The mixture is then transferred to the second reactor, cooled down and separated. The students experience that it is not at all an easy task to specify a faultless recipe. In contrast to reality, repetition of a trial is cheap and not very time-consuming and as a consequence recipe simulation systems might have a short payback. As shown in Figure 5 the tank levels are indicated on the reactors and temperature is coded into the liquid colour. Pressure is indicated on a gauge and small charts record the time dependent behaviour of the variables.

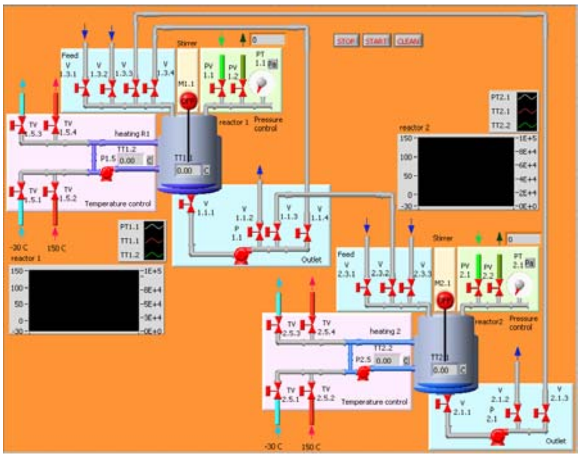

Figure 5: simple multipurpose plant

6. Feedback

The e-learning course has been used in the control engineering course at bachelor level since 3 years. It is used as introduction to PLC-Programming. Experience shows that learning the fundamentals, based on reading, is far less efficient than explaining it in a lecture. It is usually a student request that the theoretical part should be explained in lectures. The tool also reveals a large difference between students in the ability to learn and understand the principles of SFC. Usually this is also strongly correlated to the student's success in other fields. The SFC-Trainer could therefore also be used as an assessment tool.

The real-life problems of the exercises are very motivating. The large variety of solutions shows that the problems have a wide scope leaving enough freedom to creatively develop a 'personal' solution. Feedback from professionals, i.e. automation engineers, is very encouraging. They also see possibilities to use the SFC-Trainer for further training in their own environment.

Since the SFC-Editor relies on mouse position and movements, a good mouse is necessary. Notebooks with a nervous mouse pad are unserviceable with the editor. Also elderly persons exhibit problems with the graphical editor. Consequently an additional option for drawing a SFC has to be added to the editor.

7. Conclusions

This contribution presented an interactive e-learning tool for SFC. The high level interactivity makes it an attractive e-learning course and demonstrates that e-learning can be considerably enhanced if common structures with text and multiple-choice test can be surmounted. This becomes possible if a formal language to describe solution is available. The tool is based on LabVIEW, which is not an environment designed for e-learning courses. But due the capabilities of LabVIEW to develop modular applications with attractive user interfaces it is possible to build an easily expandable application. Further exercises can be added to the course, without recompiling or reinstalling the application. For the future it is envisaged to incorporate also laboratory experiments. Another very challenging aspect for future development is an automatic test of the student's solution including an automatically generated feedback.

The SFC-Trainer is available on http://www.fhso.ch/~keller.

8. References

Bloom, Benjamin S. ; Engelhart, Max D.: *Taxonomie von Lernzielen im kognitiven Bereich.* 5. Aufl. Weinheim: Beltz, 1976 (*Beltz-Studienbuch* 35). – ISBN ISBN 3-407-18296-1

Donzellini G., Ponta D., DEEDS: an Integrated Learning Environment for Electronics', Proc of SAAEI 2003, Vigo (Spain) 2003.

D. L'Her, J.L. Scharbarg, P. Le Parc, L. Marce, Proving Sequentail Function Chart Programmes using Automata, International Workshop on Implementing Automata WIA'98, Rouen, 1998

Flochova J., Liptak R. and Bachraty P., An on line course for supervisory control teaching, ACE-Conference, Oulu, 2003, 155-160.

Guerra, T. & Heffernan, D. (2004). The Guerra Scale. http://www.learningcircuits.org/2004/mar2004/

Johnsson Ch., Arzen K-E., Batch Recipe Structuring using High-level Grafchart, Proc. IFAC World Congress, San Franzisco, 1996.

Keller J.P. (2001). Programming Sequential Function Charts in LabVIEW. Proc. NIWeek 2001, Austin.

Satow, L. eLearning und eTesting http://userpage.fu-berlin.de/~satow/kap5.htm, 2002

Schulmeister R., Diversität von Studierenden und die Konsequenzen für eLearning, Carstensen, D. / Barrios, B. (eds): Campus 2004. Kommen die digitalen Medien in die Jahre? (Medien in der Wissenschaft; 29) Münster/New York : Waxmann (2004) , 133-144

Schulmeister R., Interaktivität in Multimedia-Anwendungen, http://www.e-teaching.org (2005)

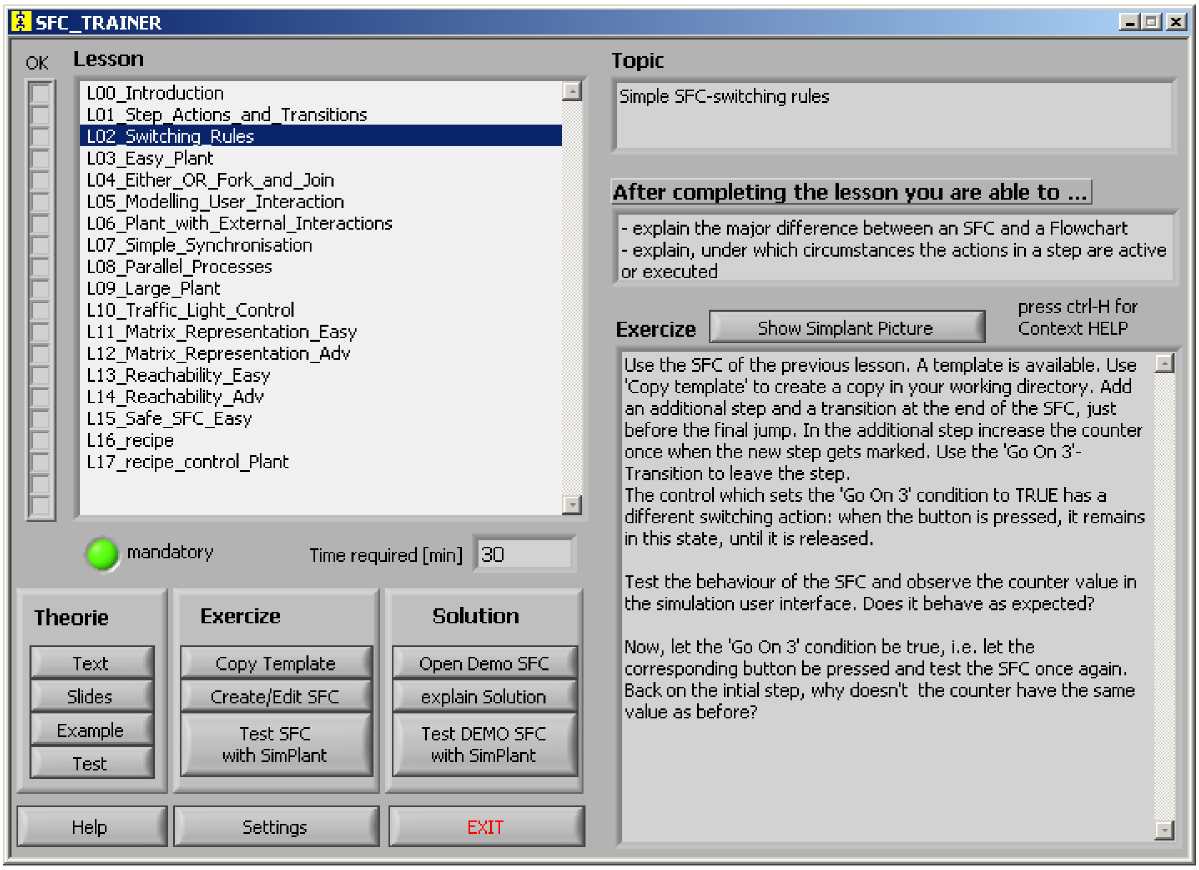

Figure 6: top-level user interface

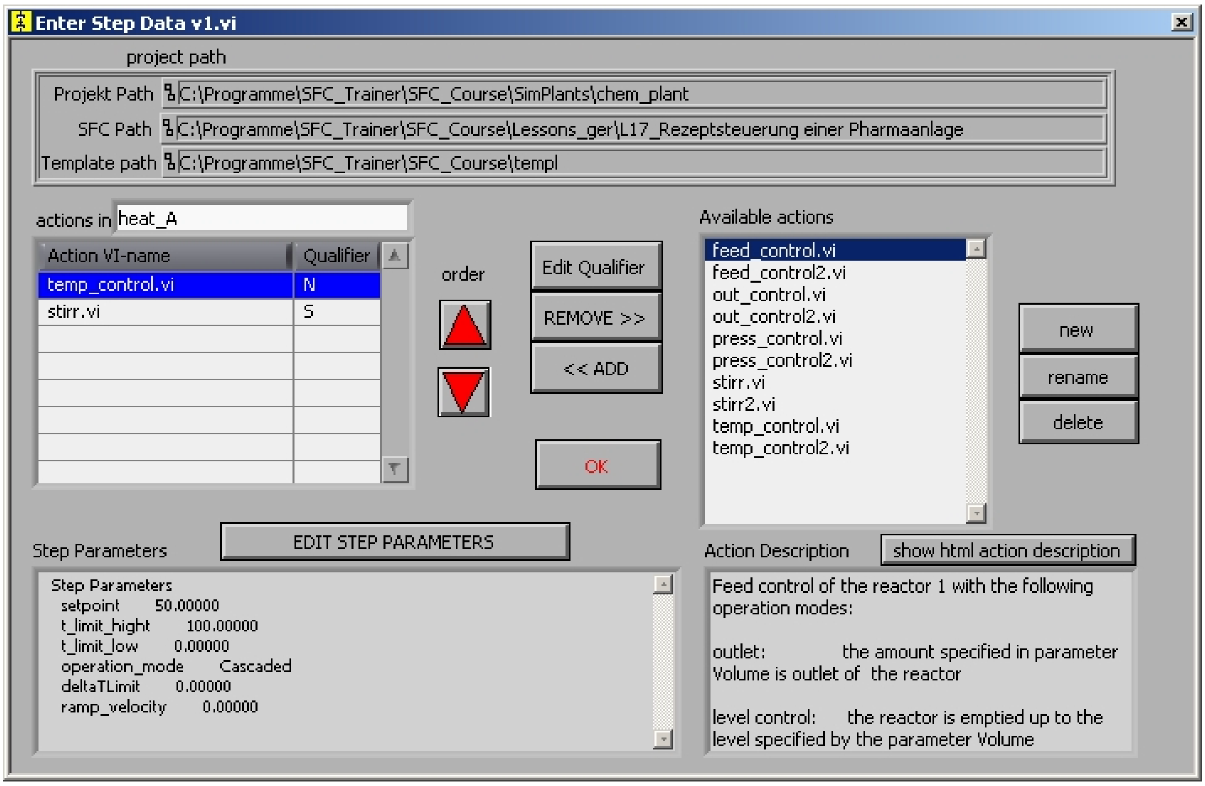

Figure 7: Specifying Actions