



498 - Modelling complex systems within TRNSYS SIMULATION STUDIO

S. Kuethe^{*}, C. Wilhelms, K. Zass, R. Heinzen, K. Vajen and U. Jordan

Kassel University, Institute for Thermal Energy Technology, 34125 Kassel, Germany

^{*} Corresponding Author, solar@uni-kassel.de

Abstract

Solar thermal systems progress in complexity as the technology advances. Development and optimization of these systems is often carried out with the help of flexible numerical simulation tools like TRNSYS. With increasing complexity, maintaining flexibility and structure of the simulation model is an important issue. Therefore, Version 16 of TRNSYS comes along with the graphical user interface SIMULATION STUDIO. It assists the user in setting up the model. Furthermore, a graphical system representation is available automatically. However, for complex systems containing a lot of components and interconnections the system understanding and usability of the model gets difficult. Due to this fact this paper describes a new structure level introduced to SIMULATION STUDIO. Like TRNSYS itself, it also pursues a modular approach by organizing a model in several subsystems, where a subsystem is a collection of components. These subsystems are connected to each other with uniform interfaces only. This results in modularity on the component as well as on the subsystem level and allows the replacement of subsystems in SIMULATION STUDIO in a simple way. The solar thermal system developed within Task 32 of the Solar Heating and Cooling Programme of the International Energy Agency (IEA-SHC Task 32) has been implemented using the new approach.

Keywords: solar thermal, simulations, modelling complex systems, TRNSYS

1. Introduction

TRNSYS [1] is the most commonly used numerical simulation tool for development and investigation of solar thermal systems. Since Version 16, it comes along with the graphical user interface SIMULATION STUDIO. Besides the assistance of generating the TRNSYS ASCII input file, the graphical representation of the model is supposed to improve the maintenance of it. In addition, it should help to decrease the time for third party users to get familiar with new models. However, with increasing complexity of a model the graphical representation gets confusing due to the crossing of many interconnections between the several components (see Fig. 1). This fact leads to difficulties of the system understanding regarding the interaction between the components.

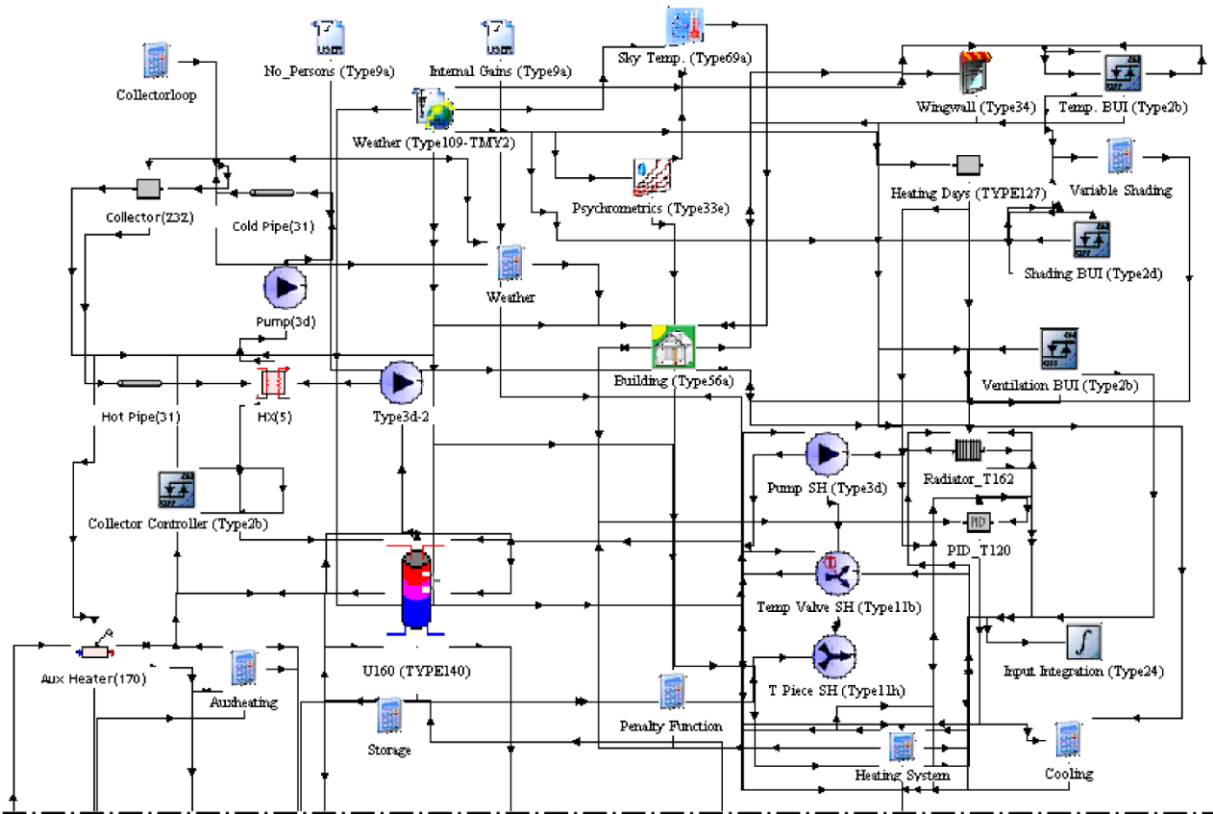


Fig. 1: Graphical representation of a complex model with a lot of overlapping interconnections between the components in SIMULATION STUDIO.

Furthermore, the replacement of system parts like a collector loop of a solar thermal system is not possible in a simple way. SIMULATION STUDIO already has got two options in terms of structuring components. On the one hand components can be arranged on several layers, which can be hidden and unhidden by demand, and on the other hand components can be collected in macros, where the input and output dialog of a macro contains all inputs and outputs of the components within the macro. In the graphical representation the collected components of the macro are replaced by one symbol only. Unfortunately, both features only help to improve the design quality but do not provide simple to handle removals or transfers of complete system parts to other models. Therefore, a new structure level is introduced to SIMULATION STUDIO. It follows the modular concept of TRNSYS by defining subsystems, in which components are grouped. The subsystems are connected via uniform implemented interfaces containing all required inputs and outputs of the subsystems. As interfaces simple EQUATION blocks are used in SIMULATION STUDIO and connections of subsystems are realized via the defined interfaces only. The interfaces are implemented in a way that a replacement of subsystems is possible without deleting or reconfiguring graphical links between the subsystems.

2. New subsystem based structure level in SIMULATION STUDIO

2.1. Concept

The concept of the new implemented structure level in SIMULATION STUDIO is based on the aim of obtaining a well structured graphical representation of the complete model on the one hand and a high level of modularity and flexibility on the other hand. This is achieved by dividing a model in several subsystems, which can be connected and replaced easily. In this way, modularity is given not only on component but also on subsystem level. A new feature realized within the subsystem approach is the possibility of removing and transferring subsystems without deleting or reconfiguring any visual connections in SIMULATION STUDIO. Fig. 2 shows a sketch of the new subsystem based structure level.

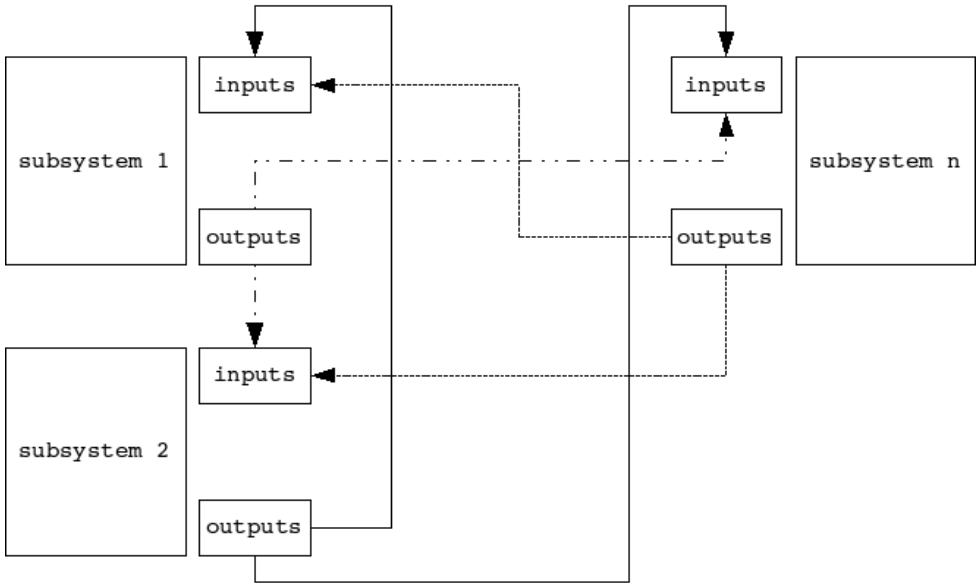


Fig. 2: Subsystem based structure level. Subsystems are connected via uniform defined interfaces only and can be removed and replaced easily. Inside the subsystems several components are connected to each other in the common TRNSYS way. To connect two subsystems with each other only two visual links (one for each direction) are needed.

2.2. Subsystems

A subsystem is a collection of several components which are connected to each other in the common TRNSYS way. In solar thermal systems the subsystems conform to the several loops in such a system, e. g. collector loop. Direct connections between components from different subsystems do not occur anymore. Instead of these connections the subsystems are connected via uniform interfaces. This gives the edge of avoiding many overlapping links in the graphical representation. Parameters that are used by the components within the subsystems are defined in separate EQUATION blocks belonging to the respective subsystem. As an example Fig. 3 shows the subsystem of a possible collector loop.

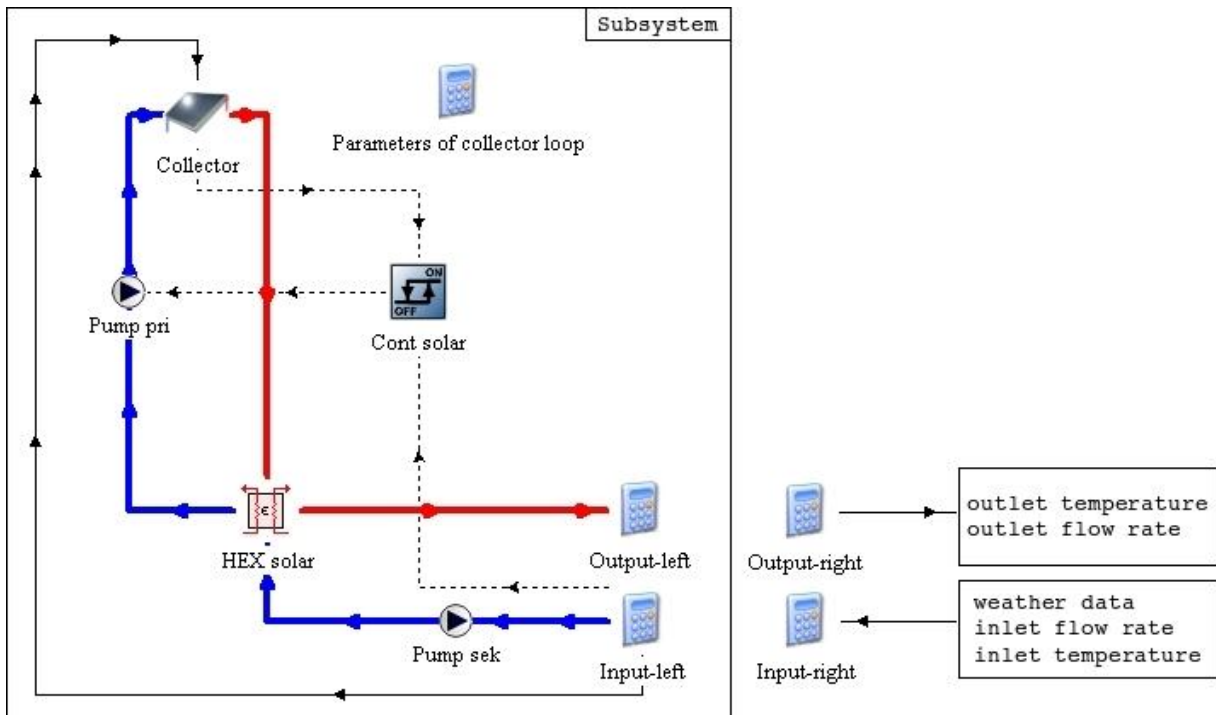


Fig. 3: Subsystem of a collector loop. Needed weather data, temperatures and mass flow rates are submitted via the interfaces only. All parameters used by the components of the subsystem, e.g. collector area, are stored in the EQUATION block “Parameters of collector loop”. There are no visual links between the equations inside and outside of the subsystem.

2.3. Interconnection of subsystems via uniform interfaces

The interconnection of subsystems is done via simple EQUATION blocks acting as interfaces. They provide all needed inputs of the components collected within a subsystem and vice versa the outputs needed by connected subsystems or components. By using EQUATION pairs as input and output interfaces, variables can be transferred without a visual link between a pair (see also Fig. 3 in section 2.2). As a result the possibility of replacing subsystems without deleting or reconfiguring any visual connection in SIMULATION STUDIO is given. The principle is based on the fact that variables defined in equations can be transferred to components or other equations without linking them visually. Furthermore, in SIMULATION STUDIO outputs from components used within an equation must be defined as auxiliary variables on the left of the dialog window of an EQUATION block¹. These auxiliary variables only appear in the graphical representation but they are not written to the input file of TRNSYS itself. Therefore, they can have the same names as the “real” variables (see Fig. 4). For the example in Fig. 4 the following section will be written to the TRNSYS input file:

```
EQUATIONS 2 "Inputs-right"
input_T_in = [UNIT No., OUTPUT No.]
input_mdot_in = [UNIT No., OUTPUT No.]
```

¹ This is necessary to be able to use time dependent values, e. g. temperatures, within equations.

```

EQUATIONS 2 "Inputs-left"
T_in = input_T_in
mdot_in = input_mdot_in
  
```

In the graphical representation the expression [UNIT No., OUTPUT No.] corresponds to the auxiliary variables on the left of the EQUATION block dialog, where outputs of components can be connected to it (see also right dialog window of Fig. 4). Because of exactly same names of the variables and the auxiliary variables used for the inputs, the transfer within the EQUATION pair in SIMULATION STUDIO is not visible for the user while connecting subsystems to each other (see Fig. 5).

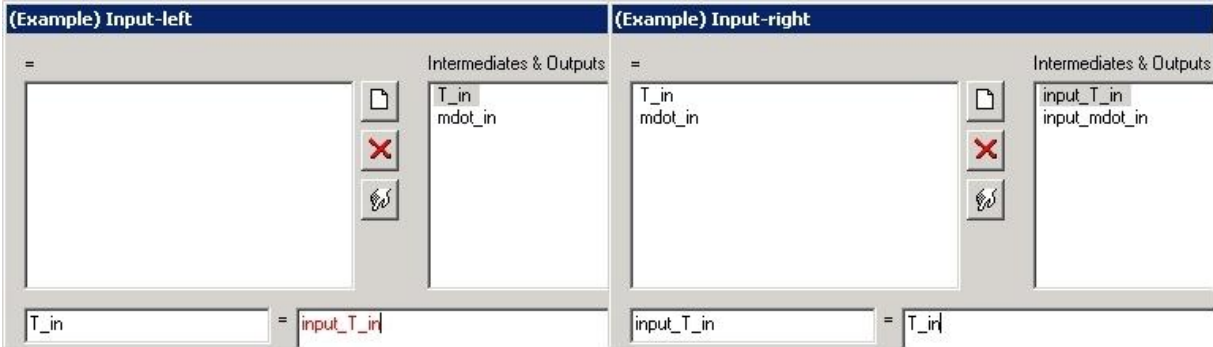


Fig. 4: Input EQUATION dialogs. The inputs of the right input EQUATION block are transferred to the left input EQUATION block without a visual connection in SIMULATION STUDIO. Variables can have same names, because variables on the left of the EQUATION dialog are only used in SIMULATION STUDIO as auxiliary variables but are not written to the TRNSYS input file.

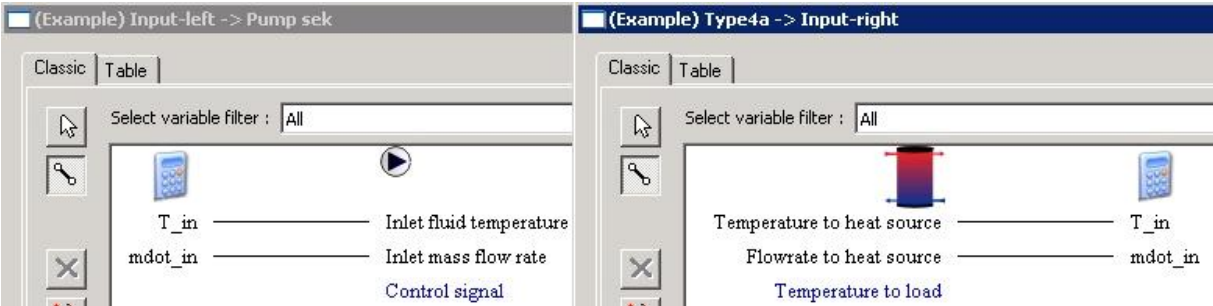


Fig. 5: Connection dialogs of EQUATION pair in SIMULATION STUDIO. The user does not realize the transfer of variables within the EQUATION pair while connecting subsystems or components because of the same names of input (auxiliary) variables (right dialog window) and output variables (left dialog window) in SIMULATION STUDIO. In this example the temperature and mass flow rate of a store is transferred to a pump, which is located within a subsystem.

3. Simulations and discussion

3.1. System description

Within the scope of IEA-SHC Task 26 and 32 a TRNSYS model of a solar thermal system for hot water preparation and space heating support was developed [2,3]. The model was implemented in the traditional way as a simple ASCII input file. To check whether the new approach obtains same results, this model has been rebuilt in SIMULATION STUDIO according to the subsystem method described in section 2. Therefore, the system shown in Fig. 6 is divided into the following subsystems, corresponding to the hydraulic loops:

- Collector loop
- Auxiliary heating loop
- Space heating loop
- Domestic hot water loop
- Weather processing
- Building representation

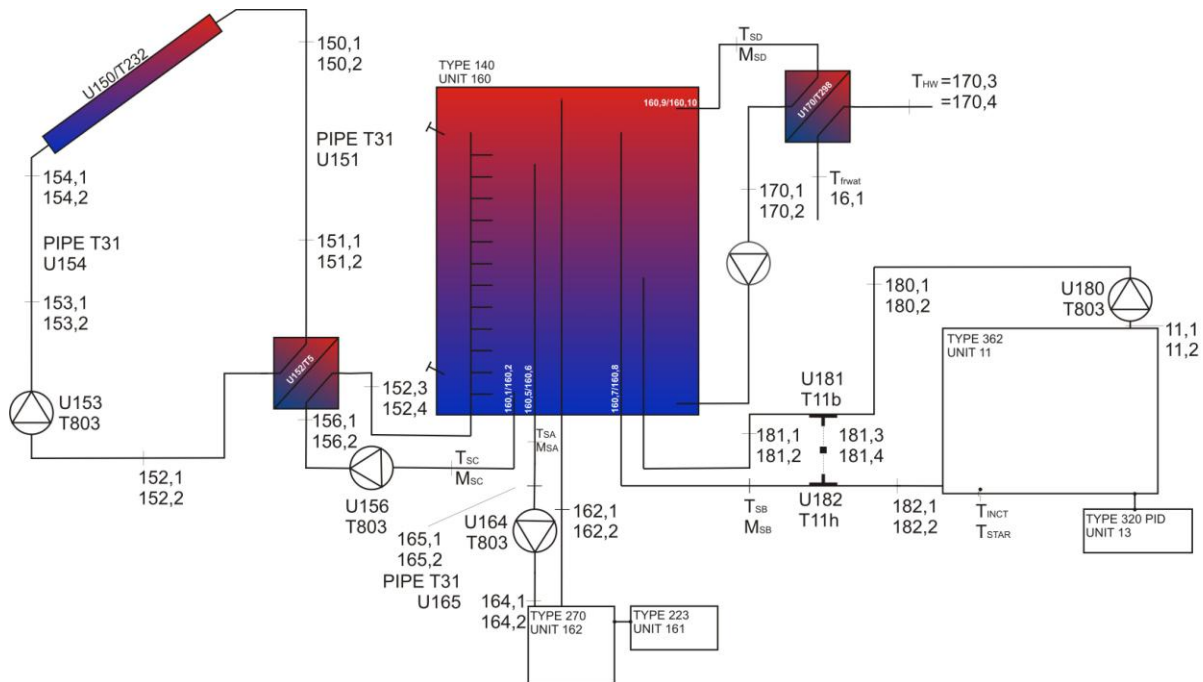


Fig. 6: Sketch of the solar thermal system for hot water preparation and space heating support modelled in TRNSYS and used for simulations within the IEA-SHC Task 32 [3].

The store as a main element of the model is connected to the subsystems as a “stand-alone” component.

The main difference of the subsystem based model in SIMULATION STUDIO to the manually as text file implemented reference system is the use of a lot of extra equations. This is caused by the above described way of implementing interfaces in order to connect the subsystems to each other. Furthermore, the possibility to optimize the component order automatically in SIMULATION STUDIO is used (see Fig. 7), which leads to another difference to the reference system.

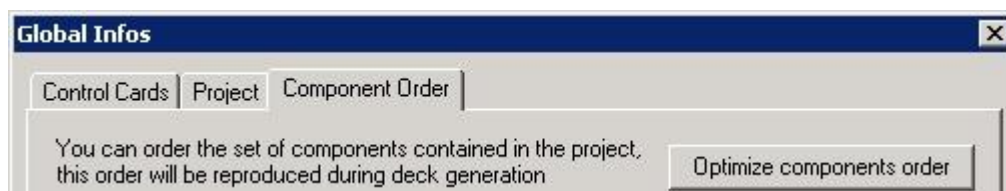


Fig. 7: Dialog window to optimize the component order in SIMULATION STUDIO.

3.2. Simulation parameters

The simulation period is one year with a time step of three minutes; weather data from Zurich is used. A single family house with a heating demand of 60 kWh/m²a is simulated. The maximum power of the burner is set to 10 kW and the set temperature of the auxiliary volume of the store is 63°C. A collector area of 20 m² and a 1 m³ store is chosen.

3.3. Compared values and comparison of results of the different modelling methods

The comparison of the results is done by comparing yearly values of energy delivered to and taken from the store by the several loops (collector, domestic hot water, space heating, auxiliary heating), the resulting energy balance as well as the fractional thermal energy savings ($f_{sav,therm}$) of the system defined as the auxiliary energy consumption of the solar thermal system (E_{sol}) compared to the final energy consumption of a reference system (E_{ref}):

$$f_{sav,therm} = 1 - \frac{E_{sol}}{E_{ref}} \quad (1)$$

The results show that the deviation of the compared values caused by the interfaces and different component order is negligible. The difference between the above named yearly values of the two systems is less than 1 %. This is also according to a simulation test made with only one controller component of the collector loop connected directly to the tank instead of the interfaces in the subsystem model in SIMULATION STUDIO. This also causes little effect compared to simulations where this component is connected via the interfaces only. Because it is only one component, in this case the difference of the compared values is less than 0.03 %.

4. Conclusion

The use of the new implemented subsystem based structure level in SIMULATION STUDIO - the graphical user interface of TRNSYS 16 - helps to improve the graphical representation of complex models as well as achieving a possibility to replace subsystems in a simple way. Regarding the educational field, the new feature leads to a shorter training time and simplified construction of models and permits students to work on projects using and maintaining complex systems, because they are not forced to get familiar with the ASCII input file. In recent simulation projects the new modular



approach was already applied. A comparison with a standard model representation via manually implemented text file showed negligible deviation, namely less than 1 %, comparing yearly values of the energy balance of the store and the fractional thermal energy savings of a solar thermal system for hot water preparation and space heating support used in the IEA-SHC Task 32. This is most likely caused by the additional EQUATIONS used as interfaces of the subsystems. Nevertheless, the new implemented feature should only be a first approach to get a new structure level to the graphical user interface of TRNSYS. It is worth striving for an integration of it in some way in newer versions of SIMULATION STUDIO without the use of extra equations as interfaces. In the future a database of subsystems could be generated like the TRNSYS component database and the subsystems could then also be linked to each other in the common TRNSYS way.

References

- [1] S.A. Klein et al, (2005). TRNSYS 16 – A Transient Simulation System. Solar Energy Laboratory, Wisconsin-Madison University.
- [2] W. Weiss et al, (2003). Solar Heating Systems for Houses – A Design Handbook for Solar Combisystems, International Energy Agency, Solar Heating & Cooling Programme – Task 26.
- [3] R. Heimrath, M. Haller, (2007). IEA-SHC-Task 32 - Project Report of Subtask A2 of Subtask A: The Reference Building, the Reference Heating System. Institute of Thermal Engineering, Graz University.