0.1 Introduction

INSEL is an acronym for INtegrated Simulation Environment Language. It provides an integrated environment and a graphical programming language for the creation of simulation applications. The basic idea of INSEL is to connect blocks to block diagrams that express a solution for a certain simulation task. INSEL was originally developed for modeling of renewable energy systems, the first versions being written at the former Renewable Energy Group at the Faculty of Physics of Oldenburg University, Germany. What makes INSEL so special?

The classical approach to computer programming is based on algorithmic programming languages like Fortran or C, for example. From a set of elementary statements a program is written with an ASCII text editor, compiled and finally linked together to build an executable. Program flow is the main aspect which dominates the whole development stage.

Graphical programming languages like INSEL use a totally different approach where data flow plays the key role. Instead of statements these languages provide graphical symbols which can be interconnected by mouse operations to build up larger structures. The graphical symbols can represent mathematical functions, real components like solar thermal collectors, photovoltaic modules, wind turbines and batteries, for example, or even complete technical systems of any kind. The graphical elements of INSEL are blocks.

The following list gives a first impression of the modular organisation and the currently main application fields of INSEL.

- The core component of INSEL is the inselEngine which is a full compiler that can interprete and execute applications written in the INSEL language or graphical preprocessors like HP VEE, for example.
- Fundamental blocks, basic operations and mathematical functions of the environment are provided in a dynamic link library called inselFB. It contains tools like blocks for date and time handling, access to arbitrary files and formats, blocks for performing mathematical calculations and statistics, blocks for data fitting, plotting routines, and so on.
- Energy meteorology and data handling is available as toolbox inselEM. This library contains algorithms, like the calculation of the position of the Sun, spectral distribution of sunlight, radiation outside atmosphere. A large data base provides monthly mean values of irradiance, temperature and other meteorological parameters. Generation of hourly radiation, temperature, wind speed, and humidity data from monthly means is possible. Further, diffuse radiation models, conversion of horizontal data to tilted are included.
- Solar electricity components like photovoltaic modules, maximum-power-point tracker, wind turbines, batteries, battery charge

regulators, hydrogen storage components, water pumps, inverters, motors, and generators, are available in the dynamic link library inselSE.

- Solar thermal components such as thermal flat-plate and vacuum water and air collectors, storage tanks. A full set of models for the simulation of thermal solar cooling plants, like desiccant and evaporative cooling systems, absorption cycles is implemented in the inselST toolbox.
- Solar thermal power plants for solar electricity generation is under development as toolbox inselPP which will contain models for parabolic trough, solar tower, dish-sterling and other solar power plant technologies.
- A building simulation tool box called inselBS is under devlopment and will contains models for walls, windows, convective and radiative heat exchange between surfaces, thermo-active components like cooling and heating floors and ceilings.
- A data processing tool box inselDP for Internet communication via different protocols is currently under development.
- A user-programmable environment inselUB in which practically all fields of engineering applications can be built in a very structured way. All standard programming languages like Fortran, C/C++ are supported.

Large data bases for simulation parameters of components that are available on the market are included in all toolboxes.

0.2 Overview

This manual/course is an introduction to programming with INSEL. No previous knowledge of INSEL or of any other graphical programming language is necessary. It covers a tutorial in three parts.

Prerequisite Many of the examples and exercises of this tutorial can be solved and tested in practise by using the inselLearningEdition software. There are however many blocks, data bases, and tools which are not part of the inselLearningEdition software, and can only be used with a licence for the respective toolbox.

All examples and exercises which require a license are marked with the INSEL icon in the margin.

Part I teaches the fundamentals of INSEL. Part II is task-oriented, so you can go directly to the sections that suit your interest. A third Part covers advanced programming techniques like implementation of user-written DLL INSEL blocks and functions and construction techniques for user interfaces.

The goal of this tutorial is to enable you – the reader – to program applications with INSEL as soon as possible. In the course you can learn to use the modular simulation tool INSEL and apply it to renewable energy systems. This knowledge can easily be applied in all other fields of numerical engineering.

First programming steps are achieved very fast. To work through the first two parts of the tutorial intensively will take about a week to complete. We have used guided examples for the most part. The exercises in Part II are a challenge to solve problems on your own. Solutions are provided with explanations. Part III is optional for the advanced INSEL programmer.

Although INSEL supports different operating systems like Windows and Linux, for example, the description given here assumes that you are working under Microsoft Windows.

Whereas INSEL is the calculation engine to solve the mathematical model, a commercial visualisation tool named HP VEE was chosen to graphically construct the model in the first place.

The tutorial is organised in modules, which treat different subject areas. Most modules also contain concrete pre-programmed examples which should be analysed and run. The user can then learn to reconstruct these examples following the given examples. In a second stage, exercises are given, where the reader has to find own solutions in model construction. However, the solutions are available at the end of the tutorial.

Part I As introductory material Module 1 provides the basic information concerning installation and start of the program. It continues with two complex examples of the INSEL software to demonstrate the combined simulation and visualisation possibilities that INSEL together with HP VEE offers. Whereas graphical programming tools such as HP VEE or LabVIEW are often limited to monitoring data visualisation and less complex simulation routines, the combination with the INSEL calculation engine offers both professional visualisation of applications and simultaneous calculation of the underlying physical model.

Module 1 is therefore meant as start-up appetizer and outlook of the performance of the tool.

Module 2 shows how to basically handle the simulation environment. The interaction between the graphical tool HP VEE and the INSEL calculation engine is explained and demonstrated in a simple example of plotting a function.

Module 3 starts with a description of the programming concepts of the graphical simulation tool INSEL. It explains the simpler block concepts such as Constant and Standard blocks and introduces the often needed Timer block concept. Module 3 then covers a range of examples using the block concepts explained before. Among the examples the performance of photovoltaic modules will be calculated.

Module 4 treats data file handling in INSEL. Reading and writing data from files and the corresponding formatting statements are explained.

Module 5 introduces a further block concept, the If blocks, and shows examples where such blocks are useful.

Module 6 starts with Delay blocks, which are necessary to solve "algebraic loops." As a last and most complex block concept, Loop blocks are introduced.

At the end of Module 6 you will be familiar with all the block concepts that INSEL offers and can – if you like – skip modules 7 and 8 and start with the second part of the tutorial, where examples from different application areas are gathered. The applications covered are solar meteorology, photovoltaics, and solar heating and cooling.

Module 7 introduces INSEL programming using a text editor. The graphical representation in HP VEE is then no longer necessary. Very few INSEL language statements need to be learned to directly program in a text editor.

Module 8 handles the topic of INSEL macro programming. Both INSEL and HP VEE macros can be built. In addition, the topic of interface programming concepts on the basis of HP VEE is presented.

Part II Part II covers three Modules with extensive applications.

Module 9 is an intensive course about many aspects of meteorological data.

Module 10 covers the topic of photovoltaic system simulation.

Module 11 completes the second part with simulation examples from the field of solar heating and cooling.

Part III Part III is meant as a supplement for the advanced INSEL programmer.

Module 12 describes how users can write and implement their own dynamic link libraries in Fortran, C++ or any other compiler language which can generate DLLs as output.

Module 13 presents the very new concept of interface programming on the basis of a general-purpose programming language like C/C++, for example.

Module 14 goes even one step further and shows how the INSEL compiler can be extended to co-operate with output of other block diagram editors like IISiBat (the graphical TRNSYS interface), LabVIEW, or Simulink, for example.