THE DEVELOPMENT AND SUPPORT OF PUBLIC DOMAIN SIMULATION SOFTWARE

R.E. Urban, W.A. Beckman, and J.W. Mitchell
Solar Energy Laboratory
University of Wisconsin—Madison
1500 Johnson Drive
Madison, WI 53706

ABSTRACT

The transient simulation program TRNSYS was originally developed to aid in the study of solar energy systems. It was first made public in 1975 when Version 6.0 wag released. Since then it has undergone continual development, and a series of versions have been documented, released, and supported. The current version, 13.1, was released in October, 1990, and is available in mainframe, IBM PC and Macintosh formats. TRNSYS capabilities have expanded as the **program has evolved:** it is now used extensively for building, HVAC, and industrial process simulation as well as for solar energy system studies. TRNSYS has flexible inputs and outputs, includes a large library of standard components, is designed to be user-modified, and is extensively documented. Approximately 500 copies of the program have been distributed world-wide.

The purpose of this paper is to briefly describe TRNSYS, review decisions made during the process of developing TRNSYS and to describe TRNSYS support. It is hoped that the experience gained from the development and support of TRNSYS will provide a base for the development and support of other simulation programs.

TRNSYS DESCRIPTION

The TRNSYS program is a component-based simulation program. Individual system components and groups of system components are described by FORTRAN subroutines. A standard library of these components is distributed with the program. If desired components are not available in the library, users can readily develop their own FORTRAN component subroutines. Additionally, a large base of user-created component models is also available. Component subroutines range from very complex differential equation descriptions to simple table look-ups.

Once all of the necessary components of a given system are available, a system information flow diagram is constructed. This is a schematic representation of the flow of information in the system, and is used to identify the connections between components. The structure of this information flow parallels the the structure of the mass, energy, and information (controls) flows in the physical system. When the information flow diagram is complete, the information can be translated from this graphical form to textual commands and data which are then processed by the TRNSYS program.

DEVELOPMENT

The desire that TRNSYS be widely accepted and used influenced a number of decisions about program structure. The program was conceived as a multi-purpose program with a wide range of possible component models. Features that would have made the program "state-of-the-art" had to be considered in light of the desire for widespread distribution, use, and support. As a result, trade-offs were made, some of which will be discussed below.

The modular approach used in TRNSYS reduces the complexity of formulating simulation models, as a large problem can be converted into a number of smaller problems, each of which can be more easily solved independently. The modular format also allows different investigators to work conveniently on various aspects of the same problem. It facilitates the addition of non-standard component models, because each component has standard interfaces to the remainder of the program. While there are a number of advantages to this modular structure, there is also an inherent disadvantage. The modularity makes it impossible to "collect" all of the symbolic representations of the equations in all of the components in order to solve these equations in a more efficient manner.

When TRNSYS was first written, FORTRAN was the obvious programming language. In spite of the advantages of newer programming languages, TRNSYS code continues to be written in FORTRAN-77. Extensions to FORTRAN-77 (such as the do-while command) are not used, nor are operating system-specific commands. For better or worse, FORTRAN-77 is a "standard" computer language; writing code in this language allows TRNSYS to be run on almost any computer, regardless of operating system. Additionally, because many researchers have at least a rudimentary understanding of FORTRAN, the choice of this language facilitates the understanding and creation of component subroutines. When additional programming language standards are finalized, it is anticipated that TRNSYS will take advantage of the additional language capabilities.

TRNSYS is distributed primarily as source code. As a result, all TRNSYS users must have FORTRAN compilers which handle the full set of FORTRAN-77 commands. The advantage of distributing source code is that users can determine exactly how sections of the program work, if necessary, and are free to modify the code to suit their needs. The drawback is that users must be reasonably computer literate to compile and link the source code into an executable version. To assist a user unfamiliar with creating executable code, an executable version sufficient for running example problems is supplied with both the IBM and the Mac versions. Detailed compiling and linking instructions are also supplied.

Prior to version 12.2, mainframe and personal computer versions of TRNSYS were separately maintained. While this was necessary due to the feeble IBM PC FORTRAN compilers available at the time, it complicated program development and maintenance, and prevented users from moving the source code from a mainframe to a personal computer and visa versa. Currently only one version of the source code is necessary. Minor changes to data statements are made to move between mainframe and personal computer versions.

A decision was made in the early development of TRNSYS to fix the timestep at a constant value throughout a simulation. The timestep can be specified by the user as any value, but it is not changed for a given simulation. This facilitates several calculations, notably those involved with conduction transfer functions for wall heat flows. However, it does not allow the equation solver routine to reduce the timestep during rapid changes and increase it during periods when little change occurs. This choice has simplified coding at the expense of possibly increased calculation time. Where necessary, an internal timestep is used to accommodate short time-constant phenomena.

Early research projects concentrated on "verifying TRNSYS." Since TRNSYS itself is only a collection of components that model various devices, verification studies can only establish the accuracy with which physical phenomena are modeled. For example, in one early study, the performance of Solar House I at CSU was modeled as a test of the accuracy of TRNSYS (Mitchell, et al., 1980). The simulation of the performance of the system showed that the individual components of the solar energy system could be modeled within the accuracy of the data available. It was also shown that the connections representing the mass and energy flows between the different components could be accurately represented. In addition, the simulation was accurate enough to detect errors in the operation of the physical system.

SUPPORT

Over the past fifteen years, a number of methods have been developed for supporting TRNSYS users. This has been essential in maintaining TRNSYS as a research tool.

The Solar Energy Laboratory (SEL) has had on its staff a TRNSYS engineer, who is most often a recent graduate of the lab and is familiar with TRNSYS from his or her thesis work. The TRNSYS engineer is responsible for responding to phone calls, letters and computer mail messages from users, updating and improving the TRNSYS code and manual, writing periodic TRNSYS newsletters for users, and, occasionally, holding TRNSYS workshops and classes. The presence of an on-going support person on staff is considered crucial for the wide-spread continuing use of TRNSYS. Support for this activity has come in part from the sales of TRNSYS.

A newsletter for TRNSYS users has appeared and disappeared sporadically over the history of TRNSYS support. For the past three years, *TRNSYS News* has been sent two to three times a year to all registered users. *TRNSYS News* contains news about TRNSYS activities and developments, information about users' TRNSYS-related research, and reports and announcements of TRNSYS workshops and other activities. As a number of regional TRNSYS groups have become active, *TRNSYS News* has reported their activities and encouraged users to contact the organizers of these activities directly.

Because of the growing use of TRNSYS in Europe, TRNSYS is now distributed and supported by a research group in Belgium and a similar group in Germany. While the ultimate responsibility for TRNSYS rests with SEL, these groups have greatly facilitated the use of TRNSYS in their regions. The difficulties of different time zones, currency and languages are considerably reduced. Both of these groups offer TRNSYS workshops and classes and are in close communication with SEL. In addition, a TRNSYS users' group in France has conducted a number of TRNSYS activities.

A variety of TRNSYS classes have been offered over the years. The original classes were oriented toward teaching researchers how to use TRNSYS. More recently the sessions have been aimed at exchanging ideas on techniques of modeling systems with TRNSYS. A beginning class in TRNSYS is planned for the ISES Meeting in Colorado in August, 1991.

Correctly translating a hand-drawn simulation information flow diagram into input text for TRNSYS can be difficult, particularly for large simulations with many components. For several years, efforts have been underway to prepare a "front-end" processor for TRNSYS to simplify this task. Not only would an easy-to-use front-end simplify the use of TRNSYS for researchers, it would also encourage other users to use TRNSYS for design work, for example. While some of these trial front-ends have been developed by the Solar Energy Lab, several more have been produced by other groups, either for their internal use or for the general public. Unfortunately, the great flexibility of TRNSYS makes the task of developing an equally flexible front-end very complex. One graphical front-end is currently commercially available (PRESIM); others are in progress. Had the possibility of a front-end been foreseen when TRNSYS was in the development stages, certain decisions could have been made that would have simplified the front-end development process. For example, the extensive use of different "modes" in TRNSYS components complicates the front end structure, as different modes of one component do not necessarily require the same number of parameters, inputs and outputs.

In addition to the "standard" library of TRNSYS components, a vast body of unofficial TRNSYS components are available. Some of these components were developed by students at the Solar Energy Laboratory; an even greater number were developed by other TRNSYS users. The most generally applicable of these components are eventually moved into the standard TRNSYS library. TRNSYS users have usually been willing to freely exchange user-developed components with the understanding that they are being made available "as is." The Solar Energy Laboratory has encouraged this exchange by making available its non-standard components and by publishing news about user-developed components in the newsletter. This free exchange of user components prevents duplication of efforts in some cases, and, in others, provides a starting point for further development. These non-standard TRNSYS components have proven to be useful in exchanging models between research groups working on similar problems.

SUMMARY

TRNSYS is a widely accepted and used computer program for simulating building energy systems. Producing a program for widespread application involved tradeoffs in its development. Some "state-of-the-art" features were judged not to be feasible. Several important development aspects have been: a modular structure; the use of FORTRAN-77; the distribution of source code rather than executable code to facilitate the development of component models by users; the development of one set of source code for all machines and operating systems; the use of fixed timesteps; and verification studies. A commitment to on-going program and user support has been essential to the use and development of TRNSYS. This includes in-house engineering support and distribution of a newsletter. The formation of regional user groups is encouraged; periodic short courses are offered; cooperative efforts in developing user-friendly front-ends is encouraged; and a library of "unofficial" component models is readily available.

REFERENCES

Klein, S.A., et al. 1990. *TRNSYS 13.1 User's Manual.* University of Wisconsin, Solar Energy Laboratory.

Mitchell, J.W; W.A. Beckman and M.J. Pawelski. 1980. "Comparisons of Measured and Simulated Performance for CSU House I." *Journal of Solar Energy Engineering,* 102, August: 192-195.