

TAYLOR II MANUFACTURING SIMULATION SOFTWARE

William B. Nordgren

F&H Simulations, Inc.
1366 South 740 East
Orem, Utah 84097, U.S.A.

ABSTRACT

Taylor II is a menu-driven simulation package mainly used in manufacturing, warehousing, and material handling. It is developed for the analysis and quantitative evaluation of complex processes, especially those with a dynamic character. A lot of applications in different industries show that there is an increasing need for simulation tools. This paper will demonstrate the process of building, analyzing and presenting models of real world systems with Taylor II.

1 INTRODUCTION

Companies spend a lot of time and money to obtain highest performance from machines and plants. Besides expensive and complicated techniques, there are catchwords like flexibility, just-in-time, lean production, etc. All these terms are based on the optimization of delivery, reliability, throughput times, stocks, utilization: i.e. costs. The problem is often it is nearly impossible to make a quantitative approach to these characteristics. Two factors are responsible: complexity and uncertainty. With an increasing number of components being in relationship to each other, it becomes more and more difficult to describe the system mathematically. When the second factor, uncertainty, is involved, even analytical methods fail. Uncertainties include: machine breakdowns, varying cycle times and batch sizes, waste and reject, etc. Throughput times, buffer sizes, the effects of resources influencing each other cannot be determined with conventional methods. Here, at least, you have to resort to assumptions, experience and intuition, although a better insight would be desirable. At least both possibilities of a bad system dimension, too high and too low, cause much costs. For these problems simulation can be a powerful tool to justify decisions or even prevent you from making wrong decisions.

Taylor II is a package for the simulation of discrete event systems. In general, you can simulate all kinds of systems in which discrete entities are to be processed, transported and stored (queuing systems).

Usually these entities are parts, tools, pallets, etc. This shows that Taylor II is not limited to specific industries. Besides manufacturing there are applications in distribution centers, harbors, airports, etc. Even in administration, simulation can be useful. Think of people or documents (information) which can be represented by entities too.

Taylor is a Dutch product, developed by F&H Simulations B.V. since 1986. The package is installed in more than 2000 companies and educational institutes. In the middle of 1996, the package, now called Taylor II for Windows received a complete new structure and integrates all necessary functions for a simulation study. There is one platform for modeling simulation, animation, analysis and presentation. In the following, all important functions will be described.

2 BUILDING A MODEL

Working with Taylor II usually starts with building a model. All model building is menu driven. A model represents a real world system which is to be studied. In practice, this representation is often simplified. The most important aspect when doing this is to build a model as exactly as necessary and not as detailed as possible. What is the reason for this? The reason is not the least performance of simulation packages, Taylor II or others, but an efficient, economical and objective-oriented way of working.

A model in Taylor II consists of four fundamental entities being related to each other: elements, jobs, routings and products. On each element, one or more operations can take place. The three basic operations are processing, transporting and storing; in Taylor II they are called jobs. Jobs are characterized by a cycle time (cycle times for storage are naturally 0), which can be random. The entities using the jobs are called products. Products can be defined freely and represent parts, tools, people, etc. They take place on an element and are sent to another element when the operation is finished. Therefore, you need a description of how the products flow through a

model. This description is stored in routings. Routings consist of a number of stages (routing records). At each stage it is defined which job-element is used and what the next stage is.

2.1 Layout and Routings

Defining a layout is the first step when building a model. A layout contains a number of different element types that you can position in different sizes and in any place on your screen. The following element types are available in Taylor II: Inout, Machine, Buffer, Conveyor, Transport, Path, Aid(operator), Warehouse and Reservoir. Choosing a type depends on the function in the real world system. A machine is the general purpose representation for any kind of operation so that a machine could be a robot, a mill, a counter, etc. Inout elements represent entries and exits (sink/sources) in a model. They can generate and 'eat' products freely. A buffer is for storage. Transports and conveyors represent continuous and discontinuous transportation. Aid elements are generally assigned to other elements. An aid could be an operator. Warehouses and reservoirs are buffers with some special functions. In a warehouse you can place products at a specific location, which is described by a number of cells horizontally and vertically. In a reservoir there is the possibility to define open/close control levels. Each element placed in the layout gets one job automatically.

When necessary, additional jobs can be assigned later. Besides the cycle time, a job also describes how an operation is done concerning input batch, additional resources, etc.

Now one or more routings are to be defined. By selecting the elements one after another you describe the path the products follow in the model. In most models the routing roughly reflect the flow of products because there are a number of product types taking (nearly) the same path and you can specify differences later. In some applications you have to handle a large number of product types all using different routings, i.e. in a job shop. In this case, Taylor II offers the modeler to import process routing descriptions from external files.

2.2 Detailing the Model

The second step is to detail the model. This means entering parameters describing the behavior of the model that differ from the default settings. Usually only some of the parameters are needed, so it is not necessary to fill a lot of masks before starting the model for the first time. Most of the parameters are related to the fundamental entities discussed above. Element parameters describe the behavior of the elements: what kind of element it is, stochastic (breakdowns) and deterministic (shifts, pauses), availability, how many products it can store, fixed and

variable costs, how it selects its next task, etc. The job parameters are similar.

Many systems seem to be quite simple. When looking at details, one often finds different kinds of rules and strategies to make decisions. That is the reason why implementing these decision points takes most of the modeling time. An important aspect in this phase is the way a simulation tool supports the user. Two items have to be discussed: high flexibility for implementing control logic and easy to test right working of the model (verification). Besides special functions for verification the on-line animation in Taylor II plays a large part. When the layout and the routings are finished, animation is available immediately. All changes in the model can be seen on the screen at once. The user is able to detail and to test the model step-by-step until the model runs according to his wish.

Every company producing simulation software is in a dilemma when developing a concept for control logic. The software should be easy to use but, the system should also allow all possible constructions. These two aspects are hard to combine. Distributors go different ways from fixed implemented strategies or decision tables to interfaces for programming languages or combinations of them. Besides a number of default strategies, Taylor II uses a macro language called TLI (Taylor Language Interface). TLI is an easy to use programming language that allows for modifying the model's behavior powerfully in combination with simulation-specific predefined and user-definable variables. TLI is used at element, job and routing level. There is also the possibility to use TLI interactively during a simulation run to make queries and updates. An interface to own routines written in C, Basic, Pascal, etc. is also available. Typical situations in which the routines have to be modified are loops (i.e. rework), selecting elements as a result of a certain status, assembly, disassembly, etc. Taylor II can send products from each stage in the model to each other in each quantity. For this purpose the user has direct access to all addresses in a routing and replaces the (in most cases) fixed values by expressions of the following form:

select number *with* condition *from* list *order*
expression *quantity* list *location* list

These so called 'select' statement are used for sending and receiving products. The italic parts in this structure can be values or other expressions. Selections can also be nested. This offers many possibilities for order picking, assembly and complex guiding and receiving strategies.

Some examples:

select 1 from 3,4,5

One product will be sent to machine 3, 4 or 5.

select 1 from 3,4,5 order - utilization [L]

The same as above, but the machine with the lowest utilization has the highest priority.

select maximum[1,F] with elqueue[L]<5 from 3,4,5 quantity 4-elqueue[L]

An (express) order uses the maximum number of machines. Only machines with a stock less than 5 are taken into consideration.

Elements are described by fixed acceptance rules for picking products or other TLI-user-definable rules and strategies for entry and exit of products. Examples are:

Entry condition: plan from file

The element chooses the next order from a list stored in a disk file.

TLI-entry condition: time|600<120

A conveyor accepts products only for the first 120 seconds in a 600 second interval, (| stands for modulo). The expression is true for 120 seconds.

Product types are described by a code (in combination with an icon) and by size (important for conveyors). Sometimes it is necessary that products get individual information or signs. Information could be a delivery date, priority, state of work, etc. For this purpose, products are coupled with attributes. Attributes can also work as a pointer to a matrix in which i.e., cyletimes are stored. This enlarges the amount of possible information enormously. Querying and updating of the attributes are done at the jobs by evaluating triggers. A trigger is one (or more) simple lines of text with a TLI-expression. Triggers can be evaluated nearly every time an event is scheduled. They can influence products (with their attributes) or the model status itself. An example is:

Trigger on Entry: curcycle[J]:=att2[C]

The cycle time of the order is stored in the second attribute and is assigned when the product enters the job.

Assume that a model is built with Taylor II. Now starts the actual simulation, apart from the (little) simulation when testing the model.

3 SIMULATION

In stochastic processes random numbers play an important part. Taylor II comes with different random number generators which normally work independently from each other. In some experiments you may want to study alternatives of a system. Then the models should run with identical conditions and for this purpose it could be necessary that all generators generate the same sequence of random numbers.

Taylor II is fully event-oriented. this means that the time between two events (i.e., beginning and ending of an operation) needs no CPU time. An internal event list makes sure that all changes in status are evaluated.

For every simulation run, the start situation is free to define. Products can be stored in any quantity at any place in the model. To reduce the warm-up period (moment of time that a model runs with stable status) you can make a warmstart. For a number of various runs you can define a batch run.

During simulation you can zoom, pan and rotate freely, stop, make modifications and then continue. A model can be simulated during a certain period of time or until a special (user-defined) condition is met.

The time representation is fully user-definable (i.e. week 8; tue; 8:30,2) and is displayed by an analog and digital clock. With the single-step-mode you can trace your model in detail or you use the conditional single-step for watching a part of the model only.

While simulating, the screen output can be interactively changed, ranging from no information (as fast as possible), via statistics and simple animation to full animation. With full animation you see each product moving from stage to stage or queuing in the system. Taylor II automatically keeps track of a large number of common statistics like utilization, waiting times, etc. Default and user-defined statistics can be displayed as numbers or as dynamically changing graphs. Especially determining the end of the warm-up period. Taylor II can also keep track of any kind of variables you want and display them after the simulation run.

4 ANALYSIS

For a correct description of a model, a number of input data is required like arrival times, cyletimes, breakdown failures, etc. Often these data exist in protocols, databases, etc. But the question is: is there enough data to characterize a process well or are they only a part of reality? That means a detailed analysis with statistical methods should be done first. Taylor can read in datasets and analyze them. Two important parameters are the average value and the standard deviation. another point is the statistical distribution which fits best. A built in routine automatically makes a distribution suggestion for the

dataset and displays this distribution depending on the type, in continuous or discrete form on the screen.

Typical information you get from a simulation study is utilizations, throughput times, production per period and costs. The results are available in different ways:

Tabular reports: contents and form are predefined, but may be changed by the user.

TLI reports: containing a mixture of explanation, results and illustration or giving specific results.

Queries: all results can be queried with TLI interactively (including minimum, maximum, average and standard deviation).

Predefined graphs: wait time histogram, queue graph, status diagram, utilization pie, etc.

User-defined graphs: representation of any kind of data like throughput times, costs, etc.

Graphs can be displayed in over 10 types of general purpose representation: histograms, bar graphs, x-y plots, pie charts, etc. including averages (whole and cumulative), standard deviation and so on. All reports and graphs can be sent to screen, disk file and printer or you can export these data for use in other programs.

5 ANIMATION

In the past animation has been treated as a nice but more or less unnecessary extra. Taylor II has an extensive built-in animation module.

There are several reasons. First, similar to the discussion in Section 2.2, the animation can be very useful (but generally not replace) for verifying a model. For this purpose the 2-D animation is available as soon as routing is created. With little additional work, 3-D animation is available too. They are useful when giving presentations to management or customers (often more convincing than rough numbers or in the educational sector. A number of ready-to-use 2-D and 3-D libraries come with the package.

With the built-in paintbox you can define your own icon-sets or draw background illustrations. The 3-D paint module allows you to change exiting or to add 3-D element icons. Background drawings are automatically translated into vector files for sealing and rotating. Existing drawings can be imported into the HPGL format.

6 ADDITIONAL FEATURES

In addition to the points above there are some special functions useful (not essentially necessary) for the daily work. When simulating a lot of alternative models with

different parameters and settings you could lose your overview. For this purpose, Taylor II generates a model documentation (text file) describing the whole model. another important aid is the context-sensitive online help with index and page search facility.

Besides stochastic arrival patterns, Taylor II can process external files (ASCII-file) with an unlimited number of deterministic product arrivals including arrival time, quantities and product attributes. This offers the possibility to simulate with real data. Special arrival patterns, think of season influences of daily peaks, can be generated with genlink. Genlink generates an external file by taking samples from different distributions.

External files can also be used to define deterministic machine planning or routing specification (see Section 2.1). The handling with ASCII-files becomes easier with the integrated text editor. DDE (Dynamic Data Exchange) links exist for windows base products that allow such links.

The ability to created self-running presentations allows you to build models which explain themselves. With a runtime module you can give all models to other people that are participating in a simulation study.

Taylor II version 4.0 includes several new that have been specially designed for windows 95. The Taylor Language Interface has been extended to include the us of functions. There are also three new modules included. These are Advanced Statistics (with an autofit module which tests distributions to fit data and an experiment module), runtime Development Kit (allowing you to create customized applications), and the Animator (special 3-D animation module with shaded animation).

7 SUMMARY

Taylor II integrates all functions necessary for a simulation study and combines in a simple manner high flexibility with ease-of-use without making concessions to functionality. many applications in different industries; manufacturing, transportation, distribution, material handling, show this. Fast modeling and online animation are the concept for the use not only by simulation experts. Taylor II is used more and more in education institutes for their practical courses in engineering.

Continuous customer feed back and our own consulting department enable F&H to keep improving Taylor II. This experience is also an advantage for the support and training courses that we offer.

AUTHOR BIOGRAPHY

WILLIAM B. NORDGREN is President of F&H Simulations, Inc. U.S.A. He received his Master of Science in Computer Integrated Manufacturing (CIM), and Bachelor of Science in Manufacturing Engineering from Brigham Young University. Bill was co-founder of

Taylor II Manufacturing Simulation Software

ProModel Corporation and served as Vice President of Customer Services and Training from 1988 until 1992. In 1993, Bill joined F&H Simulations B.V. and founded F&H Simulations, Inc. U.S.A. Through this union Taylor II Simulation Software was introduced into the United States, Canada, Mexico and South America. He is listed in the 22nd, 23rd, 24th and 25th editions of Who's Who in the West. He has done simulation projects for several large corporations including Andersen Consulting, Black & Decker, Boeing, Eastman Kodak, IBM, Pillsbury, Thiokol, TRW and Whirlpool