This chapter gives a number of criteria that can be used for the evaluation of simulation software. According to Law and Kelton (1991), simulation software can be classified as simulation languages and simulators. However, in this study the term ‘simulation software’ is mainly used for discussing data driven simulators, and all criteria are derived and described from the perspective of this type of simulation software. On the other hand, some of the criteria (e.g. criteria related to user support or financial and technical features) might also be used for the evaluation of simulation languages, and this was the main reason why the more general expression ‘simulation software’ has been used.

Criteria listed in this chapter represent a comprehensive evaluation framework that can be used for software selection by potential buyers as well as for guidance in further software development and improvement.

The literature on software evaluation analyzed in Chapter 2 provides a limited number of software evaluation criteria. This study is believed to be more comprehensive than earlier studies. These criteria were derived mainly from practical experience of automobile industry users, from simulation software developers and some of them were identified at during analysis of the literature.

3.2 CRITERIA FOR THE EVALUATION OF SIMULATION SOFTWARE
The description of the above criteria is given below:

(i) **Sources of information about the package:** This criterion examines how much famous is the software and how many different sources give information about the software.

(ii) **Sale of Software:** This criterion gives an idea about the use of the software i.e. how many companies are using the software.

(iii) **Availability of references:** This criterion gives the recommendations given by various people for the use of software. The number of recommendations will represent the faith of various users in that software.

(iv) **Quality of support for programming:** The variety and extent of programming concepts support is examined by this criterion. This evaluates whether the package supports typical programming concepts. This feature is very important because it greatly enhances the flexibility of the packages.

(v) **Efficiency of compilation:** This feature examines the level of efficiency of compilation of the simulation software. Compilation should be performed quickly and reliably, without substantial memory requirements.

(vi) **Built-in logic builder:** This facility of built-in logic builder helps a lot in logic development. Software itself gives suggestions for writing the logic for model development.

(vii) **Program generator:** This is another advanced feature provided by simulation software for program generation. A program generator is the tool to aid in the production of computer-coded representations of a logical model.

(viii) **Snippet code help:** This feature helps in writing the code while developing the models. Snippets are formally defined operative units to incorporate into larger program modules. These are used to minimize the use of repeated code.
(ix) **Built-in functions:** Providing the built-in functions when additional programming is allowed, might speed up the model development, because the user only has to call these preprogrammed functions by specifying their names and parameters.

(x) **Ease of entering text/code:** When additional programming is allowed, a substantial amount of code could be added to the model depending on the model complexity. In that case, the possibility of manipulating that text (code) is valuable.

(xi) **User-defined functions:** This feature is also connected with programming flexibility. When a user is allowed to develop his/her own functions, the logic of the model is handled more easily and modularity of modelling is enhanced.

(xiii) **Global variables:** Global variables can be accessed by any element in the model, and are often needed when additional programming is allowed to handle complex logic.

(xiv) **Interface to user written programs:** This facility also helps a lot in handling complex model development. The developer can use the already written program for model development.

(xv) **Integration with database management systems:** All data related to simulation model (input data, data related to the model’s logic or output data) can be stored in a database and retrieved by a database management system. This is particularly useful when a simulation system is connected to the control system of the factory, and uses real data to facilitate online scheduling or handle random events and errors. A database management system can perform the following functions for a simulation package: storage of models and their results for further use, retrieval of data for presentation and post-run analysis, comparison of results when multiple runs are performed or different models simulated, or it can facilitate adaptive modelling in the
form of information from previous models that can be used for development of new
models.

(xvi) **Integration with manufacturing requirements planning software:** Integration with manufacturing requirements planning software (MRP) is to ensure inputs from it. MRP is a software application that uses bill of material, routing, inventory, work order, sales order, purchase order, transfer order and other information to calculate requirements for materials.

(xvii) **Is it possible to do broad level scheduling with simulation software:** This is a feature of simulation software that allows a rough level of scheduling to be simulated.

(xviii) **Quality of manuals:** This criterion is very important because manuals make it easy for the user to understand the software. So, quality of manuals means a lot for the user.

(xix) **Run-time help:** This is another important feature because as you simulate many problems can be found. So run-time help facility should be there.

(xx) **Software maintenance facility:** It is worthwhile if the supplier provides the user with updated versions of a package that contains some additional facilities which were not available earlier.

(xxii) **Web-based support:** This is another important feature that means one can get support using the website of the company.

(xxiii) **Troubleshooting facility:** It is a logical, systematic search for the source of a problem so that it can be solved. It is needed to develop and maintain complex systems where the symptoms of a problem can have many possible causes. Using this facility, any model can be analyzed and problems can be removed.
**Demo models:** A number of different demonstration models are necessary for the first impression about the package. They could also be useful as examples of how certain features can be modeled when a need for modelling such characteristics arises.

**User group meetings:** It may be beneficial to attend user-group meetings, where it is discussed how to overcome deficiencies of the package, how to perform modelling more efficiently, to learn about plans for the release of a new software version, or behold different case studies where a particular package was used.

**Discussions group on the internet:** Discussion group of users should be there for discussion about various problems faced by the users. This is very useful for sharing the experiences of various users.

**User community web page:** Some web pages should be there for user community so that they can interact with each other and can share their problems and solutions.

**Type of contracts available:** This criterion evaluates which types of contracts are provided by the supplier.

**Frequency of update:** This criterion evaluates how often the supplier releases an updated version of the software, which is important when user’s requirements for simulation software are constantly increasing.

**Life-cycle maintenance cost:** This criterion evaluates whether a package requires considerable maintenance costs.

**Price of software:** It is not easy to judge this criterion. However, improving the quality of the packages should increase the market for their use, which should lead to reduction in price.
(xxxii) **Installation costs:** This criterion evaluates whether additional funding is needed for the installation of the package.

(XXXiii) **Cost of hardware required:** This feature relates to cost of hardware that is needed for the use of a package.

(XXXiv) **Availability of free evaluation software:** It is to help understand the software before purchase so that one can have an idea about whether he can use it to arrive at a decision for his problem.

(XXXv) **Hierarchical modelling capability:** Using this facility, user can approach a problem from different levels—Macro, Midlevel and Micro. Hierarchical modelling allows you to link these levels of models seamlessly.

(XXXvi) **Ease of installation:** This criterion assesses how easy or difficult it is to install a package.

(XXXvii) **Simulation of material handling systems:** AGV’s etc. require a path to move on and modelling software should allow flexibility for this type of modelling like for Cranes, Hoists etc.

(XXXviii) **Is it possible to use software in a network:** This criterion tells us whether it is possible to use the package in network or not.

(XXXix) **Availability of package on standard operating systems:** This criterion explores whether the package is available for standard operating systems or not.

(XXXX) **Add-on facility:** This criterion tells whether the software provides some add-on facility or not.

(XXXXi) **Fly-through mode:** This feature helps in observing the model as we are observing it from space.
(xxxii) **Type of licenses of software:** This gives the idea about the type of the license of the software whether it is a perpetual or time-based. A perpetual license is a license that is paid once and does not need to be renewed annually. A time-based license is essentially a lease of software for a shorter period of time.

(.xxxiii) **Downtime definition capability:** Any location or resource can go down either due to breakdown, rest, maintenance etc. This feature helps in defining it.

(.xxxiv) **Purpose:** Purpose relates to the application area of a package i.e. the types of systems that can be simulated by a particular package. Simulation packages are usually general purpose, or special purpose.

(.xxxv) **Representativeness of models:** This is a general criterion which evaluates to what extent the models are ‘natural’ representation of real systems. The level of representativeness is determined by several factors such as the type of visual display, the manner of modelling the elements and their behaviour, and the interaction between these elements.

(.xxxvi) **Run-time interface capability for scenario creation:** Experimentation software must allow easy experimentation without having to run the simulation as many times as changing parameters.

(.xxxvii) **Multiple branch decision making:** An entity can be routed to any number of locations depending on some condition. It is like a junction where cars can go to different destinations depending on their address. This is controlled in logic.

(.xxxviii) **Probabilistic branch decision making:** This criterion judges the extent of decision making based on the probability of an event.

(.xxxix) **Distributed simulation on network environment:** This criterion tells about using the power of Networks (LAN) to use the software by any group in LAN.
Splines, Polygon and orthogonal curve types: This criterion is for checking the availability of graphical editing features for making icons.

i. Libraries and templates of simulation objects: This criterion evaluates the assistance provided by the simulation software by providing availability of libraries and templates of simulation objects.

ii. Facility for designing reusable user defined elements: Attributes and variables are elements that can be defined and added. Their reuse, depending on the requirement, can be done.

iii. 3D models library: This is to help the user by availability of large 3D models library.

iv. Bubble help: This is another important feature for assisting the user in developing the model. It is a simple and fun visual tool that allow users to pose a question regarding some aspect of a given program and receive a response from the built-in help files included in the software.

v. Shape libraries: This feature includes graphical icon creation and categorizing them into libraries. There are readily available library for different shapes.

vi. 3D-animator: When this criterion is satisfied, then it is possible to obtain more realistic 3-dimensional graphical presentation of models. 3D text and shapes can be added. It is possible to choose one of several ready-made templates.
vii. **Scenario viewer:** This feature makes us able to view different scenarios for experimentation. This is the facility to analyze possible future events by considering alternative possible outcomes.

viii. **Library for real-time simulations:** This feature provides the facility to create a library of real-time simulation models.

ix. **Virtual reality animation:** This is also an important feature that provides animation for virtual reality.

x. **HotSpot Evaluator:** This facility is for problem reporting in context.

xi. **Print screen facility:** A print screen facility enables a printed display of the model to be obtained, as it appears on the screen. This feature might be useful for model documentation or writing a report on the simulation model developed.

xii. **Virtual screen:** A virtual screen enables the creation of a model display that exceeds the size of the screen. This feature is very useful when a model is complex, with many elements and details. In that case, the model can be created in a dimension larger than the screen size, and then reduced by a zooming function.

xiii. **Robustness:** Robust packages enable a variety of systems and characteristics to be modeled. This criterion is linked with programming flexibility because the possibility of additional coding is the best way of achieving robustness.

xiv. **Level of detail:** This criterion is connected to programming flexibility and robustness. While robustness mainly relates to types of systems that can be modeled by a particular package, this criterion is focused on a level of detail
that can be incorporated in each specific model. Both of these criteria are better satisfied when additional programming is allowed.

xv. **Reliability:** This criterion evaluates how often a package is subject to ‘crashing’, and unpredictable behaviour. For example, when a package allows programming and new code is added with some logical mistake, or when problems with memory occur, some packages are simply stopped. The only thing to do then is to reset the computer, whilst losing all changes made after the model was previously saved.

xvi. **Model execution time:** This criterion is evaluated by the time required to execute a program over a certain simulated period.

xvii. **Editing partially developed models:** This criterion examines whether it is possible to retrieve and edit partially developed models, or whether a model should be developed again from scratch.

xviii. **Interactive handling of parameters during experimentation:** This is also very important feature, which allows handling various parameters during experimentation.

xix. **Activity based costing:** This feature means cost calculated on the basis of an activity.

xx. **Display of attributes:** The dynamic display of the values of attributes might be useful for validation. If this feature is not provided by the package, but the display of variables is allowed, then an attribute value can be assigned to a variable and the value of that variable displayed.

xxi. **Replication of run-length:** This feature enables to make experiments with variable data.
xxii. **Execution trace:** This feature enables the user to trace the actual happenings at the time of execution of the logic.

xxiii. **List of used elements:** Such a feature provides information where a specific element is used or referenced within the model. This list might be valuable for checking the credibility of the model.

xxiv. **Backward clock:** This (not very common) feature can be useful for verification. For example, a long experiment can be run without animation in order to test model credibility. When the experiment is finished, and the graphical display of the current state of the model obtained, it may be realized that before the end of the experiments a logical error caused a blockage in the system or some other problem (the moment when such a problem occurs might depend on a specific combination of random number streams). It could be useful to run the model backwards to reach the moment of blockage faster, rather than to run the model from the beginning which might take several hours.

xxv. **Step function (event to event jumping):** A step function enables the observation of the change of the model’s state one step at a time. Differences that occur after each step can be examined in detail, which is very useful for verification.

xxvi. **Rejection of illegal inputs:** When this criterion is satisfied, any illegal type or format of input data is rejected, preferably with the message why the input was rejected.

xxvii. **Syntax checker:** This feature of the package is to check the syntax and to give warning if the syntax is wrong.
Antithetic numbers: This feature is based on the use of complementary values to introduce negative correlation. When these numbers are used properly, the variance can be reduced.

User pause facility: This facility is to pause the execution at any particular moment.

OLE compatibility: This feature represents the object linking capability of the package.

Display of events on the screen: This criterion examines whether the package provides a display of the events on the screen, when a model is run step by step.

Quality of error messages: Characteristics of error messages are examined by this criterion. Error messages should be comprehensive, completely documented, indicating precisely where and why a mistake has occurred, and preferably how an error can be corrected.

Stepwise simulation runs: This feature is used primarily for debugging. Normally a model runs depending on the clock speed or animation speed. However for debugging a model, it may help to step through the code. This will take user through all logic but naturally slows the process.

Accuracy check: It is useful if the software is capable to provide information about the accuracy of simulation output, obtained on the basis of several runs. This value might indicate that the run length is not sufficient, or there are some problems with the model parameters.

Shift editor: This feature is to define shift timings, break etc.
xxxvi. **Scheduled execution of scripts:** This feature is to run some predetermined event at right time.

xxxvii. **Sensitivity analysis:** Many independent factors affect a dependent factor. Which of these independent factors affect the most is what Sensitivity Analysis is all about. A small variation in the most sensitive input factor or independent factor can affect the output to a large extent.

xxxviii. **Quality of data analysis facility:** This criterion evaluates the level of data analysis, including the type of analysis that is provided, the understandability of the results of analysis and the number of different reports that is provided etc.

xxxix. **Number of theoretical statistical distributions:** When large complex models are developed, it is likely that a variety of different theoretical statistical distributions will be needed. Therefore, it might be useful if many of these distributions are provided.

xl. **Time dependent distributions:** Output that vary with time and are plotted as distributions are called time dependent distributions. These distributions should be there in the software.

xli. **Goodness-of-fit tests:** It might be useful if a package can undertake the Goodness-of-fit testing. Such tests can be used to check how real data fits a theoretical distribution, or to what extent simulation results are close to the performance of a real system.

xlii. (ix) **Output data analysis:** The ability to analyze a series of simulation results, and produce a variety of statistical reports on multiple runs, is very valuable but not a very common feature. A lot of time can be saved if, for example, a package can give mean values, variances or confidence intervals
for selected measures of performance obtained during many replications of simulation experiments.

xliii. **Confidence intervals:** This criterion examines whether a package provides an estimation of confidence intervals for relevant measures of performance obtained in multiple runs, which might be useful for the evaluation of accuracy of simulation results.

xliv. **Static graphical output:** Static graphical report relates to a graphical representation of simulation results in the form of histograms, time series, bar charts, pie charts etc. These graphical representations are obtained after simulation experiments. Graphical presentation is very valuable, because it quickly gives an impression about the measures of performance.

xlv. **Snapshot reports:** This report gives information on the number and types of the parts currently in the system, and the current state of the element. The information provided by this report is similar to that obtained by the explode function, but is not so detailed because individual values of part attributes are not displayed.

xlvi. **Dialogue boxes:** A dialogue box is a window which displays a series of controls, which usually appear in response to menu commands or points to some important condition, expecting the user to react. For example, a dialogue box can ask the user whether he/she wants to save the model before leaving the system. This feature represents a useful assistance during the use of a package.

xlvii. **Data charting:** This feature enables the user to see data in the form of charts – Pie charts, Histograms, Time Plots etc. These are easy to understand than just numbers.
Custom report generation: This facility is to create a set of charts and displays to suit user’s requirements.

Quality of output reports: This criterion examines the characteristics of output reports that is how they are presented, when they are obtained and whether they contain relevant information etc.

Understandability of output reports: Understandable output reports provide fast perception of model behaviour and credibility. Graphical presentation of results is an additional aid in presentation of results, but the results in numerical (and/or textual) form have also to be clearly presented and described.

Printed report after each simulation run: This feature generates the report, each time simulation is run.

Exchange data via internet: This allows the data to be exchanged through internet.

Task execution report: This is for output information of any task or activity that is executed and what it provides.

Queue data collection report: It means that how one can pick up information on the queue at any location.

Automatic rescaling of histograms and time series: This is to show the right plot or bar graphs without worrying about the different values of the output. This makes the tasks for a person analyzing the output results easy.
lvi. **Capability to do what-if analysis:** This is to experiment with data, that is what a simulation model can do but an excel sheet by itself or numbers cannot do.

lvii. **Conclusion making report:** Facility to help the user in taking the final decisions and drawing the conclusion based on the information.

lviii. **Optimization:** This feature is to experiment with the input factors to find the best combination for a set of desired outputs.

The Framework was presented to **The JJ Team (ProModel Partner in India),** Fibrolite Compound, Secunderabad. The comments of The JJ Team on Framework are attached as *Annexure-I.*

**REFERENCES**


