

Smart Sim Selector: A Software for Simulation Software Selection

Ashu Gupta

*Senior Lecturer
Department of Computer Applications
Apeejay Institute of Management
Rama Mandi-Hoshiarpur Road
Jalandhar-144007 (India)
Mob: 98158-91562*

guptashu1@rediffmail.com

Dr. Rajesh Verma

*Assistant Professor
Lovely School of Business
Lovely Professional University
Phagwara (Punjab) India*

dr.rajeshverma1@rediffmail.com

Dr. Kawaljeet Singh

*Director, University Computer Centre
Punjabi University
Patiala, Punjab (India)*

singhkawaljeet@rediffmail.com

Abstract

In a period of continuous change in global business environment, organizations, large and small, are finding it increasingly difficult to deal with, and adjust to the demands for such change. Simulation is a powerful tool for allowing designers imagine new systems and enabling them to both quantify and observe behavior. Currently the market offers a variety of simulation software packages. Some are less expensive than others. Some are generic and can be used in a wide variety of application areas while others are more specific. Some have powerful features for modeling while others provide only basic features. Modeling approaches and strategies are different for different packages. Companies are seeking advice about the desirable features of software for manufacturing simulation, depending on the purpose of its use. Because of this, the importance of an adequate approach to simulation software selection is apparent.

Smart Sim Selector is a software developed for the purpose of providing support for users when selecting simulation software. Smart Sim Selector consists of a database which is linked to an interface developed using Visual Basic 6.0. The system queries a database and finds a simulation package suitable to the user, based on requirements which have been specified. This paper provides an insight into the development of Smart Sim Selector, in addition to the reasoning behind the system.

Keywords: Simulation, Simulation software, Selection, Rating

1. INTRODUCTION

Growing competition in many industries has resulted in a greater emphasis on developing and using automated manufacturing systems to improve productivity and to reduce costs. Due to the complexity and dynamic behavior of such systems, simulation modeling is becoming one of the most popular methods of facilitating their design and assessing operating strategies. Increasing popularity of simulation techniques has resulted in an increase in the number of simulation packages available in the market. Currently the market offers a variety of discrete-event simulation software packages. For large international companies with their own simulation team it is often hard to select new discrete event simulation software. In simulation software selection problems, packages are evaluated either on their own merits or in comparison with other packages. The wrong selection of an inappropriate package can result in significant financial losses as well as the disruption of simulation projects. However, appropriate assistance in simulation software selection might reduce the scale of such problems. Many studies have taken place concerning the selection of simulation software.

2. RESEARCH IN SIMULATION SOFTWARE SELECTION

The starting point for the research was to review previous studies on the selection of simulation software tools. There are many studies on simulation software selection, some of which are as described below:

Banks and Gibson (1997) suggested some considerations to be made while purchasing the simulation software like accuracy and detail, powerful capabilities, fastest speed, demotion of problem, opinions of companies with similar applications, attending the user group meetings.

Hlupic and Paul (1999) presented criteria for the evaluation of simulation packages in the manufacturing domain together with their levels of importance for the particular purpose of use. They suggested general guidelines for software selection. They pointed that not to expect a particular package to satisfy all criteria. However, it is indicated which criteria are more important than others, according to the purpose of software use. These guidelines can be used both by users that are looking for a suitable simulator to buy, and by developers of such simulators who wish to improve existing versions of simulators, or who wish to try to develop a new, better manufacturing simulator.

Nikoukaran et al. (1999) created a framework of criteria to be considered when evaluating discrete-event simulation software. This framework is structured, and pays attention to a rich set of criteria on which simulation packages can be compared. It is, however, difficult to base a decision for a large multinational company on these criteria, as it is only a comparison, without weighing and without a method to determine the relative weights, and the weight differences between parts of the simulation team.

Tewoldeberhan et al. (2002) proposed a two-phase evaluation and selection methodology for simulation software selection. Phase one quickly reduces the long-list to a short-list of packages. Phase two matches the requirements of the company with the features of the simulation package in detail. Different methods are used for a detailed evaluation of each package. Simulation software vendors participate in both phases.

Seila et al. (2003) presented a framework for choosing simulation software for discrete event simulation. By evaluating about 20 software tools, the proposed framework first tries to identify the project objective, since a common understanding of the objective will help frame discussions with internal company resources as well as vendors and service providers. It is also prudent to define long-term expectations. Other important questions deal with model dissemination across the organization for others to use, model builders and model users, type of process (assembly lines, counter operations, material handling) the models will be focused, range of systems represented by the models etc.

Popovic et al. (2005) developed criteria that can help experts in a flexible selection of business process management tools. They classified the simulation tools selection criteria in seven categories: model development, simulation, animation, integration with other tools,

analysis of results, optimization, and testing and efficiency. The importance of individual criteria (its weight) is influenced by the goal of simulation project and its members (i.e., simulation model developers and model users).

Many other studies have been carried out involving the selection of simulation software such as **Bovone et al. (1989)** and **Holder (1990)**. All of these studies are geared towards a more theoretical approach of the problems involving the selection of simulation software. Smart Sim Selector attempts to put these issues into practice and aid the user in various difficulties that may arise when selecting a suitable simulation package.

This paper is structured as follows: Subsequent to a description of the criteria used for software selection, and justifications for choosing criteria, a description of the database and the way in which the database handles a query are provided. Later sections provide an insight into the software itself. Issues related to an overview of the software and the way in which a query is processed by Smart Sim Selector is addressed. The methodology which Smart Sim Selector uses to recommend most suitable packages is also discussed. Finally, conclusions outline the main findings of this research; highlight the benefits that can be derived from Smart Sim Selector as well as its future developments.

3. SELECTION CRITERIA

The type of criteria required to evaluate simulation software was an important issue that was addressed throughout the development of Smart Sim Selector. Research within this topic was undertaken and several important studies were focused upon (**Tocher, 1965**), (**Cellier, 1983**), (**Banks et al. 1991**), and (**Law and Kelton, 1991**). However, majority of these studies were theoretical and unsuitable as not many of them provided a critical evaluation of the software products under consideration. In addition, it was difficult to compare the evaluation of different studies because they did not use common criteria.

The criteria used for the Smart Sim Selector was taken from **Verma et al. (2009)** who proposed more than 210 evaluation criteria divided into 14 main groups: General Information, Coding Aspects, Compatibility, User Support, Financial & Technical Features, General Features, Modeling Assistance, Visual Aspects, Efficiency, Testability, Experimentation, Statistical facilities, Input/Output Capabilities, Analysis Capabilities. The main reason for incorporating data from such a wide range of categories is to ensure that Smart Sim Selector will be able to produce accurate results.

4. DATABASE DESIGN

The database is considered to be the “engine-room” of Smart Sim Selector. It holds the information related to evaluation details of each package. In addition, the queries concerning the suitability of packages are also generated within the database. The initial requirements of the database were largely concerned with not only the ability to store and access data, but also to ensure that the database could be easily modified and maintained. This is an issue of great importance because in the near future the number of packages which are evaluated will be increased. The data has been collected from Automobile manufacturers in North India which includes 18 automobile companies.

The database is designed in a conventional manner, with information being stored about each package. A particular data table shown in Figure 1 contains information regarding the simulation software package such as type of package, purpose of simulation, and price. Another table designed, holds information regarding the name of the package and a unique index which holds the pack-id number. A query is generated after the user has finalized the requirements and priority rating.

Software_ID	1	2	3	4	5	6	7	8	9
4	5	5	4	3	3	3	4	4	4
10	5	5	5	5	5	5	5	5	5
11	5	4	3	3	3	3	4	4	5
12	6	5	5	5	6	6	6	6	6
13	5	5	5	6	6	6	6	6	6
14	4	3	3	3	2	2	4	4	4
2	5	4	4	5	5	3	4	3	3
3	4	4	4	4	4	4	5	6	6
6	6	4	4	4	3	2	3	4	4
7	4	4	5	4	3	3	3	2	2
8	5	4	3	3	4	5	3	3	4

Figure 1: One of the Tables within the Smart Sim Selector Database

5. OVERVIEW OF SMART SIM SELECTOR

Smart Sim Selector was designed using Visual Basic 6.0, because it allowed for the creation of effective and user-friendly interface. Visual Basic 6.0 has a built in Access Engine which is easily capable of handling all database queries. This was ideal for Smart Sim Selector which required a relational database to store information regarding various packages. At the moment, Smart Sim Selector stores information about 11 packages, based on 210 different criteria. There is no bias towards a specific supplier, which is one reason why Smart Sim Selector may be seen as an effective tool in the simulation software selection process. Figure 2 shows the starting screen of Smart Sim Selector.

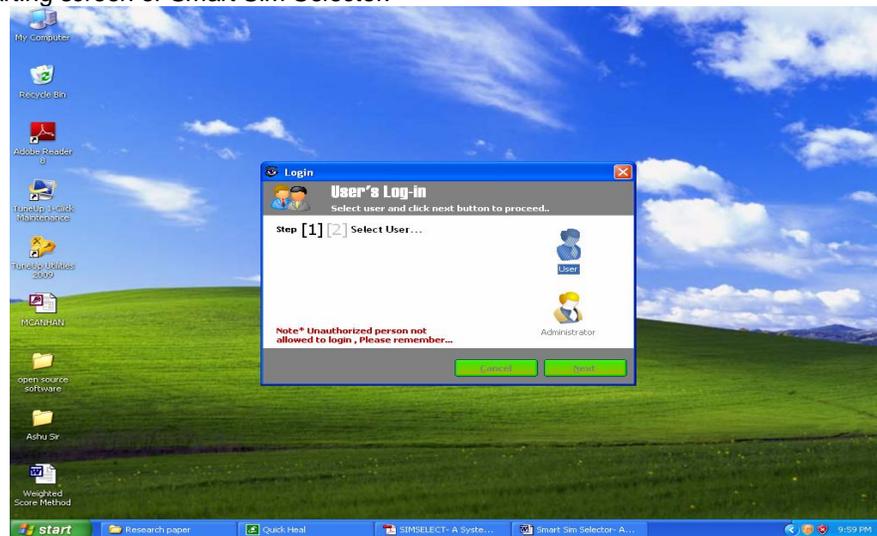


Figure 2: Starting Screen of Smart Sim Selector

5.1 Generating a Query

In order to generate a query using Smart Sim Selector, the user is required to input data regarding the requirements of the desired system. The main menu of the system is shown in Figure 3. The options provided by the menu are Administrator Work, Suggest Software, Comparison of Softwares, and Exit.

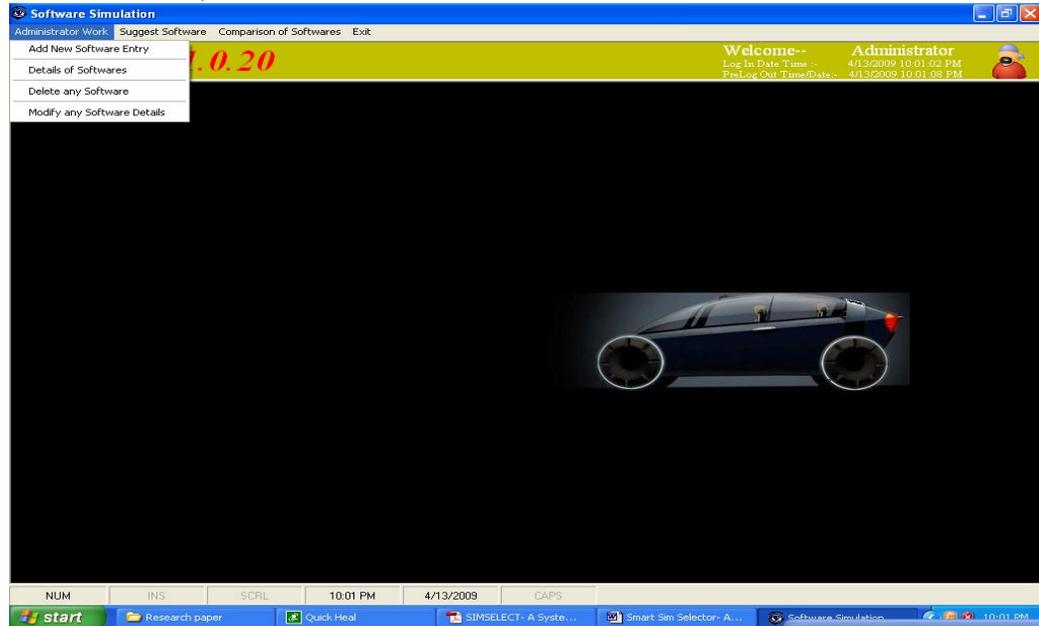


Figure 3: Smart Sim Selector Menu System

Administrator can add, delete and modify any software in the database. Also the option 'Details of the Softwares' will display the details of all the softwares contained in the database to the user and administrator. Once the administrator selects to add new software, the screen as shown in Figure 4 will be displayed.

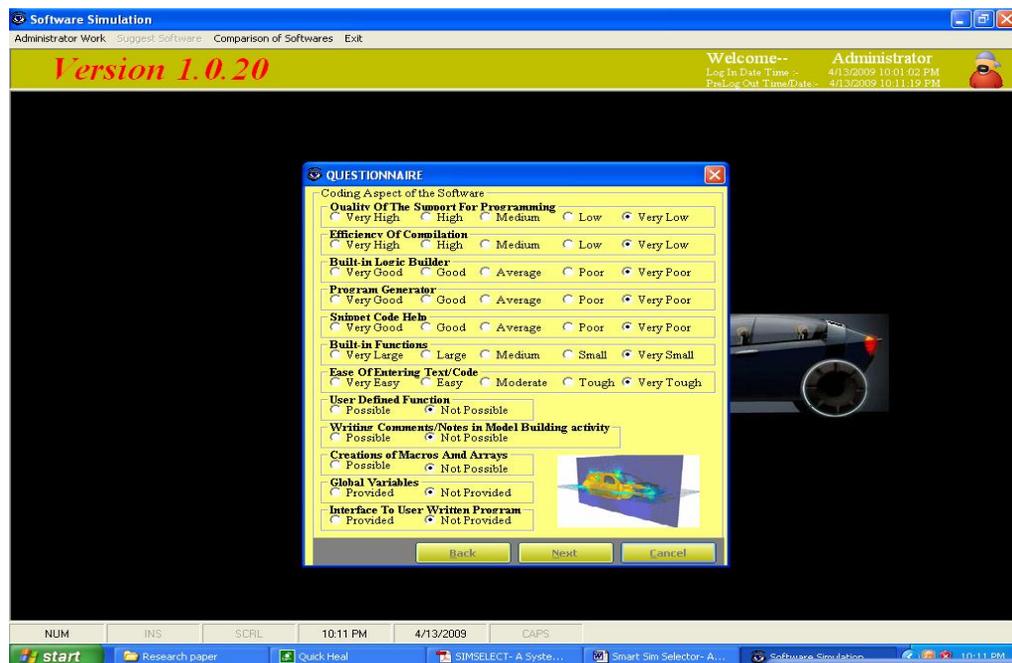


Figure 4: Screen for adding new software in Smart Sim Selector database

5.2 COMPARISON OF SOFTWARES

In order to compare the simulation softwares, a rating of these has been established. This rating is based on an analysis of the simulation softwares. As such it should be considered as a relative measure of quality of these softwares from the perspective of groups of criteria rather than as an absolute value.

There are total 14 groups of features i.e. general information, coding aspects, software compatibility, user support, general features, modeling assistance, visual aspects, efficiency, testability, financial and technical features, experimentation facilities, statistical facilities, input and output capabilities, analysis capabilities. The value (out of 10) of these groups of features is calculated for all the simulation softwares in the database. Figure 5 shows the comparison screen.

$$\text{Evaluated Value} = \frac{\text{Calculated Value} \times 10}{\text{Maximum Value}}$$

Where Maximum Value= Sum of highest possible values that can be selected in a particular group of features

and Calculated Value= Sum of actual values selected in a particular group of features



Figure 5: Screen showing Comparison of the Softwares

6. SUGGESTING THE BEST SUITABLE SIMULATION SOFTWARE

Smart Sim Selector has three different techniques for selecting the best suitable software for the user as shown in Figure 6:

6.1 Analytic Hierarchy Process

6.2 Weighted Score Method

6.3 TOPSIS (Technique of Order Preference by Similarity to Ideal Solution)

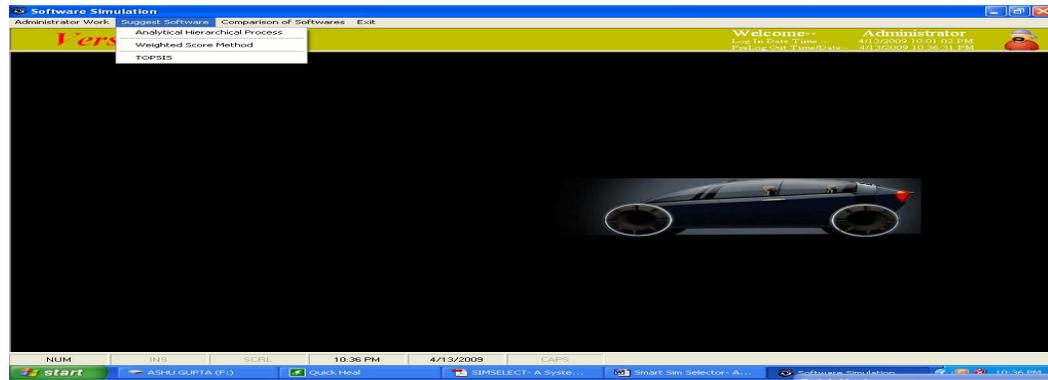


Figure 6: Different Techniques for Software Selection

6.1 The Analytic Hierarchy Process and Simulation Software Selection

The AHP separates the evaluation decision into hierarchy levels and attempts to reduce the inconsistencies in human judgment. It was originally used for socio-economic and political situations but of late it has proved useful for judgmental decision making in other areas, such as the selection of equipment for ice breakers (**Hannan et al., 1983**), the selection of materials handling equipment (**Frazelle, 1985**) and perhaps more relevant, the selection of manufacturing software (**Williams, 1987**) and scheduling software (**Williams and Trauth, 1991**). Further applications, along with a good exposure of AHP, are given by **Partovi et al. (1990)** and **Zahedi (1986)**.

In using the AHP technique all the criteria are compared in a pairwise way, using Saaty's intensities of importance (**Saaty and Roger, 1976**) shown in Table 1, in order to establish which criteria are more important than others. The values are then placed in a matrix and the normalized principal eigenvector is found to provide the weighting factors which provide a measure of relative importance for the decision maker. The next step is to make pairwise comparisons of all alternative with respect to each criterion. Final rankings of the alternatives are made by multiplying the critical weights of the alternatives by the critical weights of the criteria. The alternative with the highest score is then deemed to be the preferred choice.

Figure 7 shows the screen for AHP.

Table 1: Saaty's Intensities of Importance

Intensity of Importance	Definition	Explanation
1	Equal importance	Two activities contribute equally to the objective
3	Weak importance of one over another	The judgement is to favor one activity over another, but it is not conclusive
5	Essential or strong importance	The judgement is to strongly favor one activity over another
7	Demonstrated importance	Conclusive judgement as to the importance of one activity over another
9	Absolute importance	The judgement in favor of one activity over another is of the highest possible order of affirmation
Reciprocals of above non-zero numbers	If activity i has one of the above non-zero numbers assigned to it when compared with activity j, then j has the reciprocal value when compared with i	

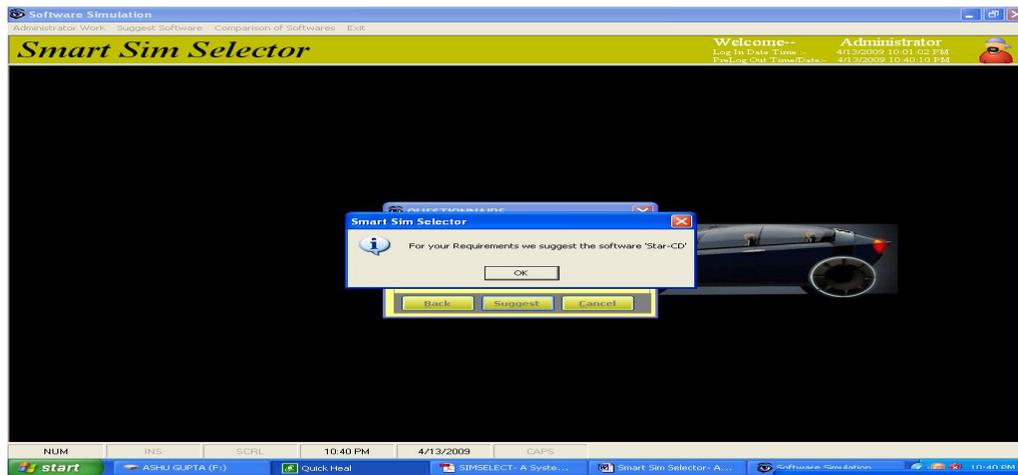


Figure 7: Best Software Suggested

6.2 Simulation Software Selection using Weighted Score Method

The steps used in Weighted Score method are:

1. **Determine the criteria for the problem:** We have 14 different groups of features to satisfy this step
2. **Determine the weight for each criterion:** This value will be taken by the user when he will select the criteria and their importance. Figure 8 shows this screen.
3. **Obtain the score of alternative i using each criteria j for all i and j as follows:**

Let S_{ij} be the score of software i using group of criteria j

W_j be the weight for group of criteria j

S_i score of alternative i is given as:

$$S_i = \sum_j W_j S_{ij}$$

The simulation software with the best score is selected.

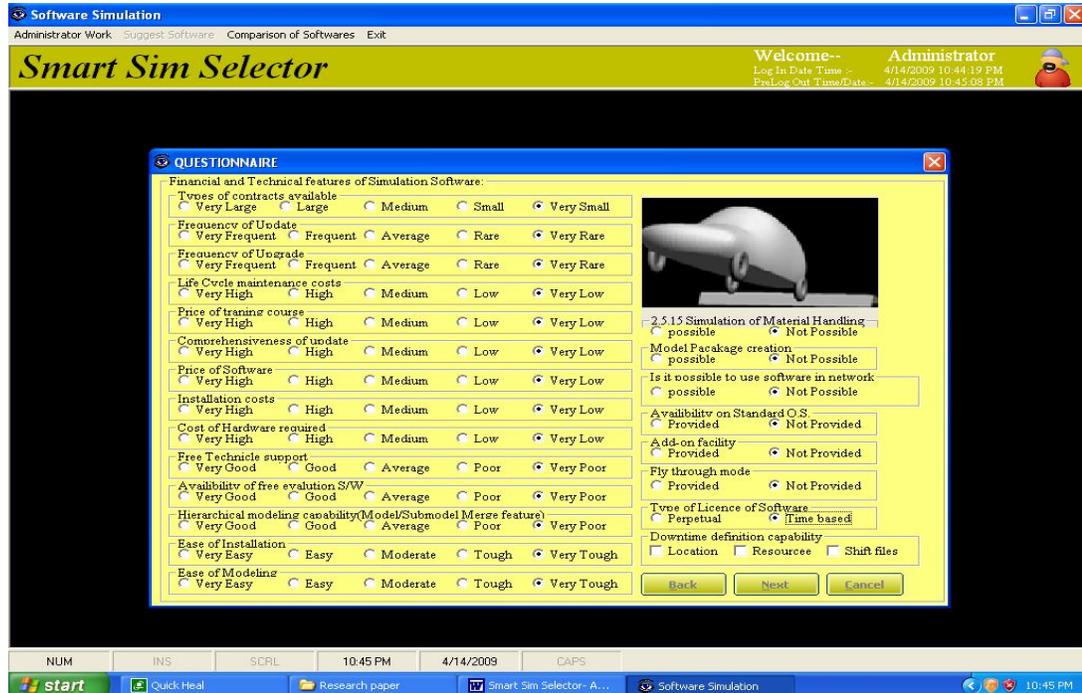


Figure 8: Weightage to each criteria given by the user

6.4 TOPSIS (Technique of Order Preference by Similarity to Ideal Solution)

In this method two artificial alternatives are hypothesized:

1. Ideal alternative: the one which has the best level for all attributes considered.
2. Negative ideal alternative: the one which has the worst attribute values.
3. TOPSIS selects the alternative that is the closest to the ideal solution and farthest from negative ideal alternative.

TOPSIS assumes that we have m alternatives and n criteria and we have the score of each alternative with respect to each criterion.

1. Let x_{ij} score of alternative i with respect to criterion j
We have a matrix $X = (x_{ij})_{m \times n}$ matrix.

2. Let J be the set of benefit criteria (more is better)
3. Let J' be the set of negative criteria (less is better)

Step 1: Construct normalized decision matrix: This step transforms various attribute dimensions into non-dimensional attributes, which allows comparisons across criteria.

Normalize scores or data as follows:

$$r_{ij} = x_{ij} / (\sum x_{ij}^2)^{1/2} \text{ for } i = 1, \dots, m; j = 1, \dots, n$$

Step 2: Construct the weighted normalized decision matrix:

- Assume we have a set of weights for each criteria w_j for $j = 1, \dots, n$.
- Multiply each column of the normalized decision matrix by its associated weight.
- An element of the new matrix is: $v_{ij} = w_j r_{ij}$

Step 3: Determine the ideal and negative ideal solutions:

- **Ideal solution:**
 $A^* = \{v_1^*, \dots, v_n^*\}$, where
 $v_j^* = \{ \max (v_{ij}) \text{ if } j \in J ; \min (v_{ij}) \text{ if } j \in J' \}$
- **Negative ideal solution:**
 $A' = \{v_1', \dots, v_n'\}$, where $v_j' = \{ \min (v_{ij}) \text{ if } j \in J ; \max (v_{ij}) \text{ if } j \in J' \}$

Step 4: Calculate the separation measures for each alternative.

- **The separation from the ideal alternative is:**
$$S_i^* = [\sum (v_j^* - v_{ij})^2]^{1/2} \quad i = 1, \dots, m$$
 - **Similarly, the separation from the negative ideal alternative is:**
$$S_i' = [\sum (v_j' - v_{ij})^2]^{1/2} \quad i = 1, \dots, m$$
- Step 5: Calculate the relative closeness (C_i^*) to the ideal solution**
$$C_i^* = S_i' / (S_i^* + S_i') , \quad 0 < C_i^* < 1$$

Select the option with C_i^* closest to 1.

7. SUMMARY AND CONCLUSIONS

Smart Sim Selector is a tool which can assist the user in the simulation software selection process. Although the system may seem limited in terms of the number of packages currently evaluated and stored within the database, it still represents an explicit attempt to combat some of the problems involved in simulation software selection.

Smart Sim Selector offers an alternative view of selecting simulation software as it is unbiased and unrelated to any software vendor or supplier. Each simulation package has been evaluated and stored in the database against criteria which covers a variety of issues. The criteria do not favor a single package, and the database will be increased to cover more criteria and simulation packages in the near future.

In addition to further expansion of the database, Smart Sim Selector will be distributed to various educational and commercial institutions involved in simulation, in order to get more feedback on possible improvements. Smart Sim Selector also requires enhancements to be implemented which will allow for the system to actually distinguish between the various levels of criteria that a simulation package can offer. One way to do this would involve creating another data table which takes into account such concepts. For example, after specifying that a statistical distribution fitting mechanism would be required, the user would be able to distinguish how many statistical distributions would be sufficient. The system would then take this into account rather than just offer a package which offered a statistical distribution fitting mechanism but not knowing how many distributions were provided.

Similarly an enhancement is required to allow Smart Sim Selector to take into account user comments at various stages so that the user actually inputs information rather than to select from a number of choices offered by the system. The user should be able to input comments regarding the type of animation that is required, the level of coding. Despite the limitations, Smart Sim Selector represents a step forward to more explicit assistance in the simulation software selection process.

Acknowledgements

Much of the information in this paper was derived from extended discussions with software vendor personnel and automobile manufacturers. The authors gratefully acknowledge the time investments in this project that were generously provided by Mr. Jai Shankar from ProModel Corporation, Mr. Kiran Vonna from CD-adapco India Pvt. Ltd. and Mr. Suru from Escorts Group, Mr. Beth Plott from Alion Science and Technology. We are thankful to Mr. D.N. Prasad from Hero Honda Motors, Mr. Sanjay Gupta from Maruti Suzuki India Ltd., Mr. Dharmendra Upadhyay from Yamaha Motor Solutions India Pvt. Ltd. for sparing their valuable time to help us and making this project successful.

References

- Banks, J., Aviles, E., McLaughlin, J. R., & Yuan, R. C. (1991). The Simulator: New Member of the Simulation Family. *Interfaces*, 21(2): 76-86.
- Banks, J., & Gibson, R. R. (1997). Selecting Simulation Software. *IIE Solutions*, 29(5): 30-32.
- Bovone, M., Ferrari, D. V. and Manuelli, R. (1989). How to Choose an Useful Simulation Software. In D. M. Smith, J. Stephenson, & R. N. Zobel (Eds.), *Proceedings of the 1989*

- European Simulation Multiconference* (pp. 39-43). SCS, San Diego: Society for Computer simulation International.
- Cellier, F. E. (1983). Simulation Software: Today and Tomorrow. In J. Burger, & Y. Jarny (Eds.), *Proceedings of the IMACS International Symposium* (pp. 3-19), Amsterdam: Elsevier Science Publishers.
- Hlupic, V., & Paul, R. J. (1999). Guidelines for Selection of Manufacturing Simulation Software. *IIE Transactions*, 31(1): 21-29.
- Holder, K. (1990). Selecting Simulation Software. *OR Insight*, 3(4): 19-24.
- Law, A. M., & Kelton, W. D. (1991). *Simulation Modeling and Analysis*. Singapore: McGraw-Hill.
- Nikoukaran, J., Hlupic, V., & Paul, R. J. (1999). A Hierarchical Framework for Evaluating Simulation Software. *Simulation Practice and Theory*, 7(3): 219-231.
- Popovic, A., Jaklic, J., & Vuksic, V. B. (2005). Business Process Change and Simulation Modeling. *System Integration Journal*, 13(2): 29-37.
- Seila, A. F., Ceric, V., & Tadikamalla, P. (2003). *Applied Simulation Modeling*. Australia: Thomson Learning.
- Tewoldeberhan, T. W., Verbraeck, A., Valentin E., & Bardonnnet, G. (2002). An Evaluation and Selection Methodology for Discrete-Event Simulation Software. In E. Ycesan, J. L. Snowdon, J. M. Charnes, & J. Wayne (Eds.), *Proceedings of the 2002 Winter Simulation Conference* (pp. 67-75). Boston: Kluwer Academic Publisher.
- Tocher, K. D. (1965). Review of Simulation Languages. *Operational Research Quarterly*, 16(2): 189-217.
- Verma, R., Gupta, A., & Singh, K. (2009). A Critical Evaluation and Comparison of Four Manufacturing Simulation Softwares. *International Journal of Science, Engineering and Technology*, 5(1): 104-120.