# Framework for simulation software selection

Luca Fumagalli[a], Adalberto Polenghi[a]*, Elisa Negri[a], Irene Roda[a]

[a]*Department of Management, Economics and Industrial Engineering, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133, Milan, Italy*

Corresponding author: Adalberto Polenghi

Details: adalberto.polenghi@polimi.it

+39 02399 9267

Via Lambruschini 4/b, 20156 Milan, Italy

Word count: **6453** words

# Framework for simulation software selection

In the last twenty years, the selection of the most suitable simulation software to correctly face industrial problems has become vital for companies. The choice should be made among a number of possibilities and many firms do not have the appropriate tools for selecting the correct one, especially SMEs. Moreover, in recent years, a change of perspective can be detected in industry. In fact, the technological evolution has shifted the problem from the identification of the "best-ever" solution, independently from the goals, to the usability of the software, depending on the objectives the software package is used for. In this context, technicians and managers act no more as receivers of the selection process, but they play as decision-makers.

The present work proposes an update on the criteria traditionally used for evaluating the most appropriate simulation software. Based on that, an AHP-based evaluation framework is defined to support the selection of an alternative among a series of simulation software packages to face the actual industrial problem.

The proposed methodology has been tested on an industrial case and its effectiveness has been proven.

Keywords: word: evaluation framework; software selection; selection criteria; simulation software; simulation

## Introduction

In a world becoming more and more competitive with continuously growing attention to details, the solution to engineering and operations management problems within manufacturing systems must be addressed efficiently, trying to optimise every step of the problem-solving process. Hence, before directly facing the problem, it is important to choose the most suitable tool(s) to address it. The technological growth of the last two decades raises the issue: the right instrument should be found out from thousands of alternatives.

The choice of the right software tool could be highly demanding; thus, many standardised selection methods have been developed for moving in such a crowded field.

In particular, these methods have been applied to the simulation software selection: among software packages, simulation software covers a primary role in terms of complexity for such analysis (Nikoukaran, Hlupic and Paul, 1998). A huge number of different types of simulation packages appeared on the market since the 80s: they mainly differ one from the other in terms of application field, computation capability and easiness in learning and implementation. This differentiation has increased the need to develop software selection criteria and methodologies (Grant, 1988; Banks, 1991; Davis and Williams, 1994; Hlupic and Paul, 1996).

The attention in the last decades has been paid toward the creation of a general framework, whose steps could be utilised by institutions and especially firms to highlight the best simulation software for their goals (Alomair, Ahmad and Alghamdi, 2015).

The present work aims to develop a new evaluation framework to select the most appropriate simulation software and to update the existing selection criteria found through a systematic literature analysis. Such analysis not only concerns the criteria but the paper aims to recall the overall evolution of a specific type of software used for problem-solving in industrial engineering problems, namely simulation software. In this context, the evaluation framework places the firm at the decision-maker level, focusing on the goals the software is used for and not merely on absolute performance.

The state-of-the-art analysis covers three different fields: i) a recap on the evolution of simulation modelling techniques since the 80s; ii) an examination of the selection techniques developed through the years, from the initial and simplified ones to the most elaborate and complex ones; and, finally, iii) a review, built on a systematic literature review, of the criteria used to select the software.

The paper is structured as follows: the first section describes the state-of-the-art analysis, divided into the aforementioned parts ((i) literature review, (ii) evaluation technique review, and (iii) selection criteria review). Secondly, a description of the new evaluation framework is proposed, describing the advantages with respect to the state-of-the-art. Thirdly, the update of the selection criteria by means of the creation of a structured hierarchy is presented. Fourthly, a real application case is described (in three parts: a brief description of the industrial problem, the presentation of the industrial case application of the proposed selection framework and the collection the results of the case). At the end, some conclusions on the proposed simulation software selection framework are drawn, providing some hints for further improvements.

## Literature review

### *Methodology*

In this section, firstly, a review about the evolution of the simulation modelling tools during the last decades is presented. Secondly, the focus is shifted towards the techniques and the criteria proposed in the literature for the selection of simulation software. To this regard, a systematic literature review was set up in order to span in the field of simulation software selection and collect the main contributions about such a topic.

The databases considered for the analysis are ISI-Web of Science and Scopus. The two databases are analysed and screened according to different combinations of the keywords: “simulation software”, “criteri*”, “feature*”, “evaluation”, “selection” and “framework”, that were combined through the logical operators, as presented in Figure 1.

------------------

Insert Figure 1

--------------------

The search involves the main part of the articles and, in particular, the combinations of keywords and operators in order to consider a document as eligible for the analysis must be found in the title or abstract or keywords of the article.

There is a huge number of documents responding to the presented combinations in the databases, as shown in Figure 2.

-----------------

Insert Figure 2

------------------

These documents are further screened in two steps according to different eligibility criteria:

1. First step screening criteria:
    a. English written documents
    b. Not grey literature (materials, usually not peer-reviewed, produced by institutions and organisations outside classical scientific distribution channels, as industrial reports, position papers, and government normative)
    c. Availability of all information (authors, title, journal/conference)
    d. Adherence of title with respect to the goals of the analysis (e.g. application of simulation software to a case study or performance evaluation of vehicle through simulation are not considered)
2. Second step screening criteria:
    a. Availability of the whole document
    b. Presence of selection software criteria

The search is not filtered against specific application area (e.g. engineering, medicine, architecture, education); however, the results show that most of the eligible documents could be applied to different fields and for different applications.

A snowball analysis is also performed to introduce significant references not found by the systematic search in the databases. Even though the described eligibility criteria imply not to include grey literature, two books result to be present in so many works that it seems to the authors mandatory to introduce them in the analysis, as an exception. These books are (Law and Kelton, 1991) and (Banks, Carson and Nelson, 1996).

The whole systematic literature review process is shown in Figure 3.

--------------------

Insert Figure 3

----------------------

***Simulation modelling tools evolution review***

Within the research stream about simulation tools, there are many classifications, and one of the most significant ones is proposed by Banks, which is based on two axes (Banks, 1991):

- on the one hand, the different criteria useful for determining the best option among many simulation tools,
- on the other hand, the distinctions between different classes of the simulation modelling tools: spreadsheets, rapid modelling tools, simulators and simulation languages.

Indeed, the work of Banks, dated back more than 25 years ago, is still of paramount importance for some of the classifications therein provided.

Starting from the most simple simulation form, it is possible to develop spreadsheet-based simulation models taking a minimal amount of time, because the system under consideration is simple; the output is of graphical nature, and usually, the results are graphs in a very attractive form. Examples are MS Excel (https://products.office.com/it-it/excel) and Gnumeric (http://www.gnumeric.org/).

Rapid modelling tools help the decision maker to have an idea about some important performance measures, related to throughput or bottleneck; it represents one step ahead with respect to a spreadsheet from the implementation complexity point of view. An example of rapid modelling tool building is proposed by (Suri and Tomsicek, 1990), based on ManuPlan/SimStarter by Network Dynamics, Inc.

Within simulators, it is possible to build data-driven simulations, which requires no programming for simple models even though for complex system it is mandatory to introduce compiled code; the outputs of these tools are very impressive and provide a first interface with spreadsheets, giving the possibility to export the final results into a form that could be easily elaborated. On the market the main examples are represented by WITNESS (https://www.lanner.com/technology/witness-simulation-software.html), Plant Simulation by Siemens (https://www.plm.automation.siemens.com/it/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml), Arena by Rockwell Automation (https://www.arenasimulation.com/), and SimFactory 11.5 (Goble, 1991).

Simulation languages are the most comprehensive tools because in principle they allow modelling any system; unfortunately, they require a very large time and resource investment and well-skilled people, because the creation of a simulation model within a simulation language requires coding it. The output goes from standardised to tailor-made.

The problem of choosing the right simulation tool seems to become hard, as it requires to identify the right tool complexity for the system at hand. Nevertheless, the first two classes (spreadsheets and rapid modelling tools) are not capable of representing the production system complexity. Thus their use in the real industrial environment is really limited, except for drawing simple graphical outputs. In the end, the two remaining classes are simulators and simulation languages (Law and Kelton, 1991), but the distinction is blurring.

Simulation languages have evolved by introducing features typically associated with simulators, as material handling characteristics, drop-down menus; in the meanwhile, simulators have moved toward the first ones by introducing the possibility to code and the association of attributes to an entity (Banks, 1991).

(Nikoukaran and Paul, 1999) states that improvements in the facilities available in simulators made them increasingly powerful, flexible and user-friendly.

For this reason, it could be stated that simulators and simulation languages have merged into one only class of “simulation software”, among which there are different families, each of them with its own characteristics: some are moving toward the original simulators features, focusing on the representation of the system; others, closer to the simulation languages, are improving the capabilities of describing every possible characteristic of the problem (Law and Kelton, 1991).

For this reasons some of the previously presented simulation tools do not exist anymore (e.g. ManuPlan/SimStarter by Network Dynamics, Inc.) or are completely changed (WITNESS is today more powerful than the original simulator version).

### *Software evaluation technique review*

The goal of this analysis is to review evaluation tecniques that could be able to address the actual problem, and to have a look to the state-of-the-art about their utilisation in scientific literature.

In the literature, the first contributions on the topic regard the development of simple weighted scoring methods, in which weights and marks are assigned to criteria and software, respectively (Banks, 1991). Nevertheless, from the review it emerged that these methods are different for each researcher that has tried to build them, and a common structure is not possible to be extrapolated.

Starting from 1994 the evaluation techniques became more and more studied, and interesting results came in. Vendors were also sometimes included in the studies about the selection process, but the strong bias during the evaluation was already noticed due to the tendency of each software supplier to sell its own package (Hlupic and Paul, 1996). Therefore, the following selection processes did not consider the vendors, leaving the negotiation part to the final user of the software.

The AHP, Analytic Hierarchy Process (Saaty, 1990), becomes the first structured method used and applied in problems of selecting the best option among software alternatives (Davis and Williams, 1994).

In the middle of 90s, a software called SimSelect (Hlupic and Mann, 1995) is developed, with the only aim of solving such a problem. The tool imposes to rank the criteria at three different levels, namely “high”, “medium”, and “low” and then, for each software, to provide a “binary” evaluation for each criterion (yes/not, possible/not possible, provided/not provided). The software automatically provides the best candidate and, if the option is enabled, also the first two best alternatives.

More recently the researchers have started to develop Fuzzy Multi-Criteria Decision-Making Process (Fuzzy-MCDM) in order to take into account possible errors and mistakes in the association of weights to a criterion or to overpass the problem of being undecided about what is the appropriate score of a software (Cochran and Chen, 2005; Azadeh, Shirkouhi and Rezaie, 2010).

The most used multi-criteria decision-making technique remains the AHP (Jadhav and Sonar, 2009), even though many methods have been created during the years to overpass it as FAHP (Fuzzy Analytic Hierarchy Process), TOPSIS (Technique for Order Preference by Similarity to Ideal Solution), PSI (Preference Selection Index) (Alomair, Ahmad and Alghamdi, 2015) and ELECTRE (Rohaninejad *et al.*, 2015).

The AHP demonstrates to be the most widely adopted evaluation technique also within the eligible documents, as presented in Figure 4.

-------------------

Insert Figure 4

---------------------

In the picture, a distinction is made between a hierarchical approach and AHP. A hierarchical approach includes any evaluation technique that considers a decomposition of the list of criteria into a hierarchy and assigns to each of them a weight (an indication of relative importance between the criteria) and a score (performance of the simulation software with respect to each criterion). The AHP is instead a well structured and defined methodology for the evaluation, so every document that describes a very general approach to evaluate the hierarchy is classified as “hierarchical approach” while the ones clearly using AHP are classified as “analytic hierarchy process”. Some eligible documents do not specify, together with the selection criteria, the evaluation technique.

Some others instead describe different options to evaluate the simulation software, as highlighted in Figure 4 by the voice “multiple”. However, AHP is always considered as a valid option in every of such a kind of paper, so the real number for the “analytic hierarchy process” may grow significantly.

***Selection criteria review***

One of the first works about simulation software selection proposes only few criteria with very general meaning, which are the following. Effective User Interface, Implemented Algorithm, Interfacing with External Software, Output Reports, and Ease of Use (Grant, 1988). The underlying reason for such a low number of criteria could be associated with the recent development of simulation techniques and the fact that the capability and type of help that they could provide were not already completely known. The in-depth creation of the hierarchy belonged to the decision maker, who knows the production system.

Even though this is true, having a comprehensive overview of the overall possible useful criteria could be of big advantage when creating the hierarchy for the decision-making process. This need has been converted in works that are considered seminal for selection criteria also nowadays, such as (Hlupic, Irani and Paul, 1999; Nikoukaran and Paul, 1999).

The first tentative of creation of a wide comprehensive classification of criteria is made in 1999 and this study brings to the definition of important selection criteria categories as Vendor, Model and Input, Execution, Animation, Testing and Efficiency, Output and User (Nikoukaran and Paul, 1999), recalled and adopted also by (Azadeh, Shirkouhi and Rezaie, 2010; Pezzotta *et al.*, 2016).

The same year a more general analysis was built, including more than two hundred criteria; among them, the most useful features of the simulation software for the actual problem could be found (Hlupic, Irani and Paul, 1999). In this latter work, the authors studied very carefully the shape of the general framework used for assessing the software selection and this work is also considered seminal for future works from this point of view, too. All the subsequent works consider the above hierarchy of criteria as the most comprehensive one (Stewart, 2004) and they operate a reworking of it to better fit with their actual case study. (Bosilj-Vuksic, Ceric and Hlupic, 2007) propose business simulation tool criteria that involve four main categories: Hardware and Software Considerations, Modelling Capabilities, Simulation Capabilities, and Input/Output Issues, whose sub-categories are recovered by (Hlupic, Irani and Paul, 1999).

Other works tried to improve the criteria to select the most adequate simulation software, and a summary of them is presented in Table 1. It reports the criteria used by the authors of the eligible documents to evaluate and select the most suitable simulation software. It must be remarked that, apart from the very first papers about this topic, a hierarchical structure of criteria is almost always used. This structure is consistent with the adoption of the AHP as the evaluation technique.

Table 1 aims to be as comprehensive as possible, but, due to a matter of space, there are some works whose number of criteria is too large to be all reported, also considering the first level of the criteria hierarchy.

Generally, for a complete look of the selection criteria, the suggestion is to refer to the original paper.

---------------------

Insert Table 1 (composed of 4 parts)

--------------------

The research literature activity has never slowed down on the software selection topic, also considering the development of new software packages and the technological growth, as demonstrated by Figure 5 that proposes an overview of the number of papers per year.

---------------

Insert Figure 5

-----------------

Especially the technological growth has shifted the selection problem from the best-performing software in absolute terms (i.e. without any look to the final goal it aims at solving), as the computational speed, to the application of the technology itself, focusing on the specific objectives the company is attempting to reach.

The usability of the software package in accordance with the firm objective is nowadays the real discriminant between different software alternatives.

This approach will also favour the SME (Small and Medium Enterprises): the selection of the software is becoming a tool which takes care of the real needs, helping in selecting software that is usable in the real environment where the SME work.

**Conclusions**

Literature analysis underlined that there is not a unique way of selecting the most suitable simulation software, even though some general frameworks were developed in order to provide at least the steps of the evaluation process.

Moreover, even though the use of simulation software is widespread in industrial engineering companies, the criteria and methods proposed in the scientific literature to evaluate it are proposed by the academic world, thus not providing the proper help for a company whose aim is to enhance the efficiency of its production system.

In particular, the methodologies present some lacks, such as a common list of generic software evaluation criteria and their meaning, and a framework comprising software selection methodology, evaluation technique, evaluation criteria (Jadhav and Sonar, 2009).

These gaps have been partially filled by (Azadeh, Shirkouhi and Rezaie, 2010), which proposes a questionnaire-based methodology, and by (Tewoldeberhan, Verbraeck and Hlupic, 2010), which instead presents a two-phase evaluation framework tested and applied at Accenture.

The need for industrial validation is continuously remarked in the recent literature to emphasise the pragmatic approach academic research should have to enhance the validation of the newly proposed models, as described by (Garousi, Petersen and Ozkan, 2016) in the software engineering field.

The actual research tries to bridge the gap between the evaluation criteria and methods proposed by academia and those relevant for industry.

From the outcomes of this work, the research institutions could take advantage from the study of a real case and from the proposal of a structured methodology to identify the evaluation criteria, whereas the company acquires new know-how in terms of software utilisation.

In the next section, the structured methodology for the interaction of these two parts is proposed. It is based on meetings, a questionnaire, and data elaboration that, together with a suitable evaluation technique, allow to come up with a final decision about the best simulation software in a manufacturing company.

**Proposed evaluation framework**

The proposed AHP-based evaluation framework for simulation software selection is a three-step decision-making process: Hierarchies Generation, Evaluation of Alternatives, and Identification of the Best Option.

The first two steps are composed of different sub-steps; in particular:

- Hierarchies generation is composed of Research Approach Hierarchy, Industrial Application Hierarchy, and Integrated Hierarchy;
- Evaluation of Alternatives is constituted by Screening, AHP Application (Saaty, 1990), and Sensitivity Analysis.

A summarising graph is presented in Figure 6.

---------------------

Insert Figure 6

--------------------

As it is possible to notice, the proposed framework has been designed to provide an agile approach to the selection of simulation software. It is composed of a sequence of steps that could be customised when needed, depending on the specific application and/or on the industrial problem the simulation software should face. Indeed, the importance of choosing the simulation software on the specific problem objectives is remarked in the literature (Taylor and Robinson, 2006).

***Methodology description***

The evaluation methodology above described should be implemented in ordered steps (with the eventual exception of the Research Approach Hierarchy and Industrial Application Hierarchy that may be performed in parallel).

*Hierarchies generation*

During this phase of the methodology, the hierarchies should be created, with the purpose of using the AHP approach to evaluate the best candidate among the possible simulation software packages. In particular:

- Research Approach Hierarchy: this hierarchy should represent the important criteria from the point of view of the research application, for example in terms of code manipulation or statistical analysis (see following section for the proposed selection criteria and the updated ones);

- Industrial Application Hierarchy: this hierarchy should be generated by the firm interested in introducing a simulation software in the IT infrastructure; during its creation, the technicians of the company could be led by means of questionnaire during meetings, for example, in case the company has not already developed an internal simulation know-how (especially referring to small and medium-sized enterprises - SME);
- Integrated Hierarchy: the generation of this hierarchy should be done respecting the criteria highlighted by both parties; for this reason, a simple four-step approach is promoted to enhance the merging process:
    1. Comparison of the hierarchies to detect common criteria, similar criteria and non-common criteria;
    2. Common criteria are inserted once into the final hierarchy, respecting the relative position: a second-level criterion is placed in the final hierarchy at the second-level; a third-level criterion at the third level of the final hierarchy and so on;
    3. Similar features are merged into the final hierarchy and allocated to the relative position;
    4. Different criteria are all inserted into the final hierarchy respecting the relative position.

The questionnaire used during the drawing up of the Industrial Application Hierarchy, should be built in order to facilitate the merging process.

*Evaluation of alternatives*

This phase is fundamental to find out which is the best option among all the possibilities taken into account. To ease the process, the Evaluation of Alternatives step is further divided into three sub-steps:

- Screening: an initial selection of a group of well-performing packages with respect to objectives is done (a low number of objectives is enough, in order to have an easy evaluation); this is useful to concentrate the attention on the most appropriate software (for this sub-step it is recommended to use a simple technique, such as a weighted scoring method); some help could be provided by the Institute of Operations Research and Management Sciences (OR/MS Today - http://www.orms-today.org/), which publishes every two years a simulation software survey, which contains also a first evaluation of some features considered important by the scientific community (Swain, 2015), thus becoming a very important source of software to be considered in the analysis (Banks *et al.*, 2010). In addition, a very discrete-event software ranking has been proposed recently (Dias *et al.*, 2016), which could provide further help during the Screening phase;
- AHP Application: AHP methodology (Saaty, 1990) is applied in order to determine the relative importance of each criterion with respect to another one; recalling briefly the method, it is composed of mainly two phases:
    1. Determination of the relative importance of each criterion with respect to the other one/s of the same level;
    2. Assignment of the scores of each simulation software package with respect to each criterion (the power of the AHP lies in being able to include both quantitative and qualitative scores);

- Sensitivity Analysis: in order to validate the robustness of the solution against a change of a set of weights associated to the criteria; this will help in understanding if an error will strongly influence the final candidate or if the solution is still confirmed.

*Identification of best option*

Once the methodology comes to its end, it should be clear which is the most suitable candidate among the studied group of software packages for the actual application in a given firm.

***Advances with respect to the state-of-the-art***

The proposed simulation software selection framework aims to overpass the previous methods in terms of easiness and to enable SME to approach high-level software packages.

In fact, SME usually do not have the economic possibility to try different solutions, as done by (da Silva and Botter, 2009). For this reason, many companies focus on the evaluation among OSS (Open Source Software) (Aversano and Tortorella, 2013) and this trend has increased in time (Franco-Bedoya *et al.*, 2017).

Considering the actual economic and technological moment of Industry 4.0 paradigm, the simulation will become more and more central in industrial systems modelling and controlling (Negri, Fumagalli and Macchi, 2017), thus providing importance also to the tool of selection of the most suitable software package (Ferreira *et al.*, 2017).

The technological growth does not only bring simulation as central, but leads also to think about the integration of this methodology in a wider perspective of design and management of assets, as production systems (Roda and Macchi, 2018). Such integration might not succeed if the research is stuck on selecting the “best-ever” solution for a certain domain, intended as the top solution in the market for that specific domain (i.e. design). In fact, the usability of the software in the specific company’s context and to support its decision-making processes is crucial. Thus, the focus is on “usable” simulation software according to specific company conditions or purposes. The SME might not afford best-in-the-field simulation software, so they can look at other possibilities on the market that balance cost and performance, focusing on the most important selection criteria for their case and itended use. For example, asset-intensive industries look for simulation software to support both systems design and management decisions, thus the purposes push towards highly integrated and customizable solution that may not always coincide with one simulation ecosystem from branded software that is specialized on one specific domain.

**Selection criteria**

Considering the eligible documents from the literature review, the list of criteria is made, including all the possible features that the software could have in order to enhance and facilitate the industrial engineering work.

Moreover, two new high-level criteria are introduced: Implementation and Validation.

The former criterion collects all the important features for coding the model in the simulation software starting from the most general ones as technical characteristics to very precise requirements in programming aspects (in particular, the considered sub-criteria are: General Characteristics, Technical Characteristics, Visual Aspects, Programming Aspects, Efficiency, Support).

The latter is fundamental for assessing that the software is able to provide significant results, providing a robust basis on which the decision maker chooses the best solution. This criterion is instead composed of all the features related to Testing, Compatibility, Input/Output and Experimentation.

All the selection criteria are presented in Table 2. The table is structured as follows: the first column indicates the high-level criterion (Implementation (I) or Validation (V)); the second column lists the second-level criteria; the third column list the third-level criteria (or leaf criteria); the last column describes what each criterion means.

As understandable by the table layout, not only a list of selection criteria is proposed, but also an already built hierarchy is here presented to favour future decision makers to create their own hierarchy. Nevertheless, the list of criteria should be intended as a list so that a complete new problem-specific hierarchy could be created with the same items.

---------------------

Insert Table 2

--------------------

The selection criteria hierarchy is the one used in the following section for the application to a real case study, and so it represents the basis to build the Research Approach Hierarchy proposed by the research group to the firm, and it contributes to the creation of the Integrated Hierarchy.

**Validation through real case**

The industrial case is about a company producing forged and laminated rolled rings of several kinds of materials, from carbon and alloyed steels to nickel, titanium, cobalt alloys, aluminium and copper. Company’s products are aimed at industrial sectors such as aerospace, Oil&Gas, power generation and general mechanics.

The Company wants to remain unknown for privacy issues; thus from now on it is addressed to as Company.

### *Industrial problem statement*

The general objective of the industrial project that is the industrial frame for the developed evaluation methodology validation is the generation of an algorithm to improve the effectiveness of the scheduling activities, by means of the adoption of a software, capable to be perfectly integrated with the managerial software owned by the Company, as described in (Fumagalli *et al.*, 2018).

The first step, before starting the development of the algorithm, is to choose the most suitable software capable of representing as close as possible the production constraints, identified to be the critical issue in the development of the project. This is the reason why the decision is to select a simulation software, in order to be able to model the complexity of the system at the shop-floor level.

### *Application of the proposed simulation software selection framework*

The next sections expose the application of the afore-described proposed framework for simulation software selection to the real case.

#### *Hierarchies generation*

Following the steps previously defined, firstly the Research Approach Hierarchy and Industrial Application Hierarchy must be generated and then the Integrated Hierarchy has to be derived.

For the sake of shortness, the hierarchies are represented in the next by means of diagrams. For what concerns the Research Approach Hierarchy (Figure 7), it has the same structure and criteria of Table 3, except for neglecting two criteria in Implementation-General Characteristics criterion, that are Model Reusability and Hybrid Modelling.

Model Reusability is excluded because the industrial project deals with a single use case and the Company is not interested in enlarging the scope of the project. Hybrid Modelling is neglected because the system is simplified since the problem (i.e. scheduling problem) does not require deeper investigation and modelling of the behaviour of different parts of the system. Industrial Application Hierarchy is instead generated after Company filled up a questionnaire (created ad hoc by the research group) and the resulting diagram is presented in Figure 8. The two hierarchies present common, similar and different criteria: these last ones are going to be analysed separately in order to bring to a full understanding, while for the first two types refer to Table 2.

The Integrated Hierarchy (Figure 9, and Table 4 for a tabular form) is built considering the four-step approach exposed in the description of the proposed evaluation framework.

One of the most critical phases in the evaluation framework is the interaction with the firm for creating its own hierarchy, including the important features from the Company's point of view.

This is critical because usually, during the development of the project, some new interesting criteria emerged, as desired by the firm, such as the possibility to interface the simulation software to external packages in order to connect their internal management software with the future simulation package.

--------------------

Insert Figure 7

--------------------

A formalisation becomes in this way mandatory, in order to promote the introduction of their requests into the already prepared hierarchy, a questionnaire is developed and asked during a meeting. The document has the goal to formalise and precisely list all the features of the software package the firm is interested in.

The development of the questionnaire proceeds taking always into account three elements:

- Criteria derived from the literature analysis;
- Features expressed by the Company in previous contacts;
- Empty spaces to include new criteria.

The Company was asked the questionnaire in a session in which experts from different departments were present, namely IT management, technical department, and Operations departments. With the support of the academic team, the Company was led step by step through the questionnaire, explaining each time the questions and how to correctly fill it. Figure 3 presents the resulting Industrial Application Hierarchy.

---------------------

Insert Figure 8

--------------------

As it is clear from Figure 3, some of the criteria are already present in the Research Approach Hierarchy. Thus a further explanation is not necessary (such as Documentation, User-friendly Interface, Code Manipulation, and Hardware Compatibility). Others are proposed by the Company and, among them, some are self-explaining (Tutorial, Transfer to Other Packages, Application Field, Import from Other Packages, Memorise in the Database, Specific Output, Export to Other Packages, Trial Version) and for the rest a brief insight is proposed in order to allow a full understanding of the hierarchy:

- Simple graphics: the output should be easy to be understood in order to facilitate its comprehension by every manager or production responsible in the firm;

- Maturity: a software whose license or functioning often changes is considered as a lack of maturity;
- Precision: the software must have algorithms capable of finding the solution in a precise way, avoiding the possibility of missing it;
- Modularity: it is similar to that expressed by "Union of Different Models" in the Research Approach Hierarchy, indicating the need to have a hierarchical and more manageable model;
- Personalized unit time: the time is not constrained to be in minutes, hours or days, but it will be set depending on the problem;
- Quick response: having the same algorithm running, the software should compute the results in a faster way than the other ones.

---------------------

Insert Figure 9

--------------------

*Evaluation of alternatives*

The first sub-step (Screening) of this main phase deals with the reduction of the number of possible candidates by means of a weighted scoring method. Some criteria must be selected, and they should be of easy evaluation, and the score may be assigned also by non-expert people in simulation because the information is all included in the online documentation for each software or in some specialised web forums could be consulted.

The Screening for the actual industrial case was set up to evaluate the software packages with respect to three different objectives: Simulation Types (DES – Discrete Event Simulation -, CTS – Continuous Time Simulation - or both), Application Field (general or Manufacturing-oriented), and Data Elaboration (possible or not).

The simulation software candidates are selected considering authors' knowledge and reviews present on the web; the candidates are presented in Table 2. It should be noted that this is not a thorough list since it derives from an Industrial Engineering background of the authors, so it may be biased, even though the methodology is valid. Ideally, the decision-maker should go through all of the possible simulation softwares on the market.

---------------------

Insert Table 3

--------------------

After this phase, the final candidates are: Arena, Lanner, MatLab&Simulink, Plant Simulation, and MapleSim.

Once the Screening is completed, the AHP could be applied. The first step is to complete the judgements matrices (refer to (Saaty, 1990) for the overall procedure and nomenclature): this step is done together with the Company departments involved (IT, technical, and Operations) in order to give priority to the most important criteria. The final weights for each level of criteria are reported in Table 4.

With the collaboration of the afore-mentioned Company departments, each simulation software receives a score for each criterion (for privacy issue these values are not shown).

Finally, the application of the AHP-based methodology figures out the best candidate for the actual industrial case.

---------------------

Insert Table 4

--------------------

To assess the robustness of the solution, a sensitivity analysis has been performed by changing the weights of the main criteria. The result of this analysis has shown that the first place belongs always to the identified best simulation software and the only change happens between the second and third places that are switched (see Figure 10).

The upper part of Figure 10 represents a changing of the weights in order to favour Research Approach criteria, so Efficiency and Programming Aspects criteria weights have been increased separately (+5% and +10%). At the bottom part of the figure, there is a change in the parameter towards the Company perspective: Users criterion is enforced against Implementation and Validation.

---------------------

Insert Figure 10

--------------------

***Industrial case results***

The industrial case regards not only a simple representation and testing of the actual system but also a demanding data elaboration activity, both for input and output data. This goal has brought to the choice of the best simulation software for the specific industrial application, whose configuration is capable of satisfying all the needs the Company has shown and explained during the meetings. Due to the privacy agreement and not to promote any of the software, the best candidate for the specific application case is not explicitly named here.

## Conclusions

The use of a structured framework to choose the best software is important for succeeding in reaching the final goal. This is valid also for what concerns the selection of the most suitable simulation software for solving manufacturing-related issues.

The proposed framework fills the three gaps highlighted by (Jadhav and Sonar, 2009): (i) AHP application, a common list of selection criteria, and a framework comprising the methodology for the software selection, (ii) evaluation technique, evaluation criteria, and (iii) a system to assist decision makers. Thus it represents a natural evolution of the most recent methodologies.

In particular, this methodology empowers and highlights the need for goals satisfaction as key-driver for the software selection rather than a method able to select the best software package in absolute terms (e.g. computational speed). In this way, the goals are dictated by the company, but a tool is necessary for selecting the most suitable simulation software with respect to their own objectives, either strategic or shop-floor related.

The developed software selection framework is proved to be solid during its application in an industrial case, and the firm recognised its ability in reaching the best option among several candidates. The solid structure helped in establishing a successful relationship with the firm that played a central role in the framework: this helped in starting a clear and true dialogue, which enhances the selection process with a continuous flow of new possibilities to be evaluated, both in terms of software and in terms of selection criteria. The novel framework claims to pave the way for better communication and cooperation between firms and academic institutions, enlightening a common field on which both are interested in moving.

In addition, its simplicity and effectiveness are enablers to involve also the world of SME, which are reluctant in using high-level simulation software, especially due to the difficulty in choosing the suitable one (Franco-Bedoya *et al.*, 2017). Moreover, SME could benefit from the application of this powerful framework because it does not require a big initial expenditure (evaluation and implementation of the most suitable simulation software are distinct): in the presented industrial case, four meetings were needed to select the software, without any costs except for the one related to the man hours (technicians and managers).

Going one step ahead, this evaluation framework may be of big help in the Industry 4.0 era, where simulation is central in operations management and control, but also with respect to higher scope as the creation of cyber-physical systems (Negri *et al.*, 2018).

Next improvements and steps will be focused on the extension of the proposed simulation software selection framework to different kinds as managerial or manufacturing control ones in order to become an adaptable framework capable of helping different types of industries and problems.


## Funding details

The presented work was not funded by any organisation/institution.


## Disclosure agreement

No potential conflict of interest was reported by the authors.

## References


Alomair, Y., Ahmad, I. and Alghamdi, A. (2015) ‘A review of evaluation methods and techniques for simulation packages’, *Procedia Computer Science*. Elsevier Masson SAS, 62(Scse), pp. 249–256. doi: 10.1016/j.procs.2015.08.447.

Alvarez, M., Rincón, G., Pérez, M. and Hernández, S. (2008) ‘Evaluation and selection of discrete-event simulation software for the oil industry’, *Latin American Applied Research*, 38(4), pp. 305–312.

Aversano, L. and Tortorella, M. (2013) ‘Quality evaluation of floss projects: Application to ERP systems’, *Information and Software Technology*. Elsevier B.V., 55(7), pp. 1260–1276. doi: 10.1016/j.infsof.2013.01.007.

Azadeh, A., Shirkouhi, S. and Rezaie, K. (2010) ‘A robust decision-making methodology for evaluation and selection of simulation software package’, *The International Journal of Advanced Manufacturing Technology*. London, 47(1), pp. 381–393. doi: 10.1007/s00170-009-2205-6.

Azadeh, M. and Shirkouhi, S. (2009) ‘Evaluating Simulation Software Using Fuzzy

Analytical Hierarchy Process', in *Proceedings of the 2009 Spring Simulation Multiconference*, p. 41.

Banks, J. (1991) 'Selecting simulation software', pp. 15–20.

Banks, J., Aviles, E., McLaughlin, J. R. and Yuan, R. C. (1991) 'The simulator: new member of the simulation family', *Interfaces*. INFORMS, 21(2), pp. 76–86.

Banks, J., Carson, J. S. and Nelson, B. L. (1996) 'Discrete-event system simulation'. Prentice-Hall.

Banks, J., Nelson, B. L., Carson, J. S. and Nicol, D. M. (2010) 'Discrete-Event System Simulation', *Prentice-Hall international series in industrial and systems engineering*, p. 640. doi: 10.2307/1268124.

Bard, J. F., Desilva, A. and Bergevin, A. (1997) 'Evaluating simulation software for postal service use: Technique versus perception', *IEEE Transactions on Engineering Management*, 44(1), pp. 31–42. doi: 10.1109/17.552806.

Bosilj-Vuksic, V., Ceric, V. and Hlupic, V. (2007) 'Criteria for the evaluation of business process simulation tools', *Interdisciplinary Journal of Information, Knowledge, and Management*, 2, pp. 73–88. doi: 10.1.1.103.9869.

Cochran, J. K. and Chen, H.-N. (2005) 'Fuzzy multi-criteria selection of object-oriented simulation software for production system analysis', *Computers & Operations Research*, 32(1), pp. 153–168. doi: http://dx.doi.org/10.1016/S0305-0548(03)00209-0.

Davis, L. and Williams, G. (1994) 'Evaluating and Selecting Simulation Software Using the Analytic Hierarchy Process', *Integrated Manufacturing Systems*. MCB UP Ltd, 5(1), pp. 23–32. doi: 10.1108/09576069410050314.

Dias, L. M. S., Oliveira, J. A., Vieira, A. A. C. and Pereira, G. A. B. (2016) 'Discrete simulation software ranking--A top list of the worldwide most popular and used tools', in *Winter Simulation Conference (WSC)*, pp. 1060–1071.

Dorado, R., Gómez-Moreno, A., Torres-Jiménez, E. and López-Alba, E. (2014) 'An AHP Application to Select Software for Engineering Education', *Computer Applications in Engineering Education*, 22(2), pp. 200–208. doi: 10.1002/cae.20546.

Ejercito, P. M. and Nebrija, K. G. E. (2017) 'Traffic Simulation Software Review', *Information, Intelligence, Systems & Applications (IISA), 2017 8th International Conference on. IEEE*, pp. 1–4.

Ferreira, F., Faria, J., Azevedo, A. and Marques, A. L. (2017) 'Industry 4 . 0 as Enabler for Effective Manufacturing Virtual Enterprises', *Workin Conference on Virtual Enterprises*, pp. 1–7.

Franceschini, R., Bisgambiglia, P., Touraille, L., Bisgambiglia, P. and Hill, D. (2014) 'A survey of modelling and simulation software frameworks using Discrete Event System Specification', *OASIcs-OpenAccess Series in Informatics*, 43(1), pp. 40–49.

Franco-Bedoya, O., Ameller, D., Costal, D. and Franch, X. (2017) 'Open source software ecosystems: A Systematic mapping', *Information and Software Technology*. Elsevier B.V., 91, pp. 160–185. doi: 10.1016/j.infsof.2017.07.007.

Fumagalli, L., Negri, E., Sottoriva, E., Polenghi, A. and Macchi, M. (2018) 'A novel scheduling framework: integrating genetic algorithms and discrete event simulation', *International Journal of Management and Decision Making*, 17(4), pp. 371–395.

Garousi, V., Petersen, K. and Ozkan, B. (2016) 'Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review', *Information and Software Technology*. Elsevier B.V., 79(September 2014), pp. 106–127. doi: 10.1016/j.infsof.2016.07.006.

Goble, J. (1991) 'Introduction to SimFactory 11.5', in *Proceedings of the 1991 Winter Simulation Conference*, pp. 77–80.

Grant, F. H. (1988) 'Simulation in designing and scheduling manufacturing systems', *Design and analysis of integrated manufacturing systems*. National Academies Press, pp. 134–147.

Gupta, A. (2014) 'How to Select A Simulation Software', *International Journal of Engineering Research and Development*, 10(3), pp. 35–41. Available at: https://www.aad.org/spot-skin-cancer/learn-about-skin-cancer/prevent-skin-cancer/how-to-select-a-sunscreen.

Gupta, A., Singh, K. and Verma, R. (2010) 'A critical study and comparison of manufacturing simulation softwares using analytic hierarchy process', *Journal of Engineering Science and Technology*, 5(1), pp. 108–129.

Gupta, A., Verma, R. and Singh, K. (2009) 'A critical evaluation and comparison of four manufacturing simulators using analytic hierarchy process', *International Journal of Engineering*, 8(1), pp. 205–216. doi: 10.1080/00207549508904843.

Hincu, D. and Andreica, M. (2012) 'The evaluation and selecting process for simulation software using fuzzy sets', *Metalurgia International*, 17(3), pp. 141–144.

Hlupic, V., Irani, Z. and Paul, R. J. (1999) 'Evaluation Framework for Simulation Software', *The International Journal of Advanced Manufacturing Technology*. London, 15(5), pp. 366–382. doi: 10.1007/s001700050079.

Hlupic, V. and Mann, A. (1995) 'SimSelect: a system for simulation software selection', pp. 720–727. doi: 10.1145/224401.224718.

Hlupic, V. and Paul, R. J. (1996) 'Methodological approach to manufacturing simulation software selection', *Computer Integrated Manufacturing Systems*, 9(1), pp. 49–55. doi: http://dx.doi.org/10.1016/0951-5240(95)00037-2.

Hlupic, V. and Paul, R. J. (1999) 'Guidelines for selection of manufacturing simulation software', *IIE Transactions (Institute of Industrial Engineers)*, 31(1), pp. 21–29. doi: 10.1080/07408179908969802.

Jadhav, A. S. and Sonar, R. M. (2009) 'Evaluating and selecting software packages: A review', *Information and Software Technology*, 51(3), pp. 555–563. doi: http://dx.doi.org/10.1016/j.infsof.2008.09.003.

Jadhav, A. S. and Sonar, R. M. (2011) 'Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach', *Journal of Systems and Software*. Elsevier, 84(8), pp. 1394–1407. doi: 10.1016/J.JSS.2011.03.034.

Kochhar, A. K. and Ma, X. (1989) 'Discrete event simulation software tools for the simulation of advanced manufacturing systems', in *Proceedings of the 1989 European Simulation Conference, Italy, Simulation Computer Society, San Diego*, pp. 13–18.

Kuljis, J. (1996) ‘HCI and simulation packages’, in *Proceedings of Winter Simulation Conference, 1996*, pp. 687–694.

Law, A. M. and Haider, S. W. (1989) ‘Selecting simulation software for manufacturing applications’, *Proceedings of the 1989 Winter Simulations Conference,* pp. 29–32. doi: 10.1109/WSC.1989.718658.

Law, A. M. and Kelton, W. D. (1991) *Simulation modeling and analysis*. McGraw-Hill New York.

Mackulak, G. T., Cochran, J. K. and Savory, P. A. (1994) ‘Ascertaining important features for industrial simulation environments’, *Simulation*. Sage Publications Sage CA: Thousand Oaks, CA, 63(4), pp. 211–221.

Negri, E., Fumagalli, L., Cimino, C. and Macchi, M. (2018) ‘FMU-supported simulation for CPS Digital Twin’, in *Procedia Manufacturing in CARV International Conference on Changeable, Agile, Reconfigurable and Virtual Production. Nantes, 8th – 10th October 2018*.

Negri, E., Fumagalli, L. and Macchi, M. (2017) ‘A Review of the Roles of Digital Twin in CPS-based Production Systems’, *Procedia Manufacturing*. The Author(s), 11(June), pp. 939–948. doi: 10.1016/j.promfg.2017.07.198.

Nikoukaran, J., Hlupic, V. and Paul, R. (1998) ‘Criteria for simulation software evaluation’, *30th conference on Winter simulation*, pp. 399–406. Available at: http://dl.acm.org/citation.cfm?id=293256.

Nikoukaran, J., Hlupic, V. and Paul, R. J. (1999) ‘A Hierarchical framework for evaluating simulation software’, *Simulation Practice and Theory*, 7(3), pp. 219–231. doi: 10.1016/S0928-4869(98)00028-7.

Nikoukaran, J. and Paul, R. J. (1999) ‘Software selection for simulation in manufacturing: a review’, *Simulation Practice and Theory*, 7(1), pp. 1–14. doi: http://dx.doi.org/10.1016/S0928-4869(98)00022-6.

Pezzotta, G., Rondini, A., Pirola, F. and Pinto, R. (2016) ‘Evaluation of Discrete Event Simulation software to design and assess service delivery processes’, *Service Supply Chain Systems: A Systems Engineering Approach*, 8(86).

Rincon, G., Alvarez, M., Perez, M. and Hernandez, S. (2005) ‘A discrete-event simulation and continuous software evaluation on a systemic quality model: An oil industry case’, *Information and Management*, 42(8), pp. 1051–1066. doi: 10.1016/j.im.2004.04.007.

Roda, I. and Macchi, M. (2018) ‘A framework to embed Asset Management in production companies’, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(4), pp. 368–378. doi: 10.1177/1748006X17753501.

Rohaninejad, M., Kheirkhah, A., Fattahi, P. and Vahedi-Nouri, B. (2015) ‘A hybrid multi-objective genetic algorithm based on the ELECTRE method for a capacitated flexible job shop scheduling problem’, *The International Journal of Advanced Manufacturing Technology*. London: Springer, 77(1), pp. 51–66. doi: 10.1007/s00170-014-6415-1.

Saaty, T. L. (1990) ‘How to make a decision: The analytic hierarchy process’, *Desicion* *making by the analytic hierarchy process: Theory and applications*, 48(1), pp. 9–26. doi: http://dx.doi.org/10.1016/0377-2217(90)90057-I.

Sahay, B. S. and Gupta, A. K. (2003) ‘Development of software selection criteria for supply chain solutions’, *Industrial Management & Data Systems*. MCB UP Ltd, 103(2), pp. 97–110. doi: 10.1108/02635570310463429.

da Silva, A. K. and Botter, R. C. (2009) ‘Method for assessing and selecting discrete event simulation software applied to the analysis of logistic systems’, *Journal of Simulation*, 3(2), pp. 95–106. doi: 10.1057/jos.2008.21.

Stewart, R. (2004) *Simulation: the practice of model development and use*. John Wiley & Sons Ltd. doi: 10.1057/palgrave.jos.4250031.

Suri, R. and Tomsicek, M. (1990) ‘Modeling manufacturing systems using Manuplan and Simstarter - a tutorial’, in *Proceedings of the 1990 Winter Simulation Conference*, pp. 168–176.

Swain, J. J. (2015) *Simulation software survey*, *OR/MS Today*. Available at: https://www.informs.org/ORMS-Today/Public-Articles/October-Volume-42-Number-5/Simulation-Software-Survey (Accessed: 2 February 2017).

Taylor, S. J. E. and Robinson, S. (2006) ‘So where to next? A survey of the future for discrete-event simulation’, *Journal of Simulation*, 1(1), pp. 1–6. doi: 10.1057/palgrave.jos.4250002.

Tewoldeberhan, T. W., Verbraeck, A. and Hlupic, V. (2010) ‘Implementing a discrete-event simulation software selection methodology for supporting decision making at Accenture’, *The Journal of the Operational Research Society*, 61(10), pp. 1446–1458.

Tewoldeberhan, T. W., Verbraeck, A., Valentin, E. and Bardonnet, G. (2002) ‘An evaluation and selection methodology for discrete- event simulation software’, *Simulation Conference, 2002.Proceedings of the Winter*. USA, pp. 67–75. doi: 10.1109/WSC.2002.1172870.

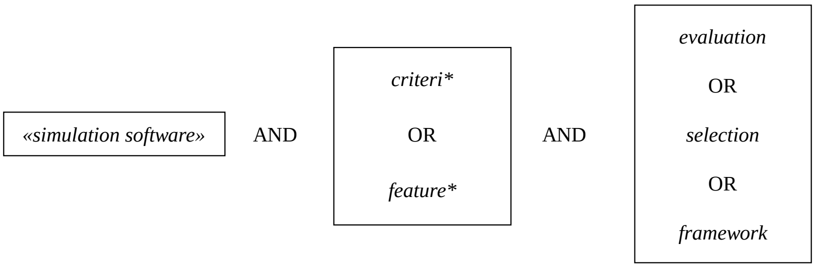



**Figure 1** Literature review setup: keywords and operators

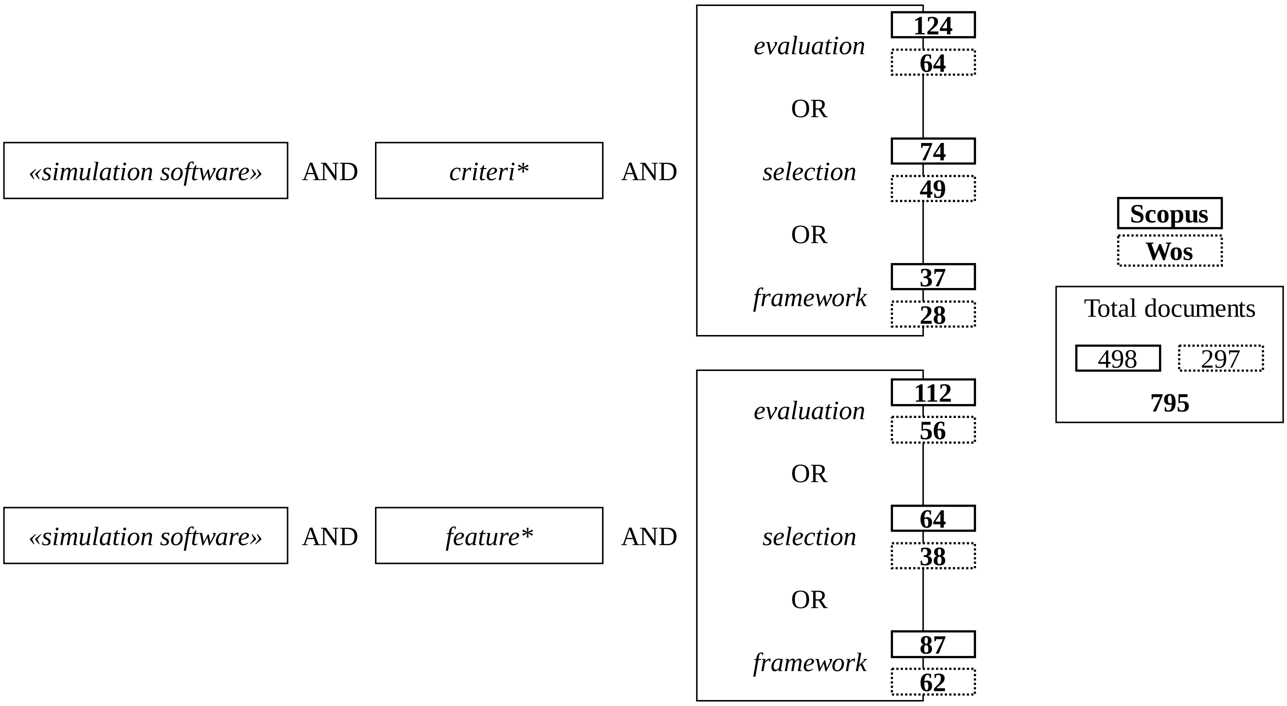



**Figure 2** Databases search results

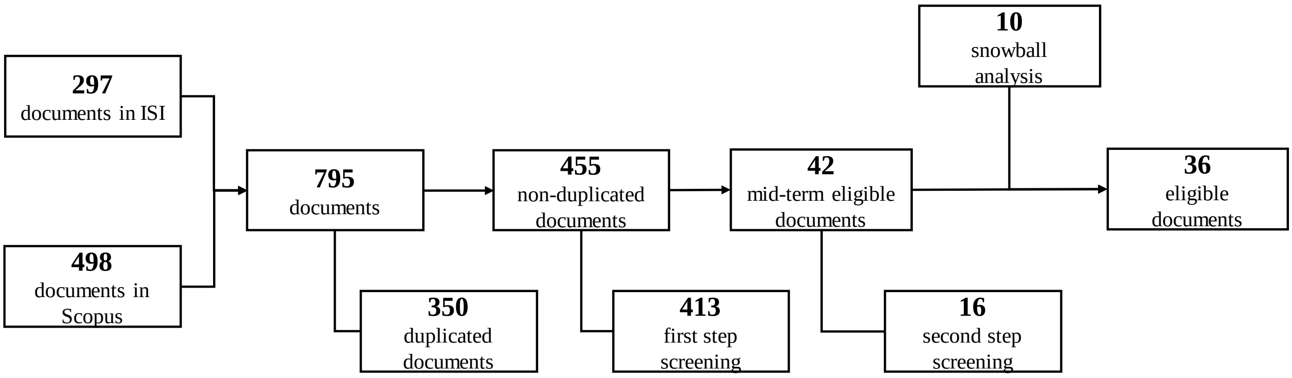



**Figure 3** Systematic literature review process

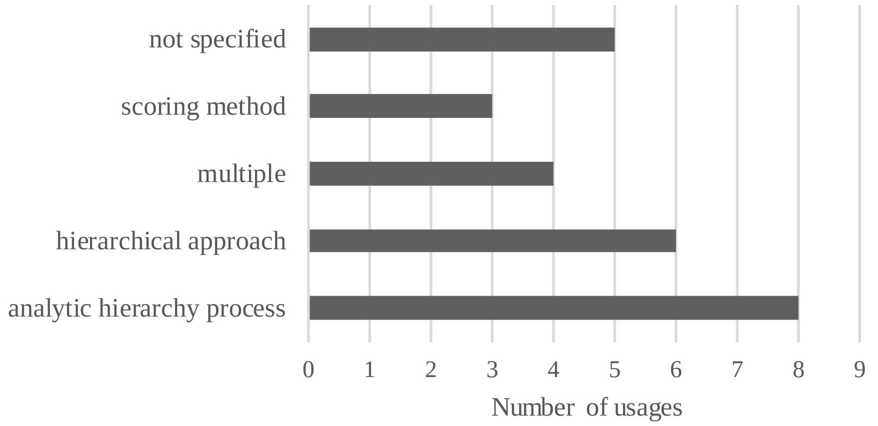



**Figure 4** Evaluation techniques adoptions in the eligible papers

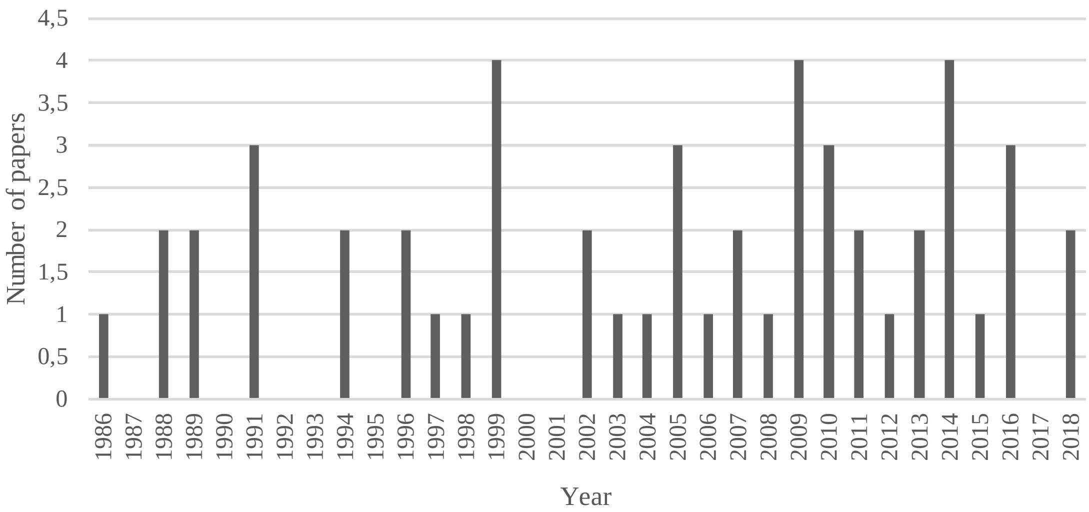



**Figure 5** Number of papers per year

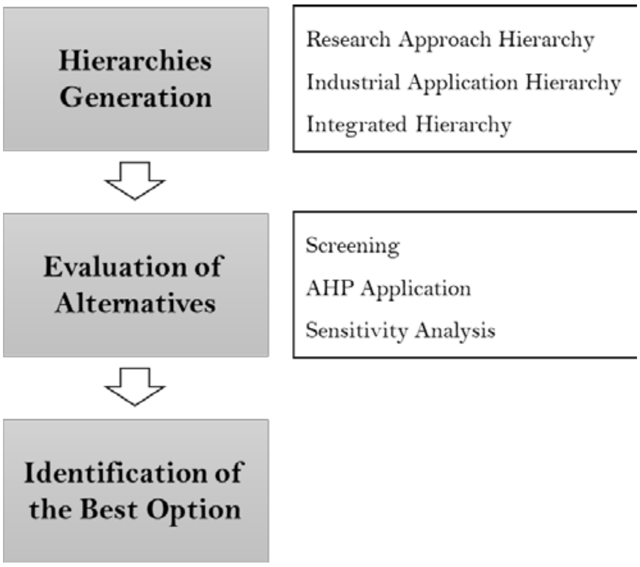



**Figure 6** Simulation software selection methodology

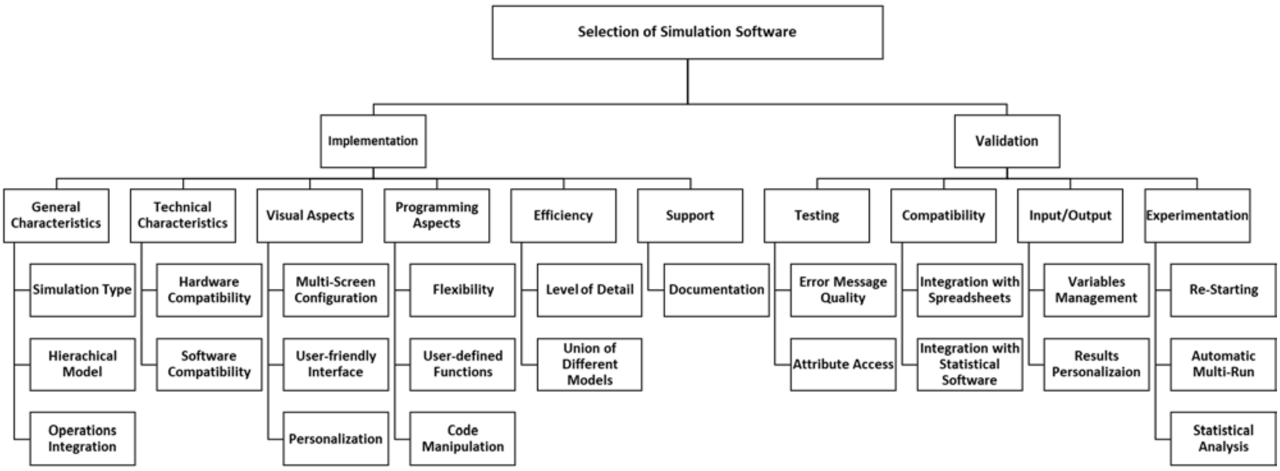



**Figure 7** Research Approach Hierarchy

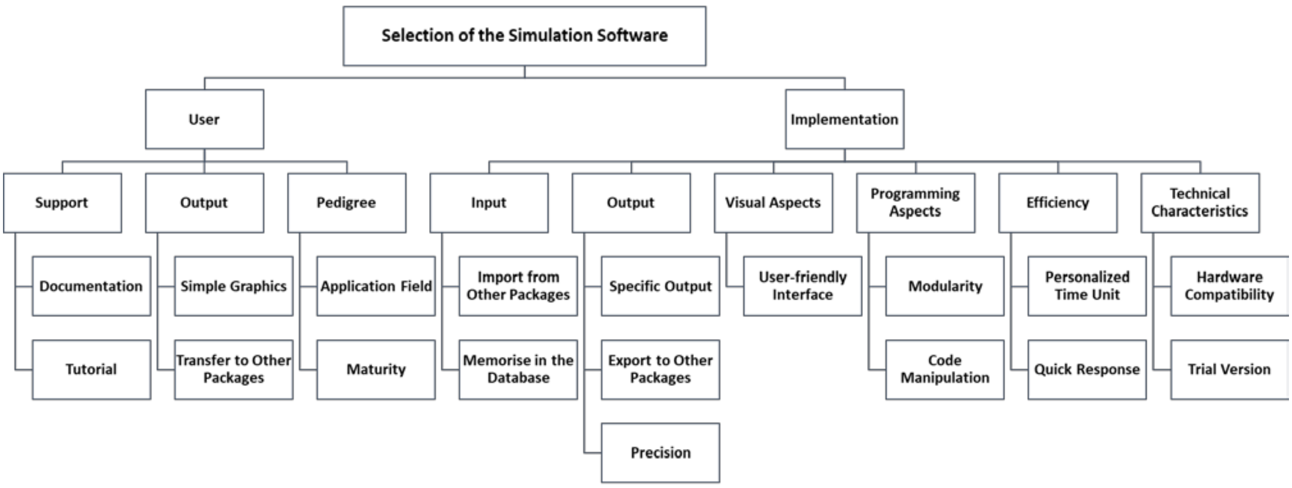


**Figure 8** Industrial Application Hierarchy

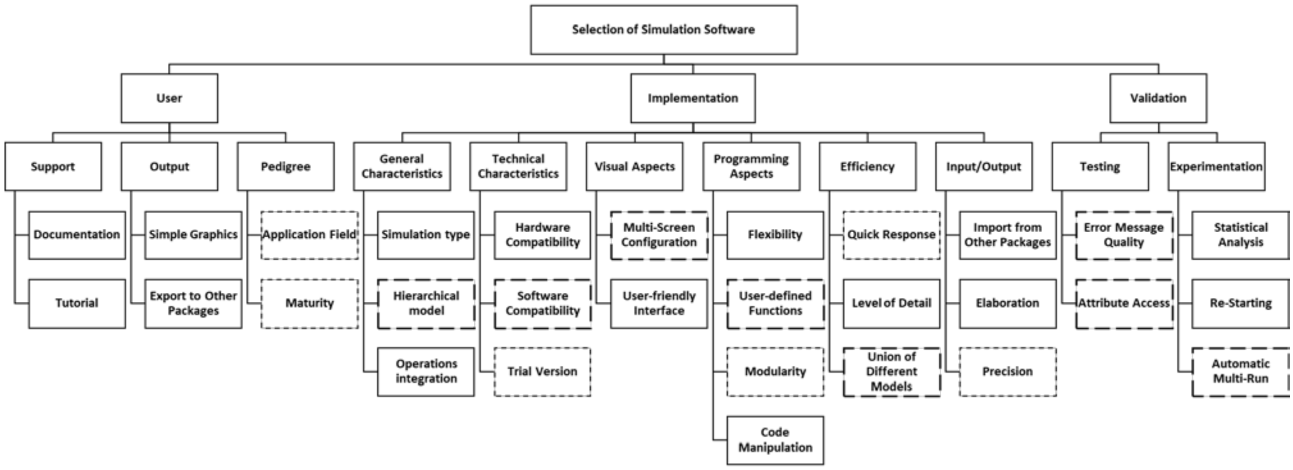
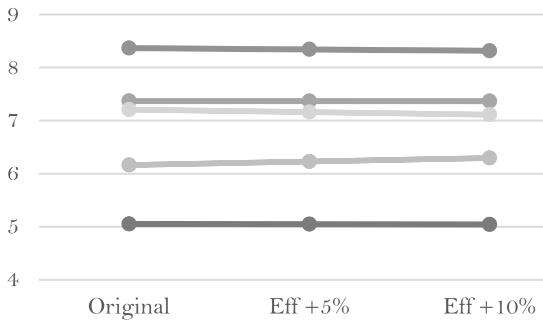
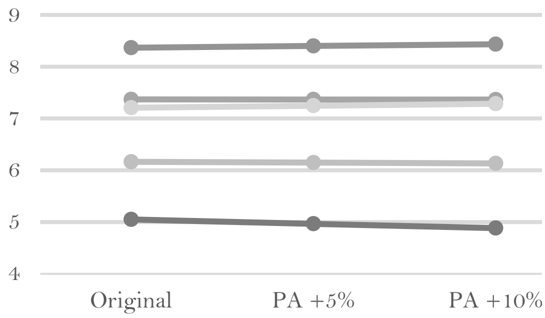


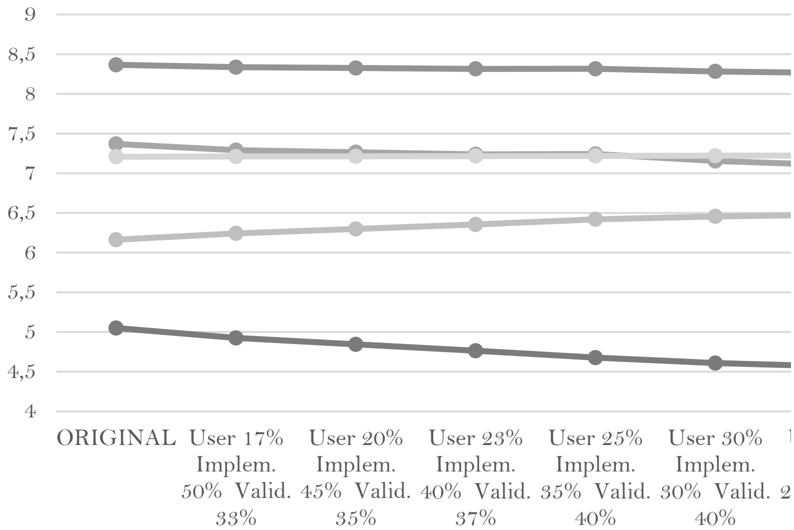

**Figure 9** Integrated Hierarchy




Different *Efficiency* weights.




Different *Programming Aspects* weights.





Different *User* weights.

**Figure 10** Sensitivity Analysis by varying some weights: the two plots at the top shows a change in weights in order to favour the research perspective (Efficiency and Programming Aspects), whilst at the bottom the change is towards Company point of view (User)

**Table 1** Selection criteria per author/s in chronological order (1988-1997)

| *1988* | *1989* | | *1991* | | | *1994* | | *1996* | | *1997* |
|---|---|---|---|---|---|---|---|---|---|---|
| **Grant** | **Law, Haider** | **Kochhar, Ma** | **Banks** | **Law, Kelton** | **Banks, Aviles, McLaughlin, Yuan** | **Mackulak, Savory, Cochran** | **Davis, Williams** | **Banks, Carson, Nelson** | **Kuljis** | **Bard, Desilva, Bergevin** |
| Physical & control system balance | General features | Modelling assistance | Input | General features | Robust features | General | Cost | Input features | Main system characteristics | Modeling features |
| System flexibility | Animation | Interactivity | Processing | Animation | Qualitative | Data acquisition | Comprehensiveness of the system | Processing features | Data input/model specification | Debugging |
| Random system behaviour | Statistical capabilities | Graphics | Output | Statistical capabilities | Cost | Model development | Integration with other systems | Output features | Simulation results | Animation |
| Animation | Material handling | Data handling | Environment | Customer support | Basic features | Validation and verification | Documentation | | Printed manuals | Report generation |
| | Customer support | Time scale | Cost | Output results | Special constructs | Model execution | Training | | On-line user assistance | Documentation |
| | Output reports | Learning curve | | | | Output analysis | Ease of use | | | Performance |
| | | Required skills | | | | Documentation | Hardware and installation | | | User |
| | | Ease of model editing | | | | Simulation project data | Confidence related issues | | | |
| | | Portability | | | | Methods of user interface | | | | |
| | | Simulation speed | | | | | | | | |
| | | Interfacing with external software | | | | | | | | |

**Table 1 (continued)** Selection criteria per author/s in chronological order (1998-2007)

| *1998* | *1999* | | | | *2002* | *2003* | *2005* | | *2007* |
|---|---|---|---|---|---|---|---|---|---|
| **Nikoukaran, Hlupic, Paul** | **Hlupic, Paul** | **Hlupic, Irani, Paul** | **Nikoukaran, Paul** | **Nikoukaran, Hlupic, Paul** | **Tewoldeberhan, Verbraeck, Valentin, Bardonnet** | **Sahay, Gupta** | **Cochran, Chen** | **Rincon, Alvarez, Perez, Hernandez** | **Bosilj-Vuksic, Ceric, Hlupic** |
| Vendor | Modeling assistance | General features | User interface | Vendor | Model development | Technology | Object-oriented features | Functionality | Hardware and software considerations |
| Model and Input | General features | Visual aspects | Database storage | Model and Input | Input modes | Cost and Pricing | Programming features | Reliability | Modelling capabilities |
| Execution | Visual aspects | Coding aspects | Debugging | Execution | Testing and efficiency | Features | Simulation features | Usability | Simulation capabilities |
| Animation | Efficiency | Efficiency | Interaction via mouse | Animation | Execution | Customisation | Environment features | Efficiency | Input/Output issues |
| Testing and Efficiency | Testability | Modelling assistance | Documentation | Testing and Efficiency | Animation | Support & Services | | Maintainability | |
| Output | Input/Output | Testability | Storage capabilities | Output | Output | Vendor vision | | Portability | |
| User | Physical elements | Software compatibility | Input data | User | User | Industry covered | | Client-supplier | |
| | User support | Input/Output | Library | | | Vendor strength | | Engineering | |
| | Coding asdpects | Experimentation facilities | Animation | | | Others | | Support | |
| | Pedigree | Statistical facilities | | | | | | Management | |
| | … (refer to article) | … (refer to article) | | | | | | Organisational | |

**Table 1 (continued)** Selection criteria per author/s in chronological order (2008-2012)

| *2008* | *2009* | | | | *2010* | | | *2011* | *2012* |
|---|---|---|---|---|---|---|---|---|---|
| **Alvarez, Rincon, Perez, Hernandez** | **Jadhav, Sonar** | **da Silva, Botter** | **Azadeh, Nazari Shirkouhi** | **Gupta, Verma, Singh** | **Azadeh, Nazari Shirkouhi, Rezaie** | **Tewoldeberhan, Verbraeck, Hlupic** | **Gupta, Singh, Verma** | **Jadhav, Sonar, Rajendra** | **Hincu, Andreica** |
| Functionality | Functional | General features | User | Hardware and software considerations | Vendor | Model development and input | Hardware and software considerations | Functional | Comprehensiveness of the system |
| Usability | Personalizability | Data input | Testing and efficiency | Modeling capabilities | Model and input | Animation | Modeling capabilities | Technical | Integration with other systems |
| Efficiency | Portability | Model development | Vendor | Simulation capabilities | Execution | Output | Simulation capabilities | Quality | Ease of use |
| Fit to purpose | Maintainability | Data Output | Model and input | Input/Output issues | Animation | Vendor | Input/Output issues | Vendor | Hardware and installation |
| Interoperability | Usability | Efficiency and testing | Output Execution | | Testing and efficiency | Testing and efficiency | | Output | Confidence-related issues |
| Security | Reliability | Execution | Animation | | Output | User | | Cost and benefit | |
| Easy of understanding and learning | Efficiency | Technical support | | | User | | | Opinion | |
| Graphic interface | Vendor | | | | | | | | |
| Operability | Cost | | | | | | | | |
| Execution performance | Benefits | | | | | | | | |
| Resource utilization | … (refer to article) | | | | | | | | |

**Table 1 (continued)** Selection criteria per author/s in chronological order (2013-2018)

| *2014* | | | *2016* | *2018* |
|---|---|---|---|---|
| **Dorado, Gómez-Moreno, Torres-Jiménez, López-Alba** | **Franceschini, Bisgambiglia, Touraille, Bisgambiglia, Hill** | **Gupta** | **Pezzotta, Rondini, Pirola, Pinto** | **Ejercito, Nebrija, Feria, Lara-Figueroa** |
| Cost | Software version | Hardware and software considerations | Vendor | Open source and free use |
| Educational usability | Programming language | Modeling capabilities | Model and Input | Operating system portability |
| Learning effectiveness | Quality of documentation | Simulation capabilities | Execution | Creating traffic networks and associated vehicle patterns |
| | Simulator algorithms | Input/Output issues | Animation | Quality of GUI and documentation |
| | Model sets | | Testing and efficiency | Simulation output (data and files) |
| | Network management | | Outputs | Ability to simulate very large traffic networks |
| | Design tools | | User | Additional capabilities |
| | Visualisation | | | CPU and memory performance |

**Table 2** Selection criteria

| *First-level criterion* | *Second-level criterion* | *Third-level criterion* | *Brief description* |
|---|---|---|---|
| I | General Characteristics | Simulation Type | Possibility of the software to perform DES (Discrete Event Simulation) or CTS (Continuous Time Simulation). Some software packages are able to perform both. |
| | | Hierarchical Model | Possibility to reproduce the production system in a hierarchical way to better control it during the implementation phase. |
| | | Operations Integration | Ability to perform other operations as data gathering or output analysis. |
| | | Model Reusability | Possibility to use the built model for new projects in order to reduce the time to create new models for future projects |
| | | Hybrid Modelling | Possibility to link the simulation model with other model types in order to create a hybrid model that can be useful to describe different behaviours of parts of the system |
| I | Technical Characteristics | Hardware Compatibility | In every work, there is a cooperation of different people, such as programmer, user, reviewer, and supervisor: each should access to all the part of the code and should be able to change or adjust some mistake or incompliance, despite possible different software or hardware. |
| | | Software Compatibility | |
| I | Visual Aspects | Multi-Screen Configuration | Easily controlling and managing modeled systems. |
| | | User-friendly Interface | Easiness in using the software package. |
| | | Personalization | Adaptable working environment. |
| I | Programming Aspects | Flexibility | Indicator of the easiness of generating code for addressing a task. An example is whether the user is forced to only one way to model a task, or whether he is allowed to choose from a series of alternatives or even whether there is not a pre-fixed way and the user can combine the functions provided by the package to code its own way for obtaining a certain result. |
| | | User-defined Functions | The software offers the possibility to save the user-defined task model. |
| | | Code Manipulation | It deals with the possibility of highly managing the code and the degree with which the user can manipulate the code and enter the details of the |

**Table 2** Selection criteria

| *First-level criterion* | *Second-level criterion* | *Third-level criterion* | *Brief description* |
|---|---|---|---|
| | | | software. In case there is no possibility of code manipulation, the user must use what provided to him. |
| I | Efficiency | Level of Detail | The level of detail is referred to the possibility to access faster to the needed part of the model and to configure it in the most appropriate way. |
| | | Union of Different Models | The creation of a complete model could take advantage also of the possibility of integration of more models into one, enhancing the chance to focus on some aspects of each model and then to generate the more comprehensive one with a simple copy-and-paste operation. |
| I | Support | Documentation | Availability of articles, papers, documents, handbooks about the programming language. |
| V | Testing | Error Message Quality | The quality of the error message is not dependent on the availability of documentation, in fact, the message error should be self-explicative and reach such a level of detail that the documentation may be not required to fix it. |
| | | Attribute Access | Capability of changing some parameters, as the processing time of a machine, without the need to have a full knowledge of the model, as an example having interface grids eases the access. |
| V | Compatibility | Integration with Spreadsheets | Interaction of the software with other packages, in particular spreadsheets and statistical software for pre- and post-processing. |
| | | Integration with Statistical Software | |
| V | Input/Output | Variables Management | Capability of the software in managing the variables, without the possibility of interchanges, especially when more models are created in the same working environment. |
| | | Results Personalization | Possibility to have tailor-made outputs. |
| V | Experimentation | Re-Starting | Capability of the software to re-set the initial conditions, allowing a new start of the simulation in a fast way. |
| | | Automatic Multi-Run | Possibility of starting a second run after the current one in order to create a set of results, which will be subsequently analysed. |

**Table 2** Selection criteria

| *First-level criterion* | *Second-level criterion* | *Third-level criterion* | *Brief description* |
|---|---|---|---|
| | | Statistical Analysis | Possibility of launching statistical analysis by the software itself, without exporting the data into other environments. |

**Table 3** Simulation software candidates

| *Name* | *Reference website* |
|---|---|
| LMS Amesim | https://www.plm.automation.siemens.com/en/products/lms/imagine-lab/amesim/ |
| AnyLogic | https://www.anylogic.com/ |
| Arena | https://www.arenasimulation.com/ |
| Dymola | http://www.dofware.com/prodotti/dymola/ |
| EcosimPRO | https://www.ecosimpro.com/ |
| Enterprise Dynamics | https://www.incontrolsim.com/product/enterprise-dynamics/ |
| ExtendSim | https://www.extendsim.com/ |
| Flexsim | https://www.flexsim.com/ |
| GoldSim | https://www.goldsim.com/Home/ |
| Lanner | https://www.lanner.com/ |
| MapleSim | https://www.maplesoft.com/products/maplesim/ |
| MatLab&Simulink | https://it.mathworks.com/ |
| Plant Simulation | https://www.plm.automation.siemens.com/it/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml |
| Pro Model | https://www.promodel.com/ |
| Simcad Pro | http://www.createasoft.com/ |
| Simio | https://www.simio.com/index.php |
| Simul8 | https://www.simul8.com/ |
| VisSim | http://web.solidthinking.com/vissim-is-now-solidthinking-embed |
| Wolfram SystemModeler | http://www.wolfram.com/system-modeler/ |

**Table 4** Weights for each criterion

| *First-level criterion* | *Second-level criterion* | *Third-level criterion* | *Weight [%]* |
|---|---|---|---|
| User | | | 10.95 |
| | Support | | 18.84 |
| | | Documentation | 83.33 |
| | | Tutorial | 16.67 |
| | Output | | 73.06 |
| | | Simple Graphics | 12.50 |
| | | Transfer to Other Packages | 87.50 |
| | Pedigree | | 8.10 |
| | | Application Field | 16.67 |
| | | Maturity | 83.33 |
| Implementation | | | 58.16 |
| | General Characteristics | | 2.54 |
| | | Simulation Type | 10.47 |
| | | Hierarchical Model | 63.70 |
| | | Operations Integration | 25.83 |
| | Technical Characteristics | | 4.32 |
| | | Hardware Compatibility | 25.83 |
| | | Software Compatibility | 63.70 |
| | | Trial Version | 10.47 |
| | Visual Aspects | | 7.42 |
| | | Multi-Screen Configuration | 75.00 |
| | | User-friendly Interface | 25.00 |
| | Programming Aspects | | 22.08 |
| | | Flexibility | 5.87 |
| | | User-defined Functions | 31.56 |
| | | Modularity | 48.14 |

**Table 4** Weights for each criterion

| *First-level criterion* | *Second-level criterion* | *Third-level criterion* | *Weight [%]* |
|---|---|---|---|
| | | Code Manipulation | 14.42 |
| | Efficiency | | 20.74 |
| | | Quick Response | 42.86 |
| | | Level of Detail | 14.29 |
| | | Union of Different Models | 42.86 |
| | Input/Output | | 42.91 |
| | | Import from Other Packages | 73.06 |
| | | Elaboration | 18.84 |
| | | Precision | 8.10 |
| Validation | | | 30.90 |
| | Testing | | 16.67 |
| | | Error Message Quality | 25.00 |
| | | Attribute Access | 75.00 |
| | Experimentation | | 83.33 |
| | | Statistical Analysis | 65.86 |
| | | Re-Starting | 15.62 |
| | | Automatic Multi-Run | 18.52 |