UHASSELT

KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

Masterthesis

Simulation Software Characteristics: An Integrated Framework and the Users' Perspective

Gerhardus van Hulzen

Scriptie ingediend tot het behalen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

PROMOTOR :

dr. Niels MARTIN

UHASSELT

KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

2018
2019

UHASSELT
KNOWLEDGE IN ACTION

Faculteit Bedrijfseconomische Wetenschappen

master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

Masterthesis

Simulation Software Characteristics: An Integrated Framework and the Users' Perspective

Gerhardus van Hulzen

Scriptie ingediend tot het behalen van de graad van master in de toegepaste economische wetenschappen: handelsingenieur in de beleidsinformatica

PROMOTOR :
dr. Niels MARTIN

"A journey of a thousand miles begins with a single step."

LAOZI (4^{th} – 6^{th} century B.C.E.)

HASSELT UNIVERSITY

Abstract

Faculty of Business Economics
Hasselt University

Master of Business and Information Systems Engineering

Simulation Software Characteristics: An Integrated Framework and the Users' Perspective

by Gerhardus A. W. M. VAN HULZEN

SIMULATION is a powerful method for measuring and predicting the effect of changes to a process without disrupting and committing actual resources. Currently, a great variety of simulation software packages have been developed to aid practitioners during their simulation study. Each of these packages has its own strengths and weaknesses depending on the domain of the simulation study. Therefore, it is important to gain insight into the desirable characteristics of simulation tools. In literature, various studies have been devoted to constructing evaluation frameworks, establishing important features, and the comparison of such tools. However, only limited research has been conducted on the integration of simulation software features described in literature into a framework and its validation by practitioners. This paper presents such an integrated framework and the results of interviews with simulation practitioners about the most important characteristics of simulation software packages and innovative functionalities which are currently not included in these tools.

Keywords: evaluation criteria, simulation, simulation features, simulation software, simulation software opportunities, software selection

May, 2019

UHASSELT

Acknowledgements

This master's thesis is the final piece for obtaining the degree of Master of Business and Information Systems Engineering at Hasselt University. The choice for this topic stems from the interest in the field of simulation that arose during the Process Simulation course which I had last year. The development of the master's thesis was a fascinating and enriching period, the result of which I am pleased to present to you.

First and foremost, I would like to express my deep gratitude to dr. Niels MARTIN, my research supervisor, for his patient guidance, enthusiastic encouragement and useful critiques of this research work. It was a very pleasant experience to work with him and his useful advice helped me to significantly improve the quality of this work. Secondly, I wish to express my gratitude to the simulation experts and academic researchers who have participated in my research. Their expertise in the field of simulation, as well as their experiences and insights were particularly valuable and without their contribution, it would not have been possible to complete this work. Finally, I would like to thank my immediate environment for all support, not only during the elaboration of this master's thesis but during my entire study career. In particular, I address my parents as they gave me the opportunity to complete my studies.

Contents

List of Figures

List of Tables

Dedicated to my parents

1 Introduction

Banks et al. [4] define simulation as "the imitation of the operation of a real-world process or system over time." A simulation model is a computer model used to analyse the behaviour of this process. This model can be used to measure and predict the effect of changes to the process. If an unexisting process is analysed, the performance under various circumstances can be predicted using simulation [4]. One advantage of simulation is that the operation of the real process does not have to be disrupted to study the process under various circumstances. Another advantage is that no actual resources have to be committed during the study. "What if"-scenarios are also often easier to simulate than to actually test them in the real world [34]. Simulation can be applied in a wide variety of areas, e.g. manufacturing applications, construction engineering and project management, logistics, transportation and distribution, business processing, health care, military applications, and many more [4].

A great variety of simulation software packages have been developed to provide simulation practitioners with a solid set of functionalities. These packages include Commercial off-the-Shelf (COTS) software, such as AnyLogic, Arena, Simio, etc. and programming libraries without a Graphical User Interface (GUI), such as SimPy [41] or simmer [44]. However, not every package provides all the functionalities one requires for a particular simulation study. Additionally, one particular package could provide better and more support in one application context than another. Therefore, selecting the right simulation software package is an important step when starting a simulation study, or when moving from one package to another. Various papers have been written about evaluation frameworks, required features, or the comparison of simulation tools [1, 3, 5, 6, 8, 10, 11, 13, 14, 15, 17, 19, 20, 23, 25, 29, 30, 31, 35, 36, 40, 42, 43]. However, very little research has been carried out which integrates the required features into a literature-based evaluation framework and validates this framework by asking the opinion of practitioners.

This paper presents a literature-based framework outlining simulation software characteristics. Moreover, interviews with simulation practitioners were conducted to validate this framework. With these interviews, the author tried to determine how simulation software packages are used by simulation experts in the industry and academic researchers. They indicated which features are considered important during simulation studies. In addition, the practitioners were also asked to indicate whether they felt some features were lacking in the tools they currently use and whether they have looked for other packages that might overcome these limitations. The results of these interviews are also presented in this paper. With this work, the author extends the research on the evaluation frameworks and required features of simulation software packages by asking the opinion of practitioners. In addition, the insights from this paper can support the future development of simulation tools.

The remainder of this paper is structured as follows. Section 2 defines the methodology used during the research. Section 3 provides a structured overview of the research conducted so far. Section 4 describes an integrated framework on the characteristics of simulation software packages based on prior research. The results of this study are presented in Section 5, whereas Section 6 discusses these results. Section 7 concludes this work with a summary, limitations of the study, and recommendations for further research.

2 Research Methodology

In this section, the proposed methodology that was used for this study is described. The following subsections give a structured in-depth overview of the different steps that were needed to conduct this study.

2.1 Literature Review for Framework Development

The first step was to identify the most important characteristics of a simulation software package according to prior research. A literature study was conducted to formulate an integrated framework of these characteristics.

Various sources were consulted for obtaining the necessary background information. These sources included the UHasselt University Library, Google search engine, and Google Scholar. Search keywords included "simulation & software & criteria", "simulation & software & selection", "simulation & software & selecting", "simulation & software & features", "business process simulation", "criteria for the evaluation of business process simulation tools", "methodology & simulation & software", "decision & methodology & evaluation & selecting & simulation & software", "discrete event simulation & selection", "business process & simulation", etc.

Relevant papers w.r.t. the topic were selected and the references of those papers were analysed, including citation chaining using Google Scholar, to find additional literature on this topic. Papers were considered relevant if the authors listed simulation software characteristics, proposed selection methods for such tools, or compared these tools. This resulted in a total of 23 papers [1, 3, 5, 6, 8, 10, 11, 13, 14, 15, 17, 19, 20, 23, 25, 29, 30, 31, 35, 36, 40, 42, 43]. These papers were analysed and the mentioned characteristics were assembled in an integrated framework. This framework consists of four main categories of criteria, which are further divided into fifteen different sub-criteria and 40 individual features. Section 3 describes the literature analysed for this study and Section 4 gives an in-depth overview of the framework.

2.2 Interviews for Framework Validation

The following subsections discuss the preparation of the interviews, how these were conducted and how the results were analysed.

2.2.1 Preparatory Work for the Interviews

To provide methodological support for the interviews, an interview protocol is developed. This protocol provides a guideline throughout the search for respondents, contacting them, conducting the interviews, analysing the results, and reporting these results [47]. The author of this study argues that in-depth, qualitative interviews provide more useful insights than other questioning methods – e.g. a survey – for this study. Individual, face-to-face interviews are generally the preferred choice in qualitative interviewing, because of the interpersonal contact, enhanced context sensitivity, and flexibility – i.e. it is possible to ask follow-up questions – it provides to the interviewer [7]. Kvale calls the qualitative research interview "a construction site for knowledge" [26].

Two main categories of subjects were contacted with the question whether they would be willing to participate in this study:

1. Simulation experts, i.e. his or her main job function consists of modelling, simulating, and analysing the results of simulation studies, and

2. Academic researchers whose main research field consists of simulating and analysing the results of simulation studies.

Both of these groups are thus active in the field of simulation studies. However, it is most likely that they use different approaches when conducting these studies, because of their knowledge and expectations. Therefore, it would be interesting to analyse to what extent these two groups require the same set of features when conducting a simulation study.

During the interviews, the two main categories of subjects were asked how they use simulation tools and what their opinion is about the most important characteristics of such tools. For a more detailed overview of the interview protocol and the proposed questions to ask during the interview, the interested reader is referred to Appendix A.

2.2.2 Selecting Interview Respondents

The respondents for the group of academic researchers were selected based on their field of study. University connections were used to find researchers active in the domain of simulation. The other group, the simulation experts, were selected from various companies which are active in the domain of simulation. Google search engine and university connections were used to find respondents who would be willing to participate in this study.

All respondents were initially contacted via email. Before the interviews took place, interested respondents received a couple of example questions to prepare themselves for the interview.

2.2.3 Conducting the Interviews

In total, six in-depth interviews were conducted: three with simulation experts and three with academic researchers. All interviews took place in March 2019 and the average length of these interviews was around 80 minutes.

A semi-structured interview style was used during these interviews. This implies that the interviewer uses a guideline in the form of a list of questions about topics the interviewer wants to cover. This is an inclusive, but not an exhaustive list of questions. It is inclusive because it contains most of the topics the interviewer want to cover, but it is not exhaustive in the sense that it does not list the only possible questions the interviewer may ask during the interview. Kvale and Brinkmann define the semi-structured qualitative research interview as "[...] an interview with the purpose of obtaining descriptions of the life world of the interviewee in order to interpret the meaning of the described phenomena" [27]. This interview style was therefore chosen because it provides a structured overview of the topics to cover, and – at the same time – allows the interviewer to let the interviewee elaborate more on the given answers by using follow-up questions [26]. In addition – as Kvale and Brinkmann describe – the semi-structured qualitative research interview is well suited for interpreting and understanding phenomena described by the interviewees.

During the interviews, two main categories of questions needed to be answered:

1. How does the respondent use the simulation software package(s) and which specific characteristics are important to him or her?

2. Which of these characteristics are currently underdeveloped or missing entirely?

The questions that were asked within each of these categories are listed in Appendix A.

2.2.4 Analysing the Interview Results

After all the interviews had been conducted, the answers to each question were analysed and compared with each other to formulate conclusions.

One question asked the interviewee to rank the sub-criteria of the framework presented in this study w.r.t. the importance of each sub-criterion. Despite the fact that specific ranking methods for Multiple-Criteria Decision-Making (MCDM) problems exist, such as Analytic Hierarchy Process (AHP) [38] or Technique of Order Preference by Similarity to Ideal Solution (TOPSIS) [21], the author of this study opted for a different method. MCDM methods are designed to structurally organise and *solve* a decision problem [46]. The goal of this study, however, is to determine which characteristics of a simulation software package are considered important, and especially *why* it is believed to be more important, rather than determining which software tool is considered the best based on its features.

In addition, AHP relies on a Pairwise Comparison Matrix (PCM) which is constructed from comparing each criterion to the others. A PCM with m criteria needs $m(m-1)/2$ pairwise comparisons. This methodology suffers therefore from the "curse of dimensionality" – i.e. for small decision-making problems, this matrix can be easily construed – but for problems where m is larger, the amount of needed pairwise comparisons increases quadratically [24]. For this study, where $m = 15$ (i.e. the amount of sub-criteria in the framework presented in this study), the total amount of pairwise comparisons equals to $15(15-1)/2 = 105$. This would not only be very cumbersome for the interview respondent, but it would also take up the majority of the time available for the interview itself.

Table 2.1 was used to aid the interviewee in ranking the sub-criteria of the framework. This consists of a simple table in which the respondent can place small cards containing the different sub-criteria of the framework. Placing a card on the top of the table means that the respondent thinks that the characteristic is indispensable. On the other hand, placing a card on the bottom means that the characteristic has no added value and could be omitted. This scale was inspired by the nine-point scale designed by Saaty for the AHP [38], but inverted so that a score of "1" corresponds to the *most* important category and "9" to the *least* important category. This makes the ranking process more intuitive because the most important characteristic is placed at the top of the table. The same principle was also used by Guimarães et al. [13] in which a score of "1" meant that the feature is "completely indispensable" and and "9" "completely dispensable".

1	*Extreme importance*
2	*Very, very strong importance*
3	*Very strong importance*
4	*Strong plus importance*
5	*Strong importance*
6	*Moderate plus importance*
7	*Moderate importance*
8	*Weak or slight importance*
9	*No importance*

TABLE 2.1: Nine importance categories[1].

[1] This table has been scaled down to save space. The table presented to the respondents during the interviews expanded two A4-format pages.

Another question during the interviews was used to determine which of the 40 individual features are considered most important. The interview respondents were given a table including these features (see Appendix A for the visual representation of this table) and nine coins – two "golden", three "silver" and four "bronze" coins. They were then asked to "configure" their personalised simulation software package by placing coins on the features they would really like to have. Each feature would "cost" the same and each coin has the same "value". The difference between the coins should be interpreted as: "If your supervisor has to bring you the unpleasant news that the budget has been reduced, which feature would you let go first?" This feature should get a "bronze" coin. The material of the coin, therefore, represents the relative importance between features.

To analyse the results from this question and rank the features, each coin was given a numerical value. Table 2.2 gives an overview of these values.

Coin	Value
Golden	3
Silver	2
Bronze	1
No coin	0

TABLE 2.2: Coin values.

Next, the coin values were summed for each feature over all responses to create a score for the ranking. However, for some of the features, the scores were identical making it impossible to make a strict ranking. To solve this, the standard deviation and maximum value were also included in the calculation of the score. A high standard deviation was penalised because this means that the experts disagree about the importance of that feature. On the other hand, a high maximum value – i.e. at least one respondent gave that feature a "golden" coin – was rewarded, because this feature was judged indispensable. The score of each feature was calculated with Equation 2.1:

$$S_i = \sum_{j=1}^{N} s_{i,j} + \frac{1}{\sigma_i} + \max_{1 \leq j \leq N} s_{i,j} \tag{2.1}$$

where S_i is the final score of feature i, $s_{i,j}$ is the numerical value of the coin of feature i given by respondent j, and σ_i is the standard deviation of the scores $s_{i,j}$ of feature i.

3 Literature Review

Four main streams can be found on the topic of comparing simulation software packages in literature: (1) criteria listings, (2) comparisons performed by the authors of the study, (3) selection methodologies, and (4) comparisons with input from simulation practitioners. The following subsections present several studies for each of these four streams.

3.1 Criteria Listings

The first stream, i.e. criteria listings, mainly contains papers describing possible features of simulation software packages. The most comprehensive list of features for any general or special purpose simulation package is given by Hlupic et al. [19]. They have analysed several studies in the field of simulation software criteria and presented a list of over 300 different criteria in thirteen different categories. The authors do not specify, however, which features would be an absolute necessity, and which are optional. Nevertheless, this study was referenced many times by other studies due to its comprehensiveness. Bosilj-Vukšić et al. [5] later refined the feature list by Hlupic et al. [19] to define the main simulation features specifically for Business Process Simulation (BPS) tools. The criteria were divided into four main categories: hardware and software considerations, modelling capabilities, simulation capabilities, and input/output issues; each further classified into subcategories. The authors' own experience was used to reduce the number of criteria to a more manageable number of 70.

An earlier study performed by Nikoukaran et al. [30] presents a comprehensive list of criteria described in literature in a structured hierarchical framework. They identified seven main categories of criteria: vendor, model and input, execution, animation, testing and efficiency, output, and user. Each category was further divided into criteria, combining to a total of 50 criteria.

3.2 Comparisons by the Authors

The limitation of the criteria listings in Subsection 3.1 is that the mentioned criteria are not used to compare actual simulation software packages, nor are they validated by simulation practitioners. The second stream, i.e. comparisons performed by the authors of the study, use these criteria lists to actually compare simulation tools. Bradley et al. [6] compared four Business Process Re-engineering (BPR) tools. BPR tools are simulation packages specifically designed for analysing business processes. To compare these four tools, they used seven categories: tool capabilities, tool hardware/software, tool documentation, user features, modelling capabilities, simulation capabilities, and analysis capabilities, which were further divided into 50 individual sub-criteria. However, no list of characteristics described in literature was used to evaluate these tools and no particular decision-making method was used to rank the tools based on their features.

Gupta et al. [14] performed a similar study. They compared four manufacturing simulators: NX-IDEAS, Star-CD, Micro Saint Sharp, and ProModel using the AHP. This method was developed by Saaty [37] as a structured technique for making complex decisions. During the comparison, Gupta et al. [14] considered four main categories of criteria: hardware and software considerations, modelling capabilities, simulation capabilities, and input/output issues and over 200 sub-criteria. The authors selected these criteria from various studies in literature and supplemented these with their own experience.

da Silva and Botter [40] also used the AHP for assessing and selecting Discrete-Event Simulation (DES) software applied in the logistics industry. They identified a set of over 100 criteria, classified into seven categories: general features, data input, model development, data output, efficiency and testing, execution and technical support. The authors selected these criteria from various studies in literature and supplemented these with relevant criteria for modelling and analysing logistic systems. Next, they tested three simulation packages (Arena, ProModel, and @Risk) to determine which tool performed best.

Another comparison method was used by Damij et al. [10]. This was one of the largest comparison studies, in which the authors compared a total of 33 different BPS tools using the qualitative MCDM method DEX and Qualitative to Quantitative (QQ) methodology. They ranked each tool on three dimensions: visual aspects, simulation facilities, and statistical facilities, according to five different categories found in literature: statistical facilities, experimental capabilities, testing capabilities, efficiency, and visual aspects.

Jansen-Vullers and Netjes [25] also compared various BPS tools. A total of six software tools were evaluated, divided into three different categories that could be used for BPS studies: (1) Business Process Modelling tools (Protos and ARIS), (2) Business Process Management (BPM) tools (FLOWer and FileNet), and (3) general-purpose simulation tools (Arena and CPN Tools). Each tool was evaluated on the modelling, simulating and output analysis capabilities, which are – according to the authors – the most important capabilities of a BPS tool. A total of thirteen individual criteria were used to compare the six BPS tools. They showed that the general-purpose simulation tools are appropriate for BPS studies and even outperformed the other tools on most criteria.

Pidd and Carvalho [36] suggested that software developers of simulation tools should adapt their software to an on-demand service, which end-users can customise to their needs. End-users would only have to pay for the functionalities they need in their current simulation studies. The authors compared programming simulation languages and concluded that the DotNetSim prototype illustrates such a component-based architecture.

3.3 Selections Methodologies

Various methodologies for selecting software have been developed over the years. The third stream, i.e. selection methodologies, contains papers addressed specifically to selection methodologies for simulation software packages. Nikoukaran and Paul [31] acknowledged the need for a list with standard definitions of criteria to be considered for selecting the proper simulation software package, as well as a standardised selection methodology.

Alomair et al. [1] extended on the research of Nikoukaran and Paul [31] and summarised different evaluation methods for simulation software packages found in the literature. These included MCDM methods (AHP, Fuzzy Analytic Hierarchy Process (FAHP), TOPSIS, and Preference Selection Index (PSI)), hierarchical frameworks, simulation selection software tools, and two-phase evaluation and selection methodology.

To aid users selecting the right package, Hlupic and Mann [20] developed SimSelect; a database linked to an interface created using Visual Basic 3.0. The system queries the database and finds a simulation package suitable to the user, based on a list of 40 different criteria (taken from Hlupic [18]) which ranged from the following groups: general features, visual aspects, coding aspects, efficiency, modelling assistance, testability, input/output, software compatibility, experimental facilities, statistical facilities, and financial and technical features. No specific decision-making technique is mentioned in the paper, but the authors assure that when the user rates a feature as "high", it is included in the suggested software packages. Later, Smart Sim Selector was developed by Gupta et al. [15] for the same purpose. This tool was able to guide the user to the right simulator using more than 200 evaluation criteria divided into fourteen main categories (taken from Gupta et al. [14]). The user could also choose among three different selection techniques: AHP, Weighted Score Method, and TOPSIS. Although these tools provide a quick way of selecting simulation software based on different criteria, their databases were never updated after publication.

Another selection methodology is FAHP. FAHP is an extension of AHP with fuzzy sets initially proposed by Van Laarhoven and Pedrycz [45]. Hincu and Andreica [17] assessed this method for selecting simulation software. The authors argue that fuzzy sets logic provides a more convenient way of working with imprecisely defined data, which is often the case with ranking alternatives.

3.4 Comparison with Input from Experts

The fourth and last stream also deals with comparisons of simulation software packages. However, in contrast with the studies listed under Subsection 3.2, the input from simulation practitioners was asked, whereas in Subsection 3.2 the opinion of the authors of the study was used to perform the comparisons. A two-phase evaluation and selection methodology was proposed by Tewoldeberhan et al. [43] to select the right simulation software based on criteria extracted by a selection team, who asks from the input from the simulation team. Phase one of the methodology quickly reduces a long list of potential software packages to a short-list by checking whether the tool satisfies the hard criteria (i.e. the most important ones). Then, in phase two, the hard criteria from phase one are further established and weighted based on the level of importance. The simulation packages from the short-list are then scored on each criterion individually resulting in a general score. The package with the highest score is then assumed to be the best suitable simulator. Later, Tewoldeberhan et al. [42] applied this methodology for supporting decision-making related to the selection of a new DES package for Accenture's worldwide simulation team. The criteria categories they used were: model development, input modes, testing and efficiency, execution, animation, output, and user. The simulation experts of Accenture indicated that for their projects modelling flexibility, maintenance support and documentation, a good debugger, and export functionalities were among the most important features of a suitable simulation tool.

Guimarães et al. [13] also proposed a two-stage method for evaluating and selecting DES software used to develop models in the industry. In the first stage, the processes of the company are analysed using maturity models. These models can be used to measure the quality of the processes of an organisation and classify them based on their stage of development. This classification can then be used to identify whether the basic conditions for conducting simulation are satisfied. The authors of the study argue that, in order to apply simulation to a process, the process must be structured and documented, and statistical data must be available about it, e.g. arrival rates, throughput times, fault rates, etc. The second stage evaluates the features of the potential simulation software and compares these against the requirements as indicated by the company. AHP is then used to solve the MCDM problem. The main criteria were based on previous studies and other important criteria considered by the authors, based on experience in the manufacturing industry: data input and output, development of models, execution of models, evaluation and efficiency, technical support, and costs. This methodology was then applied to four different companies operating in different market segments. The companies indicated that technical support, specialised training, statistical information on input and output data, consistency checking, and batch input belong to the most important features of a simulation software package.

Another study using the AHP technique was conducted by Davis and Williams [11]. To evaluate and select simulation software they identified eight criteria with sub-criteria (taken from previous studies and the experience of the authors) that need to be addressed when choosing manufacturing software and simulation packages: cost, comprehensiveness of the system, integration with other systems, documentation, training, ease of use, hardware and installation, and confidence-related issues. Each criterion was ranked by relative importance and five simulation packages were assessed with respect to each criterion and sub-criterion. The package with the highest ranking was recommended for purchase. A case study using their proposed method was conducted for a medium-sized engineering company in the UK. The criteria which were considered the most important were: comprehensiveness of the system, containing flexibility, statistical facilities, and graphical capabilities; and ease of use, containing required simulation expertise, and user-friendliness.

Simulation tools are also often used in an academic environment. Jadrić et al. [23] designed an experiment in which they asked 24 students who enrolled in the BPS course to create a simple tollbooth model and analyse the results. The students first used ExtendSim, and then Arena, or vice versa, and were asked to complete an online survey. The authors compared the results of the two simulation packages on three main criteria (taken from [5, 6, 25]): modelling capabilities, simulation capabilities, and output analysis capabilities, further divided into 23 sub-criteria. The results showed that the students preferred ExtendSim over Arena, although the objective measurements (e.g. faster model building and fewer clicks needed) suggested that Arena was far superior to ExtendSim. This would suggest that the perceived capabilities are more important than the actual capabilities of a simulation package.

Azadeh et al. [3] used fuzzy theory for selecting simulation software. They verbally questioned ten different experts to derive the weights of various criteria, such as user, testing and efficiency, vendor, model and input, output, and execution. By using fuzzy theory, they were able to reduce ambiguities and uncertainties that are inherent in the criteria of selection. The respondents indicated that the most important features included: required experience, financial costs, general purpose or domain specific purpose, validation and verification, coding flexibility, output reports, graphics, and statistical analysis capabilities. Cochran and Chen [8] used a similar approach to compare three programming/simulation languages: C++, SIMAN, and SIMPLE++. Their focus was, therefore, more on the programming capabilities of simulation packages. The results showed that re-usability, flexibility, statistical ability, and documentation were considered most important.

Mackulak et al. [29] were among the first who identified that little research had been done on identifying and prioritising features from the viewpoint of a simulation practitioner. Therefore, they conducted a survey on the most important simulation software features to evaluate the views of simulation practitioners. In total, 29 responses were recorded. According to their research, the most important feature a simulation tool must possess is a consistent and user-friendly interface, followed by the capability to store input data, and an interactive debugger for error checking and code tracing. To date, this study involved most experts in determining important criteria for simulation software packages.

Over the years, a great interest has been shown around simulation software packages and how to select the most appropriate one. However, only a little amount of studies have validated whether these techniques also work in practice. In addition, very little research has been conducted on what simulation practitioners think are the most important features of simulation software tools. Every study used its own list of criteria based on previous studies and the experience of the authors. In 1999, Nikoukaran and Paul [31] acknowledged the need for a list of standardised criteria and a well-established selection methodology. To date, no such standardised list and selection methodology has been accepted by the community. This study will try to extend the knowledge available on the domain by proposing a literature-based framework on the characteristics of simulation software packages which will be validated by the end-users of such tools, i.e. simulation practitioners. In addition, these experts will also be asked to indicate the most important criteria, limitations of the current generation of simulation tools and opportunities for future improvement.

4 Simulation Software Characteristics Framework

This section describes an integrated framework on the characteristics of simulation software packages based on a thorough analysis and synthesis of the literature [3, 5, 6, 8, 10, 11, 13, 14, 15, 17, 19, 20, 23, 25, 29, 30, 35, 36, 40, 42, 43]. Figure 4.1 gives a structural overview of this framework and will be validated by practice experts, which is currently a limitation of the literature in this domain.

The framework presented in this work consists of four main categories of criteria, which are further divided into sub-criteria:

- Modelling Functionalities: describe the tool's ability to *create* simulation models.
- Simulation Functionalities: describe the tool's ability to *simulate* the created simulation models.
- Integration Functionalities: delineate the tool's ability to *interact* or *integrate* with data sources and other software tools, as well as the ability to present and use the insights gained from simulation.
- Hardware and Software: describe other characteristics of the tool with regards to hardware requirements, licence costs, etc.

This subdivision is directly in line with literature [5, 14, 23], with one distinction. The mentioned studies refer to "Input/Output Issues", whereas this study refers to this group as "Integration Functionalities". The author argues that this label better captures the input and output capabilities, as well as the analysis capabilities to "integrate" the real world with the abstracted simulation model.

Other studies [3, 30, 42, 43] define seven categories: vendor, model and input, execution, animation, testing and efficiency, output, and user. The author of this study argues that each of these categories could be captured by one of the categories defined above, as shown in Table 4.1. Therefore, the author has opted for the framework with four categories.

Seven categories	Four categories
Vendor	Hardware and Software
Model and Input	Modelling Functionalities and Integration Functionalities (resp.)
Execution	Simulation Functionalities
Animation	Simulation Functionalities
Testing and Efficiency	Simulation Functionalities, and Hardware and Software (resp.)
Output	Integration Functionalities
User	Hardware and Software

TABLE 4.1: Each category from the framework with seven categories [3, 30, 42, 43] can be mapped to the framework with four categories [5, 14, 23].

The following subsections will further define the criteria of each main category to eliminate confusion and ambiguity.

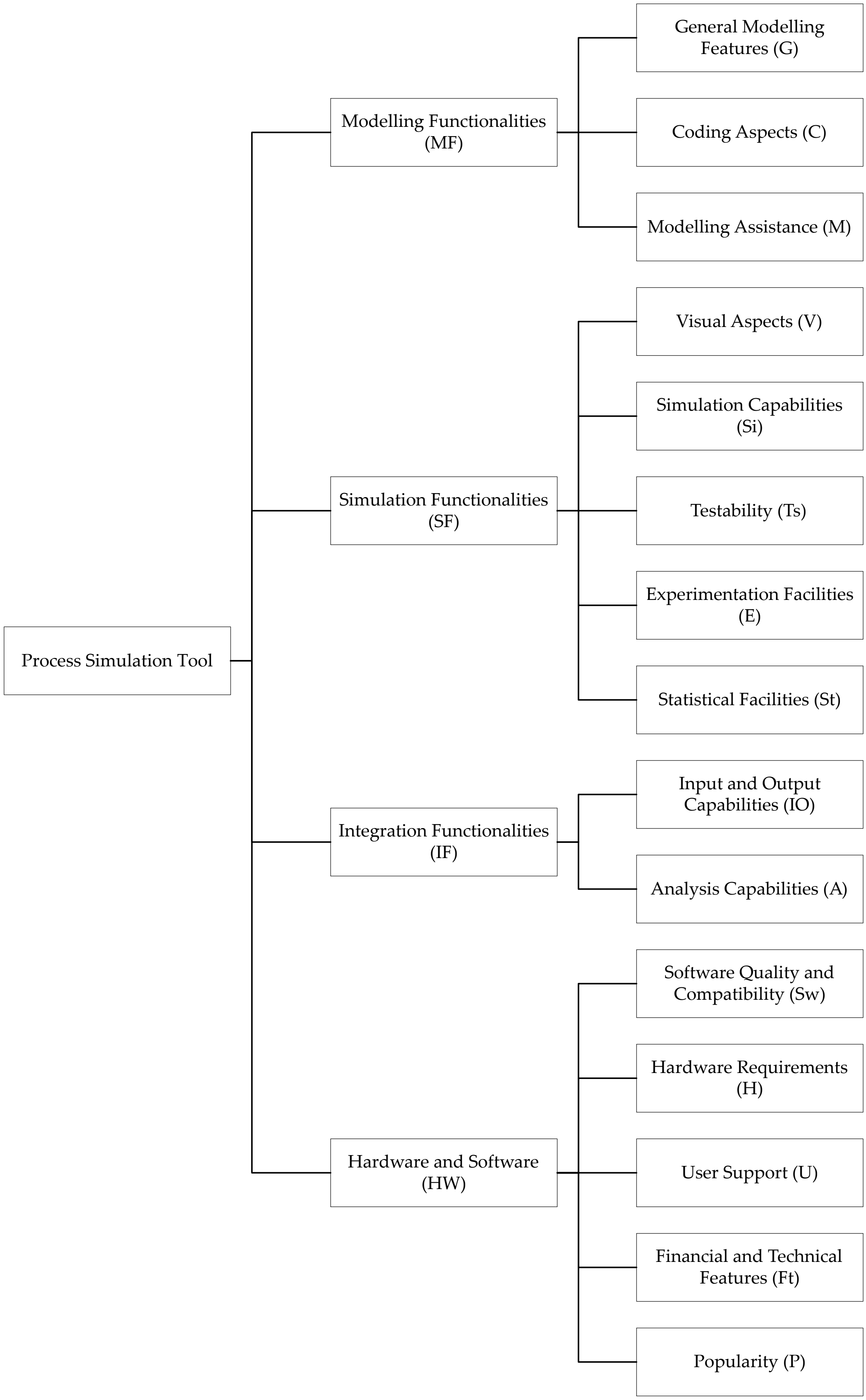

FIGURE 4.1: Simulation software package characteristics framework.

4.1 Modelling Functionalities

The *modelling functionalities* [3, 5, 6, 14, 23, 25, 29, 36, 40, 42, 43] of a simulation software package allow the user to create simulation models, which can be later executed. Generally, there are two ways of modelling simulation models [36]:

1. Using a Graphical User Interface (GUI): allowing the user to simply drag and drop various building blocks, each with its own purpose, and connecting these blocks to form a process flow.

2. Using a programming language: either a simulation programming language, which is specifically designed to model and simulate simulation projects (e.g. SIMAN [33] or SIMULA [9]), or a general-purpose programming language, which can be used for a great variety of applications, including simulation (e.g. C++, C#, R, Python, etc.).

4.1.1 General Modelling Features

The criterion *general modelling features* [5, 14, 15, 19, 20, 23, 29] of a simulation software package contain some general characteristics of the tool. For instance, the ability to model *process flow diagrams* [5, 6, 13, 14, 19, 25, 29, 30, 35, 36, 42, 43] using building blocks. For BPS studies a modelling language, such as Business Process Model and Notation (BPMN) [32], would be suitable. BPMN is a modelling language designed to display business processes graphically so it is readily understandable by all business users [32]. However, BPMN does not contain all components required for simulation purposes as, e.g. queues are not included. The BPSim [12] extension could be used for that. A closely related characteristic is the *modelling flexibility* [5, 11, 14, 17, 19, 23, 25, 30, 36] which describes for how many domains the tool can be used. The more flexible, the more domains can be modelled with the tool. There usually is, however, a trade-off between flexibility and the level of detail that can be achieved within one specific domain.

Another characteristic applicable for almost any software tool is *user-friendliness* [3, 5, 6, 8, 10, 11, 14, 17, 19, 23, 25, 29, 30, 35, 36, 40]. This is not only limited to the amount and quality of available documentation but also concerns the layout of the user interface and whether that is logical and intuitive.

The *level of detail* [5, 6, 10, 14, 19, 20, 23, 25] of models, *model re-usability* [5, 8, 14, 19, 20, 30, 35, 36, 40, 42], the ability to *chain* [5, 8, 14, 19, 20, 35, 36] multiple models (so that the output of one model can be fed as input to another model) are also examples of features in this criterion.

4.1.2 Coding Aspects

Programming flexibility [3, 5, 8, 10, 13, 14, 19, 20, 30, 36, 40, 42] is the counterpart of modelling flexibility when using a programming language to model. General-purpose programming languages are generally more flexible that simulation programming languages, which are often designed for one specific domain. Simulation programming languages, on the other hand, provide more power in the domain they were designed for. Two examples of simulation frameworks using general-purpose programming languages are SimPy [41] (Python) and simmer [44] (R). In addition, most COTS simulation software packages, e.g. Arena or Simio, provide support for extending the graphical model with programmed scripts for enhanced flexibility.

Access to the source code [5, 13, 19, 29, 30, 40, 42, 43] can also be very practical for extending the tool being used. However, this requires knowledge in both the programming language the tool was written in as well about the software architecture.

These two features define the criterion *coding aspects* [3, 5, 8, 10, 13, 14, 15, 17, 19, 20, 23, 30, 36, 40, 42, 43].

4.1.3 Modelling Assistance

The criterion *modelling assistance* [3, 5, 6, 14, 15, 19, 20, 23, 30, 36] goes beyond the documentation and training provided by the software vendor and is closely linked to user-friendliness. A "virtual assistant" would guide the user through the various steps of a simulation study with prompts and dialogue boxes easing the entire process [30].

4.2 Simulation Functionalities

The criterion *simulation functionalities* [5, 6, 8, 10, 14, 23, 25, 29, 36, 40, 43] enable the user to simulate the created (or imported) models. Different simulation approaches include [36]: Discrete-Event Simulation (DES), Continuous Simulation, System Dynamics, and Agent-Based Simulation. The framework presented in this study is more targeted towards DES tools, but it is also applicable for other types of simulation.

4.2.1 Visual Aspects

The criterion *visual aspects* [3, 5, 6, 10, 11, 13, 14, 15, 17, 19, 20, 23, 25, 29, 36, 40, 43] of the simulation tool cover the graphical presentation of simulation models and *animation* [3, 5, 6, 8, 10, 13, 14, 17, 19, 20, 23, 25, 29, 30, 36, 40, 42, 43] of the simulation [5]. Animations and *dynamic display* [3, 5, 10, 13, 14, 19, 23, 29, 30, 40, 42, 43] of the values of variables and attributes can be very practical for debugging models by analysing the state of the system at any time. This also gives a nice impression of the system to aid consultants to explain problems and opportunities to their clients.

4.2.2 Simulation Capabilities

Model reliability [5, 14, 19, 23, 36] is linked to the criterion *simulation capabilities* [5, 6, 14, 19, 23, 25, 36] of a simulation tool. This means that the model is reliably simulated according to the theory of simulation.

Other features, like simulating *queueing policies* [3, 5, 10, 13, 14, 19, 30, 40, 42, 43] (FIFO, LIFO, priority, etc.) and calculating *costs* [5, 6, 13, 14, 23, 25, 40] for processing tasks and assigning resources also belong to this category.

4.2.3 Testability

Testing [3, 5, 6, 10, 14, 15, 17, 19, 20, 23, 29, 30, 40, 42, 43] models is an important step during any simulation study to increase the model's credibility. Every model should undergo two types of tests [4]:

- Verification: the model should be analysed and tested to make sure that it operates as intended ("[...] building the model correctly" [4]), and
- Validation: the model's behaviour and output should correspond to the reality ("[...] building the correct model" [4]).

Verification [3, 5, 6, 10, 13, 14, 19, 23, 25, 29, 30, 35, 40, 42] focusses, therefore, more on the syntax, whereas validation centres around the semantics of the model. Syntax debugging, just like in any IDE, can relatively easily be automated since it is based on predefined rules. Debugging of semantics, however, is a lot more complex and may require more human input and experience [28].

In DES *event stepping* [5, 14, 19, 20, 30, 40, 42] is commonly used to verify a model. This feature allows forwarding the clock to the next event, allowing the user to "step" from event to event. *Breakpoints* [3, 5, 10, 14, 19, 29, 30, 40, 42, 43] can be used to pause the simulation when a certain condition is met.

Trace files [3, 5, 10, 14, 19, 20, 30, 40, 42, 43] can be used to statistically verify the model's output. These files contain data collected about the state of the model for each task executed during the model run [30].

4.2.4 Experimentation Facilities

Experimentation facilities [5, 10, 13, 14, 15, 19, 20, 23, 35, 36] are required for comparing various alternative models. In this subcategory are classified: *warm-up period* [3, 5, 10, 14, 17, 19, 20, 25, 30, 40, 42, 43], which is used to advance the simulation to the steady state of the system before collecting output data, *automatic batch run* [3, 5, 10, 14, 17, 19, 20, 25, 30, 40, 42, 43], which runs the simulation multiple times to reduce variability, and *automatic determination of run length* [5, 10, 14, 19, 42, 43], which optimises the trade-off between variability and the time needed to run the system.

4.2.5 Statistical Facilities

A simulation tool also requires *statistical facilities* [3, 5, 6, 8, 10, 11, 14, 15, 17, 19, 20, 23, 25, 30, 35, 36, 40, 43]. To simulate real-life variations we can use *statistical distributions*, which could be *theoretical* [3, 5, 6, 10, 14, 19, 20, 23, 25, 30, 40, 42, 43] or *user-defined* [3, 5, 19, 20, 30, 40] (empirical). To draw a sample from these distributions *random number streams* [5, 10, 14, 19, 25, 30, 40, 42, 43] are used, which are necessary for analysing experiments. *Confidence intervals* [3, 5, 10, 17, 19, 25, 29, 30, 40] can be used for proving statistical significance.

4.3 Integration Functionalities

The criterion *integration functionalities* [3, 5, 11, 17, 20, 36, 42, 43] can be used to "integrate" the information gained from the real world into the model. The next step would be simulating the model with the tools described in Subsection 4.2. The final step is to "integrate" the knowledge discovered by simulating the model back in the real world by putting these insights into actions.

4.3.1 Input and Output Capabilities

The ability to *import data* [3, 5, 6, 8, 13, 14, 17, 19, 23, 29, 30, 35, 36, 40, 42, 43] allows the user to work with data gathered from the real world. *Distribution fitting* [5, 14, 19, 20, 23, 29, 30, 36] could be used to convert the data into a parametrised distribution, introducing variability into the system. The gathered data from simulating the model could also be *exported* [3, 5, 13, 14, 17, 19, 20, 23, 29, 30, 35, 36, 40, 42, 43] for further analysis. These features belong to the criterion *input and output capabilities* [3, 5, 11, 13, 14, 15, 17, 20, 23, 25, 30, 35, 36, 40, 42, 43]

4.3.2 Analysis Capabilities

The *analysis capabilities* [3, 5, 6, 10, 11, 14, 15, 23, 25, 29, 35, 36, 40, 42] aid the user to interpret the results from simulating the model, e.g. *output data analysis* [3, 5, 6, 8, 10, 13, 14, 19, 23, 25, 29, 30, 35, 36, 40, 42, 43], *what-if-analysis* [5, 6, 10, 14, 25, 29, 42] to compare different scenarios, *optimisation of parameters* [5, 6, 13, 14, 29, 30, 36, 40, 42, 43], or *decision-making support* [5, 6, 14, 25] facilitating the user to make the right decisions.

4.4 Hardware and Software

The fourth main category of criteria, *hardware and software* [5, 6, 8, 11, 14, 23, 25, 29, 40, 42], contains features with regards to hardware requirements, software, financial costs, etc.

4.4.1 Software Quality and Compatibility

Software quality and compatibility [3, 5, 6, 13, 14, 15, 20, 23, 25, 29, 30, 36, 40, 42, 43] evaluates the interfaces to other software packages (e.g. spreadsheet software, ERP systems, statistical software, etc.) and the general quality of the software package itself. The *robustness* [5, 6, 10, 14, 19, 36] of the software reflects the “capability of the software package to run consistently without crashing” [22] (which can be very frustrating for the user, especially when data was lost). *Performance* or *efficiency* [5, 6, 10, 11, 14, 17, 19, 20, 23, 25, 30, 36, 40, 42, 43] measures the ability to process the input data into output data in an acceptable amount of time w.r.t. the amount of data to process [22].

4.4.2 Hardware Requirements

Various *hardware requirements* [3, 5, 6, 11, 17, 19, 25, 29, 30, 36, 40, 42, 43], such as internal memory, storage space, operating system, etc. are also very important to keep in mind when selecting simulation software. Choosing software that requires new hardware could significantly increase the Total Cost of Ownership (TCO).

4.4.3 User Support

User support [3, 5, 6, 14, 15, 19, 20, 23, 25, 30, 40, 43] of a software tool contain *documentation* [3, 5, 6, 8, 11, 13, 14, 17, 19, 23, 25, 29, 30, 40, 42, 43] for additional help and information, *tutorials* [3, 5, 6, 8, 11, 13, 14, 17, 19, 23, 25, 29, 30, 40, 42, 43] or *training courses* [3, 5, 6, 11, 13, 14, 17, 19, 30, 40] to learn how to use the software package, or *consultancy* [3, 5, 6, 11, 13, 14, 17, 19, 30, 40] provided by the vendor for technical assistance in case of problems with the software, or additional help in using the software correctly.

4.4.4 Financial and Technical Features

The *TCO* [3, 5, 8, 11, 17, 19, 20, 30, 40, 42, 43] of the software package contains all the direct and indirect costs of the system. These *financial features* [3, 5, 8, 15, 19, 20, 23, 30, 40] includes license costs (the price for each user to work with the software), installation and implementation, maintenance, training (for training the user to work with the system), upgrading (the cost when a newer version is released and it is decided to upgrade), and potential hardware costs as explained before [22].

The amount and frequency of *updates* [5, 6, 13, 14, 19, 40, 42] introducing new features and fixing old problems belong to the criterion *technical features* [5, 15, 19, 20, 23] of the tool.

4.4.5 Popularity

Reputation of the software vendor [3, 5, 6, 11, 17, 19, 23, 30, 40, 42, 43], the *spread* [5, 17, 19, 30] – i.e. the amount of users working with the software – and the *age* [5, 19, 30] of the tool could be used to measure the *popularity* [3, 5, 6, 11, 17, 19, 23, 30] of the simulation software tool.

5 Results

The results of the interviews are described here. For the sake of preserving the anonymity of the respondents, only masculine pronouns are used in all quotes. The structure of questions of the interview protocol (see Appendix A) is followed in this section.

5.1 Interviewee Background

All interviews were started with a couple of short questions about the professional experience of the interviewee. On average, the respondents were eight and a half years active within the domain of simulation, with a minimum of three years and a maximum of nineteen years. Table 5.1 gives an overview of the years of experience of each respondent broken down between the two categories. The number of simulation projects they have each worked on – or were otherwise involved – ranged from a couple to over 200 projects.

Respondents	Years of Experience			Average
Simulation experts	5	11	19	**11.67**
Academic researchers	3	6	7	**5.33**
			Average	**8.5**

TABLE 5.1: Years of experience of the respondents within the domain of simulation.

5.2 Simulation Software

The respondents use – or have used – several simulation tools. Table 5.2 gives an overview broken down between the two categories of respondents. No respondent indicated to use programming libraries or frameworks, such as SimPy (Python) [41] or simmer (R) [44].

Simulation Software	Simulation Experts	Academic Researchers	Total
Arena	3	3	**6**
Simio	3	-	**3**
AnyLogic	1	2	**3**
MATLAB	1	-	**1**
Custom developed	1	-	**1**

TABLE 5.2: Simulation software usage.

Some respondents use multiple tools. In that case the problem at hand will determine which tool is most suitable for the simulation study. One respondent uses a custom developed simulation software package. He indicated that COTS tools are "usually too limited and provide little control over the functionality, [i.e.] they are like 'black-boxes'". Another respondent also indicated a potential limitation of COTS tools: "COTS tools are generally developed for model developers and not for end-users". This statement refers to the fact that usually, a custom developed user interface or dashboard is needed to present the results of the simulation study to clients, i.e. the end-user.

5.3 Modelling Functionalities

The following questions related to the modelling functionalities of simulation tools and whether the respondent preferred working visually in a graphical interface, i.e. with building blocks, or programming the model. One respondent commented that this "[. . .] generally depends on personal preference and background". His experience tells that people with an IT background tend to program the model rather than to work visually. On the other hand, people with an industrial background are more inclined to work visually. They generally have less programming experience and that deters them. He also commented that models are sometimes "[. . .] built visually to enhance maintainability for people with a non-IT background".

There is a consensus among the interviewees that complex models – or parts of models – almost always have to be programmed. Programming the model provides a "[. . .] higher degree of flexibility". One respondent commented that "[. . .] some problems are very cumbersome to model with building blocks". He argued that programming that part of the model is much more convenient and organised than modelling it with "[. . .] a lot of building blocks".

The interviewer also asked whether the interviewees are familiar with Business Process Model and Notation (BPMN) and if they have used it during simulation studies. Only two out of six respondents were familiar with BPMN. One respondent indicated that "BPMN is used to describe processes on a high level. It is a modelling notation, not a simulation language". Another interviewee commented that he could see the benefit of using BPMN in simulation studies. "BPMN has 'modelling rules', which could make modelling more efficient. Each tool uses its own notations [– i.e. no standardisation as with BPMN –] and sometimes it is unclear which notation means what."

5.4 Simulation Functionalities

With regards to the aesthetic aspect of simulation models, the answers of the respondents differ. One academic researcher commented that the results are more important. He argues that "[. . .] visuals don't contribute to the quality of the academic research; [. . .] 'functionality prevails over visuals' ". The simulation experts have a different opinion. A respondent from this group said that "it really depends on the customer; some customers don't care about visuals, others care everything about it". Usually, custom user interfaces are built to show dashboards and animations of the process. Another respondent of this group commented that "[. . .] impressive visuals are often used as a selling point of the results towards higher management".

The following question asked the interviewees which techniques they use when testing their models. Indeed, validating the simulation model is one of the most important tasks during a simulation study [4]. Two categories of testing methods can be identified with some concrete examples:

- Basic testing, mainly for verification:
  - Event stepping and breakpoints, which are mostly used when something seems to be wrong,
  - Animations for visually inspecting the flows of entities in the process, and
  - Variable watches[1] to monitor various variables over time.
- Advanced testing, both for verification and validation:
  - Trace files which can be used to construct events out of the process flow. Domain experts recognise these events and are able to indicate anomalies,

[1] Watches are often used to display the value of variables when debugging an application using an IDE. The same principle is also applied in simulation software.

- Animations and face validation ("[...] expert walk-through [...]"),
- Key Performance Indicator (KPI) comparison to compare the simulation output with historical data, and
- Sensitivity and extreme case testing to analyse the performance of the process under various conditions. Domain experts can use this data to indicate strange behaviour.

Especially event stepping and breakpoints, animations, trace files, and KPI comparison were mentioned most by the respondents.

Other important features of simulation software tools according to the respondents are user-friendliness, re-use of model components (which "[...] could save a lot of time during modelling"), comprehensive output statistics and KPIs ("[...] the pre-built reports usually provide too little information", which means that manual output extraction and analysis is often needed), and good integration with programming languages for "[...] more modelling flexibility".

5.5 Integration Functionalities

The next part of the interview dealt with other software tools the respondents use in addition to the simulation software package. Table 5.3 provides an overview of the used software, the number of respondents using it, and the purpose of the tool during simulation studies. This table shows for instance that all interviewees regularly use Microsoft Excel to prepare their input data and to analyse the results of the simulation study with graphs and tables. However, when datasets are large, programming languages, such as R and Python are used mainly for performance reasons. The Arena Input Analyser was only regularly used by one respondent to fit distributions on input data. Other respondents preferred to rely on other tools that offer extensive functionalities.

Software Tool	Respondents	Purpose
Microsoft Excel	6	Input and output analysis (graphs, tables, etc.), data cleansing, etc.
R	5	Input and output analysis (graphs, tables, etc.), data cleansing, etc.
SPSS	3	Statistical analysis on mainly output data.
Tableau	3	Visualisation and analysis of data.
Custom developed	3	ETL, dashboards, views, scenario management, etc.
Spotfire	2	Visualisation and analysis of data.
Python	2	Statistical analysis on mainly output data.
JavaScript	2	Describing process flows and decision tables.
Arena Input Analyser	1	Input analysis (distribution fitting).
Microsoft Visual Studio	1	Automatic running of models and output data gathering.

TABLE 5.3: Additional software tools usage.

One general remark given by a respondent was that "simulation packages usually have little links with other software tools". Other interviewees indicated that they do not mind using multiple software tools. They assert that another tool specifically designed for a particular purpose is "[...] usually better and contains more useful features".

5.6 Hardware and Software

The respondents were also asked whether they have ever contacted the simulation tool distributor in case they had a question or problem. Table 5.4 summarises the context of the inquiry, the number of respondents, and an indication of the frequency of the inquiry. Technical problems – such as software bugs, failing licence activations, problems with installing the software, etc. – was a common phenomenon for which contact was made with the supplier of the software.

Context of Inquiry	Respondents	Frequency
Technical problems (e.g. bugs, licences not working, installation problems, etc.)	6	Regularly
Lacking features	3	Occasionally
Support questions during modelling (constructs)	3	Occasionally
Giving constructive feedback	2	Occasionally

TABLE 5.4: Interactions with simulation software vendors.

With regards to the hardware that is being used during simulation studies, all respondents indicate that they use a laptop rather than a powerful desktop. They all indicated that "a laptop is generally fast enough for modelling". The actual simulation, i.e. running the model, however, is usually performed on a powerful server to lower the run time. The academic researchers mainly use desktops for running the models, because they do not have access to a powerful server supporting their simulation software package.

The last question in this section queried the respondents for the reasons why they – or their company – chose for the particular simulation software package they are currently using. Their answers ranged from "[the tool] was already in use when I started working here", price, and available documentation. One respondent using a custom developed simulation software package indicated that this allows for "[...] more flexibility and customisation". He additionally indicated, however, that "[...] fast model prototyping is usually faster in COTS tools".

5.7 Characteristics Framework

All interviewees could support the four main categories of the proposed framework described in Section 4; as "[...] every feature [or characteristic] probably fits within one category". However, one respondent emphasized that "[...] not all categories have the same priority". He considered *Modelling Functionalities (MF)* more important than the other categories.

Next, the respondents were asked to complete Table 2.1 (see Subsection 2.2.4). Figure 5.1 illustrates the average importance given by the experts to each criterion of the framework presented in Section 4. The criteria on the left side of the graph are "extremely" to "very strongly" important according to both groups of experts. In contrast, the criteria on the right side of the graph are less easy to categorise according to importance because the academic researchers and simulation experts do not share the same opinion. It is also noteworthy that the simulation experts tend to assign "extreme" importance values (e.g. "1" – "extremely important" and "9" – "not important at all"), whereas the academic researchers gave more nuanced values.

General Modelling Features (G), *Simulation Capabilities (Si)*, *Testability (Ts)*, *Coding Aspects (C)*, *Software Quality and Compatibility (Sw)*, and *User Support (U)* are considered to be the most important characteristics of simulation software packages according to both groups. On the other hand, *Input and Output Capabilities (IO)*, *Statistical Facilities (St)*, *Hardware Requirements (H)*, *Analysis Capabilities (A)*, *Experimental Facilities (E)*, and *Financial and Technical Features (Ft)* are not important at all in the opinion of the simulation experts, whereas *Popularity (P)*, *Visual Aspects (V)*, and *Modelling Assistance (M)* are not very important as stated by the academic researchers.

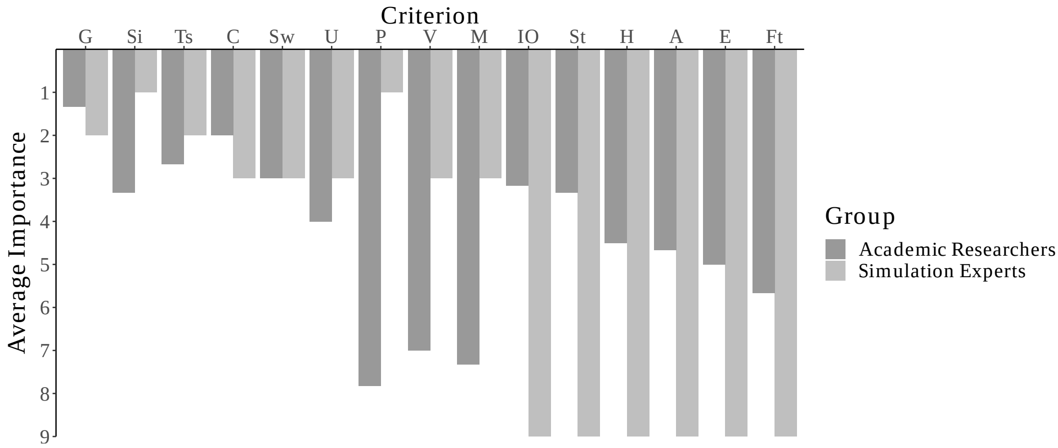

FIGURE 5.1: Average importance weights of simulation software characteristics. "1" indicates that the criterion is considered "extremely important" and "9" means that it is "not important at all".

Then, the respondents were asked to indicate the most important individual features by assigning coins (see Subsection 2.2.4 for a detailed explanation of the ranking process). Table 5.5 gives the top-10 most important features of simulation software packages. Both *Modelling* and *Programming Flexibility* are considered to be the most important characteristics, followed by *Documentation and Tutorials*, *Model Reliability*, and *Model Verification*. This question was also used to validate the criteria ranking displayed in Figure 5.1. If the respondents made their ranking for Table 5.5 in such a way that the most important individual features belong to criteria more located on the right side of Figure 5.1, then this would indicate that either: (1) the respondents were inconsistent in the ranking of the criteria and individual features, or (2) the definitions of the criteria was unclear and, therefore, the respondents answered from their interpretation – which would indicate that the proposed framework is unclear according to the experts. Only *Input Files* (*Input and Output Capabilities (IO)*) and *What-if-Analysis / Scenarios* (*Analysis Capabilities (A)*) showed this inconsistency. However, these features were only considered important by the group of academic researchers, who gave an average importance score of 3.17 and 4.67 on *Input and Output Capabilities (IO)* and *Analysis Capabilities (A)* respectively, indicating a "strong" to "very strong importance" on the scale presented in Table 2.1. Therefore, the author concludes that the experts agree with the proposed framework.

Rank	**Feature**
1	Modelling Flexibility
2	Programming Flexibility
3	Documentation and Tutorials
4	Model Reliability
5	Model Verification
6	User-Friendliness
7	Performance / Efficiency
8	Input Files
9	Access to Source Code
10	What-if-Analysis / Scenarios

TABLE 5.5: Simulation software feature ranking.

5.8 Closing the Interview

The last part of the interview gave the respondents the chance to indicate opportunities for improvement for the current generation of COTS simulation software packages. One of the biggest remarks of the interviewees was the limited output flexibility ("[...] the pre-built reports usually provide too little information" and "extracting data from [the tool] is not very easy"). Another observation was that "simulation packages usually have little links with other software tools". Other frequent mentioned limitations of current tools (and opportunities for improvement) included lacking features, "[...] limited number of modules [– i.e. building blocks –], which makes programming necessary", better integration with popular programming languages for more modelling flexibility, "[...] limited documentation on the use of programming languages", scalability of models ("[...] models are often never used again after the simulation study has ended"), "[...] poor simulation performance [– i.e. running speed –] for complex models", "[...] stability issues, including bugs and errors", keeping track of scenarios for what-if-analysis, and "in general there is little information and help available on the internet [about simulation and building models], especially in comparison with [general-purpose] programming languages".

The respondents were also encouraged to give their creativity free rein w.r.t innovative new features for the future development of simulation software packages. One example is the ability to collaborate in real-time on the same model ("It is not possible to work together on the same model at the same time. This makes modelling a 'solo job' "). Another interesting new feature is the ability "[to] go back in time". To the knowledge of the respondents and author of this study, it is currently only possible to advance the simulation clock and not to rewind it. W.r.t. testing the models one interviewee suggested that "[simulation] tools could do more testing, for example, unit testing[2]". Lastly, the majority of the respondents felt that the current simulation software packages have insufficient links with other tools. Therefore, integration with Application Programming Interface (API)s, e.g. Google Maps or HERE Maps for finding routes, optimisation algorithms, impact and collision detection, data input and output from and to databases, etc. would greatly increase the flexibility of simulation tools.

[2] Unit testing is a software testing method extensively used in computer programming. It is the "lowest" level of testing to assess "units" of software w.r.t. the implementation [2].

6 Discussion

From the results in Section 5, it follows that the interviewees agreed with the proposed framework presented in Section 4. No respondent indicated that he or she felt that particular criteria were missing. They also supported the division with four categories; as "[. . .] every feature [or characteristic] probably fits within one category".

Ranking the characteristics of the framework described in Section 4 shows that both groups – simulation experts and academic researchers – indicate that *General Modelling Features (G)*, *Simulation Capabilities (Si)*, *Testability (Ts)*, *Coding Aspects (C)*, *Software Quality and Compatibility (Sw)*, and *User Support (U)* are considered the most important characteristics of simulation software packages, as shown in Figure 5.1. This is also supported by Table 5.5 – i.e. the top-10 most important features of simulation software packages – which contains mostly individual features belonging to these characteristics. The different opinion on the aesthetic aspect of simulation models is also illustrated in Figure 5.1. The simulation experts indicate that *Visual Aspects (V)* is an important characteristic, whereas the academic researchers suggest that this is not important at all. Another interesting difference is *Popularity (P)*. Whereas the academic researchers are more interested in detailed documentation and references, the simulation experts indicate that a higher spread of a particular simulation software package makes it "[. . .] easier to share models and results".

When it comes to modelling, all respondents agree that flexibility is of paramount importance. This is also reflected in the feature ranking in Table 5.5 in which *Modelling Flexibility* is ranked number one. This higher degree of flexibility is, however, mostly obtained by programming parts of the model, or even the entire model. Sometimes a lot of building blocks are needed for what could be done with just a few lines of code. Programming that part not only increases the interpretability, but also the maintainability of the model. Therefore, *Programming Flexibility* received second place.

The use of BPMN could also enhance the interpretability of a model. However, there are only a limited amount of simulation software packages which support modelling with BPMN and even if they do, it is not always clear what the definitions of the symbols are. This is probably mainly due to the fact that BPMN does not include all components that are required to model a simulation model and consequently each tool defines its own implementation. The fact that the majority of the respondents are unfamiliar with this modelling notation confirms that BPMN is not a well-known paradigm within the domain of simulation. Nevertheless, within a business context Business Process Management (BPM) is getting more adopted [16]. The integration of BPMN in simulation software packages could, therefore, narrow the gap between business and simulation.

The aesthetic aspect of simulation models is where the two groups – simulation experts and academic researchers – differ substantially in opinion. In an academic setting, the results predominate the visual aspect; "[. . .] functionality prevails over visuals". In contrast, the simulation experts indicate that it depends on the customer's wishes and expectations. In a business setting, impressive visuals are often used to convince senior management.

The most popular simulation tool in this study is Arena: five out of six respondents use it actively. In addition, most interviewees have tried only a limited amount of different simulation software packages. This might bias their views w.r.t. limitations of the current generation of COTS simulation software packages, although respondents who used several tools seem to experience the same limitations.

All interviewees indicated to use various other software tools in addition to the simulation software package. Microsoft Excel is regularly used for analysing and cleansing input and output data. The simulation experts also often use data visualisation tools like Tableau and

Spotfire to present their findings to their end-users. This indicates that simulation tools rarely stand alone and additional tools are often used to compensate for shortcomings or supplementary functionalities.

The results of this study are in line with the findings in previous studies presented in Subsection 3.4, in which the input from simulation practitioners was included. Table 6.1 gives an overview of important characteristics of this study in comparison with other studies. It should be noted, however, that the authors of these studies did not always clearly define the definitions of these features. Given the context in which the features were mentioned, the author of this study determined which terminology best matched the one used in this work. The results are very similar; six characteristics of Table 6.1 are also included in the top-10 of Table 5.5. Especially *Model Verification* and *Flexibility* are considered to be the most important characteristics of simulation software packages.

Characteristic	**Studies**
Flexibility	[3, 11, 13, 42]
Model Verification	[3, 13, 29, 42]
Documentation and Training	[13, 29, 42]
Visual Aspects	[11, 13, 29]
Statistical Facilities	[3, 11, 13]
User-friendliness	[11, 29]
Input and Output Capabilities	[29, 42]
Access to Source Code	[13]
Analysis Capabilities	[11]

TABLE 6.1: Comparison of the results with other studies.

7 Conclusion

The goal of this study was to identify differences between evaluation frameworks and required features of simulation software packages on the one hand and on the other hand the opinion of practitioners. First, prior research was consulted to formulate an integrated framework on the required characteristics. Next, six in-depth interviews were conducted to assemble the views and thoughts of simulation practitioners – both simulation experts and academic researchers. Lastly, the results from these interviews were summarised and compared with prior research.

This study confirms that the most important characteristics of simulation software packages include flexibility (both modelling with building blocks and programming), facilities for testing the models, the overall user-friendliness, and simulation capabilities. In addition, the results of the interviews revealed two interesting differences between the two groups of respondents – i.e. simulation experts and academic researchers – and provided opportunities for improvement.

First, in an academic setting, the results are much more important than the aesthetic aspect of simulation models, whereas, in a business context, visuals are often used as a selling point towards senior management. Second, the popularity of the tool is considered important by the simulation experts, but not important at all by the academic researchers. The spread of a specific tool eases sharing models and results, however, academic researchers prefer clear and complete documentation.

The biggest opportunity for improvement for the current generation COTS simulation software packages indicated by the interviewees is more and better integration with APIs and other software tools, e.g. Google Maps or HERE Maps for finding routes, optimisation algorithms, impact and collision detection, data input and output from and to databases, etc. This would not only greatly extend the capabilities of these tools, but also make it easier to import and extract data from simulation models.

Other opportunities indicated by the respondents were real-time collaboration on the same simulation model, more advanced testing (e.g. unit testing), and further integration with popular programming languages to increase the flexibility for both modelling and output analysis. This would also facilitate the definition of KPIs, because the results from pre-built reports are generally too limited and additional data extraction is almost always needed.

The fact that only six in-depth interviews have been conducted could be seen as a limitation of this study. It must be emphasized that in this study, in-depth interviews were chosen to collect richer information. Furthermore, the results obtained from Table 2.1 during the interviews cannot be used for an AHP MCDM problem. The respondents tended to cluster their rankings when using this table, whereby the transformation of the results into a PCM may be questioned.

Further research could, therefore, extend the work from this study by replicating the described methods on a much larger group of experts in the domain of simulation, using e.g. a survey to determine the most important characteristics of simulation software packages. In addition, it could also be interesting to analyse the modelling behaviour of those experts to gather more insights on how the implementation of programming can increase modelling flexibility and which programming languages are preferred.

A Interview Protocol

It is good practice to plan an interview carefully before starting to contact possible respondents. This provides a guideline throughout the search for respondents, contacting them, conducting the interviews, analysing the results, and reporting these results [47].

The remainder of this appendix displays the protocol used for conducting the interviews. The structure described in [26, 39] was used as a guideline.

A.1 What, Why, and How

The *aim* of the interviews will be identifying the most important characteristics of a simulation software package and opinions about characteristics that are currently "underdeveloped" from the perspective of two main categories of subjects: (1) simulation experts, and (2) academic researchers.

These two categories may require different features, because of their knowledge, expectations, and angle of approach. Therefore, it is interesting to get the opinions of both groups and analyse to what extent they are similar and different.

The *reason* for choosing for interviews, instead of another questioning method originates from the very subjective nature of ranking requirements. Using, for instance, an online questionnaire limits the respondent's ability to elaborate more on why one feature would be more important than another. Therefore, in-depth interviews are assumed to deliver more useful insights [7].

The interview is adequately *supported by the literature* by a comprehensive literature study (see Section 3) that has been carried out on the interview's topic.

Given the in-depth nature of the interviews, the *number of respondents* will be smaller compared to, e.g. survey research; preferably around six or seven, depending on the willingness of respondents to participate. According to Kvale, this is common in interview studies. He suggests that $5 - 25$ respondents for a qualitative interview study is a perfectly acceptable amount, depending on the time and resources available [26].

The *selected respondent* should fulfil one of following conditions:

- The respondent is a simulation expert, i.e. his or her main job function consists of modelling, simulating, and analysing the results of simulation studies (within the domain of process simulation), or
- The respondent is an academic researcher whose main research field consists of simulating and analysing the results of simulation studies (within the domain of process simulation).

The respondents who agreed to participate in the study received a couple of *introductory questions* prior to the interview itself to prepare themselves – e.g. "Which simulation tools do you use (or have you used in the past)? Which of these is your favourite and why?" – allowing to smooth out the process during the interviews.

During the interviews, a *semi-structured interview* style will be used. This type of interview gives the interviewer a clear guideline of the topics he or she wants to cover and, at the same time, leaves openness for the interviewee to elaborate more on the given answers [26]. *Manual notes* will be made by the interviewer. If the interviewee agrees, the interview could be *recorded* for later in-depth analysis. The interviews will take one to maximum one and a half hour.

The *deadline for conducting the interviews* will be the end of March 2019, so that the interviews can take place in February or March 2019, allowing the respondents to receive and review a draft of the report before the final submission in May 2019.

A.2 Interview Questions

This subsection contains an overview of the questions asked during the interviews. As stated before, a semi-structured interview style was used. Therefore, the questions listed below merely formed a guideline for the interviewer. Depending on the answers given by the interviewee, follow-up questions were asked for a more elaborated answer. Some questions contain possible follow-up questions in *italics*.

A.2.1 Introduction

First of all, I would like to thank you for participating in my research. During this interview I will ask you some questions about simulation software and how you use it. The goal of my research is to identify the most important characteristics of tools to perform process simulation studies. This is guiding in order to better align such tools with the wishes and expectations of users. Hopefully this can also shed light on characteristics that deserve more attention.

A.2.2 Questions

A. Interviewee Background

I would like to start with a couple of short questions about your professional experience:

- How many years do you have this position within the company?
- How many years are you already active in the domain?
- How many simulation projects have you worked on (approximately)?

B. Simulation Software

Now, I would like to go into more detail about the specific simulation software you use, and how you use them.

- Which simulation tools do you use (or have you used in the past)?
- Which of these is your favourite and why?

C. Modelling Functionalities

- Would you like to work visually, in an editor with "building blocks", or would you rather program your simulation study?
  When would you choose one over the other?
- If you work visually, have you already carried out a simulation study with Business Process Model and Notation (BPMN)?
  Did you encounter problems during modelling? What were these problems and how did you solve them?

D. Simulation Functionalities

- Do you attach a lot of importance to the aesthetic aspect of your simulation models (possibly at the request of the customer)?
- With which techniques do you test your simulation models (e.g.: trace files, event stepping, breakpoints, animations, ...)?
- During simulation, what do you think are really necessary properties that a simulation tool must have (e.g.: reuse of model components, queue algorithms, costs of processes, distributions, animations, ...)?

E. Integration Functionalities

- In addition to the simulation tool, do you use other software or does your simulation tool offer all the necessary functions (e.g.: cleaning data, analysing output, ...)?
  Which tools do you use specifically and why?

F. Hardware and Software

- Have you ever called the seller of your simulation tool to ask for an explanation if the manual was not sufficient and you could not solve the problem yourself?
- Do you use a powerful desktop for your simulation studies or is a laptop sufficient?
  Desktop / Laptop
- Was the choice for the current simulation tool that you use mainly based on the available features, or was the price also very important?
  Why did you, or your company, choose exactly that tool over another?

G. Characteristics Framework

The previous questions were divided into four categories:

- "Modelling Functionalities": The abilities to model a simulation model (via editor or programming);
- "Simulation Functionalities": The abilities to simulate and verify a simulation model;
- "Integration Functionalities": The capacities to integrate with other software and data, as well as analyses and scenarios;
- "Hardware and Software": The required hardware to use the tool, the price of the software itself and the available documentation or training.

Questions:

- These form the four major pillars that a simulation tool must meet. Do you agree with this statement? Why, or why not?
- I would like you to rank the following sub-criteria from 1 (extremely important), 4 – 5 (moderately important) to 9 (not important at all). You do not have to use every number between 1 and 9, and you are allowed to assign the same number to multiple criteria (i.e. they share the same importance). You may place the sub-criteria side by side to indicate that they are equally important or arrange them within each level to indicate subtle differences. Appendix A.2.3 shows a list of characteristics that belong to each sub-criterion.
  The interviewee is presented a table with nine importance categories (see Table A.1) and fifteen flashcards containing the sub-criteria. Table A.2 is used by the interviewer to record the answers of the interviewee.

1	*Extreme importance*
2	*Very, very strong importance*
3	*Very strong importance*
4	*Strong plus importance*
5	*Strong importance*
6	*Moderate plus importance*
7	*Moderate importance*
8	*Weak or slight importance*
9	*No importance*

TABLE A.1: Nine importance categories[1].

Criterion	**Rank**
Modelling Functionalities (MF)	
General Modelling Features (G)	
Coding Aspects (C)	
Modelling Assistance (M)	
Simulation Functionalities (SF)	
Visual Aspects (V)	
Simulation Capabilities (Si)	
Testability (Ts)	
Experimentation Facilities (E)	
Statistical Facilities (St)	
Integration Functionalities (IF)	
Input and Output Capabilities (IO)	
Analysis Capabilities (A)	
Hardware and Software (HS)	
Software Quality and Compatibility (Sw)	
Hardware Requirements (H)	
User Support (U)	
Financial and Technical Features (Ft)	
Popularity (P)	

TABLE A.2: Sub-criteria ranking.

[1]This table has been scaled down to save space. The table presented to the respondents during the interviews expanded two A4-format pages.

- You have two gold coins, three silver coins and four bronze coins. If the simulation package is freely configurable, which tools do you absolutely need (gold), which are very important to you (silver) and which tools would I like to have (bronze)?

  Note: It is possible that certain characteristics cannot operate without each other. If you think this is the case, can you indicate why you think this is the case?

  The interviewee is presented a table with the fifteen sub-criteria, each with the specific features that belong to that category (see Figure A.1). The interviewer uses Table A.3 to record the answers.

Gold	Silver	Bronze

TABLE A.3: Individual features ranking (interviewer).

H. Closing the Interview

- Are there any opportunities for improvement you can think of with simulation tools you are using? What would you like developers of simulation tools change to improve upon these points?
- Before closing the interview, do you have any other comments or topics that have not been discussed and that you would like to discuss now?

General Modelling Features (G)			**Statistical Facilities (St)**		
User Friendliness	Process Flow Diagram	Modelling Flexibility	Theoretical Statistical Distributions	User-Defined Distributions	Random Number Streams
Level of Detail	Model Reusability	Model Chaining / Modularity	Confidence Intervals		
Coding Aspects (C)			**Input and Output Capabilities (IO)**		
Programming Flexibility	Access to Source Code		Distribution Fitting	Input Files	Output Files
Modelling Assistance (M)			**Analysis Capabilities (A)**		
Visual Aspects (V)			Output Data Analysis	What-if-Analysis / Scenarios	Conclusion / Decision Making Support
Animation	Dynamic Display		Optimisation		
Simulation Capabilities (Si)			**Software Compatibility (Sw)**		
Model Reliability	Queueing Policies	Costs	Robustness	Performance / Efficiency	
Testability (Ts)			**Hardware Requirements (H)**		
Model Verification	Trace Files	Event Stepping	**User Support (U)**		
Breakpoints			Documentation and Tutorials	Consultancy and Training	
Experimentation Facilities (E)			**Financial and Technical Features (Ft)**		
Warm-up Period	Auto Run Length	Auto Batch Run	Pricing and TCO	Updates	
			Popularity (P)		
			Age	Spread	Reputation of Supplier

FIGURE A.1: Individual features ranking (interviewee).

A.2.3 Simulation Software Characteristics Framework

- Modelling Functionalities (MF):
  - General Modelling Features (G):
    - User-Friendliness, Process Flow Diagram, Modelling Flexibility, Level of Detail, Model Re-usability, and Model Chaining / Modularity.
  - Coding Aspects (C):
    - Programming Flexibility, and Access to Source Code.
  - Modelling Assistance (M).
- Simulation Functionalities (SF):
  - Visual Aspects (V):
    - Animations, and Dynamic Display.
  - Simulation Capabilities (Si):
    - Model Reliability, Queueing Policies, and Costs.
  - Testability (Ts):
    - Model Verification, Trace Files, Event Stepping, and Breakpoints.
  - Experimentation Facilities (E):
    - Warm-up Period, Auto Run Length, and Auto Batch Run.
  - Statistical Facilities (St):
    - Theoretical Statistical Distributions, User-Defined Distributions, Random Number Streams, and Confidence Intervals.
- Integration Functionalities (IF):
  - Input and Output Capabilities (IO):
    - Distribution Fitting, Input and Output Files.
  - Analysis Capabilities (A):
    - Output Data Analysis, What-if-Analysis / Scenarios, Conclusion / Decision-Making Support, and Optimisation.
- Hardware and Software (HS):
  - Software Quality and Compatibility (Sw):
    - Robustness, and Performance / Efficiency.
  - Hardware Requirements (H):
  - User Support (U):
    - Documentation and Tutorials, and Consultancy and Training.
  - Financial and Technical Features (Ft):
    - Pricing and TCO, and Updates.
  - Popularity (P):
    - Age, Spread, and Reputation of the Supplier.

Glossary

A

AHP	Analytic Hierarchy Process 5, 7–9, 25
API	Application Programming Interface 22, 25

B

BPM	Business Process Management 8, 23
BPMN	Business Process Model and Notation 13, 18, 23, 28
BPR	Business Process Re-engineering 7
BPS	Business Process Simulation 7, 8, 10, 13

C

COTS	Commercial off-the-Shelf 1, 13, 17, 20, 22, 23, 25

D

DES	Discrete-Event Simulation 8, 9, 14

E

ERP	Enterprise Resource Planning 16
ETL	Extract, Transform and Load 19

F

FAHP	Fuzzy-Analytic Hierarchy Process 8, 9
FIFO	First In, First Out 14

G

GUI	Graphical User Interface 1, 13

I

IDE	Integrated Development Environment 14, 18

K

KPI	Key Performance Indicator 19, 25

L

LIFO	Last In, First Out 14

M

MCDM	Multiple-Criteria Decision-Making 5, 8, 9, 25

P

PCM Pairwise Comparison Matrix 5, 25
PSI Preference Selection Index 8

T

TCO Total Cost of Ownership 16
TOPSIS Technique of Order Preference by Similarity to Ideal Solution 5, 8

Bibliography

[1] Yasmeen Alomair, Iftikhar Ahmad, and Abdullah Alghamdi. "A Review of Evaluation Methods and Techniques for Simulation Packages". In: *Procedia Computer Science* 62 (2015), pp. 249–256. ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.08.447.

[2] Paul Ammann and Jeff Offutt. *Introduction to Software Testing*. 2nd ed. University Printing House, Cambridge CB2 8BS, UK: Cambridge University Press, 2017, p. 364. ISBN: 978-1-107-17201-2. DOI: 10.1017/9781316771273.

[3] Ali Azadeh, Salman Nazari-Shirkouhi, and Kamran Rezaie. "A Robust Decision-Making Methodology for Evaluation and Selection of Simulation Software Package". In: *International Journal of Advanced Manufacturing Technology* 47.1-4 (2010), pp. 381–393. ISSN: 1433-3015. DOI: 10.1007/s00170-009-2205-6.

[4] Jerry Banks et al. *Discrete-Event System Simulation*. 5th ed. Edinburgh Gate, Harlow, Essex, UK: Pearson Education Ltd., 2014, p. 560. ISBN: 978-1-292-02437-0.

[5] Vesna Bosilj-Vukšić, Vlatko Ceric, and Vlatka Hlupic. "Criteria for the Evaluation of Business Process Simulation Tools". In: *Interdisciplinary Journal of Information, Knowledge, and Management* 2 (2007). Ed. by Kathy Lynch, pp. 73–88. DOI: 10.28945/101.

[6] Padraig Bradley et al. "Business Process Re-engineering (BPR) — A Study of the Software Tools Currently Available". In: *Computers in Industry* 25.3 (1995), pp. 309–330. ISSN: 0166-3615. DOI: 10.1016/0166-3615(94)00044-q.

[7] Svend Brinkmann. *Qualitative Interviewing*. Understanding Qualitative Research. 198 Madison Avenue, New York, NY, USA: Oxford University Press, 2013, p. 177. ISBN: 978–0–19–986139–2.

[8] Jeffery K. Cochran and Hung-Nan Chen. "Fuzzy Multi-Criteria Selection of Object-Oriented Simulation Software for Production System Analysis". In: *Computers and Operations Research* 32.1 (2005), pp. 153–168. ISSN: 0305-0548. DOI: 10.1016/S0305-0548(03)00209-0.

[9] Ole-Johan Dahl and Kristen Nygaard. *SIMULA: A Language for Programming and Description of Discrete Event Systems. Introduction and User's Manual.* Norwegian Computing Center, 1966.

[10] Nadja Damij et al. *Ranking of Business Process Simulation Software Tools with DEX/QQ Hierarchical Decision Model*. Electronic Article. 2016. DOI: 10.1371/journal.pone.0148391.

[11] Lesley Davis and Glyn Williams. "Evaluating and Selecting Simulation Software Using the Analytic Hierarchy Process". In: *Integrated Manufacturing Systems* 5.1 (1994), pp. 23–32. ISSN: 0957-6061. DOI: 10.1108/09576069410050314.

[12] Lloyd Dugan et al. *Business Process Simulation Specification*. Electronic Article. 2016. URL: http://www.bpsim.org/specifications/2.0/WFMC-BPSWG-2016-01.pdf.

[13] Alexandre Magno Castañon Guimarães, José Eugenio Leal, and Paulo Mendes. "Discrete-Event Simulation Software Selection for Manufacturing Based on the Maturity Model". In: *Computers in Industry* 103 (2018), pp. 14–27. ISSN: 0166-3615. DOI: 10.1016/j.compind.2018.09.005.

[14] Ashu Gupta, Kawaljeet Singh, and Rajesh Verma. "A Critical Study and Comparison of Manufacturing Simulation Softwares using Analytic Hierarchy Process". In: *Journal of Engineering Science and Technology* 5.1 (2010), pp. 108–129. ISSN: 1823-4690.

[15] Ashu Gupta, Rajesh Verma, and Kawaljeet Singh. "Smart Sim Selector: A Software for Simulation Software Selection". In: *International Journal of Engineering (IJE)* 3.3 (2009), pp. 175–185. ISSN: 1985-2312.

[16] Paul Harmon and Celia Wolf. *The State of Business Process Management – 2018*. Electronic Article. 2018. URL: https://www.redhat.com/en/resources/state-of-bpm-2018-bptrends-analyst-paper.

[17] Daniela Hincu and Marin Andreica. "The Evaluation and Selecting Process for Simulation Software using Fuzzy Sets". In: *Metalurgia International* 17.3 (2012), pp. 141–144. ISSN: 1582-2214.

[18] Vlatka Hlupic. "Simulation Modelling Software Approaches to Manufacturing Problems". Thesis. 1993, p. 333.

[19] Vlatka Hlupic, Zahir Irani, and Ray J. Paul. "Evaluation Framework for Simulation Software". In: *International Journal of Advanced Manufacturing Technology* 15.5 (1999), pp. 366–382. ISSN: 0268-3768. DOI: 10.1007/s001700050079.

[20] Vlatka Hlupic and Amardeep Mann. "SimSelect: A System for Simulation Software Selection". In: *Proceedings of the 1995 Winter Simulation Conference*. WSC '95. Arlington, VA, USA: IEEE Press, 1995, pp. 720–727. ISBN: 0-78033018-8. DOI: 10.1109/WSC.1995.478849.

[21] Ching-Lai Hwang and Kwangsun Yoon. "Methods for Multiple Attribute Decision Making". In: *Multiple Attribute Decision Making*. Vol. 186. Lecture Notes in Economics and Mathematical Systems. Heidelberg, Germany: Springer-Verlag Berlin Heidelberg, 1981. Chap. 3, pp. 58–191. ISBN: 978-3-540-10558-9. DOI: 10.1007/978-3-642-48318-9_3.

[22] Anil S. Jadhav and Rajendra M. Sonar. "Evaluating and Selecting Software Packages: A Review". In: *Information and Software Technology* 51.3 (2009), pp. 555–563. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2008.09.003.

[23] Mario Jadrić, Maja Ćukušić, and Antonia Bralić. "Comparison of Discrete Event Simulation Tools in an Academic Environment". In: *Croatian Operational Research Review* 5.2 (2014), pp. 203–219. DOI: 10.17535/crorr.2014.0008.

[24] Eugene Rex Jalao, Teresa Wu, and Dan L. Shunk. "An Intelligent Decomposition of Pairwise Comparison Matrices for Large-Scale Decisions". In: *European Journal of Operational Research* 238.1 (2014), pp. 270 –280. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.03.032.

[25] Monique H. Jansen-Vullers and Mariska Netjes. "Business Process Simulation – A Tool Survey". In: *Seventh Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*. Ed. by Kurt Jensen. Vol. 38. Eindhoven University of Technology, 2006, pp. 77–96.

[26] Steinar Kvale. *Doing Interviews*. Ed. by Uwe Flick. The SAGE Qualitative Research Kit. 1 Oliver's Yard, 55 City Road, London, UK: SAGE Publications Ltd., 2007, p. 157. ISBN: 978-0-7619-4977-0.

[27] Steinar Kvale and Svend Brinkmann. *InterViews: Learning the Craft of Qualitative Research Interviewing*. 2nd ed. Los Angeles, USA: Sage Publications Ltd., 2009. ISBN: 978-0-7619-2541-5.

[28] Kam Cheong Li et al., eds. *Engaging Learners Through Emerging Technologies. International Conference on ICT in Teaching and Learning*. Vol. 302. Communications in Computer and Information Science 1. Hong Kong, China: Springer-Verlag, Berlin, Heidelberg, Germany, July 2012, p. 228. ISBN: 978-3-642-31398-1. DOI: 10.1007/978-3-642-31398-1.

[29] Gerald T. Mackulak, Jeffery K. Cochran, and Paul A. Savory. "Ascertaining Important Features for Industrial Simulation Environments". In: *SIMULATION* 63.4 (1994), pp. 211–221. DOI: 10.1177/003754979406300402.

[30] Jalal Nikoukaran, Vlatka Hlupic, and Ray J. Paul. "Criteria for Simulation Software Evaluation". In: *Proceedings of the 1998 Winter Simulation Conference*. Vol. 1. WSC '98. Washington, DC, USA: IEEE Computer Society Press, 1998, pp. 399–406. ISBN: 0-7803-5133-9. DOI: 10.1109/WSC.1998.745014.

[31] Jalal Nikoukaran and Ray J. Paul. "Software Selection for Simulation in Manufacturing: A Review". In: *Simulation Practice and Theory* 7.1 (1999), pp. 1–14. ISSN: 0928-4869. DOI: 10.1016/s0928-4869(98)00022-6.

[32] Object Management Group (O.M.G.) *Business Process Model and Notation (BPMN)*. Electronic Article. Version 2.0.2. Needham, MA, USA, 2013. URL: https://www.omg.org/spec/BPMN/2.0.2/PDF.

[33] Claude D. Pegden. "Introduction to SIMAN". In: *Proceedings of the 15th Conference on Winter Simulation*. Vol. 1. WSC '83. Arlington, Virginia, USA: IEEE Press, 1983, pp. 231–241.

[34] Claude D. Pegden, Robert E. Shannon, and Randall P. Sadowski. *Introduction to Simulation Using SIMAN*. 2nd ed. McGraw-Hill International Editions: Industrial Engineering Series. New York, NY, USA: McGraw-Hill Inc., 1995. ISBN: 0070493200.

[35] Michael Pidd. "Guidelines for the Design of Data-Driven Generic Simulators for Specific Domains". In: *SIMULATION* 59.4 (1992), pp. 237–243. ISSN: 0037-5497 1741-3133. DOI: 10.1177/003754979205900403.

[36] Michael Pidd and Adelaide Carvalho. "Simulation Software: Not the Same Yesterday, Today or Forever". In: *Journal of Simulation* 1.1 (2006), pp. 7–20. DOI: 10.1057/palgrave.jos.4250004.

[37] Thomas L. Saaty. *The Analytic Hierarchy Process*. New York, NY, USA: McGrew Hill, 1980, p. 324.

[38] Thomas L. Saaty. "Decision Making with the Analytic Hierarchy Process". In: *International Journal of Services Sciences* 1.1 (2008), pp. 83–98. ISSN: 1753-1446. DOI: 10.1504/IJSSCI.2008.017590.

[39] Irving Seidman. *Interviewing as Qualitative Research: A Guide for Researchers in Education and the Social Sciences*. 3rd ed. 1234 Amsterdam Avenue, New York, NY, USA: Teachers College Press, 2006, p. 162. ISBN: 13 978-0-8077-4666-0.

[40] André Koide da Silva and Rui Carlos Botter. "Method for Assessing and Selecting Discrete-Event Simulation Software Applied to the Analysis of Logistic Systems". In: *Journal of Simulation* 3.2 (2009), pp. 95–106. DOI: 10.1057/jos.2008.21.

[41] Team SimPy. *SimPy: Discrete Event Simulation for Python*. Computer Program. Version 3. 2002–2018. URL: https://simpy.readthedocs.org.

[42] Tamrat W. Tewoldeberhan, Alexander Verbraeck, and Vlatka Hlupic. "Implementing a Discrete-Event Simulation Software Selection Methodology for Supporting Decision Making at Accenture". In: *Journal of the Operational Research Society* 61.10 (2010), pp. 1446–1458. DOI: 10.1057/jors.2009.119.

[43] Tamrat W. Tewoldeberhan et al. "An Evaluation and Selection Methodology for Discrete-Event Simulation Software". In: *Proceedings of the 2002 Winter Simulation Conference*. Ed. by E. Yücesan et al. Vol. 1. WSC '02. IEEE Press, 2002, pp. 67–75. ISBN: 0-7803-7614-5. DOI: 10.1109/WSC.2002.1172870.

[44] Iñaki Ucar, Bart Smeets, and Arturo Azcorra. "simmer: Discrete-Event Simulation for R". In: *Journal of Statistical Software* accepted for publication (2018). URL: https://arxiv.org/abs/1705.09746.

[45] Peter J. M. Van Laarhoven and Witold Pedrycz. "A Fuzzy Extension of Saaty's Priority Theory". In: *Fuzzy Sets and Systems* 11.1-3 (1983), pp. 229–241. ISSN: 0165-0114. DOI: 10.1016/S0165-0114(83)80082-7.

[46] Ling Xu and Jian-Bo Yang. *Introduction to Multi-Criteria Decision Making and the Evidential Reasoning Approach*. Manchester, UK: Manchester School of Management Manchester, 2001, p. 21. ISBN: 978-1-86115-111-7.

[47] Robert K. Yin. *Case Study Research: Design and Methods*. Ed. by Vicki Knight. 5th ed. Los Angeles, CA, USA: Sage Publications Inc., 2014, p. 378. ISBN: 978-1-4522-4256-9.

Registered Trademarks

AnyLogic®is a registered trademark of AnyLogic North America, LLC. Computer Program. Oakbrook Terrace Tower, 1 Tower Ln, Suite 2655 Oakbrook Terrace, IL 60181, USA, 2015 – 2019. URL: https://www.anylogic.com/.

Arena®is a registered trademark of Rockwell Automation, Inc. Computer Program. 1201 South Second St., Milwaukee, WI 53204, USA, 1993 – 2019. URL: https://www.arenasimulation.com/.

ARIS®is a registered trademark of Software AG. Computer Program. Uhlandstraße 12, Darmstadt, 64297, Germany, 1996 – 2019. URL: https://resources.softwareag.com/aris.

BPMN™and Business Process Model and Notation™are trademarks of the Object Management Group, Inc. Other. 109 Highland Ave, Needham, MA 02494 USA, 2019. URL: https://www.omg.org/.

CPN Tools™is a trademark of the AIS group. Computer Program. Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands, 2010 – 2019. URL: http://cpntools.org/.

FileNet®is a registered trademark of International Business Machines Corporation ("IBM"). Computer Program. 1 New Orchard Road, Armonk, NY 10540, USA, 1985 – 2019. URL: https://www.ibm.com.

Google Maps™is a registered trademark of Google LLC. Digital Service. 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA, 2019. URL: https://www.google.com/maps/.

Google Scholar™is a registered trademark of Google LLC. Digital Service. 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA, 2019. URL: https://scholar.google.com/.

Google™is a registered trademark of Google LLC. Brand. 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA, 1998 – 2019. URL: https://www.google.com/.

HERE Maps®is a registered trademark of HERE Global B.V. Digital Service. Kennedyplein 222-226, 5611 ZT Eindhoven, Netherlands, 2012–2019. URL: https://wego.here.com/.

JavaScript™is a registered trademark of Oracle America, Inc. Computer Programming Language. 500 Oracle Parkway, Redwood Shores, CA 94065, USA, 2000 – 2019. URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

MATLAB®is a registered trademark of The MathWorks, Inc. Computer Program. 1 Apple Hill Drive, Natick, MA 01760, USA, 1994 – 2019. URL: https://www.mathworks.com/.

Micro Saint®Sharp is a registered trademark of Alion Science and Technology Corporation. Computer Program. 1750 Tysons Boulevard, Suite 1300, McLean, VA 22102, USA, 2004 – 2019. URL: http://www.microsaintsharp.com/.

Microsoft®Excel®is a registered trademark of Microsoft Corporation. Computer Program. One Microsoft Way, Redmond, WA 98052, USA, 1985 – 2019. URL: https://products.office.com/en/excel.

Microsoft®is a registered trademark of Microsoft Corporation. Brand. One Microsoft Way, Redmond, WA 98052, USA, 1975 – 2019. URL: https://www.microsoft.com/.

Microsoft®Visual Studio®is a registered trademark of Microsoft Corporation. Computer Program. One Microsoft Way, Redmond, WA 98052, USA, 1998 – 2019. URL: https://visualstudio.microsoft.com/.

OMG®and Object Management Group®are registered trademarks of the Object Management Group, Inc. Brand. 109 Highland Ave, Needham, MA 02494 USA, 2010 – 2019. URL: https://www.omg.org/.

ProModel®is a registered trademark of ProModel Corporation. Computer Program. 7540 Windsor Drive, Suite 300, Allentown, PA 18195, USA, 1991 – 2019. URL: https://www.promodel.com/.

Python™is a trademark of the Python Software Foundation. Computer Programming Language. 9450 SW Gemini Dr., ECM 90772, Beaverton, OR 97008, USA, 2001 – 2019. URL: https://www.python.org/.

@Risk®is a registered trademark of Palisade Corporation. Computer Program. 130 East Seneca Street, Suite 505, Ithaca, NY, USA, 1987 – 2019. URL: https://www.palisade.com/risk/.

Simio®is a registered trademark of Simio LLC. Computer Program. 504 Beaver St Sewickley, PA 15143, USA, 2009 – 2019. URL: https://www.simio.com/.

Spotfire®is a registered trademark of TIBCO Software, Inc. Computer Program. 3303 Hillview Avenue, Palo Alto, CA 94303, USA, 1999 – 2019. URL: https://www.tibco.com/products/tibco-spotfire.

SPSS®is a registered trademark of International Business Machines Corporation ("IBM"). Computer Program. 1 New Orchard Road, Armonk, NY 10540, USA, 2004 – 2019. URL: https://www.ibm.com/analytics/spss-statistics-software.

STAR-CD™is a registered trademark of Siemens Product Lifecycle Management Software Inc. Computer Program. 5800 Granite Parkway, Suite 600, Plano, TX 75024, USA, 1998 – 2019. URL: https://mdx.plm.automation.siemens.com/star-cd.

Tableau®is a registered trademark of Tableau Software, Inc. Computer Program. 1621 N 34th St., Seattle, WA 98103, USA, 2003 – 2019. URL: https://www.tableau.com/.

Visual Basic®is a registered trademark of Microsoft Corporation. Computer Programming Language. One Microsoft Way, Redmond, WA 98052, USA, 1993 – 2019. URL: https://docs.microsoft.com/en-us/dotnet/visual-basic/.

All other trademarks are the property of their respective owners.