Available online at www.sciencedirect.com
SCIENCE DIRECT®
Information & Management 42 (2005) 1051–1066
INFORMATION & MANAGEMENT
www.elsevier.com/locate/dsw

A discrete-event simulation and continuous software evaluation on a systemic quality model: An oil industry case

Gladys Rincon [a,*], Marinelly Alvarez [b], Maria Perez [c], Sara Hernandez [a]

[a] Universidad Simón Bolívar, Valle de Sartenejas, Edf. MYS, Piso 3, MYS-318, Apartado Postal 89000, Caracas 1080-A, Venezuela
[b] INTEVEP PDVSA, Departamento de Refinación y Comercio. Apdo. 76343. Caracas 1070-A, Venezuela
[c] Universidad Simón Bolívar - LISI. Apdo. Postal 89000, Caracas 1080-A, Venezuela

Received 2 March 2003; received in revised form 19 December 2003; accepted 16 April 2004

Abstract

This paper proposes a specific set of criteria for evaluating discrete-event Simulation Software capable of simulating continuous operations. A quality specifications model was developed; it identified 40 sub-characteristics and 131 metrics to assess the quality of this type of software; it is then to be used in the selection process. The application was demonstrated in one organization that provides consulting services in the logistics' area of the Venezuelan oil industry and it was used to examine four commercial software systems that might fulfill the technical requirements established by the organization. The selection and evaluation technique successfully identified the software that best suited their needs.

Keywords: Discrete-event; Continuous operations; Simulation; assessment; Software-quality; Oil industry

1. Introduction

The application range of simulation techniques has increased in recent years and, consequently, a great deal of high-quality simulation software has emerged in the marketplace, with different characteristics and specific purposes [18]. Therefore, the following question arises: How can we tell which software is the one that best meets the goals of the organization? Sometimes corporations make decisions concerning technology they need and, unknowingly, use approaches that may underestimate or ignore important aspects in the selection and future use of the technology they are purchasing [20].

An improperly selected simulation software may result in wrong operational and/or strategic decisions, with subsequent economic damage to the organization. It is not easy to obtain a set of criteria that can be generally applied to evaluate all software because the benefits of their use are difficult to assess. Nikoukaran,

* Corresponding author. Tel.: +58 2129063305; fax: +58 2129063303.
E-mail addresses: grincon@usb.ve (G. Rincon), marinellyalvarez@cantv.net (M. Alvarez), movalles@usb.ve (M. Perez), hernandezsl@cantv.net (S. Hernandez).

0378-7206/$ – see front matter
doi:10.1016/j.im.2004.04.007

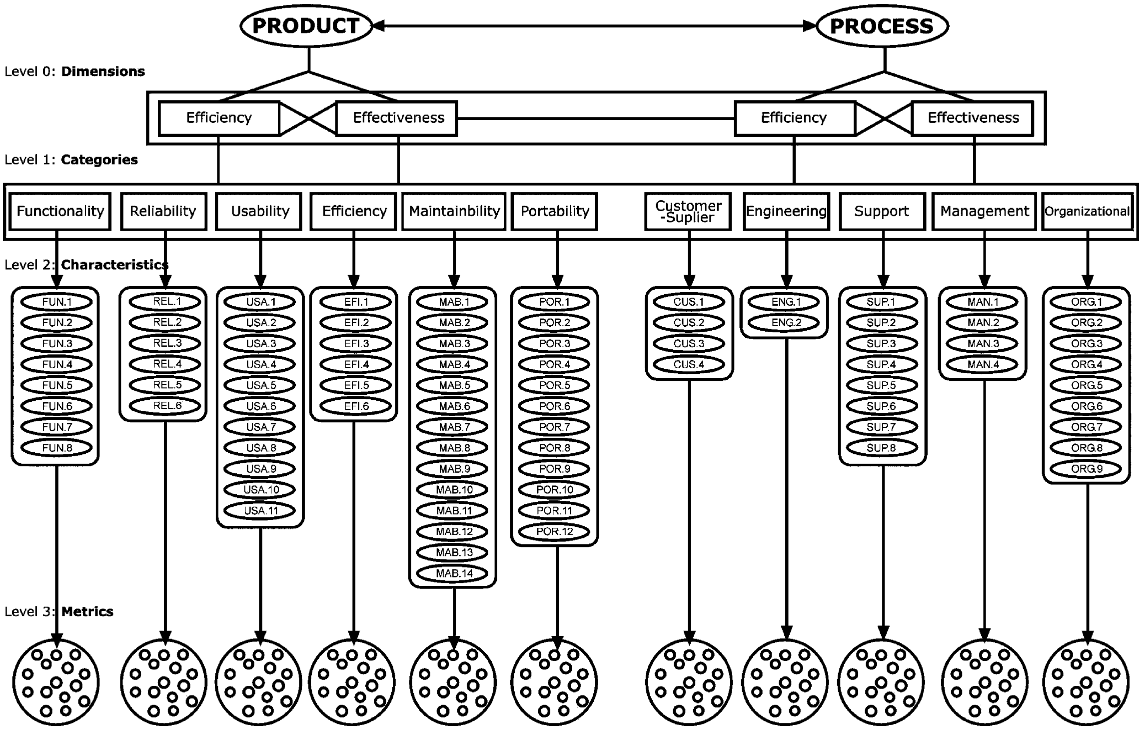

Fig. 1. Diagram of the systemic quality model (SQMO).

Hlupic and Paul stated that it is imperative to have a list of assessment criteria for the selection of suitable simulation software, which, once identified, must be structured in a decision-making model.

Our research was aimed at proposing a set of attributes to evaluate *D*iscrete-*E*vent *S*imulation *S*oftware capable of simulating Continuous operations (identified by its acronym, DESS-c) and to support the decision-making process associated with their selection.

2. The systemic quality model (SQMO)

A *Systemic Quality Model* (SQMO) [13,15] was developed in 2001 by the Universidad Simón Bolívar (Venezuela). Since then, the University adopted the SQMO for software evaluations. The work of Ortega et al. [19], Alvarez [1], Diaz [3], and Martín [14] provides examples of successful implementations of SQMO, which is aimed at estimating the systemic quality within an organization engaged in software development. It is based on a systemic quality model proposed by Callaos and Callaos [2]. As seen in Fig. 1, SQMO consists of two sub-models (a *Product* and a *Process* submodels).

The SQMO can use either the *Product* submodel, the *Process* submodel, or both. The first is intended to evaluate fully developed software, while the second is used to evaluate the development process. The different levels that form the SQMO are:

2.1. Level 0: dimensions

This considers both *Product* and *Process* software evaluation in its different stages (analysis, design, implantation, and operation). The four dimensions are: *Efficiency* and *Effectiveness* of the *Product*; and *Efficiency* and *Effectiveness* of the *Process*. For this level, the *Efficiency* involves computation of the relation between the quantity obtained and the quantity of resources used. However, the *Effectiveness* measures the ratio between the obtained and the obtainable quantities.

Table 1
SQMO – categories for the *Product* submodel

Category	Definition
Functionality (FUN)	Functionality is the capacity of the software product to provide functions that meet specific and implicit needs, when software is used under specific conditions
Reliability (FIA)	Reliability is the capacity of a software product to maintain a specified level of performance when used under specific conditions
Usability (USA)	Usability is the capacity of the software product to be attractive, understood, learned and used by the user under certain specific conditions
Efficiency (EFI)	Efficiency is the capacity of the software product to provide appropriate performance, relative to the amount of resources used, under stated conditions
Maintainability (MAB)	Maintainability is the capacity of the software to be modified. Modifications can include corrections, improvements or adaptations of the software to adjust to changes in the environment, in terms of functional requirements and specifications
Portability (POR)	Portability is the capacity of the software product to be transferred from one environment to another

2.2. Level 1: categories

This has six elements corresponding to the *Product* and five to the *Process* parts of the software system development cycle. Tables 1 and 2 summarize the definitions of the categories associated with the *Product* and *Process* submodels, respectively.

The efficiency category differs from the *Product Efficiency* dimension (in the *Dimensions* level) because the quality standard is that of the intermediate products, considering only their specific traits. At the category level the quality is measured by product-use.

2.3. Level 2: characteristics

SQMO states that each category has a set of associated characteristics, which define the key areas that must be fulfilled in order to guarantee and control the *Product* and/or *Process* quality. A list of the characteristics recommended by SQMO and their respective definitions was presented by Mendoza et al.

2.4. Level 3: metrics

Each characteristic has a group of metrics for the attributes to be evaluated in the software and/or process. These metrics must be defined for the case study.

3. Adoption of the systemic quality model (SQMO)

SQMO was adopted as reference because it takes the best concepts from Dromey [4], reinforced by the work of Voas [24]; McCall [16]; ISO/IEC 9126 [8] and ISO/IEC 15504-2 [9]. In addition, the fact that the SQMO was developed by our University made it possible to count on direct assistance of its authors and to have access to all model documentation. In order to define which aspects

Table 2
SQMO – categories for the *Process* submodel

Category	Definition
Client–supplier (CUS)	Is made up of processes that have an impact on the client, support the development and transition of the software to the client, and give the correct operation and use of the software product or service
Engineering (ENG)	Consists of processes that directly specify, implement or maintain the software product, its relation to the system and documentation on it
Support (SUP)	Consists of processes that can be used by any of the processes (including support ones) at several levels of the acquisition life cycle
Management (MAN)	Consists of processes that contain practices of a generic nature that can be used by anyone managing any kind of project or process, within a primary life cycle
Organizational (ORG)	Contains processes that establish the organization's commercial goals and develop process, product and resource goods (values) that will help the organization attain the goals set in the projects

are useful in the evaluation of DESS-c, the levels of this model were analyzed. The considerations taken into account for the definition of the SQMO levels are:

3.1. Submodel selection

Here, only the *Product* submodel of SQMO was considered. The *Process* submodel was excluded, because our intent was to evaluate the fully developed DESS-c that are marketed, and not to consider its development process.

3.2. Level 0: dimensions selection

Only *Effectiveness* (not *Efficiency*) was considered for Level 0; the assessment of efficiency would have required the source code of DESS-c, which was not available (it is considered proprietary information). Additionally, special attention was centered on the evaluation of the software quality features observed in its execution.

3.3. Level 1: categories selection

The starting point of the application of SQMO for Level 1 is the *functionality* category. They also stated that, out of the remaining five SQMO categories, the two essential categories for the stakeholders, *Usability* and *efficiency*, had to be chosen. The *usability* category was selected because its characteristics were considered crucial by large organizations, who intended to reduce the time and costs associated with personnel training and model development and maintenance [23]. The *efficiency* category was selected because it was fundamental to investigate the performance of the DESS-c under stated conditions. Particularly, they require a significant memory resource.

3.4. Level 2: characteristics selection

All of the characteristics suggested by SQMO were used, except for those lacking relevance within our context:

3.4.1. Functionality

The characteristics selected were: *Fit to Purpose* (FPU), *Interoperability* (INT) and *Security* (SEC), former *FUN.1*, *FUN.3* and *FUN.4*, respectively. The *characteristics* from *FUN.5* to *FUN.8* were discarded, because they are associated with the *Efficiency* dimension, which was excluded in Level 0. *Precision* (*FUN.2*) was also excluded because it is very difficult to measure the accuracy for a model using real data [21]. Also, the random nature of the variables characterizing stochastic simulation models produced random outputs and, therefore, only represented an *estimate* of the true characteristics of the system [11].

3.4.2. Usability

One of the key characteristics when a DESS-c was adopted is the ease with which the user obtains the knowledge required to employ it and understand its features, operations, and concepts. However, since its learning and understanding are closely related, *Ease of Understanding* (former *USA.1*) and *Learning Ability* (former *USA.2*) characteristics of SQMO were merged into a single factor, known as *Ease of Understanding and Learning* (*EUL*).

The characteristics chosen for this category are therefore *Ease of Understanding and Learning* (*EUL*), *Graphical Interface* (*GIN*), former *USA.3*, and *Operability* (*OPR*), former *USA.4*. Characteristics from *USA.5* to *USA.10* were discarded.

3.4.3. Efficiency

This was evaluated using *Execution Performance* (*EPE*), former *EFI.1*, and *Resource Utilization* (*RUT*), former *EFI.2*. The *characteristics* from *EFI.3* to *EFI.6* were excluded since they are related to the *Efficiency* dimension.

3.5. Level 3: metrics definition

When the product quality was measured, each characteristic had associated metrics, which were related to qualities or attributes of the software to be evaluated. Complexity of the software resulted in modifications in SQMO to gather the attributes in a coherent, manageable, and understandable fashion. To this end, the assessment of the selected characteristics was restated, and this resulted in an improved model: the SQMO+.

4. Systemic quality model – SQMO+

Authors such as Kitchenham [10] have pointed out that when the attributes identified for the assessment are complex, they can often be divided into simpler items, which, if desired, can be further subdivided. However, the excess of items can result in a need to invest substantial time in their assessment. Therefore, a balance between assessment depth, the desired confidence level, and practical difficulties must be established.

The study of the aspects that should be included in a DESS-c led to the identification of a different set of attributes. SQMO+ requires an additional level, so that the selected characteristics can be evaluated through a number of sub-characteristics that, in turn, are subdivided into metrics. The sub-characteristics and metrics proposed were based on the contributions of Nikoukaran et al., Hlupic [7] and information gathered from works executed from the industry sector, advertising literature, and software documentation literature. Fig. 2 shows the SQMO+ structure.

5. Sub-characteristics and metrics for the DESS-c evaluation

The set of sub-characteristics and metrics for DESS-c evaluation, correspond to Levels 3 and 4 in Fig. 2.

5.1. Sub-characteristics and metrics for assessing functionality

The commercial simulation software must be able to recognize both continuous and discrete systems in order to be appropriate for a wide range of applications. Thus, it is essential to add features to deal with continuous processes. It was necessary to assess the DESS-c ability to simulate *hybrid* systems. For this reason, this category contains criteria that operate with continuous systems such as: *Fluids*, *Pipelines,* and *Storage Tanks* (*FPU 12*, *FPU 13,* and *FPU 14*, respectively). The metrics related to sub-characteristics are shown in Table 3:

5.1.1. Characteristic: fit to purpose (FPU)

This assesses whether the software is capable of providing a proper set of features according to the user-specific tasks and goals. Sub-characteristics *FPU01* to *FPU05* examine the facilities present in the software for entering data into a model, cloning simulation elements, and building models in modules. They also assess the facilities provided by the software that allow the user to describe the system

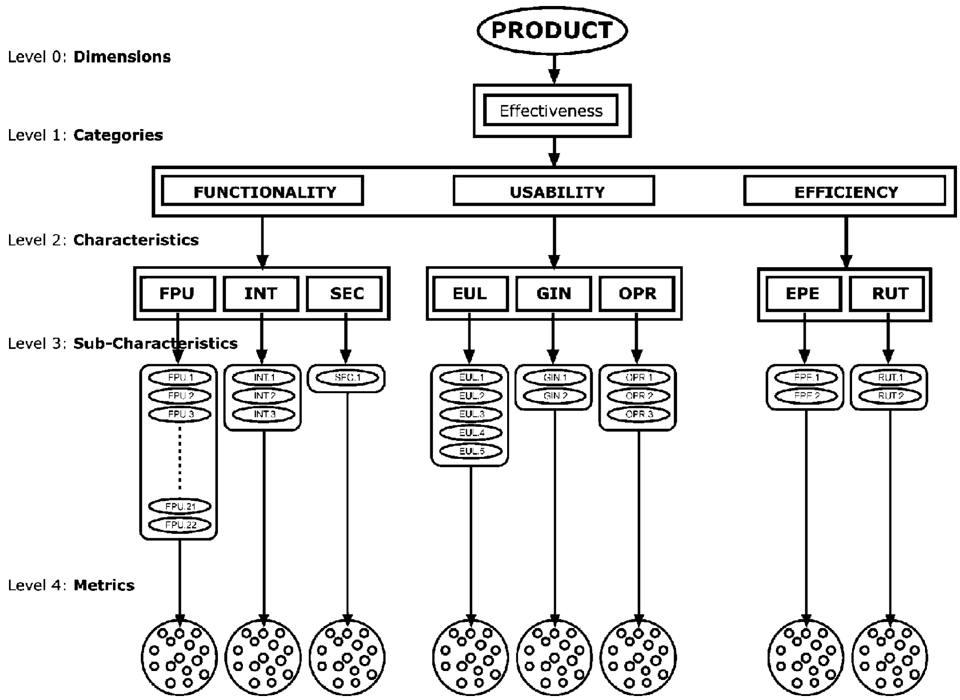

Fig. 2. Diagram of the systemic quality model (SQMO+).

Table 3
Characteristics, sub-characteristics and metrics assessed by SQMO+ – *functionality* category

Characteristics	Tags	Sub-characteristics	Metrics tags	Metrics
Fit to purpose (FPU)	FPU 01	Input data	FPU 011	Entering input data manually
			FPU 012	Reading data from an external file
	FPU 02	Cloning	FPU 021	Creating identical elements from an original one
	FPU 03	Modularity	FPU 031	Grouping and storing simulation elements
			FPU 032	Saving modules for future use
			FPU 033	Creation of modules (hierarchical model building option)
	FPU 04	Logical facilities	FPU 041	Standard functions library
			FPU 042	Creating user-defined functions
			FPU 043	Setting attributes to simulation elements
			FPU 044	Defining variables
			FPU 045	Assigning priorities to simulation elements
			FPU 046	Using arithmetic operators
			FPU 047	Using logical operators
			FPU 048	Using conditional operators
	FPU 05	Programming languages	FPU 051	Using programming languages
	FPU 06	Random numbers	FPU 061	Random number control
			FPU 062	Selecting among regular and antithetic random numbers
	FPU 07	Probability distributions	FPU 071	Standard probability distributions collection
			FPU 072	Definition of empirical probability distributions
			FPU 073	Probability distribution fitting
	FPU 08	Simulation clock	FPU 081	Analog simulation clock
			FPU 082	Digital simulation clock
			FPU 083	Time unit specification
	FPU 09	Entities	FPU 091	Specifying entities from an unlimited supply
			FPU 092	Specifying the maximum number of entities
			FPU 093	Specifying arrivals at specific time intervals
			FPU 094	Entities arrival in lots (specified number of entities)
			FPU 095	Specifying the simulation time at first entity arrives
	FPU 10	Queue	FPU 101	Specifying the maximum entities that enter in a queue
			FPU 102	Specifying the types of entities that can enter in a queue
			FPU 103	Queuing policies
			FPU 104	Holding entities in a queue for a minimum amount of time
			FPU 105	Removing entities from a queue after a period of time
	FPU 11	Operations	FPU 111	Specifying the time it takes to perform a task
			FPU 112	Scheduling maintenance and turnarounds
			FPU 113	Scheduling cleaning operations
			FPU 114	Assigning shifts to elements
	FPU 12	Fluids	FPU 121	Specifying fluid flow from an unlimited supply
			FPU 122	Specifying the maximum fluid flow supply
			FPU 123	Fluid blending
			FPU 124	Component concentration for a blend in a storage tank
	FPU 13	Pipelines	FPU 131	Specifying the maximum fluid volume
			FPU 132	Specifying the maximum fluid flow rate
			FPU 133	Cleaning/purging pipelines when fluid quality changes
			FPU 134	Reverse flow in pipelines
	FPU 14	Storage tanks	FPU 141	Specifying the storage tank capacity
			FPU 142	Specifying safety levels
			FPU 143	Specifying initial fluid type and fluid volume
	FPU 15	Experimentation	FPU 151	Specifying initial model conditions
			FPU 152	Specifying model warm-up period
			FPU 153	Replications of simulation runs
			FPU 154	Sensitivity analysis
			FPU 155	Automatic optimization of model parameters

Table 3 (*Continued*)

Characteristics	Tags	Sub-characteristics	Metrics tags	Metrics
	FPU 16	Output statistical analysis	FPU 161	Automatic calculation of statistics for selected elements
			FPU 162	Automatic calculation of statistics for replications outputs
			FPU 163	Confidence intervals estimation
			FPU 164	Goodness-of-fit test
	FPU 17	Cost analysis	FPU 171	Automatic calculation of the operating costs
	FPU 18	Saving the model	FPU 181	Saving the model structure to disk
			FPU 182	Saving experiments
			FPU 183	Saving model and status
			FPU 184	Automatic saving of an open model every few minutes
			FPU 185	Automatic creation of a backup file
	FPU 19	Report generation	FPU 191	Automatic standard report generation
			FPU 192	Generating reports for selected elements
			FPU 193	Gather in a single report the results obtained in replications of simulation runs
			FPU 194	Sending a model via e-mail
			FPU 195	Saving reports in HTML format
	FPU 20	Graphics	FPU 201	Graphical display of simulation results
	FPU 21	Images and icons	FPU 211	Library of standard icons
			FPU 212	Creating new or modifying existing icons
			FPU 213	Saving customized icons
			FPU 214	Importing images from other programs
			FPU 215	Saving images imported from other programs
			FPU 216	Importing AutoCAD drawings into the DESS-c
	FPU 22	Animation	FPU 221	Displaying entities as dynamic icons
			FPU 222	Color changes to indicate state changes of elements
			FPU 223	3D animation
			FPU 224	Automatically updating graphics display during the simulation
			FPU 225	Synchronizing the model to real time
			FPU 226	Displaying storage tanks level
			FPU 227	Turning animation on and off
Interoperability (INT)	INT 01	Operating system	INT 011	Operating systems support
	INT 02	Data exchange	INT 021	Links to Microsoft® Excel
			INT 022	Links to text files
	INT 03	Use of models by third parties	INT 031	Creating executable models
			INT 032	Runtime and player versions
Security (SEC)	SEC 01	Security devices	SEC 011	Password protection

operations in the model (such as setting attributes and assigning priorities to simulation elements). Sub-characteristic *Modularity* (*FPU03*), for example, evaluates whether it is possible to develop models of complex systems from simpler subsystems (modules). Being so, its metrics evaluate if the DESS-c allows the user to group and store elements in modules and if these can be reused. This sub-characteristic also investigates the possibility of using a hierarchical modeling structure (creation of modules within other modules).

Sub-characteristics *FPU06* and *FPU07* explore the facilities offered by the software for controlling the generation of random numbers and using probability distributions. This sub-characteristic, for instance, assesses the software's ability to use probability distributions to indicate the relative frequency of the events within the system [17]. Some of its metrics investigate which standard probability distributions are provided, whether it allows the user to define empirical distributions, and whether a distribution-fitting package is included.

Sub-characteristics *FPU08* to *FPU10* consider aspects related to the operation with typical elements of discrete-event systems, such as entities, queues, and a simulation clock. Sub-characteristic *Queues* (*FPU10*), for example, consists of metrics that investigate whether it is possible for the user to specify: (a) the number and type of entities that can be incorporated into a queue; (b) the type of queue; and (c) whether entities are held or removed from a queue after a period of time has elapsed.

Sub-characteristics *FPU11* to *FPU14* evaluate the DESS-c capability of working directly with continuous operations. For the sub-characteristics *Fluids* (*FPU12*), the metrics explore whether the DESS-c allows the user to specify an infinite fluid supply or to constrain its supply. Other metrics examine if the DESS-c can deal with fluid blends and if it is able of calculating the components concentration for a given fluid blend contained in a storage tank.

Sub-characteristics *FPU15* to *FPU17* explore the facilities provided to execute certain phases involved in simulation projects, such as experimentation, output statistical analysis, and cost analysis. Particularly, the sub-characteristic *Output statistical analysis* (*FPU16*), assesses the software ability to perform statistical data analysis. Its metrics examine whether the DESS-c automatically calculates: (a) statistics for selected elements; (b) statistics for multiple run outputs; (c) confidence interval; and (d) goodness of fit.

Sub-characteristics *FPU18* to *FPU20* are related to the saving process of the simulation model and to report generation. *Report generation* (*FPU 19*), for instance, involves metrics that evaluate the DESS-c ability to automatically generate: (a) standard reports; (b) reports for selected elements; and (c) a single report gathering the results obtained in each multiple run. Other metrics investigate whether the DESS-c enables the user to send models as attachments via e-mail and to issue reports with HTML format.

Sub-characteristics *FPU21* and *FPU22* evaluate the software features associated with visualizing and animating the model. *Animation* (*FPU 22*) includes metrics that examine the DESS-c ability to: (a) display entities in motion; (b) change color in order to indicate state changes of elements; (c) display 3D animation; (d) automatically update graphics display during the simulation; (e) synchronize the model to real time; (f) display storage tanks level as they rise and fall; and (g) activate or suspend the animation manually.

5.1.2. Characteristic: interoperability (INT)

This evaluates the ability of the DESS-c to interact with one or more systems.

Operating System (*INT01*) has a single metric that shows whether the DESS-c supports the operating system required by the organization.

Data exchange (*INT02*) is related to the exchange between the DESS-c and other applications. Its metrics evaluate the ability of receiving and sending data through links to Microsoft Excel® and text files.

Use of models by third parties (*INT03*) evaluates the facilities given by the licensor for third party users to execute or change parameters in pre-assembled models without purchasing the full version of the package. Its metrics assess whether it is possible to create executable models and if the licensor offers RunTime and Player versions of the software.

5.1.3. Characteristic: security (SEC)

This evaluates if the software is capable of protecting information so that unauthorized persons cannot access it.

Sub-characteristic *Security devices* (*SEC01*) involves whether the models are protected by means of a password.

5.2. Sub-characteristics and metrics for assessing the usability category

The metrics related to each of the sub-characteristics are shown in Table 4:

5.2.1. Ease of understanding and learning (EUL)

This characteristic assesses the software's ability to make it easier for the user to understand and to use the software. It also evaluates the facilities that enable the user to learn the application.

Learning time (*EUL01*) takes into account the average time that a new user requires to get acquainted with the use of the software so as to be able to develop a simple model. *Browsing facilities* (*EUL02*) is related to the search commands and functions in the DESS-c. Its metrics evaluate: (a) how fast commands can be located in the menu; (b) the availability of toolbars with buttons to activate functions; (c) the possibility of

Table 4
Characteristics, sub-characteristics and metrics assessed by SQMO+ – *usability* category

Characteristics	Tags	Sub-characteristics	Metrics tags	Metrics
Ease of understanding and learning (EUL)	EUL 01	Learning time	EUL 011	Average learning time
	EUL 02	Browsing facilities	EUL 021	Speed at which commands can be located in the menu
			EUL 022	Toolbars
			EUL 023	Consistency between icons in the toobars and their actions
			EUL 024	Displaying right-click menus
	EUL 03	Terminology	EUL 031	Ease of understanding the terminology
	EUL 04	Help and documentation	EUL 041	User manual
			EUL 042	On-line help
			EUL 043	Finding topics in the documentation
			EUL 044	Example models
			EUL 045	Troubleshooting guide
			EUL 046	Introduction to simulation concepts
			EUL 047	Introduction to statistical concepts in simulation
	EUL 05	Support and training	EUL 051	Availability of introductory training courses
			EUL 052	Availability of advanced training courses
			EUL 053	Availability of tailor-made training courses
			EUL 054	Phone technical support
			EUL 055	On-line support
			EUL 056	On site training at the organization facilities
			EUL 057	Availability of consulting services
			EUL 058	Response time of the vendor
Graphical interface (GIN)	GIN 01	Windows and mouse interface	GIN 011	Selecting elements with a single click
			GIN 012	Editing model elements by double-clicking
			GIN 013	Removing selected elements by pressing the Delete or Backspace keys
			GIN 014	Cutting, copying and pasting with the clipboard
			GIN 015	Dragging and droping elements to the modeling window
	GIN 02	Display	GIN 021	Color display on screen
			GIN 022	Resizing simulation windows
			GIN 023	Creating and editing the screen layout
			GIN 024	Zoom-in and zoom-out
Operability (OPR)	OPR 01	Versatility	OPR 011	Resetting the simulation clock to the start of the run
			OPR 012	Specifying whether to run the model until a particular time is reached, or until a specified event takes place
			OPR 013	Running the model backwards
	OPR 02	Interaction	OPR 021	Prompting the user to enter values for variables
			OPR 022	Stopping the simulation run at the current simulated time
			OPR 023	Automatically displaying alert messages
			OPR 024	Running the model event by event
	OPR 03	Multitasking	OPR 031	Working with another application or program while a simulation is running in the background
			OPR 032	Editing a model while another model is running
	OPR 04	Animation speed control	OPR 041	Animation speed control

displaying right-click menus; and (d) the consistency between icons and their actions.

Since the lack of a common terminology in the field of discrete-event simulation influences the learning time, the sub-characteristic *Terminology* (*EUL03*) is used to evaluate whether the terms used by the software are easily understandable. Its only metric assesses the ease of understanding the terminology.

Help and documentation (*EUL04*) examines the facilities to assist the user in learning and using the software. Its metrics assess the availability of help material, such as: user manual, on-line help, troubleshooting guide, and introductory information about simulation and statistical concepts. Likewise, other metrics investigate whether topics are easily found in the documentation and if example models are aimed at the specific application domain.

The metrics of sub-characteristic *Support and training* (*EUL05*) explore the availability of: (a) introductory training courses; (b) advanced training courses; and (c) training courses tailored to meet the requirements of the organization. Other metrics look at alternatives available for technical support and the response time of the vendor.

5.2.2. Graphical interface (GIN)

This characteristic is associated to those software attributes that render it more attractive for the user, such as the use of color and the nature of the graphic design.

Sub-characteristic *Windows and mouse interface* (*GIN01*) assesses whether the software is operated similarly to that of a Windows environment. Its metrics are related to the selection of elements by clicking the mouse and the use of the clipboard to cut, copy, and paste. Other metrics examine whether it is possible to remove elements from a model by pressing the Delete or Backspace keys, or to use *drag and drop* functionality to incorporate elements to the modeling window.

Display (GIN02) is related to the software's presentation on screen. Its metrics assess color display and examine the possibility of resizing windows, creating and editing the screen layout, and zooming in and zooming out.

5.2.3. Operability (OPR)

This characteristic evaluates if the software is capable of enabling the user to operate it and control it. *Versatility* (*OPR01*) investigates the facilities provided to control the process of running a model. Its metrics evaluate the possibility of resetting the simulation clock to the start of the run, running the model backwards, and indicating whether to run the model until a particular time is reached, or until a specified event takes place.

Interaction (*OPR02*) is related to the communication between software and user. Its metrics investigate if the DESS-c prompts the user to enter values for variables and if it automatically displays alert messages. Other metrics interrogate if it allows the user to stop the simulation run at the current simulated time or run the model event by event.

Multitasking (*OPR03*) is related to the software's ability to perform simultaneous operations. Its metrics assess the possibility of working with another application or program while a simulation is running in the background, and of editing a model while another model is running.

Animation speed control (*OPR04*) is related to the control of the animation speed in the simulation. Its metric is binary (*yes* or *no*).

5.3. Sub-characteristics and metrics for assessing the efficiency category

The metrics for the sub-characteristics for the *efficiency* category are shown in Table 5:

5.3.1. Execution performance (EPE)

This characteristic is used to assess if the software is capable of providing proper responses and processing times under specific conditions.

Table 5
Characteristics, sub-characteristics and metrics assessed by SQMO+ – *efficiency* category

Characteristics	Tags	Sub-characteristics	Metrics tags	Metrics
Execution performance (EPE)	EPE 01	Compilation speed	EPE 021	Compilation speed
Resource utilization (RUT)	RUT 01	Hardware requirements	RUT 011	CPU (processor type)
			RUT 012	Minimum RAM
			RUT 013	Hard disk space required
	RUT 02	Software requirements	RUT 021	Additional software requirements

Sub-characteristic *Compilation speed (EPE02)* assesses how fast the software creates an executable version of the model. Its metric is compilation speed.

5.3.2. Resources utilization (RUT)

This characteristic is aimed at evaluating whether the software uses the resources properly when it is performing its functions under specific conditions.

Sub-characteristic *Hardware requirements (RUT01)* refers to the viability of setting up and running the software in the organization. Its metrics take into account hard disk space required, CPU needed, and minimum RAM.

Sub-characteristic *Software requirements (RUT02)* assesses whether it is necessary to install additional software in the PCs as well as the DESS-c, e.g. a compiler.

6. DESS-c evaluation and selection

The method involves the creation of two multi-disciplinary work-teams: *Analysis and Selection Team* and *Experts Team*. The first is the group responsible for executing the evaluation and selection of the software. The second is the set of consultants, experts, and users working in the application area and simulation field.

6.1. General objectives definition

The *Analysis and Selection Team* defines the general objectives of the evaluation and selection project based on the overall organizational requirements. The objectives are: (1) areas of the application and use of the software, (2) particular aims of the organizational unit that uses them, and (3) identification of the required attributes (for it to be as functional, usable, and efficient as possible).

6.2. Mandatory and Non-mandatory definition (importance level and evaluation scale)

The *Experts Team* is required to answer a ‘‘***questionnaire***’’ that defines whether each metric should be Mandatory or Non-Mandatory. In this context, the metric which is Non-Mandatory is desirable.

In the ***questionnaire***, the Mandatory metrics are established by a binary scale, which contains the option YES (when the attribute is Mandatory). When a DESS-c produces a negative return for any Mandatory metric, it must be removed from the list of candidates. The ‘‘importance-level’’ of the Non-Mandatory metrics are established by means of a Likert type scale. The *Analysis and Selection Team* sets an ‘‘evaluation scale’’ that will make it possible to measure capacity of the DESS-c in satisfying each Non-Mandatory metric.

6.3. Prescreening

Prescreening allows the reduction of the number of considered products, to those that remain evaluated using the Non-Mandatory metrics. The activities involved in this stage were inspired by the methodology for selecting software proposed by Le Blanc [12]. They are: (a) elaborating a long list (LL) of DESS-c available in the market; (b) reducing this to a medium list (ML) containing the DESS-c that comply with the stated general objectives 1 and 2; (c) producing the short list (SL) of DESS-c that provide all the Mandatory metrics.

6.4. Evaluation and data analysis

The *Analysis and Selection Team* then: (a) evaluates the DESS-c in the SL using the metrics defined as Non-Mandatory; (b) quantifies the results obtained in the evaluation activity; and (c) scrutinizes the final results of the evaluation.

The method of quantifying the results involves:

- Assigning a value to each Non-Mandatory metric, according to the ‘‘evaluation scale’’.
- Multiplying each of these by the ‘‘importance-level’’ of the related metrics.
- Adding these to calculate the value for each evaluated category.
- Computing a percentage parameter denominated *Quality Rate*, by which it can be seen how each DESS-c behaves against the *ideal situation*.

The *Quality Rate* is the result of dividing the total value obtained by the DESS-c in a category by the maximum total value (ideal situation) that the DESS-c can reach in that category.

6.5. Selection of the DESS-c

Next, the *Analysis and Evaluation Teams* determines the hierarchy of the DESS-c contained in the SL. By determining the rank, it is possible to select the software that better suits the requirements. The Weighed Global Quality Rate Strategy (identified by its acronym in capital letters, WGQR) analyses the behavior of the software within the hierarchy, because it quantifies the influence of the ‘‘weight’’ assigned to each category. The WGQR is estimated as a function of the *Quality Rate* and of the ‘‘weights’’ of the categories. The WGQR is defined as:

$$\mathrm{WGQR} = \sum_i (\mathrm{QR}|_i \times \mathrm{Weight}|_i)$$

where, $\mathrm{QR}|_i$ is the quality rate in category i (i: functionality, usability, efficiency); $\mathrm{Weight}|_i$ is the ‘‘weight’’ of the category (i: functionality, usability, efficiency). The ‘‘weights’’ of the all categories sum to 100%.

7. The Venezuelan oil industry: a case study

A DESS-c assessment was carried out to demonstrate the applicability of the Systemic Quality Model SQMO+. The entire process was intended to support the specific needs of an organization that renders consulting services in the decision making for the Venezuelan oil industry. This is a competitive business that requires a continued and sustained effort aimed at optimizing its operations and reducing costs in the productive process chain to maintain profit margins and market competitiveness. Consequently, the decision making process related to the industry’s logistics has to be analyzed from the perspective of its impact on the business as a whole.

The term ‘‘*Logistics*’’ here refers to freight or transportation, working capital, and distribution assets, including pipelines, tank farms, fleets, terminals and warehouses. Logistics, excluding raw material costs, is the single largest cost element in the oil and chemical industry, ranging within 15–25% of the product cost. In addition, logistics assets, whether wholly owned or not, can be over 30% of the process companies total asset base. The logistic assets are used at 50–60% of capacity [5].

On the other hand, it is estimated that roughly 40–50% of the total capital investment in the oil and chemical industry projects is spent on offsites and logistics facilities. Consequently, it is important to differentiate which facilities have direct impact on offsites and logistics, so that the correct balance is achieved between operational and capital costs [6].

The supply and distribution chain of hydrocarbons in the oil industry are characterized by processes in which a series of factors interact. As a result, they produce probabilistic phenomena with discrete-event occurrence. The arrivals and departures of tankers at marine ports, for loading and unloading products, are examples of discrete-events. Likewise, these systems involve continuous operations such as filling and emptying tanks as well as crude oil and product transfers through pipelines. This situation requires that commercial simulation software be capable of recognizing both continuous and discrete systems. The organization involved in this study case, was acquiring commercial software for this specific domain in order to implement applications sufficiently quickly to respond to pressure from competitors and to keep development time and cost associated with new models and maintenance low.

7.1. General objectives

Priority was given to the DESS-c intended for: selection of packages and not languages and faculty to model hybrid systems.

7.2. Mandatory and non-mandatory definitions (importance level and evaluation scale)

Once each member of the *Experts Team* had answered the ***questionnaire***, the answers were processed and analyzed by the *Analysis and Selection Team*. The result was that out of a total, 77 metrics were considered Non-Mandatory (59%) while 54 were Mandatory. The answers also established the ‘‘importance level’’ of the Non-Mandatory metrics. The Mandatory metrics were distributed as follows: functionality: 37 metrics (43%); usability: 13 metrics (33%), and efficiency: 4 metrics (66%). They are highlighted in gray in Tables 3–5.

Additionally, the *Analysis and Selection Team* determined the evaluation scales for each Non-Mandatory metric.

7.3. Prescreening

The survey devised by Swain [22] was used, because it collects the different alternatives of DESS-c that the market offers. Based on this information, 44 possible DESS-c were found. It was assumed that the DESS-c contained in the survey OR/MS Today, constituted the universe of the software (LL).

As a result of the prescreening step, the ML consisted of five DESS-c, and this was *reduced* to a Short List (SL) of four: Extend 5.0 (Imagine That Inc.), Witness 2000 (Lanner Group), AutoMod 9.1 (Brooks Automation) and ProModel 2002 (ProModel Corporation). To preserve confidentiality of the vendors these will be designated as *A*, *B*, *C* and *D*.

7.4. Evaluation and data analysis

The DESS-c in the SL were assessed by means of the non-mandatory items using manuals and demos supplied by the vendors. Likewise, a model of a marine terminal of a refinery was developed. This was simple but important, because it clearly illustrated basic concepts related to discrete-event simulation.

The strategy used to evaluate the metrics was based on: (a) the algorithm of Mendoza et al. to measure the quality of the software according to SQMO; (b) the risks related to the Feature Analysis-Screening Mode.

Fig. 3 shows the *Quality Rates* obtained by each of the DESS-c. For the functionality category, *B*, *C* and *D* DESS-c present high values (close to the 84–88% range), which reveals that these systems have most of the functions demanded from a DESS-c needed in the area of application and industry. On the other hand, DESS-c *A* exhibits a lower value (69%).

For the usability category, *B* predominates in the group (87%). This DESS-c has favorable attributes related to: (a) ease of use and understanding of its functions, operations, and concepts; (b) more efficient on-line help; and (c) a greater number of model examples related to the logistics of the Venezuelan oil industry.

For the efficiency category, the four DESS-c have a high *Quality Rate*, and thus any could be successfully installed in the technological platform of the Venezuelan oil industry to provide adequate performance.

The differences among the DESS-c in the functionality category show negligible parameters for any worthwhile ranking to be made. Only in the usability category does one DESS-c have preeminence over the others.

7.5. Selection of the DESS-c

The Analysis and Evaluation Team used a WGQR Strategy. Because the four DESS-c complied 100% with the metrics of the efficiency category (Fig. 3), the computing of the WGQR was based only on the "weights" of the functionality and usability categories. Fig. 4 illustrates the results for the four DESS-c, in the two categories: the WGQR runs along the y axis, and the *x*-axis shows the "weights" of the

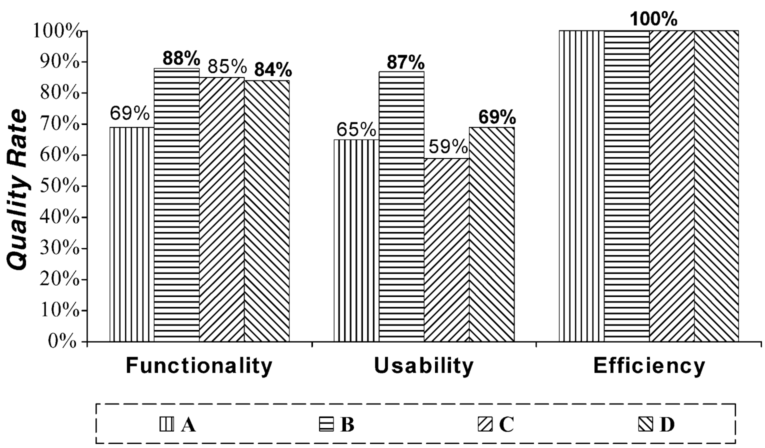

Fig. 3. Quality rate in the three categories assessed by SQMO+.

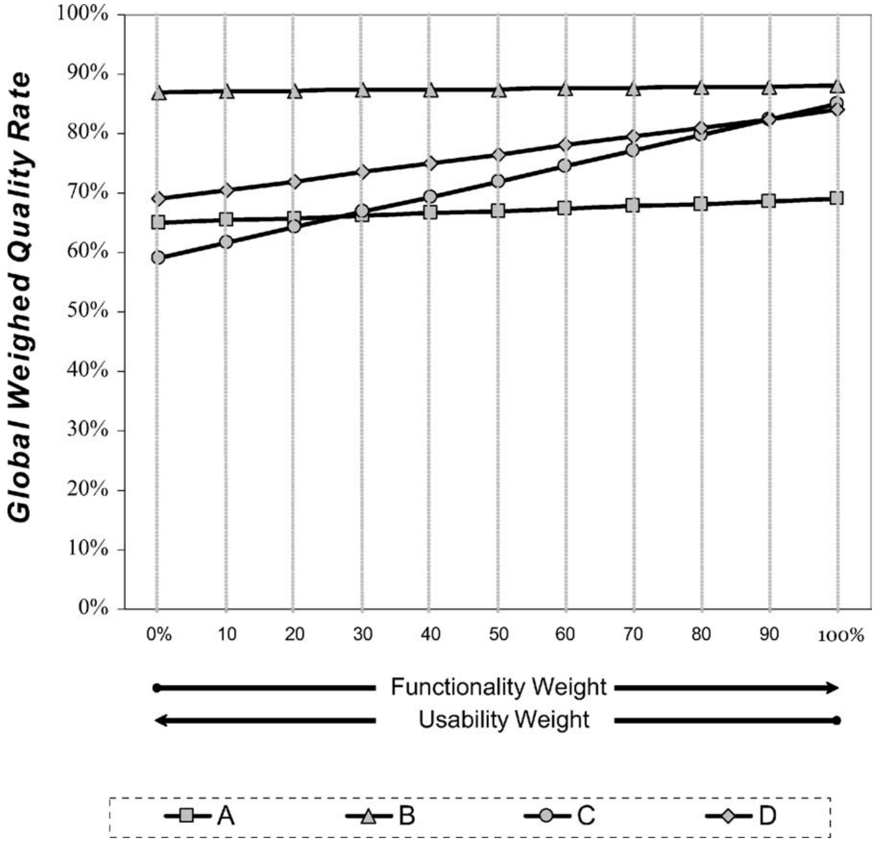

Fig. 4. WGQR for the four DESS-c assessed by SQMO+.

functionality and usability categories represented in complementary scales. The functionality category increases from left to right (from 0% to 100%); and the usability category flows in the opposite direction.

As seen in Fig. 4, software *B* retained its first place, independent of the variation of the ‘‘weight’’ assigned to the functionality and usability categories. Furthermore, the rank order of the DESS-c was maintained in the range of 30 and 90% for the functionality category. As a result, software *B* ended in first place, followed by software *D*, *C*, and *A*, respectively.

8. Conclusions

A model was defined to assess the quality of software systems used for hybrid simulation: SQMO+. This consisted of a set of sub-characteristics to measure whether a DESS-c could be employed in logistics. A set of sub-characteristics (40) and metrics (131) supported the selection process. It made the decision making process objective. The application of the Systemic Quality Model (SQMO) was also demonstrated in the development of SQMO+.

Acknowledgement

This research was sponsored by and conducted with the aid of the Research Deanship of the Universidad Simón Bolívar through the project *Herramientas de Soporte en Logística para la Industria Petrolera* (DI-CAI-004-02) and PDVSA-INTEVEP. The authors wish to thank the vendors who kindly provided demos and documentation of their products and who took part in the development of the case study.

References

[1] A. Álvarez, Modelo para la Evaluación de la Calidad del Proceso de Desarrollo de Sistemas, LISI — Universidad Simón Bolívar, Venezuela, 2000.
[2] N. Callaos, B. Callaos, Designing with a systemic total quality, Proceeding of the International Conference on Information Systems Analysis and Synthesis, ISAS'96, 1996, pp. 15–23.
[3] G. Díaz, M. Pérez, L. Mendoza, A. Grimán, Ampliación de MOSCA para la Evaluación de Software Educativo, LII Convención Anual de AsoVAC, Barquisimeto, Venezuela, vol. 53 (1), 2002, pp. 361–362.
[4] G. Dromey, Comering the Cimera, IEEE Software (1996) 33–43.
[5] F. Erzinger, M. Norris, Meeting the e-Chain Challenge: The Catalyst for Large Scale Change, NPRA 2000 Computer Conference, Chicago, 2000.
[6] D. Falconer, B. Guy, Truly Optimal Offsites. The Chemical Engineer (1998) 28–33.
[7] V. Hlupic, Discrete-event simulation software: what the users want, SIMULATION 73(6), 1999, pp. 362–370.
[8] ISO/TEC 91261.2, Information Technology—Software Product Quality, Part 1, Quality Model, ISO/IEC JTC1/SC7/WG6, Canada, 1998.
[9] ISO/IEC TR 15504-2, Information Technology—Software Process Assessment, Part 2, A Reference Model Processes and Process Capability, ISO/IEC JTC 1/SC 7, Canada, 1998.
[10] B. Kitchenham, Evaluating software engineering methods and tools, Part 5, Principles of Feature Analysis, Department of Computer Science, University of Keele, England, 1996 .
[11] A. Law, D. Kelton, Simulation Modeling and Analysis, McGraw-Hill, New York, 1991.
[12] M. Le Blanc, M. Tawfik Jelassi, DSS software selection: a multiple criteria decision methodology, Information & Management 17(1), 1989, pp. 49–65.
[13] J. Martínez, Modelo Sistémico de Calidad – MOSCA, LISI - Universidad Simón Bolívar, Venezuela, 2001.
[14] C. Martín, Propuesta de Modelo Sistémico de Calidad de Software (MOSCA) en la Dimensión Usuario-Cliente, LISI – Universidad Simón Bolívar y Universidad Católica Andrés Bello, Venezuela, 2003.
[15] L. Mendoza, M. Pérez, A. Grimán, T. Rojas, Algoritmo para la Evaluación de la Calidad Sistémica del Software, 2das Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC 2002), Salvador, Brasil, 2002, pp. 1–11.
[16] J. McCall, P.K. Richards, G.F. Walters, Factors in Software Quality, vol. I–III, AD/A-049-014/015/055, National Technical Information Service, Springfield, VA, 1977.
[17] T. Naylor, D. Balintfy, D. Burdick, K. Chu, Computer Simulation Techniques, Wiley, New York, 1966.
[18] J. Nikoukaran, V. Hlupic, R. Paul, in: D. Medeiros, E. Watson, J. Carson, M. Manivannan (Eds.), Criteria for simulation software evaluation, Proceedings of the 1999 Winter Simulation Conference, Department of Information Systems and Computing of Brunel University, United Kingdom, 1998, pp. 399–406.
[19] M. Ortega, M. Perez, T. Rojas, Software Quality Journal 11, 2003, pp. 219–242.
[20] T. Rojas, L. Mendoza, M. Pérez, Indicadores organizacionales para comparación de herramientas CASE en Venezuela, Revista de la Facultad de Ingeniería de la U. C. V 16(1), 2001, pp. 95–112.
[21] R. Shannon, Systems simulation: the art and science, Prentice Hall, New Jersey, 1975.
[22] J. Swain, Simulation software survey, OR/MS Today. 2001.
[23] S. Umeda, A. Jones, Simulation in Japan: State-of-the-art update, Technical report, National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Technology Administration, 1997.
[24] J. Voas, Software quality's eight greatest myths, IEEE Software 16(5), 1999, pp. 740–745.

Gladys Rincón Polo Chemical Engineer and M.Sc. Operational Research, Universidad Central de Venezuela. Professor at Universidad Simón Bolívar. She worked as a model analyst for PDVSA-Intevep (Venezuelan Oil Company). *Publications on*: IDEAS'03, Paraguay; Revista Facultad de Ingeniería'2003; AMCIS'04, USA; CISCI '04, USA.

Marinelly Alvarez Massieu Chemical Engineer, Universidad Simon Bolivar and MSc. Operational Research, London School of Economics. Over 20 years of experience in the areas of modeling, planning and refining economics in PDVSA-Intevep (Venezuelan Oil Company). She has held management positions in engineering projects and provided specialized technical services to refineries in Venezuela and USA.

María Angélica Pérez de Ovalles Member of the Association of Information Systems. Titular Professor at Universidad Simón Bolívar. Ph.D. Computer Science (1999), Universidad Central de Venezuela. Her current research interests are in process improvement, software engineering, methodologies, case tools, information technology. Expertise areas: information systems, methodologies and software engineering. *Publications on*: ISACC'95, Mexico; AIS'96, USA; AIS'97, USA; AIS'98, USA; AMCIS '99, USA; CLEI '99, Uruguay; JOOP 12(6); AMCIS '00, USA; Journal Information & Software Technology, 2000; JOOP, 2000; SCI '00, USA, AMCIS '01, USA; JIISIC '01, Argentina; Information Management Systems, 2002.

Sara Hernández Holds a B.Sc. in Chemical Engineering from Universidad Simón Bolívar, Venezuela. She works as a model analyst in economic planning for PDVSA-Intevep. Concurrently, she is working on proposals for using discrete-event simulation to identify and solve potential bottlenecks that may be affecting logistics operations at Venezuelan marine ports.