Accepted for publication in IADIS International Conference (November 2011)
Author's Copy

EVALUATION OF USABILITY IN A REMOTE LEARNING SYSTEM UTILIZING MARKOV MODELS

Janaina Rodrigues[1], Morganna Diniz[1], Simone Bacellar Leal Ferreira[1], Denis S. Silveira[2], Eliane Capra[1]

[1] *Universidade Federal do Estado do Rio de Janeiro (UNIRIO)*
Av. Pasteur, 296 – Urca – CEP 22290-240 – Rio de Janeiro – RJ – Brazil

[2] *Universidade Federal de Pernambuco (UFPE)*
Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária - CEP 50740-560 – Recife PE – Brazil

ABSTRACT

With the growth of web-based systems, usability has gained visibility in the Human-Computer Interaction area (HCI). Usability evaluation methods consist of systematic processes of data collection with the purpose of analyzing how users handle a product in order to execute tasks in a computer environment. The present research aims at analyzing the usability of a remote learning system in its initial development phase, using a quantitative usability evaluation method through *Markov* models. These models allow a series of measures of interest to be calculated and they have been successfully utilized in the evaluation of computing and communication systems, but they have not been much used in the evaluation of system usability.

KEYWORDS

Usability, Evaluation, Markov Decision Process, E-Learning.

1. INTRODUCTION

The success of a site rests in the quality of the services it provides and in the way these services are offered and not just in its esthetic value. The widespread use of web-based applications, including Remote Learning, highlighted the importance of usability (Preece *et al*., 2005). With respect to computer systems, it is important to point out that the interface is the most accessible part in the system, since this is the tool through which he/she performs tasks and the accessible services. However, few organizations consider this fact when planning their applications (Ferreira & Leite, 2003).

One way to provide this usability is by evaluating the interfaces. This evaluation allows the detection of problems in system use. In order to check usability, several methods can be utilized. These methods consist of a systematic data collection with the purpose of analyzing how users handle the product to execute their tasks in a computational environment (Preece *et al*., 2005). These processes can be *empirical*, which are those in which the evaluators involve the users in data collection; or *analytical*, when evaluators examine interface features without user participation. With regard to analytical evaluation processes, it became evident that model-based inspection has not been much explored. This is the reason for choosing this method for this research.

Some of the analytical methods are adequate for the evaluation of usability of the system in its initial phases, allowing detecting and solving problems in the interaction, reducing the cost of changing the system after it has been implemented. Thus, Markov models, for being analytical and predictive, are able to provide quantitative comparisons about usability when the project is still in the initial phase (Kitajima *et al.*, 2005; Thimbleby *et al*., 2001).

Therefore, this research, being quantitative and exploratory, aims to apply usability evaluation criteria in a Remote Learning system basing on Markov models, in order to contribute to making decisions related to system design and to help in improvement proposals to be implemented in the system. It is expected that this contribution, although focused on Remote Learning systems, can be extended to other systems, so as to help in the construction and adaptation of interfaces better suited to the recommendations of system usability.

2. USABILITY EVALUATION

The evaluation of interfaces allows detecting usability problems in a system. This evaluation consists of a systematic data collection process aimed at analyzing how users handle a product to perform their tasks in a computing environment (Preece *et al.*, 2005). Among its main objectives the following ones stand out: evaluating the quality of an interface design; detecting possible interaction problems; verifying conformity with usability standards among others (Barbosa & Silva, 2010). Usability evaluation methods vary according to the system development stage in which they are applied and according to the way the data necessary for the analysis is collected. With regard to the development stage and to the interface design, the evaluation can be classified as formative or additive. The formative evaluation has to do with undertaking the evaluation at the beginning of the project and is intended to detect interaction problems thus allowing corrections before the product is finalized and implemented. The additive evaluation has to do with the evaluation of the finished system or website, with the purpose of finding out whether it meets the defined requirements and standards (Barbosa & Silva, 2010).

2.1 *Markov* Models for Usability Evaluation

The utilization of Markov models for usability evaluation was proposed by (Thimbleby *et al.*, 2001). The authors used the technique to check the usability of finite state machine devices (FSM), such as microwave ovens or cell phones before creation of a prototype. After the study about usability and Markov models, the author presented a tool that can be used by designers in the evaluation of usability by way of a cost/knowledge chart, which does not require heavy mathematical knowledge.

Another research, realized in 2005, proposed a method to quantitatively evaluate the usability of Remote Learning systems, analyzing its impact in the development of these systems. For this purpose the inspection method known as “Cognitive Walkthrough” and the Markov chain were used. The results demonstrate that the utilization of Markov chains returns simple and fast results, aiding the utilization of the inspection method (Kitajima *et al.*, 2005). The advantage of utilizing Markov models in the analysis of usability is the possibility of getting metrics that provide quantitative comparisons that help to make a quick evaluation of usability when the system is still in pre-design phase. Some users have to make choices in each operation. It is important to consider the number of possible ways they can achieve their objectives. The number of states/transitions that a user takes is an obvious and simple form to measure the difficulties faced by the user (Thimbleby *et al.*, 2001).

Markov models can be utilized in a great number of scientific areas (Cunha *et al.*, 2010; Marinho *et al.*, 2010; Morais *et al.*, 2010). In order to utilize them in usability evaluation, it is necessary to choose a criterion that is more adequate to evaluate the results after the application of the formulas used in the model. Since literature presents several techniques to evaluate usability evaluation, this research was based on a technique that took in account the several methods of different authors, in other words, the criteria presented by Gassenferth (Gassenferth *et al.*, 2008).

Based on a qualitative and comparative study of the methods utilized for usability evaluation by respected authors in the HCI area, such as Shackel (Shackel, 1986), Bastien and Scapin (Bastien & Scapin, 1993), Jordan (Jordan, 1998), Shneiderman (Shneiderman, 1998), Quesenbery (Quesenberry, 2001), the research of Gassenferth *et al*. (2008) summarized a set of rules to measure the degree of usability of systems, thus creating six converging criteria to evaluate usability, as shown in Figure 1 (Gassenferth *et al.*, 2008).

Figure1. Usability Evaluation Criteria

Criterion	**Concept**
Ease of learning	This criterion sets forth that the system should not only be easy to learn and to handle, but that it also has to react in the context in which it is being used and that it should be clear in the presentation of the expressions to keep the user reacting well to the system, thus helping learning.
Ease of recall	This criterion has to do with remembering the operations performed in the system, even if the operation is executed only sporadically or after a long time.

Error Control	This criterion has to do not only with handling operation errors, but also that the user has to be clearly informed about what caused the error and is able to easily correct it.
Efficiency	This criterion refers to the user being able to use the system and perform the tasks in a fast and secure way. This also includes the speed in which the user is able to complete a task.
Effectiveness	This criterion refers to the idea of doing the right thing in the best possible way.
Satisfaction	This criterion has to do with the user’s perception in the use of the system. It is directly related to the user’s opinion with respect to the system’s user-friendliness.

Note Source: Santos, R. C., Gassenferth, W., Machado, M. A. S. (2008)..

3. MARKOV MODELS AND MARKOV DECISION PROCESSES

The creation of models is a technique used in several engineering modalities. In Software Engineering, the models can be created for several activities and in different development stages, allowing an early visualization, communication and validation of the system before its implementation (Pressman, 2004).

A stochastic model is a probabilistic model that has observed parameters and that, even though it is not possible to precisely predict the values they will assume in the future, it is possible to evaluate the probabilities associated to the future values (Haan, 1997). A stochastic model is said to be a Markov model if the system behavior depends solely on the system’s current state, i.e., the state of the model provides all the necessary information about the system (Kleinrock, 1975).

A system evolves through the transitions in its states (Blooch, 1994), where the occurrence of an action sets the transition from state n to the next state, *n + 1*. Thus Markov models can be utilized to represent changes or transitions between states, where the previous state is irrelevant to the prediction of the following states (Kitajima *et al.*, 2005). Considering that the actions, depending on their respective conditions, may indicate a change of state, it is necessary to define the probabilities connected to these changes (transitions). It is also possible that the occurrence of an action keeps the system in the same state, and in this case there is no occurrence of transition (Kleinrock, 1975). To complete the model it is necessary to provide a distribution of probabilities to the initial state, i.e., to define the initial probabilities of each state in the system (begin of observation). The π(0) vector which represents this distribution is defined by the formula in Figure 2.

Figure 2. Formula Representing a Distribution of Probabilities in the Initial State

$$\boxed{\pi(0) = [\pi_0(0), \pi_1(0), \pi_2(0), ...]}$$

Note Source: Kleinrock, L. (1975)..

where π(0) represents the probability of the system being in state *i*, before any transition in the system. Considering that *P* is the matrix of probabilities of transition between system states, the probabilities of the states after n transitions, represented by the formula in Figure 3 (Kleinrock, 1975) are calculated.

Figure 3. Formula Representing a Distribution of Probabilities to Any State

$$\boxed{\pi(\mathrm{n}) = \pi_0(0)\ \mathrm{P}_\mathrm{n}}$$

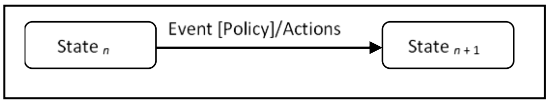
Note Source: Kleinrock, L. (1975)..

Markov decision processes can be used to model systems with several states, actions that may (possibly) modify the system’s state and perceive (albeit indirectly) the result of each executed action (Puterman, 1994). In Markov decision processes, transitions between states are probabilistic and each action has a reward (cost) which depends on the state in which the process is at. The processes obey the Markov property where the effect of an action in a given state depends solely on the action and on the system’s current state. Differently from Markov chains, decision processes model the possibility of an agent (or a decision maker) periodically interfering in the system by undertaking actions. Normally, a state captures all the relevant information available to the decision making process (Papadimitriou & Tsitsiklis, 1987).

The dynamics of a system modeled by Markov decision processes assumes that the agent, who in a Software Engineering context, can also be seen as player, at given moments (decision making occasions), and based on a system policy (business rules), is able to execute an action able to change the system state

(Jacobson, 1993). Figure 4 shows the dynamics of the way a system functions, made up of only two states, modeled through a state machine. In it, the event generated by the decision maker (who can be a user, or an event generated by the system itself or even a robot) will validate the business policy to execute or not the actions that will affect the change of system state. Business policy is understood to be the set of business rules associated to the transition from one state to another. However, it is worthwhile to point out that the state change, as defined in (Harel, 1987), will only process if the policy is validated.

Figure 4: System Dynamics Modeled by a Markov Process



Note: Source Torres, J. V. (2006)..

Therefore, the probabilities of transition are functions that depend only from the current state and from the subsequent action. Another important note is that external events may cause transitions in the system, as well as an action. Events may correspond to the natural evolution of the process or to the actions of other agents/players. For the purpose of predicting the transitions resulting from a decision, as in the case study presented in this research, these events may be treated as implicit, when the effects of an action executed by an agent/player in a given state happen together with an external event, resulting in that both are combined and modeled as a single action. If the process is in state n, and an event or action is chosen, then the system's next state (n + 1) is determined according to the probability of transition $p_{ij}(a)$. In other words, the probability of going from one state (i), to another (j) is determined according to action (a). If an action has no effect when executed in a given state, this characteristic can be recorded in this action's transition matrix. The transition matrix is shaped as a distribution for all actions a, states n, and stages t (*time*) (Torres, 2006).

Whenever necessary, the agent reviews the policy and determines the next action. When executing a policy, the agent is rewarded for each occasion of decision. This is done by the definition of a value function. Through this function, the agent/player is able to judge whether a given behavior is good or bad depending on its effect on the system path (Papadimitriou & Tsitsiklis, 1987).

4. CASE STUDY: PROJECT OF A REMOTE LEARNING SYSTEM

The case study method was chosen, as it allows understanding how and why some phenomena occur, in addition to disclosing the working of cause-effect relations (Yin 2005). A case study of the examination of a learning system's pre-project data in a private sector company was performed with the objective of implementing a system offering IT-related courses to its employees. The courses cover six learning lectures with classes at different levels.

The initial survey was performed with the company's management and its IT area to define the tools to be provided in the project. The system would be web-enabled and would be available to all company employees (close to 150 persons). One of the features of the site to be developed would be that in addition to being accessible within the company, the system should also be accessible through the Internet.

In a second stage the Moodle free software (Moodle, 2011) was chosen as the tool best equipped to the requirements set forth in the initial survey. Moodle is a system that allows producing courses based on the web with resources that make the management of the learning process and of team-based work feasible. The third state in this research consisted in identifying the functions/rules defined by the company's IT area and approved by company management, as shown in Figure 5.

Figure 5. Rules Defining the Proposed Site

require the user to log in due to security reasons
present the available courses to the user
provide direct access to e-mail through own tool
allow the user to present his/her doubts and contributions through the system forum;
allow the user to download information, whenever requested to do so;
allow the user to upload information, whenever requested to do so;

allow the user to write tests through the tool;
require the user at the end of each class, to fill out a multiple choice quiz.

Note: Source: Data Collection

By utilizing the rules defined for the project (Figure 5) it was possible to list the different system states making up a total of nine states. Figure 6 shows the functions of the system changed into states, assuming for this purpose a finite state matrix (*S*) for all possible system states.

Figure 6. Functions of the System to be Transformed into States

Functions	Description
Login	Enter the system;
Courses	Courses available to be chosen by each user;
Email	Access to e-mail;
Forum	Doubts and contributions to the lectures;
Download	File download;
Upload	File upload;
Tests	Course tests;
Quiz	Quizzes at the end of each lecture;
Lectures	Course lectures.

Note: Source: Data Collection.

Modeling through Markov is supposed to represent a set of the model being studied where only the utilization period of the system is considered, thus ignoring the period when the user is not logged to the system. Figure 7 shows the nine states listed in the system's pre-project phase and the set of available actions.

Figure 7. Sequence of Actions for the Decision Maker

State	Sequence of Actions
Login	Enter name; Enter password; Click on OK; Open link to course;
Courses	Choose course; Open link to lecture;
E-mail	Read e-mail; Reply e-mail; Forward e-mail; Delete e-mail;
Forum	Read messages; Post messages;
Download	Choose download; Download file;
Upload	Submit file;
Tests	Submit test;
Quizzes	Submit quiz;
Lectures	Choose lecture;

Note: Source: Data Collection.

Figure 8 shows the matrix of transition of system states in the initial state, derived from the calculation of probabilities according to the rules specified by the responsible area (Figure 6). The figure shows that starting from the <*login*> state, the process is directed to the <*courses*> state in 100% of cases. According to the Figure 2, the system's initial state (*i*) at this point is <login> and the next state (*j*) is <*courses*>. The probability of the system going from <*login*> to <*courses*> is calculated through the formula $p_{ij}(a)$, where (*a*) is the decision maker's action, which, in this case, only shows the opening of the <*courses*> link.

Figure 8. Matrix of Probabilities Referring to Transitions in the System

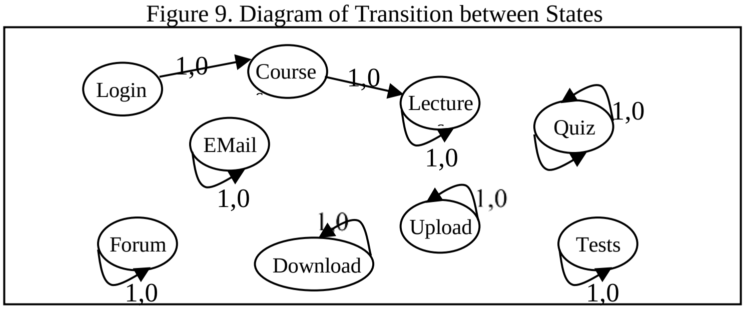
	Login	Courses	E-mail	Forum	Download	Upload	Tests	Quiz	Lecture
Login	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Courses	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Email	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Forum	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Download	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
Upload	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Tests	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
Quizzes	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Lectures	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Note: Source: Data Collection.

From the <*courses*> state the user is directed to the <*lecture*> state in 100% of the cases. This happens because in the sequence of actions shown in Figure 8, the <*courses*> state has only one exit link to the <*lecture*>

state. At this point, it is possible to identify an error in system development which repeats in other states. In the <*e-mail*>, <*forum*>, <*download*>, <*upload*>, <*tests*>, <*quiz*> states, the entries and exits only happen within their own states. Thus, no transitions occur from these states to other system states, i.e., once a user gets to one of these states, this user will remain in it. In the <*lecture*> state, requests coming from the <*courses*> state are received, according to the defined policy; however there is no exit definition and the probability of the user remaining in this state is 100%. Figure 9 shows the transition diagram referring to the matrix on Figure 8.

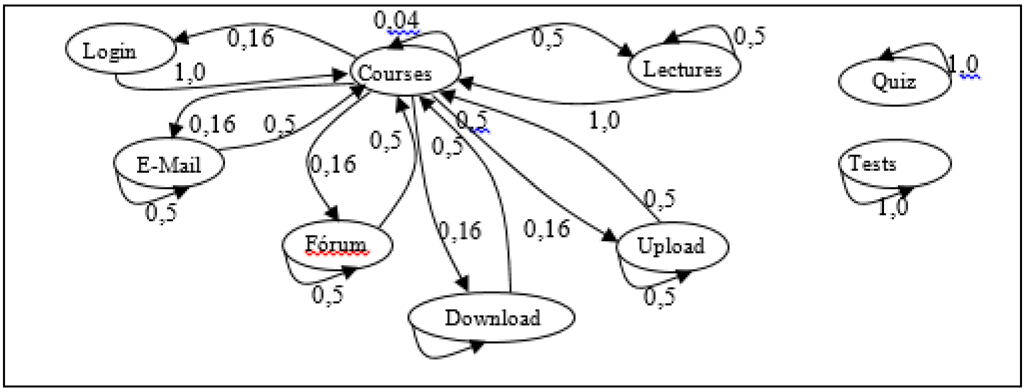
Figure 9. Diagram of Transition between States



Note: Source: Data Collection.

Figure 10 shows the diagram after a simple change such as setting links after logging in the system. The transition between the <*login*> and <*courses*> states continue at 100% (according to the rules). However, the other states in the system had their transition probabilities changed, correcting the usability errors related to the ease of learning.

Figure 10. Transition Diagram Referring to the Matrix



Note: Source: Data Collection.

It is still possible to see a design error. The <*quiz*> and <*tests*> states are still not accessed from any state, except themselves, with a 100% probability of staying in the same state. According to the system functions, the student is forced to take a quiz at the end of each lecture. Therefore, it is more sensible that the student is not allowed to go to <*courses*> and choose another lecture before taking a <*quiz*>. With one more change, a new sequence of actions is generated, and, consequently a new matrix of probabilities (Figures 11 and 12).

Figure 11. Changed Sequence of Actions

State	Sequence of Actions
Login	Enter name; Enter password; Click on OK;
Courses	Choose course; Open link to lecture; Open link to e-mail; Open link to Forum; Open link to Download; Open link to Upload; Logoff;
Email	Read e-mail; Reply e-mail; Forward e-mail; Delete e-mail; Go back to Course;
Forum	Read messages; Post messages; Go back to Course;
Download	Choose download; Download file; Go back to Course;
Upload	Submit file; Go back to Course;
Tests	Submit test; Go back to Course;
Quiz	Submit quiz; Go back to Course;
Lectures	Choose lecture; Open link to tests; Open link to quizzes;

Note: Source: Data Collection.

Figure 12. Matrix of Transition Probabilities

	Login	Courses	E-mail	Forum	Download	Upload	Tests	Quiz	Lecture
Login	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Courses	0.16	0.04	0.16	0.16	0.16	0.16	0.0	0.0	0.16
Email	0.0	0.5	0.5	0.0	0.0	0.0	0.0	0.0	0.0
Forum	0.0	0.5	0.0	0.5	0.0	0.0	0.0	0.0	0.0
Download	0.0	0.5	0.0	0.0	0.5	0.0	0.0	0.0	0.0
Upload	0.0	0.5	0.0	0.0	0.0	0.5	0.0	0.0	0.0
Tests	0.0	0.5	0.0	0.0	0.0	0.0	0.5	0.0	0.0
Quiz	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.5	0.0
Lectures	0.0	0.0	0.0	0.0	0.0	0.0	0.33	0.33	0.34

Note: Source: Data Collection.

5. ANALYSIS OF THE RESULTS

After generating the matrix of probabilities among states and actions, the evaluation of usability based on Markov models allows the analysis of four of the six criteria specified in (Gassenferth *et al.*, 2008), shown in the Figure 1. Thus, the "*Ease of Recall*" and "*Satisfaction*" criteria can be verified using other evaluation methods proposed in literature.

The following results are based on the possible criteria:

- Ease of learning: With the use of states and probability diagrams it is possible to understand how a user will be directed through the system. For example: it is possible to visualize that a user in <*login*> state will be directed to the courses <*state*> in 100% of the cases. It is also possible to understand how a change in the system's initial phase affects the user's "path", creating new probabilities among the system's states;

- Error Control: The analysis of the diagrams with the system states, shows that there is no error-handling state. This state can be created, and according to the probabilities, error frequency can be determined (keeping in mind that errors can occur in any system state);

- Efficiency: According to the rules pre-defined in the pre-project, the user will always start the system through the <*login*> function and be sent to the <*courses*> state. The evaluation of the case study in the first survey (Figure 9), shows that if a link is placed so that the user after performing the <*login*>, is required to go to the <*courses*> state, response time is reduced and the speed of access to system information is increased;

- Effectiveness: According to the states generated for the system, it is possible to predict how the user will interact with the system and the corresponding interaction probabilities. This information helps to better evaluate which tasks should be included in the system so that it can execute the tasks demanded from it with the lowest possible level of user interaction.

6. FINAL CONSIDERATIONS

The success of a site does not only depend on its esthetics, but on the quality of the services it offers. In Remote Learning systems, usability is an essential requirement so that the system can meet its users' needs.

In the evaluation of usability it is possible to apply different evaluation methods. One of them consists of Markov models which help to make quantitative comparisons about usability when the project is still in its initial phase. Thus, this research aimed to investigate the utilization of Markov models in the evaluation of usability in a Remote Learning system in its pre-project phase and to help in proposing improvement suggestions. To this end, a case study of a Remote Learning system to be utilized by a private sector company was performed.

This study allowed the generation of a probability matrix among states and actions, which was utilized in the evaluation of usability based on Markov models, where the usability criteria specified by Gassenferth *et al.* (2008) were analyzed. The Markov models allow four criteria in the evaluation of usability: ease of learning, error control, efficiency and effectiveness.

The study showed that the utilization of Markov models brought more flexibility to the developer in making the right decisions for the project. The utilization of Markov models also allowed to easily confirm improvements in system usability based on the metrics utilized by the authors most respected in the area.

The obtained results demonstrated that the utilization of this model furnished simple and quick results. Future research topics include the modeling of user errors, the verification of the increase in the number of actions within the states and to compare the results with another methods of usability evaluation already consecrated in the literature, such as Nielsen (Nielsen, 1994), Wharton (Wharton *et al.*, 2004), Rubin and Chisnell (Rubin & Chisnell, 2008) and Snyder (Snyder, 2003).

REFERENCES

Barbosa, S., Silva, B. ,2010. Interação Humano-Computador (1a ed.). Rio de Janeiro: Elsevier.

Bastien, J. M. C.; Scapin, D. L. (1993). Human factors criteria, principles, and recommandations for HCI: methodological and standatdisation issues. França: INRIA.

Cunha, U. S.; Machado, S. A. ; Figueiredo Filho, A.; Sanquetta, C. R. (2010). Predição da estrutura diamétrica de espécies comerciais de terra firme da Amazônia por meio de matriz de transição. Ciência Florestal, Santa Maria, v. 12, n. 1, p. 109-122.

Ferreira, S. B. L.; Leite, J. C. S. P. (2003). Avaliação da usabilidade em sistemas de informação: o caso do sistema submarino, RAC – Revista de Administração Contemporânea, Vol. 7, número 2, p. 115 – 136, Abril-Junho.

Gassenferth, W.; Santos, R. C.; Machado, M. A. S. (2008). Systems usability evaluation metrics review. Global Business and Technology Association Conference - GBATA, Systems Usability Evaluation Metrics Review, Madri, Espanha

Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming. [s.l.]: No 8, p. 231 – 274.

Haan, C. T. (1997). Statistical Methods in Hydrology. The Iowa State University Press, Chapter 15, p. 289-312, 1st ed.

Jacobson, I. (1993). Object-Oriented Software Engineering - a Use Case Driven Approach: 1933: 333 – ACM Press ; Wokingham, Eng. ; Reading, Mass. : Addison-Wesley Pub., c1993.

Jordan, P. W. (1998). An Introduction to Usability. London, UK: Taylor & Francis.

Kleinrock, L. (1975). Queueing Systems - Theory, Vol. I. Wiley-Interscience.

Kitajima, M; Kariya, N., Takagi, H. and Zhang, Y. (2005). Evaluation of website usability using Markov chains and latent semantic analysis. IEICE Transactions on Communications. E88B (4),1467-1475.

Marinho, A.; Cardoso, S. S.; Almeida, V. V. (2010). Disparidades nas filas para transplantes de órgãos nos estados brasileiros. Cad. Saúde Pública [online]. vol.26, n.4, pp. 786-796. ISSN 0102-311X.

Moodle. (2011). Course Management System (CMS) [Software]. Recouped in 21, January, 2011 de www.moodle.org.

Morais, I. A. C.; Bertoldi, A.; Anjos, A. T. M. (2010). Estimativa de um modelo não linear para as exportações brasileiras de borracha no período 1992-2006. Rev. Econ. Sociol. Rural [online]. vol.48, n.3, pp. 679-704. ISSN 0103-2003.

Nielsen, J., Heuristic Evaluation. In: R. Mack & J. Nielsen (eds.), Usability Inspection Methods. New York, NY: John Wiley & Sons, pp25-62, 1994;

Papadimitriou, C. H.; Tsitsiklis, J. N. (1987). The complexity of Markov decision processes. Mathematics of Operations Research, v. 12, n. 3, p. 441–450.

Preece, J., Rogers, Y., Sharp, H. (2005). Design de Interação: além da interação homem-computador (1a ed). Porto Alegre: Bookman.

Pressman, R. S. (2004). Software Engineering: a Practitioner's Approach (6a ed.). McGraw-Hill.

Puterman M. L. (1994). Markov Decision Processes: Discrete Stochastic Dynamic Programming. New York, NY: Wiley-Interscience. ISBN 0471619779.

Quesenbery, W. (2001). What does usability mean: Looking beyond 'ease of use'. In: 48th Annual Conference Society for Technical Communication. Chicago.

Rubin, J., & Chisnell, D. (2008). Handbook of usability testing: how to plan, design, and conduct effective tests (2a ed). Nova York: John Wiley & Sons.

Shackel, B. (1986). Ergonomics in design for usability. In: HCI 86 Conference on People and Computer. New York: Cambridge University Press.

Shneiderman, B. (1998). Designing the User Interface: Strategies for Effective Human-Computer Interaction. 3. ed. EUA: Addison Wesley.

Snyder, C. Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces. Son Francisco, CA: Morgan Kaufmann, 2003.

Thimbleby, H.; Cairns, P.; Jones, M. (2001). Usability analysis with markov models. ACM Trans. Comput. Human Inter. 8, 2, 99–132.

Torres, J. V. (2006). Representações compactas para processos de decisão de markov e sua aplicação na administração de impressoras - Dissertação apresentada a Escola Politécnica da Universidade de São Paulo para obtenção de título de Mestre em engenharia.

Yin, R. (2005) Estudo de caso: planejamento e métodos. 3.ed. ISBN: 85-363-0462-6. Porto Alegre: Bookman.

Wharton, C.; Rieman, J.; Lewis, C.; Polson, P., The Cognitive Walkthrough Method: A Practitioner's Guide. In: R. Mack & J. Nielsen (eds.), Usability Inspection Methods. New York, NY: John Wiley & Sons, pp25-62, 1994;