Modeling and Simulation as a Quantitative Pedagogical Approach to Teaching E-Commerce to Diverse Audiences

Dan Tappan and Tiffany Blount
Department of Mathematics and Computer Science
Western New Mexico University, Silver City, NM, USA

Abstract - *Simultaneously teaching a vast and complex subject like e-commerce to a diverse audience of undergraduate and graduate students in computer science and management information systems in face-to-face and online environments is a daunting challenge due to the many conflicting needs and expectations. This work describes a stochastic discrete-event simulator grounded to a pedagogical framework, which together seamlessly expose students to an end-to-end process of analyzing, designing, building, evaluating, and refining e-commerce models, as well as reporting on all these aspects. It especially focuses on analytical and communication skills. The simulator provides the basis for designing and executing controlled experiments on almost any aspect of e-commerce. It generates quantitative results that allow students to make informed, justifiable, and persuasive decisions, from which they can not only solve actual problems, but also learn from the experience.*

Keywords: modeling, simulation, analytical methods, pedagogy

1 Introduction

E-commerce is a subject of great breadth and depth. As such, it often draws a variety of students from disparate programs with different—and often conflicting—needs, expectations, skill sets, and even prerequisites. In the case of this work, the programs are Computer Science (CS) and Management Information Systems (MIS) at Western New Mexico University, the students are undergraduates and graduates, and the delivery is face to face and online. The course is also designated as writing intensive, which imposes composition requirements.

Teaching such a complex subject to such a diverse audience is extremely challenging. Furthermore, any realistic e-commerce solution would likely be too large and complex to implement to a reasonably functional degree, even for a CS-only audience. It is thus not uncommon for e-commerce courses to take a predominantly non-technical approach by having students read and write about the subject, but not actually do it [1,2,3]. Unfortunately, many students, especially undergraduates, lack the analytical and communication skills to benefit from this approach [4,5,6]. They also lack a quantitative basis for making, evaluating, and justifying their decisions.

This paper describes a prototype software system that combines the technical concepts of modeling and simulation with a pedagogical framework targeted to the needs of the students, the requirements of the course, and the expectations of the real-world work environment.

2 Pedagogical framework

Technical courses tend to teach in a bottom-up manner, in that they introduce a vast array of discrete solutions without necessarily considering their contextual application to overall problem solving [7]. As a result, in the top-down process toward an e-commerce solution, it may not be clear to students, given all the alternatives in their toolboxes of resources, how to get from the problem (*what* to do) to the implementation (*how* to do it). The pedagogical framework of this work helps bridge this gap. It addresses the progression of holistic understanding of e-commerce subject material over time as students accumulate *data*, *information*, and *knowledge* throughout their studies, and eventually *wisdom* over their careers. This DIKW model in Figure 1 helps students connect the dots within the subject of e-commerce and throughout their curriculum [8,9]:

a. Data: no associativity or context
b. Information: associativity within one context
c. Knowledge: associativity within multiple contexts
d. Wisdom: generalization of principles based on knowledge from different sources over time

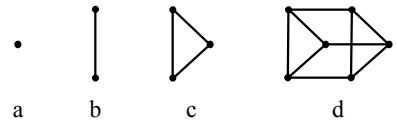

Figure 1: Data, Information, Knowledge, and Wisdom

2.1 Relationship to teaching philosophy

The pedagogical framework derives heavily from the way experienced professionals—those having achieved "wisdom" in Figure 1—commonly solve real-world engineering problems [10]. The goal is to decompose a problem in both a top-down and bottom-up manner into its isolated components with well-defined behaviors, connections, and interactions. This breakout facilitates clearly addressing the independent, dependent, and interdependent multidimensional roles of *who*, *what*, *when*, *where*, *why*, and *how* (W^5H) [11]. Professionals can often

address multiple roles simultaneously, but students appear to benefit more from a targeted, piecemeal approach that avoids conflation and the resulting confusion, misunderstanding, and impediment to learning [10]. To this end, the following subsections touch on how assignments associated with this system reflect a carefully staged introduction of new material while circularly revisiting and strengthening previous material from both this course and others. Individually, assignments consist of a series of relatively small, focused questions on a specific topic for depth of coverage, and collectively they provide a representative cross-section of the general subject matter for breadth. Section 4 will address how the system aligns with this philosophy.

2.1.1 Problem analysis

Analyzing a problem involves establishing a holistic understanding of what needs to be solved and why. The running example here is to estimate the number of servers and their processing delay, or latency, to satisfy a supply of customers at a hypothetical online bookstore when no supporting data are available. Students must define in their own terms what customers are, how many there could be, how they behave, what their effect is on servers, and so on. In support of their decisions, they are expected to draw upon resources like use cases, case studies, related solutions and research, and their own experience.

2.1.2 Solution design

Designing a solution involves proposing and specifying the abstract components for a hypothetical working system. In this context, abstract means describing in English what the necessary components are and what they can do without addressing the concrete technical details of how to make them actually function. Given reasonable justification, there are no right or wrong answers at this point. However, there is also rarely a defensible basis or degree of confidence behind them due to the lack of quantitative supporting data. For example, students estimated needing from 1 to 10 servers based on contrived numbers of customers. The range itself is indicative of coarse understanding because nobody estimated hundreds of servers, but a finer understanding of the differences within this range is lacking.

2.1.3 Solution virtual implementation

Implementing the design involves creating a corresponding representation that can actually be executed. Ordinarily, in a CS-only environment, this step would incorporate a variety of web-application technologies, with programming as the glue logic. However, given the constraints of the audience, this approach is not a viable option. Nevertheless, realistic execution in some form is still critical to understanding an e-commerce solution holistically. Modeling and simulation bridge this gap. A well-designed virtual implementation of the solution captures the essence of its lower-level operation while keeping its programmatic and technical aspects at a higher level.

The model is a concrete, computable definition of the abstract components in the design. Mapping from the abstract to the concrete is a complex thought process, which this framework facilitates. The key concept is similitude, or the bidirectional correspondence between the components in the real world and the virtual world [12]. No model of any kind ever perfectly reflects its real-world counterpart; it is always an abstraction. Deciding what to include (the breadth) and to what degree of realism (the depth) is a skill that students need to develop because most fields, especially CS and MIS, actually work with the virtual world, not the real one [12].

2.1.4 Solution simulation

Executing the model involves running a rich set of simulated experiments on it to observe its operation and establish ranges for its performance with a reasonable degree of confidence. For practicality, especially as this work is still in progress, students are not expected to implement the model or simulation, or even run the simulation. Rather, their role is in carefully articulating—as nominal subject-matter experts and consultants—how someone else (here, the instructor) can actually build and run it. This opportunity gives them valuable first-hand knowledge of how communication can directly affect the outcome of a project, for better or worse.

2.1.5 Solution evaluation

Evaluating the solution involves answering questions of interest related to the problem analysis. At this point, for example, students can actually see the performance of their estimates for the number of servers and their latency. They can then iteratively refine their decisions based on justifiable, quantitative grounds. There are also countless other analytical opportunities, such as determining maximum operating loads, bottlenecks, critical paths with single points of failure, redundancy requirements, up-time expectations, and the effects of simplifying assumptions on the model.

2.1.6 Reflective inference and reasoning

Evaluating the solution involves more than just answering questions of interest; it also involves learning from their relationships so next time students can pose better questions. As a result, they can perform better analyses and designs earlier in the process, with less need for later refinement. This reflective meta-analysis meshes well with the DIKW model by helping students connect the dots with respect to the many and varied facets of an e-commerce solution. It also helps them address not only current questions like server count, but also future questions like the scalability and extensibility of their solution based on projected growth.

2.1.7 Solution reporting

Solving a problem and learning from the process are important accomplishments for individual students, but in a real work environment, they would also be expected to communicate among many other stakeholders about all aspects of their work. They must therefore understand their audiences and target their writing appropriately. For example, a report to other team members is usually far more technical and detailed than one to managers or clients.

2.2 Relationship to Bloom's Taxonomy

This teaching philosophy developed in both an ad hoc and empirical manner over significant time spent by one of us (Tappan) in industry and academia. Although it is not a formalized pedagogical approach, it nevertheless aligns very closely with the classic Bloom's Taxonomy, which considers the following ordered stages of learning [13]:

- *Knowledge* is applying learned facts, terms, concepts, and existing solutions to the current problem in a relatively direct, bottom-up manner. For example, the online bookstore is obviously a web application, and therefore it needs a web server and network connection.
- *Comprehension* is interpreting and understanding the significance of the knowledge-based decisions to refine them in context. For example, based on what students know about their bookstore model, one to five servers is a reasonable estimate, whereas 500 would not be.
- *Application* is using existing knowledge in new ways based on a new context. For example, connecting the servers through a network switch is an obvious requirement, but there are many ways to do so beyond what students may have ever seen in the classroom.
- *Analysis* is reasoning over the holistic combination of everything involved in a solution to make informed decisions. For example, three servers might be currently adequate, but the projected peak traffic might suggest an additional server to buffer against demand.
- *Synthesis* is combining existing and new aspects of a solution to satisfy the requirements that students determined for it. This solution could be the actual e-commerce site, but in this work, it is a model of it.
- *Evaluation* is determining the success of a solution, and then communicating it in a justifiable and persuasive manner to the stakeholders. For example, adding the fourth server may provide valuable additional processing at relatively little extra cost.

3 Related work

This work is based on analysis of e-business processes through modeling and simulation under an umbrella of pedagogical considerations. All these elements have a long history of study and usage. Even a superficial review of related work would be beyond the scope of this paper. Nevertheless, several were of particular value in framing it. Pateli and Giaglis [14] and Ree [15] provide a comprehensive comparison of frameworks for understanding and analyzing e-business models. Parker and Swatman [16] bridge this view of practical application to the educational environment. From this point, Pastor, et al. [17] serves as a technical transition into conceptual aspects with respect to modeling and simulation. Dort [18] then offers an extensive literature review of pedagogy-oriented simulation. Although dated (1989), its overall analysis applies equally well to today's technology, which Guralnick and Levy [19] and Gilliot and Rouvrais [20] address. In addition, Bouhadada and Laskri [21] consider distance learning, and Cao, et al. [22] formally studies the educational value of simulation in terms of measurable learning outcomes.

4 System description

The system is a general-purpose simulator for any interconnected components that can be modeled in a network of propagated events. While the emphasis in this work is on high-level processes like e-commerce, nothing precludes lower-level processes like software and hardware architectures, or even digital circuits. This flexibility allows the same pedagogical foundation to be applied in, or connected with, other courses.

The implementation is in Java, with JavaCC for parsing the modeling language. The portability of Java is especially useful in the online teaching environment, where supporting students' computers remotely would be troublesome. The entire system, including examples and documentation, will be available as open-source software at `shelby.wnmu.edu` once it is stable.

4.1 Model

The model defines the components of a solution and their interconnection network. The pedagogical emphasis is on their data aspects (what components are in terms of their inputs and outputs) and their control aspects (what they can do in terms of their processing). Figure 2 illustrates the example client-server model with the following informal definition:

- The *customer generator* (CG) issues a customer event at random intervals.
- The *network switch* (NS) distributes the customer to the next server in a round-robin manner.
- The *arrival logger* (AL_n) records the customer's arrival at the designated server.
- The *server* (S_n) holds the customer for a certain time.
- The *departure logger* (DL) records the customer's departure from the server.

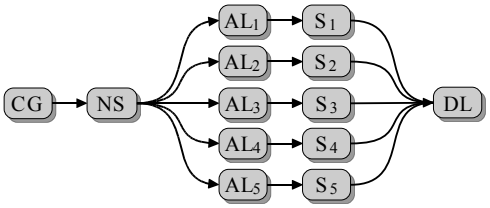

Figure 2: Example Client-Server Model

4.1.1 Language description

The formal definition of a model uses a hybrid functional language within templated programming structures for each component and the network. The functional approach allows almost any aspect of a definition to be evaluated dynamically, which results in powerful, yet small, clean, and easily understood, pieces of the overall solution. It supports a wide range of typical programming constructs such as assignments, conditionals, arithmetic and logical expressions, set operations, loops, input and output, and logging.

4.1.2 Components

The templated programming structure for any component reflects the input, processing, and output model in Figure 3. There are any number of inputs from other components, each of which feeds into its own input queue (IQ_n) and then to a processing unit (P), which, in turn, produces any number of outputs, each into its own output queue (OQ_n). The inputs and outputs can be untyped or typed as integer, real, boolean, string, or enumeration. They can also have optional ranges. The queues play the customary data-structure role of holding events, as well as imposing timing constraints by retarding flow through the component. The processing unit has access to a local symbol table (ST) to hold state information as variables and manage recursive function calls.

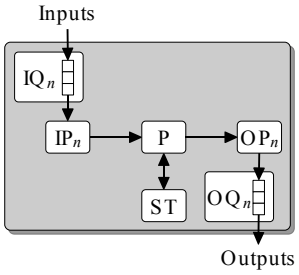

Figure 3: Component Framework

Between the queues and the processing unit are optional input and output perturbers (IP_n and OP_n, respectively). Their role is to introduce probabilistic errors into the event stream to model realistic operating conditions. In this client-server model, however, they are not used, and they are beyond the scope of this paper.

Figure 4 shows an abridged definition of the round-robin switch that distributes one input to one of five outputs.

```
(COMPONENT switch_round_robin
  (ARGUMENTS $server_count)
  (INPUTS
    (INPUT $input1))
  (OUTPUTS
    (OUTPUT $output1)
    (OUTPUT $output2)
    (OUTPUT $output3)
    (OUTPUT $output4)
    (OUTPUT $output5))
  (BEHAVIOR
    (SET $server_index (ADD $server_index 1))
    (LOG 'switch' 'arrived' $input1 'sent_to' (CONCAT 'server_' $server_index))
    (SET $output1 $input1)
    (CASE $server_index
      (1 (SET $output1 $input1))
      (2 (SET $output2 $input1))
      (3 (SET $output3 $input1))
      (4 (SET $output4 $input1))
      (5 (SET $output5 $input1)))
    (IF (EQ $server_index $server_count) (SET $server_index 0))
  )
)
```

Figure 4: Abridged Round-Robin Switch Definition

4.1.3 Network

The network defines the interconnections between components as links from their outputs to inputs. It also propagates configuration and instantiation arguments from the simulation to the components, such as `$server_count` in Figure 4. Figure 5 shows an abridged definition that corresponds to Figure 2.

```
(NETWORK arch_server_switch_round_robin
  (ARGUMENTS $customer_rate $server_count $latency)
  (COMPONENTS
    (COMPONENT g  customer_generator $customer_rate $server_count $latency)
    (COMPONENT sw switch_round_robin $server_count)
    (COMPONENT a1 arrival   'a1')
    (COMPONENT s1 server    's1' $latency)
    (COMPONENT d1 departure 'd1' $customer_rate $server_count $latency)
  )
  (CONNECTIONS
    (CONNECTION g.$output1  sw.$input1)
    (CONNECTION sw.$output1 a1.$input1)
    (CONNECTION a1.$output1 s1.$input1)
    (CONNECTION s1.$output1 d1.$input1)
    (CONNECTION sw.$output2 a2.$input1)
  )
)
```

Figure 5: Abridged Network Definition

4.2 Simulation

The simulation provides the behavior, or operational context, to complement the data and control of the model. It provides a vast array of fine-grained flexibility in addressing different analytical aspects of the same network. For example, simulations of the client-server model can determine the maximum number of customers per hour, average wait time per customer, optimal switch strategy, optimal server count, and any bottlenecks.

4.2.1 Controlled experiments

The basis of simulation is controlled experiments in the scientific sense. The first execution of the model establishes its baseline performance, against which subsequent test executions allow direct comparison. The

difference between each execution should be one—and only one—change to the model, such as modifying a parameter or substituting a component. This single perturbation demonstrates a clear cause-and-effect relationship in the results. This Monte Carlo methodology is extremely powerful in that it simultaneously helps a student refine a model, learn about it, and develop wisdom about running thoughtful experiments in general [23].

4.2.2 Overview of execution

Executing a simulation is a process of propagating events throughout the network from the outputs of components to the inputs of others. For some models, like the client-server example, propagation is end to end with no loops; i.e., from customer generator to departure logger. More complex models may propagate outputs back into earlier inputs for feedback. The trickiest part of execution is usually in setting the initial conditions and allowing the network to ramp up to a stable operating mode before logging its performance data. In addition, a simulation needs terminating conditions to stop it after a fixed amount of time or when an arbitrary logical expression is satisfied.

A single system clock coordinates the propagation of events. Each tick corresponds to an arbitrary, fixed period of real-world time; e.g., one minute in the client-server model. It also updates independent countdowns on all input and output queues, which forward their pending head event only at a specified threshold. This triggering mechanism allows for great flexibility by controlling the processing time of components; e.g., the latency in a server.

5 Model evaluation

The purpose of simulation is to make the model perform under controlled conditions of interest. From its quantitative results, students can evaluate whether the model works at all, and, if so, how well, and then they can iteratively propose and test refinements.

5.1 Data logging

The quantitative results for evaluation derive from judiciously placed log statements in components, such as in Figure 4. The choice of what to log and how is nontrivial, but the system accommodates the process in a flexible manner. For example, it supports any number of synchronized logs. It also automatically tags each log entry with meta-information like the identifier of the component making it and the clock tick. Log statements can also call user-defined Java code to manage complex data structures and formatting of file output. Export formats for Excel and the freely available gnuplot are built in.

5.2 Presentation of results

Appropriately presenting the vast, multidimensional breadth and depth of quantitative results from a simulation is a difficult process. Even a relatively simple simulation like the client-server model produces far too much raw data to present without additional processing. Therefore, part of the students' learning process is first to decide what to include and exclude, and then to decide how to refine and present what remains. This task is both a mathematical science and an art [23].

Figure 6 illustrates equivalent tabular and graphical representations of the results for customer throughput on a single server. Server latency, as the independent variable on the *x*-axis, varies from 1 to 10 minutes. Throughput, as the dependent variable on the *y*-axis, is the percentage of customers who left within the hour of simulation time. The customer-generation rate is fixed at a 10% chance of a customer arriving during any minute.

Rate	Latency	Arrivals	Departures	Throughput
0.1	1	6	6	100%
0.1	2	6	6	100%
0.1	3	6	6	100%
0.1	4	6	6	100%
0.1	5	6	6	100%
0.1	6	6	6	100%
0.1	7	6	5	83%
0.1	8	6	5	83%
0.1	9	6	4	67%
0.1	10	6	4	67%

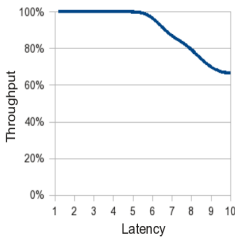

Figure 6: Single-Server Throughput Based on Latency

This presentation is adequate for one independent and one dependent variable. However, any reasonable simulation has many such variables. For example, the client-server simulation actually varies the customer rate from a 10 to 100% chance per minute (by steps of 10), the number of servers from 1 to 5, and their latency from 1 to 10 minutes. This combination produces 50 variants of Figure 6, which are obviously excessive in a report. Distilling them into a three-dimensional graph with gnuplot, however, as in Figure 7, with two independent variables (latency and customer rate) presents all the combinations in only five such graphs; i.e., one for each server count.

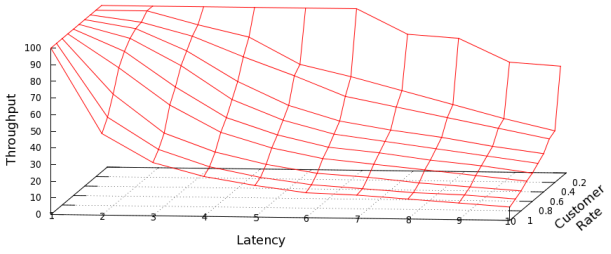

Figure 7: Single-Server Throughput Based on Latency and Customer Rate

5.3 Analysis of results

The purpose of presenting results is to be able to analyze them to make informed decisions about the problem to be solved, to understand the model, and to learn from the process of modeling and simulation. To this end,

students are expected to interpret what they see. For example, Figure 6 (and the far line in Figure 7) shows that throughput is 100% (six of six customers) until latency exceeds six minutes, after which there is a somewhat shallow drop-off in performance. The same analysis with a customer-generation rate of 1.0 (one per minute) has a severe drop-off, as shown in the nearest line in Figure 7. Such inspection and pattern discovery help students to see what happens and then to reason about when, where, how, and why it happens. This process of connecting the dots between data, information, knowledge, and wisdom provides a valuable multidimensional. holistic educational perspective [7,9].

5.4 Reporting of results

The purpose of analyzing results is to be able to report the observed patterns and conclusions. Writing concise and coherent professional reports is a skill that very few graduates with technical degrees truly possess [5,6]. Part of the problem is that most of their educational time is spent on *solving* problems, but not on *communicating* about the solutions [7]. This situation is understandable, given the limited time available in most courses. However, it is a major liability in a real work environment, where quality communication skills are essential. This system and its underlying pedagogical foundation attempt to strike a balance between problem solving and communicating by seamlessly integrating the two.

6 Results and discussion

This system is a work in progress. Its preliminary results are perhaps best considered as part of a pilot study at this point because they derive from only one semester of student contact. Nevertheless, it is based on countless experiments to develop, refine, and test its capabilities, and its underlying pedagogical framework has been in successful use for years.

A diverse audience presents challenges, but it also provides a valuable synergistic opportunity. In an educational setting, students are normally isolated within their own fields. In the real world, however, they are also usually expected to work with other fields. The breadth and depth of this system exposes everyone to every aspect of all the stakeholders to some degree. Formal and anecdotal student feedback definitely shows that they are learning to appreciate these roles in this holistic view.

Students are also learning to connect the dots within their own curriculum. The MIS students, for example, are seeing how modeling and simulation relate to operations research, supply-chain logistics, process optimization, and many other business concepts [23]. The CS students are seeing connections to software engineering, computer architecture, networking, web applications, and databases, for example. For both groups, the process of formulating questions, answering them, evaluating their answers, and iteratively refining their solutions is proving to be very educational, enlightening, and practical.

Among the notable observations is a curious lack of true understanding of commonplace technical terms and concepts by many students within their own fields. Faculty may take it for granted that students understand them, and students may not recognize or acknowledge such deficiencies. Nevertheless, this system has brought them to light and provided an opportunity to discuss the situation. One example involved the role of the power supply in a server. The scenario was to estimate and model the collective power requirements for an entire e-commerce site in order to specify its primary and backup electrical needs. A number of interesting misconceptions arose about the relationships between software and hardware, and the overall concept of power was almost universally misunderstood. A similar situation occurred with networking.

7 Future work

As a work in progress that is being iteratively developed, tested with students, and refined at the same time, this system has known deficiencies in its design and usage. The main limitation is that students do not know how to use the modeling language, which therefore requires the instructor to perform some of the pedagogical functions that should be their own tasks. A graphical user interface will make the system much easier to manipulate. In addition, it will also incorporate some of the visualization for presenting data internally to reduce the effort of exporting to external tools like Excel.

Another limitation is the lack of prebuilt components. The instructor must build everything from scratch on demand, which hinders the flow of teaching. A library of low-level support components like servers and loggers should reduce design time and increase students' creativity by allowing them to entertain more alternatives to their models. Similarly, a library of high-level components like e-commerce subsystems and even full, working e-commerce models should allow this system to integrate more tightly in the classroom, thereby increasing its effectiveness.

8 Conclusion

E-commerce is a vast and difficult subject to teach to a diverse audience. This system provides students with balanced, manageable exposure to its breadth and depth. It is based on a pedagogical approach of what to do and why, which then seamlessly flows into a practical approach of how to do it. It derives from broad industrial and academic experiences, as well as established educational foundations. It walks students through an iterative process of analyzing a problem, designing a solution, implementing and simulating a faithful model of it, analyzing, evaluating, and communicating the results, and reflecting on the experience. Every step of the process is orchestrated to

build upon the previous steps and lead to the next steps. Not only are students performing the steps, but they are also learning why they are performing them. This process results in knowledge and experience that they can apply for themselves in countless other contexts throughout both their studies and careers.

9 References

[1] Anewalt, Karen. *Utilizing interdisciplinary teams in teaching e-commerce*, Journal of Computing Sciences in Colleges, Vol. 19, No. 2, Dec. 2003.

[2] Bloss, Adrienne. *Teaching fundamentals for web programming and e-commerce in a liberal arts computer science curriculum*, Journal of Computing Sciences in Colleges, Vol. 16, No. 2, pp. 297–302, Jan. 2001.

[3] Mohammad, Rob A. *System Development and Management Methodology in Teaching E-Commerce Technology*, Communications of ACM, Vol. 9, No. 16, 2002.

[4] Spolsky, Joel. Joel on Software, And on Diverse and Occasionally Related Matters That Will Prove of Interest to Software Developers, and Managers, and to Those Who, Whether by Good Fortune or Ill Luck, Work with Them in Some Capacity, Apress, 2004.

[5] Gruba, Paul and Reem Al-Mahmood. *Strategies for Communication Skills Development*, 6th Conference on Australasian Computing Education, Vol. 30, 2004.

[6] Singh-Gupta, Vidya and Eileen Troutt-Ervin. *Assessment of Workplace Writing and Incorporation into Curriculum*, Journal of Vocational and Technical Education, Vol. 13, No. 2, Spring 1997.

[7] Computing Curricula 2001 Project, final report, Association for Computing Machinery, Dec. 2001.

[8] Tappan, Dan. *ShelbySim: A Transparent, Pedagogy-Oriented Simulator for Computer-Based Systems*, International Journal of Engineering Education, Vol. 25, No. 4, pp. 755–762, 2008.

[9] Rowley, Jennifer. *The wisdom hierarchy: representations of the DIKW hierarchy*, Journal of Information Science, Vol. 33, No. 2, pp. 163–180, 2007.

[10] Ramo, Simon and Robin K. St. Clair. The Systems Approach: Fresh Solutions to Complex Problems Through Combining Science and Practical Common Sense, KNI: Anaheim, 1998.

[11] Van Gaasbeek, J. R. and J. N. Martin. *Getting to Requirements: The W5H Challenge*, 11th Annual Symposium of INCOSE, Melbourne, Australia, 2001.

[12] Raczynski, Stanislaw. Modelling and Simulation: The Computer Science of Illusion, Wiley: Chichester, 2006.

[13] Bloom, B. S. Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain, David McKay: New York, 1956.

[14] Pateli, Adamantia G. and George M. Giaglis. *A research framework for analysing eBusiness models*, European Journal of Information Systems, Vol. 13, No. 4, pp. 302–314, 2004.

[15] Ree, Peter. *Arena: Simulating E-Commerce Agent Strategies*, Master's thesis, MIT, May 2000.

[16] Parker, Craig M., John Liman, and Paula M.C. Swatman. *TRECS: Developing a Web-based e-Commerce Business Simulation*, 2nd Annual CollECTeR Conference on Electronic Commerce, 1998.

[17] Pastor, Oscar, Silvia Abrahão, and Joan Fons. *Building E-Commerce Applications from Object-Oriented Conceptual Models*, ACM SIGecom Exchanges, Vol. 2, No. 2, Spring 2001.

[18] Dorn, Dean S. *Simulation Games: One More Tool on the Pedagogical Shelf*, Teaching Sociology, Vol. 17, pp. 1–18, 1989.

[19] Guralnick, David and Christine Levy. *Putting the Education into Educational Simulations: Pedagogical Structures, Guidance and Feedback*, International Journal of Advanced Corporate Learning, Vol. 2, No. 1, 2009.

[20] Gilliot, Jean-Marie and Siegfried Rouvrais. *A Pedagogical Canvas for On-line Simulation-based Lessons*, World AACE Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 3727-3732, June 2004.

[21] Bouhadada, Tahar and Mohamed-Tayeb Laskri. *SAPIENCE: A Simulator of Interactive Pedagogical Activities for Distance Learning Environment*, 7th International Conference on Information Technology Based Higher Education and Training, 2006.

[22] Cao, Ming-Liang, Yi Li, and Josephine Csete. *A Pedagogical Study of an E-learning Case Computer Simulation for Enhancing Students' Learning Outcomes in Clothing Functional Design*, Conference of Textile Bioengineering and Informatics Society, 2010.

[23] Fishman, G. S. Monte Carlo: Concepts, Algorithms, and Applications, Springer: New York, 1995.