Developments in Business Simulation and Experiential Learning, Volume 31, 2004

CONTROLLING THE COMPLEXITY AND ORIENTING TARGET GROUPS BY A MODULAR, SERVER-BASED BUSINESS GAME SYSTEM

Fischer Helge
Georg-August-University of Goettingen
hfische1@uni-goettingen.de

Jackstein Michael
Georg-August-University of Goettingen

Biethahn Joerg
Georg-August-University of Goettingen

ABSTRACT

This paper describes our approach in the design and development of a non-commercial modular business game system. Problems like the control of the degree of complexity, level of abstraction and reuse of existing business games, and spatiotemporal restrictions are mentioned. The core concepts of modularization and solving these problems are presented.

This article introduces a framework programmed in Java and XML. On the one hand, this would open up the possibility to create a temporary wireless network for participants to use their own hardware for the input/output process. On the other hand, it could be used as a server based platform for online and distance business games.

KEYWORDS: business game; modularity; java and xml

INTRODUCTION

For the past 30 years business games have become an inherent part of departments of business sciences at German universities. There are several hundreds games with various orientations and degrees of complexity. They are mostly in-house developments, with a specialized focus on corporate divisions or areas of application, regarding the students' level of education.

The main reason for the use of business games, besides the transformation of theoretical to practical knowledge, is the promotion of holistic thinking and the exercise of decision-making. Using business games is primarily problematic because of the necessary time expenditure, organizational problems and particularly a subject-specific missing orientation (see Bronner & Kollmannsperger, 1998).

Therefore many lecturers lack business games which fit their teaching context. This makes in-house developments necessary. However, these cause high time and cost expenditure. For this reason, some lecturers find business games uninteresting as an active teaching method, and alternatively use case studies (see Fischer & Protil, 2003).

Since the nineties a steady change has taken place through the rapid development of Internet technologies in university teaching. An economical communication structure is now available (see Mohsen, 2002). Moreover, the latest programming languages (e.g. Java in connection with XML) offer the possibility of designing platform-independent applications with well-defined and standardized interfaces.

For the solution of the afore-mentioned hindrances, a server based, modular business game system offers its services. Technical orientations can be achieved with an individual choice or through the development of components containing functional areas of a company.

With regard to educational points of view, modules can be developed and used according to the knowledge level of the game participants. Moreover, a system which is programmed in Java can be used on a great variety of hardware platforms and operating systems as a client- or as a server-based Internet application. Thus, it is possible to build an arbitrary local *business-game-network* for the duration of the seminar.

In standardized interfaces, the imbedding of standard software is also possible, in addition to the use of Java-applets via XML for in- and output processing.

PROBLEM DEFINITION

The games treated in this paper are business management, computer based games whose field of application is the learning and orienting education.

A business game can be characterized by its degree of complexity. On the amount of the large number of variables and relations among these, one must attempt to illustrate the reality in as much detail as possible. Unfortunately, both the model and the decision-making-process, in a realistic environment, are connected to a high complexity. To not

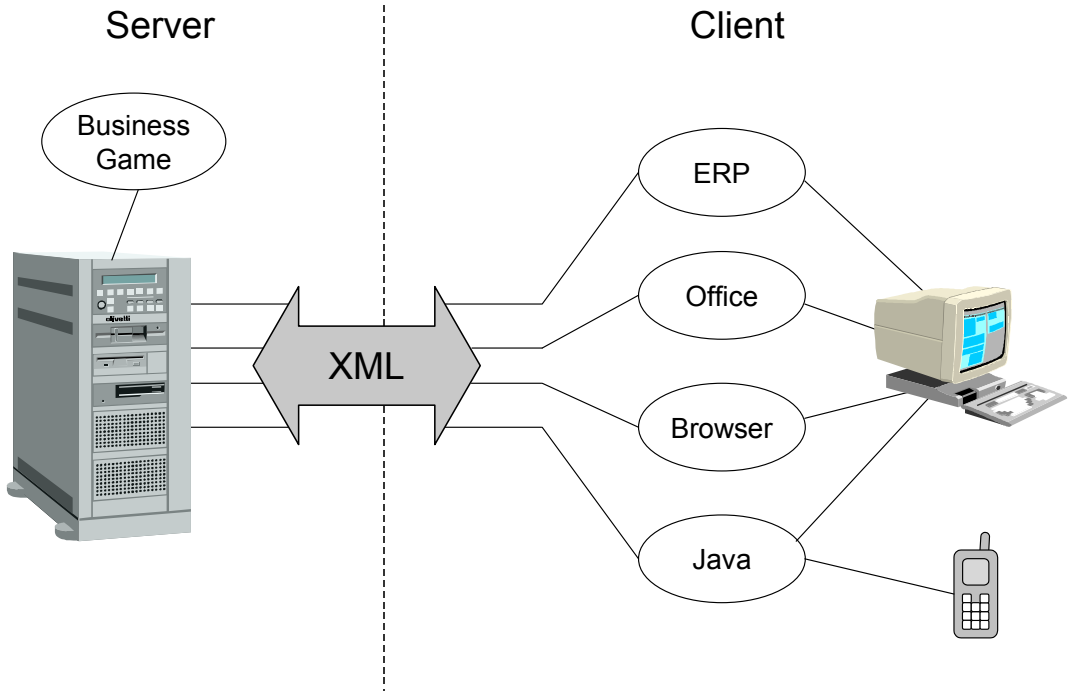

FIGURE 1 Client-server-based, local business game network

overtax game participants with a an excessive level of difficulty, the degree of complexity must be adjusted. This leads to a deterioration of the realism. In this context, one recognizes a *complexity dilemma* (see Vagt, 1983).

The designer of a business game must find an intermediate course between an adequately complex yet still playable simulation. A flexible control of the degree of complexity is usually only accomplished with the control of firmly predefined variables. This means that the game manager has the possibility of activating or deactivating these variables, but the algorithms used for calculation remain fixed. However, one should question whether such a control satisfies the claims or whether a more extensive solution is necessary.

The scale of a business game can be controlled by adding or removing decision variables, but their orientation is usually rigid. This means that the majority of business games are united through a firm focus or a specification on certain corporate divisions. Business games often show a similar structure or a comparable basic framework. This is valid both with regard to content and formal points of view. There are often similar areas, such as procurement, accountancy and sales modeled in simulated companies. The business games are, moreover alike in processing formal construction regarding game definition, input, output and information processing. It would be of advisable to reuse components of existing business games.

Although such similar structures exist, it is difficult to adapt existing solutions to one's own targets in the case of a conventional business game. A lecturer cannot take on other games if they fail to directly meet his special requirements. A development or a variation is only possible on the program code level in conventional business games by a direct intervention. The teaching content therefore, is adapted to the standard business game, but the business game is not adapted - though it would be meaningful for didactical purposes - to the teaching. Since business games are not isolated, but rather integrated into the teaching curriculum, the failure to adapt business games to teaching curricula is a great deficit of the last decades.

Through the development of the Information Technologies more complex and extensive models could be simulated by computer based business games. Nowadays it is possible for every almost little computer to simulate business game models in high speed processing. A hardcopy in- and output wastes not only environmental resources, but also ties the game participants to a certain place. Decisions must be handed in and result sheets must be collected. Besides the educational support the game manager has to control the logistics of the information flow. That means, for instance that he has to organize the distribution and collection, and provide for the data input into the computer system. This leads to restrictions in usability with large user groups. Shifting the input of the decisions for the game participants directly into the simulation computer, for use as a terminal, will cause spatiotemporal restrictions.

The use of online business games, which have been propagated since the spread of Internet Technology decreases these restrictions and permits the use of mass business games (see Klabbers, 2001). However, the blind confidence in the new technologies holds quite a number of dangers. A business game exists not only from a computer simulation model, but also from the interaction of the game participants and the game manager with one other. The manager vitalizes the

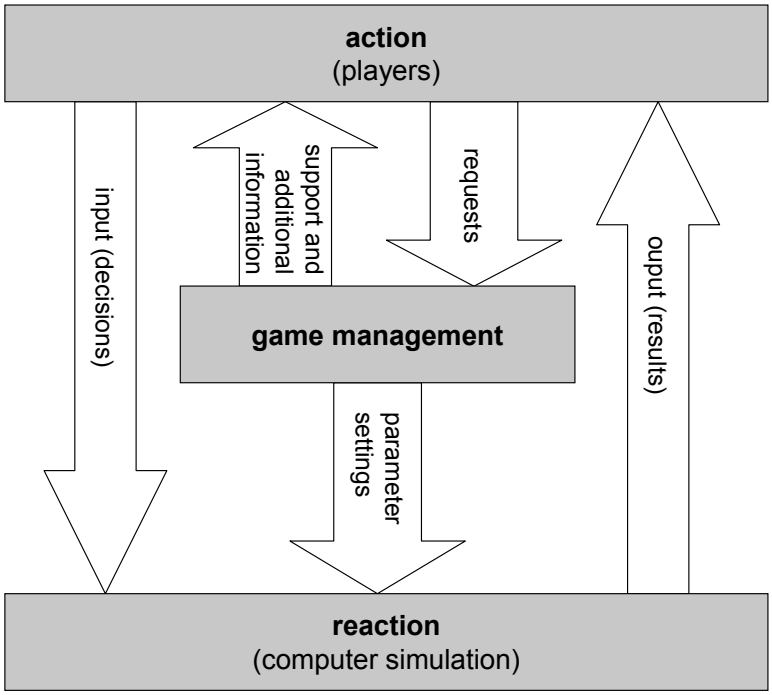

FIGURE 2 Structure of a business game (see Rohn, 1964)

business game by letting virtual surroundings arise in the imaginations of the game participants.

In distance games another reference to the game arises. The participants often do not leave the playful and entertainment level. The scientific and educational claim, therefore, is not satisfied. These business games are hardly different from common economy simulations of the entertainment industry. The promotion of team ability is one of the most important aspects and reasons for using business games. This aim is difficult to maintain with online distance games. Game participants may learn about how to cooperate over the Internet, but direct human contact and interaction at work is nevertheless the most important thing, even in Information Age.

To summarize, it is necessary that the balance between the use of online resources and personal contact receive attention. This is the only way how online business games could be justified from educational points of view.

SOLUTIONS THROUGH A MODULAR APPROACH

The following section represents a development approach. The problems described in the section above and limited use of business games can be decreased. Our research group, which deals with the programming of the modular business game system, will soon complete a first prototype; at present no empirical knowledge in usage exists yet. For this reason only a short description of the implementation of this prototype follows.

The business games used at the University of Goettingen are too inflexible in the control of the complexity and the orientation of the target group. The idea for the development of a modular business game has arisen from this fact. A system should be created, which allows one to switch several divisions of a company on and off. New and self-developed modules, containing own algorithms, should be added or existing ones should be overwritten. For the better integration into existing data processing programs, like Microsoft Office or ERP systems, interfaces should be tied together based on open standards. In addition, usability via Internet or communication with TCP/IP-support should be required.

For the development of a business game system, the principle of modularization could be used. The goal is to reduce the complexity. To keep the construction or editing as simple as possible, a complete system is divided into pieces during the modeling process. These pieces should be modules with a clear and self-contained structure.

An interface oriented perception should be considered for the development. A relatively independent module should be generated through the design of the module interfaces (see Biethahn, 1997). There are some restrictions if the arrangement of the modules is star-shaped.

The modules could only influence other modules indirectly through the *system kernel*. It must be avoided that any module influences another directly. This should be ensured both with respect to the modeling logic and the programming. This would mean, with respect to a business game system that adding a special corporate division would not have direct influence on the functionality of other existing modules. The removal of a module would not mean a restriction of the functioning ability of other corporate divisions. This is an indispensable premise for system stability.

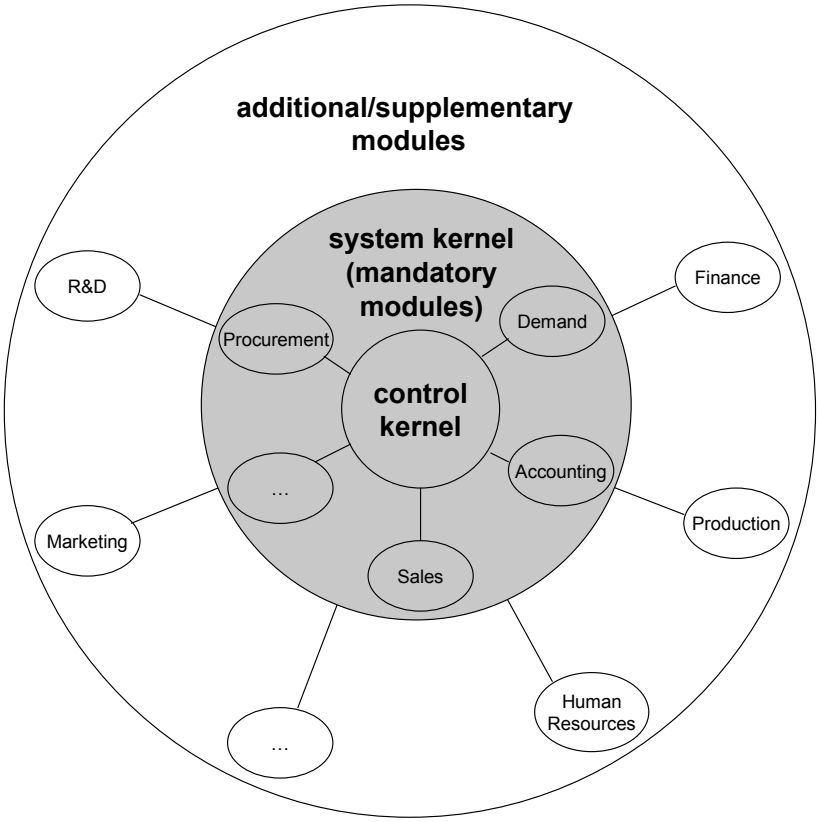

FIGURE 3 Processing component: star-shaped arrangement of the modules

To be able to ensure the run ability of a business game system, a *minimal system* must be identified. All corporate divisions necessary for the course of business activities are included. They are the so-called *mandatory modules*. In addition it contains the *control kernel* which regulates the activation of the modules. The *mandatory modules* could not be deactivated, but could be replaced by in-house developments.

The *additional* or *supplementary modules* outside the *system kernel* could be added or removed. The game manager could decide about their importance and replace them also during the course of the game with similar modules of different degrees of abstraction or complexity.

PROTOTYPICAL IMPLEMENTATION

For the prototypical implementation we have selected the combination of Java and XML (for further information about Java and XML, see McLaughlin, 2002).

Java offers many advantages as an object-oriented programming language. Its major fields of application are Distributed Systems, such as one the Internet or local networks. The principle of encapsulation is one of the generally-known advantages of the programming language Java. For instance being able to hide the data and algorithms and protect them from access by other modules.

Considering the idea of flexible use in networks with different hardware, e.g. the construction of a network, which consists of laptops, PDAs and cell phones, the most convincing arguments are portability and standardized interfaces. Since Java contains no system-dependent elements, Java programs are portable on every hardware device. Java supports applications which use resources distributed in data networks. Therefore, standardized programming interfaces simplify communication over the Internet. In addition, client-server applications could be easily realized.

XML is suitable for the structured storage of the data of business games, such as balance sheets, indicators, game parameter etc. Since both data and tags are included in an XML document which contains just ASCII characters, it can be read by any tool, which is able to process text files. Thus, XML supports the exchange of documents and data across diverse hardware, operating systems and applications very well. It is a well defined, self-describing and simple text based format which can be used without problems in every common system environment. Therefore, it is also platform-independent, such as Java.

In the future, XML will always serve more applications as a language for the description of standardized interfaces. It is freely available and specified by the World Wide Web Consortium (W3C) as a standard. For this reason, it is mostly independent from any manufacturer.

The data of an XML document can be transformed into corresponding output formats for its representation, such as with the help of the transformation language XSLT (Extensible Stylesheet Language Transformation) into HTML files. ERP systems (e.g. SAP) or Office products (e.g. Excel) also offer interfaces for the processing of XML files.

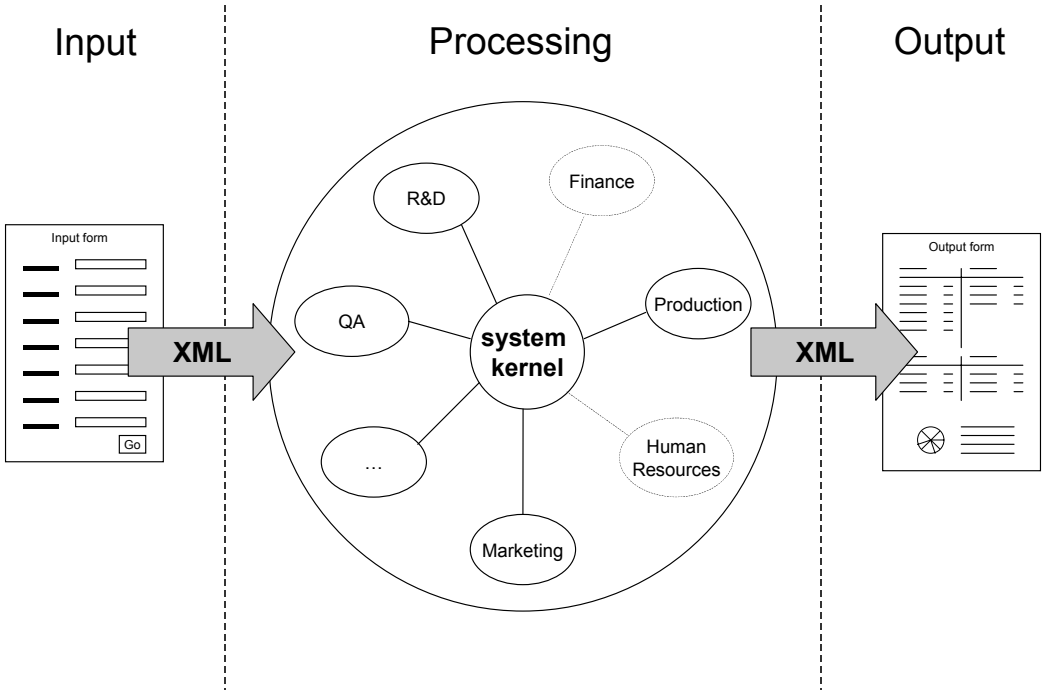

FIGURE 4 Information flow and interfaces

The data communication of the input and output components with the *processing component* is carried out by XML files containing the necessary data. Therefore, the processing of the information and the simulation of the business game model is completely detached from input and output. The results can be transformed into almost all output formats, such as HTML, PostScript, PDF, Excel, or into XML files. The necessary results for the game participants and the game manager would thus be available. The medium of this transformation is just as varied. It can be a paper or a screen visualization. A change of medium can moreover be avoided by direct digital transmission.

The possibilities of data input are also extensive. This could be realized via Web browser or a Java application on a PC, PDA or cell phone. The connection to standard software is also possible. For this reason the developers and users are free to create their individual input and output design, in accord with their own wishes and requirements.

The modules of the *processing component* are implemented in special Java classes or objects. Every module gets its own, internal processing data and parameters stored into private variables, which the other modules may not access. General data, such as balance sheets, parts lists or indicators for the economy, are stored in public variables. All modules may access and also change these variables.

The *control kernel* regulates data filing into XML structures (e.g. JDOM API, see Harold, 2003), as well as the linkage of the individual modules. It specifies which modules are included in the current simulation run of the business game.

Since the *processing component* is programmed completely in Java, it can be employed and executed in an arbitrary way. For example, the simulations can be carried out both on a mainframe and on a PDA. It is only a question of the interfaces of communication.

There are four major XML files for each game. Beside a short explanation of the workflow of the processing component, their importance will be resumed.

- definitions.xml: The definitions file has to be prepared by the game manager. It must exist before the beginning of the game. This file contains a list of the chosen modules, their primary processing sequence, the companies' names and the initial values (e.g. seed capital).

- variables.xml: The variables file is principally available and contains all existing variables. Among other things, these variables include common data like the balance sheets, profit and loss accounts and parts lists as well as input-/output and modules' own data.

- game.xml: This game file is generated from the two files above at the compilation of a new game. It contains information about the activated modules, the participating companies, all period results at that time and all existing variables. Only the variables used by an activated module are filled with values.

- input.xml: The input file is a subset of the game file. In contrast to this file, it contains only the input variables. The benefit of this separation is the possibility of an independent data input by the participants.

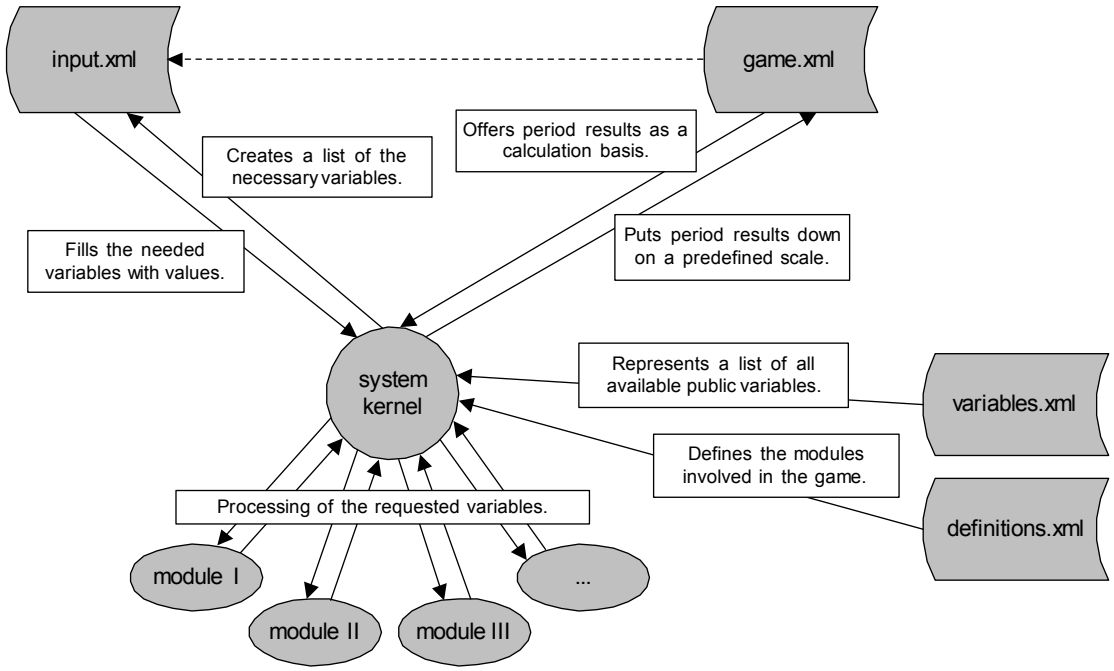

FIGURE 5 Workflow of the processing component

A new business game with a game file (see game.xml) and an input file (see input.xml) is generated by the kernel. For this purpose, a game definition (see definitions.xml) and a list of common data variables (see variables.xml) must be chosen. After this generation, the kernel identifies the actual game and period by a specified game file. The modules request decisions from the participants and period results stored in common data variables for processing. The kernel creates a list of the necessary variables (see input.xml) for the input.

In the simulation process, the modules are executed for each company in primary processing sequence. If some modules manipulate the same variables (e.g. a decrease of the liquid assets) a dynamic adjustment will be executed by an iterative loop. The calculated results are stored in the game file. The processing is finished by the generation of a new input file for the next period. This file could be initialized with default values like final values from the previous period.

The results could be edited in an adequate format for the participants.

CONCLUSION AND PERSPECTIVES

The development of a modular, server based business game system has been introduced in this article. Solutions and essential features of the prototypical implementation of a game system followed the representation of select problem definitions such as the *complexity dilemma*, inflexible target group orientation and infrastructural restrictions.

After the completion of the prototype divers ranges of application arise.

The system should be tested and then established in the teaching curriculum by primary development of first modules and the porting of the algorithms of an existing business game.

Moreover, the use of new communication structures is planned. Therefore, in addition to the Internet, a local wireless network should be build via Bluetooth. The game participants could perform the data in- and output on their own hardware, such as cell phones, PDAs or notebooks which support Bluetooth.

Another topic is the support of business start-ups and entrepreneurship. It can be completed meaningfully, using a modular business game system. Thus, it should be examined, whether a modification of the module quantity or the degree of abstraction during the runtime can simulate business start-ups. Is it possible to represent the scenario of the expansion of a back-room company to a global player with a steady-growing complexity? A meaningful objective would be to develop and combine such a scenario with special seminars for business start-ups.

REFERENCES

Biethahn, Mucksch & Ruf (1997*)* "Ganzheitliches Informationsmanagement, Band II: Entwicklungsmanagement", Munich / Vienna, p .319-324

McLaughlin (2002) "Java and XML, 2nd Edition: Solution of Real-World Problems", Beijing: O'Reilly

Klabbers (2001) "Framing Internet Games: Potentials and Limitations" In: Loebe & Severin (editor): "Planspiele im Internet: Konzepte und Praxisbeispiele fuer den Einsatz in Aus- und Weiterbildung" *Schriftenreihe der beruflichen Fortbildungszentren der Bayrischen Wirtschaft (Wirtschaft und Weiterbildung)*, Volume Twenty-Six, Bielefeld, p. 13-39

Fischer & Protil (2003) "Utilização de Simuladores Empresariais no Ensino de Ciências Sociais Aplicadas: Um Estudo na República Federal da Alemanha." *Proceeding: XXVII ENANPAD 2003 (published on CD-ROM)*, Atibaia / Sao Paulo, 2003

Harold (2003) "Processing XML with Java : a guide to SAX, DOM, JDOM, JAXP, and TrAX", Boston: Addison-Wesley

Mohsen (2002) “Internetbasierte Lehr-/ Lernmethoden fuer die wirtschaftswissenschaftliche Hochschulausbildung”, Goettingen, p. 54-73

Rohn (1964) “Fuehrungsentscheidungen im Unternehmensplanspiel”, Essen, p. 12

Vagt (1983) “Planspiel – Konfliktsimulation und soziales Lernen”, Heidelberg, p. 23 et sqq.

Bronner & Kollmannsperger (1998) “Planspiele als hochschuldidaktische Lehrmethode. Ergebnisse einer Umfrage.” *Wirtschaftswissenschaftliches Studium*, 8 (1998), p. 218-220