Journal of Internet Banking and Commerce

An open access Internet journal (http://www.arraydev.com/commerce/jibc/)

Journal of Internet Banking and Commerce, December 2010, vol. 15, no.3
(http://www.arraydev.com/commerce/jibc/)

Barriers And Profits Of Distance Education In Operations Research Based Decision Analysis

---

Tomasz Szapiro, PhD
Head of Division of Decision Analysis and Support in the Institute of Econometrics, Warsaw School of Economics, Poland
Postal Address: **Tomasz Szapiro, Szkoła Główna Handlowa, al. Niepodległości 164, 02-554 Warszawa, Poland**
Author's Personal/Organizational Website: **akson.sgh.waw.pl/~tszapiro/**
Email: **tszapiro (at) gmail.com**
Professor Tomasz Szapiro is the Head of Division of Decision Analysis and Support in the Institute of Econometrics, Warsaw School of Economics (WSE - SGH). He is Adjunct Professor at the Carlson School of Business, University of Minneapolis, Minnesota.

Przemysław Szufel, MSc
Assistant, Warsaw Division of Decision Analysis and Support in the Institute of Econometrics, Warsaw School of Economics (WSE - SGH), Poland
Postal Address: **Przemysław Szufel , Szkoła Główna Handlowa, al. Niepodległości 164, 02-554 Warszawa, Poland**
Author's Personal/Organizational Website: **akson.sgh.waw.pl/~pszufe/**
Email: **pszufe (at) gmail.com**
Przemysław Szufel is a PhD candidate and lecturer in Division of Decision Analysis and Support in the Institute of Econometrics, Warsaw School of Economics (WSE - SGH).

Abstract

Distance education and blended learning allow universities to lower costs, have more return on economies of scale, better control educational process and provide higher quality services for students. This process however requires specialized software to be used by a university.

The paper tackles issues related with this approach and presents guidelines for construction of a framework for creating web-based education support software in quantitative area – the meta-framework that allows for a rapid construction of web-based educational decision analysis software.

The framework design in paper covers three aspect: (1) information system modeling, (2) decision problem modeling and (3) teaching process modeling. The information system modeling includes system architecture, servers, programming languages and technical design. The decision problem modeling provides means to present an abstract decision problem (and more generally - a problem where quantitative methods can be used) and to model interaction between software and decision maker (i.e. a student learning a decision algorithm). Teaching process modeling includes support for linear teaching algorithm in the decision making process, further this can be extended into non-linear and hybrid teaching algorithms.

As a proof of concept a working framework prototype named Combine! will be presented. The software's usefulness was evaluated through a survey carried out on 234 students at Warsaw School of Economics.

Keywords: **e-learning; decision support; operations research; distance education**

© Tomasz Szapiro, Przemysław Szufel, 2010

1. Introduction

The goal of the paper is threefold. Firstly, the goal is to identify barriers in distance support of education in the field of operations research and decision analysis. Secondly, the goal is to propose design aspects of an integrated software framework for construction of decision support systems (DSS) that will allow to overcome this limitation. Thirdly, the goal is to verify the usefulness of the presented approach as survey carried out among students.

Distance education and blended learning allows universities to lower costs, have more return on economies of scale, better control educational process and provide higher quality services for students. This process however requires specialized software to be used by university.

Popular Learning Management Systems (LMS) (e.g. open source – Moodle, commercial – Blackboard, WebCT) enable to support students in several layers of educational process including organizing teaching material, exchange of information and providing self-study materials. The LMS systems are oriented at presenting teaching materials as set of files (PDF, HTML, multimedia) that are download or viewed by students on server. Usually educational materials are custom organized along classroom meetings. An electronic education is interactive through providing discussion forums, homework, online grading mechanisms, chats and lecture broadcasts in real-time.

However in the education of quantitative methods in economics – it is important to teach the mathematical problem modeling as well as problem solving methods. Quantitative problems are usually solved with specific software tools (e.g. decision analysis software). That means that electronic teaching of quantitative methods requires to provide a problem modeling software that would allow to illustrate taught materials. Unfortunately LMS-type software supports only the way that educational materials can be distributed to students but does not provide any framework for allowing to create an interactive educational software.

The paper tackles issues related with this approach and presents guidelines for construction of a framework for creating web-based education support software in quantitative area – the framework that allows for a rapid construction of web-based educational decision analysis software.

The paper consists of three parts. Firstly, we present some information on distance education and identify barriers in supporting distance education process in the field of quantitative methods and specifically operations research and decision analysis methods. Secondly, we present design aspects of a framework for supporting education processes in the field of operations research and decision analysis. Thirdly, a proof-of-concept - a working system is presented and results of a satisfaction survey carried out among students are described. The paper is ended by conclusions and references.

2. Distance education – barriers in operations research teaching

The goal of this section is to present barriers in distance education of quantitative methods and especially operations research. The first part focuses on the definition of the distance education while the second part focuses on barriers. In the section three we will present a methodology that allows to overcome these barriers, while in section four we evaluate a working prototype that is constructed along instructions in this paper.

The OECD (2004) classification uses the following degrees in defining teaching process:
1. Enriched with online content but without limiting the number of traditional classes.
2. Network dependent – students need to have a network access while performing key tasks. The number of traditional classes is not limited.
3. Mixed mode – an amount of traditional classes was decreased in favour of online classes (blended learning).
4. Fully online – students do not attend classes but are being thought by a software system.

E-learning has a broader meaning than distance education. Bates (2004) defines distance learning in a narrow sense and proposes the following three types for e-learning: 1. Enriched with online content but without limiting the number of traditional classes. 2. Mixed mode (blended learning, b learning) – the amount of classes is reduced. 3. Distance learning – only online classes exist. Another classification by Horton (2000) splits the trainings into traditional, fully online; fully based on electronics materials given to students off-line and two hybrid types: (1) traditional learning mixed with online educational materials and (2) off-line mixed with online educational materials.

In the paper we will define distance learning in a broader meaning – as any teaching being done through a computer network. This includes enriched mode, mixed mode as well as online-only approaches. The presented tools can support teachers in all above scenarios.

After defining meaning of distance education let's move to barriers in its development. As noted in the introduction the present e-learning solution focus on means of providing of communication framework rather than providing software for adding interactivity for online teaching materials.

However teaching operations research and decision analysis requires students to try specialized analytical tools. The educational software should allow to combine the teaching functionality with the problem solving functionality.

Obviously teaching of different areas of knowledge requires different toolsets. In the paper we focus on designing a decision analysis toolset for teaching quantitative methods and specifically operational research and decision analysis.

Barriers in introduction of distance education can be of financial (insufficient funds) or technical (insufficient knowledge) nature. In the paper we focus on the technical side of introduction of interactive distance education. We can identify three technical barriers in creating of interactive distance education materials: ability, complexity and flexibility barrier.

The *ability barrier* means that creating interactive educational materials requires the creator to possess a large technical knowledge. Insufficient technical knowledge results in poor software design, exceeded project budget or even an abandoned project (see Brooks, 1995).

The *complexity barrier* covers the problems with system upkeep and building new functionalities. This is specifically a problem when a complicated computational code is written with a general-purpose language rather than a specialized computational language like Matlab.

The third *flexibility barrier* is related to code maintenance. After some time even minor changes can be very complicated to implement. A reason for that could be a poor design methodology and – again – the complexity of computational costs.

In the next section a framework will be presented that will allow to overcome the barriers presented.

3. Framework for teaching operations research

The framework design covers three aspect: (1) *teaching process modeling* (2) *decision problem modeling* and (3) *information system modeling*. Teaching process modeling includes support for linear teaching algorithm (Skinner, 1958) in the decision making process, further this can be extended into non-linear and hybrid teaching algorithms. The decision problem modeling provides means to present an abstract decision problem (and more generally - a problem where quantitative methods can be used) and to model interaction between software and decision maker (i.e. a student learning a decision algorithm). The information system modeling includes system architecture, servers, programming languages and technical design.

3.1. Teaching process modeling

The goal of this section is to present basics of learning process modeling. Presented approaches can be applied in creation of an interactive learning software.

Learning process models where first created as a way to increase efficiency of learning processes. The machine learning concept was first created by Pressey (1927) who constructed a learning machine. During the learning process student was using the question number in a multiply criteria question form. When the answer was correct the machine would move to the next question. Otherwise the student would answer the same question as long as the answer is not connect. Pressy (1927) has proved that his machine actually teach students.

The learning process models are a continuation of works of Skinner (1958). Skinner (1958) proposed a linear education process model. The model was a result of research on machine assisted teaching. Skinner stated the following postulates on the learning process design: teaching supported by the fear of punishment does not lead to efficient results, better results can be obtained by teaching based on students cognitive activity; the teaching material should be divided into small independent parts; each proper student's answer should be immediately confirmed; particular steps should not be too difficult as a feeling of success motivates a student to learn.

As a learning process that fulfils the above arguments Skinner (1958) proposed program teaching and specifically a linear teaching program. The goals for program teaching are the following: ensuring a continuous interaction between a student and a software; allowing moving forward to the further teaching material after understanding the current material; presenting pieces of teaching material that are easy enough for a student to learn; helping student to find a right answer by properly setting up teaching program; keeping up student's interest by immediate providing of a return response after every answer, see Kupisiewicz (1975).

In order to fulfil the requirements above Skinner (1958) proposed a linear teaching program. The linear teaching program is based on the following rules: (1) the rule of small steps – teaching material should be divided into small pieces – teaching frames; (2) the rule of an immediate confirmation of an answer – immediately after an answer to a question is being given by a student it is being compared with a proper answer; (3) the rule of an individual learning speed – a student defines the speed that he wants to learn; (4) the rule of gradating of difficulty level – the number of hints is being gradually decreased; (5); the rule different information sources – each key information should be repeated in different context in different parts of an educational material; (6) the rule of standardized instrumental learning process – teaching information is presented in a particular order, in each step an answer is constructed and immediately evaluated, the knowledge is presented in small steps, see Kupisiewicz (1975).

The linear teaching program assumes that every student goes through the same material and fills in answer for the same questions. This approach met with criticism. Crowder (1961) pointed out that division of the teaching contents on equal small parts is artificial and that information presented to particular students should be individualized depending on their abilities. The result of this critics is a branching teaching program. The main differences between linear and branching teaching program include: (1) the teaching material is divided into small portions that fit subchapters in a handbook rather than equally sized frames; (2) after an improper answer to a question a student is presented with an additional teaching material that has a content adjusted to student's abilities; (3) an answer is chosen from available alternatives rather than written; (4) student has a possibility to move to a previous step.

Kae et al. (1968) point out that the linear teaching program and branched teaching program can be combined and thus a mixed teaching program can be created. In the design of a mixed teaching program it is assumed that: (1) the teaching process is too complicated to base it on just choosing a single answer from a list – therefore textual answers can be introduced; (2) an immediate assessment of an answer – however a student can decide on his own whether he wants to see additional teaching material or not; (3) both the speed and contents being thought are individualize – weaker students can use hints in critical moments of learning process.

The approaches presented above on teaching process modeling can be applied directly in software design. The algorithm can be implemented as a computer algorithm that manages the teaching process. As our goal is to define a teaching framework in the next subsection we make some remarks on decision process modeling to combine later this two approaches in a software framework.

3.2. Decision problem modeling

We assume that the decision problem can be presented as a finite set of variables, arrays and matrices containing either numbers or textual data. Further we assume that problem decision problem can be solved by carrying out a number of (possibly interactive) steps on a decision model. An example of such approach could be a multi-criteria decision modeling (MCDM). In the MCDM it is assumed that the decision maker has to evaluate several pay-offs need to be analysed for each feasible decision. In a MCDM it is assumed that several decision variables exist, where x_i is i-th decision variable. Thus an alternative can be presented as a vector $\mathbf{x}=[x_1 \ldots\ldots x_n]$ in $\mathbf{R}^n$ decision space. We define a set of feasible decisions $\mathbf{X}\subset\mathbf{R}^n$. Each decision x has several outcomes that can be calculated with an array of functions $f_j(x)$ where $f_j(x)$ is an outcome of option x with the j-th criterion, i=1,…,k. Thus $f_j:\mathbf{X}\rightarrow\mathbf{R}$ - the j-th objective function can be presented as a vector multi-objective function $f:\mathbf{X}\rightarrow\mathbf{R}^k$. Next we define the domination relation as $\forall j\in\{1,\ldots,k\}\ f_j(x_1) \leq f_j(x2) \wedge \exists j\in\{1,\ldots,k\}\ f_j(x_1) < f_j(x_2) \Leftrightarrow f(x_2)$ dominates $f(x_1)$ The problem in MCDM is to find the most preferred option, i.e. the option whose outcome dominates all other feasible outcomes. Usually the solution to such defined problem does not exist. Thus when dealing with the multi-criteria decision problem we try to find solutions that are not dominated by any other feasible alternatives – i.e. we try to find a set of Pareto-optimal solutions. In case of a linear decision model set of feasible decisions can be described as $\mathbf{Ax} < \mathbf{b}$ and vector objective function can be represented as $\mathbf{F}(\mathbf{x})=\mathbf{Cx}$. In this case, structural elements of linear multi-criteria optimisation can be described as set $<\mathbf{A},\mathbf{b},\mathbf{C}>$. The assumption of linearity can be dropped for the constraints and replaced by assumption that the set of feasible solutions is concave.

One can see that a MCDM problem can be described as a set of variables, vectors and matrices. We will call this representation *a standard form of a decision problem.* Standard form can be applied to various classes of decision problems, also, after modifications to problems of group decision making (see Szufel, Wojewnik, 2008).

3.3. Information system modeling

When designing an architecture of a web based educational decision support platform it is natural to base the implementation on the Model-View-Controller (MVC) design paradigm. The MVC paradigm assumes that software functionalities are split into three independent parts: Model, View and Controller (see Gammal et al., 1995). The model contains the representation of information that is processed by the application. The view is responsible for rendering graphical user interface. The controller processes user action generated by the view and invokes changes on the data represented by the model.

On the basis of previously presented postulates we propose that a universal platform is built along the MVC architecture. We propose that an DSS implemented within the framework is formed of the following three parts: (1) decision model definition file; (2) independent view files (we shall define them as view frames) responsible for each phase of the decision process; (3) files with code responsible for performing computations and generating graphs to support the decision process.

We propose that a decision model is defined in a model definition file. The decision model definition should contain at least two parts: (1) the model of the decision process; (2) the model of the data required by the decision process (i.e. the standard form of a decision problem) that includes those variables, vectors and matrices required to form the decision problem.

The framework should be responsible for all aspects of data management: storing the data of a decision model (this can be achieved through one of the object persistence models e.g. object serialization to an embedded database), managing user accounts and storing the changes made to the model in the view layer.

The view layer in web-based systems is usually developed as set of HTML (or DHTML) files with server side processing provided by merging HTML with general programming code (e.g. PHP, Active Server Pages, Java Server Pages). In practice, this design structure usually means that the View code partially becomes responsible for data processing – which does not conform to the MVC design paradigm. Therefore we propose a different approach: (1) use the HTML to define the user interface layout, (2) implement dedicated tags (being processed on the server side) that are responsible for displaying various parts of the decision model as HTML/Javascript forms i.e. a tag inserted in an HTML document will result in the generation of an HTML form; (3) implement automated processing of forms generated in previous phase. Following these proposals will allow the DSS designer to focus exclusively on the layout of user interface, thereby leaving the burden of form processing and data persistence on the decision platform.

The controller functionality is, from the DSS designer's perspective, the most difficult part of any DSS implementation. Usually it is implemented by the same programming language as the entire DSS. As we noted earlier the main role of the controller layer of an DSS is to provide the decision support functionality. Implementing this functionality requires performing computations and generating graphs. Therefore the most appropriate tool for construction of the controller layer of an DSS is a specialised numerical computing language (e.g. Matlab, GNU Octave) rather than a general-purpose programming language. Implementation of numerical optimisation or graph generation is much easier in numerical computing languages as these are the tasks these languages are designed for. As the numerical computational languages are inappropriate for implementing the proposed framework – the framework's base must be implemented with a general-purpose programming language, thus the educational DSS framework is an example of a heterogeneous software system.

4. Educational decision analysis software in Warsaw School of Economics

The goal of this section is to present a proof of concept a working framework prototype named Combine! will be presented. The prototype software has been successfully used for 5 years in classes at Warsaw School of Economics for teaching decision analysis and multi-criteria optimisation. The software's usefulness was evaluated through a survey carried out on students. The results will be presented in the paper.

4.1. Software

We applied design guidelines presented in this articles and technology proposed in previous Section to develop a working prototype of an educational meta-DSS platform.

Our prototype was designed as an expansion for a web based multi-tier decision support system (run at Warsaw School of Economics) named Combine!. Combine! is a software framework supporting rapid construction of educational software. This system is used for teaching operations research, optimisation and decision analysis in the Warsaw School of Economics. The Combine! framework was presented in the Technical Reports of the Institute of Econometrics e.g. Gawroński et al. (2004), Koloch et al. (2007) as well as in conference papers e.g. Bielińska et al. (2001), Szufel (2004). The main goals in the implementation of the educational decision support system was to enable the users to implement their own analytical tools with minimal programming effort, that could be immediately deployed as a web application. To enable easy and independent expansion a modular plug-in architecture was implemented. New system functionalities are built into the core system as small independent plug-ins that can be provided by independent developers – contributors.

Building an educational DSS with Combine! requires from the developer a basic knowledge of HTML, XML and being familiar with a computational language – Matlab or Gnu Octave. DSS development does not require any knowledge of low-level programming. HTML is used to specify GUI layout, additional tags have been designed allow user to place vectors and matrices from standard form of the decision making model at any location in a view frame. All programming needed for advisor or mediator functionalities of a DSS can be done in GNU Octave. The same applies to graphs that can be generated by code written in GNU Octave and placed at the desired location in a view frame. User management, data persistence, HTTP request processing is done automatically by the prototype meta-DSS Combine! platform.

Building a new educational DSS with Combine! meta-platform requires undertaking the following steps: (1) defining standard form, (2) defining decision/teaching process model, (3) building graphical user interface, (4) programming application logic, (5) debugging and (6) deploying to the server. The definition of standard form and decision process is done in an XML file.

The DSS developer defines each variable, vector and matrix needed for the standard form (e.g. <attribute class="Double" name="A" dimensions="2" /> defines a two-dimensional matrix of real numbers named A). The definition of the decision process is also done in an XML file – the decision process is represented by a set of view frames representing different decision phases. The graphical user interface development is done with the HTML, additionally a set of Combine! tags is available (e.g. <ct:BeanTable tableName="A"/> will display an editable contents of matrix A – Combine! framework will take care displaying the matrix and storing the data after it has been changed by the user). The application logic can be programmed with Java or Octave numerical computation language. The Octave code can be placed either within an HTML file or in an external file – all variables defined in the standard form can be accessed and modified within the Octave code. After a DSS is implemented it needs to be tested and debugged. Combine! supports reporting errors in Octave code, what makes them easy to find. Deployment requires copying the DSS files into an appropriate folder in the production server – plug-in architecture will take care of displaying new DSS in the list of available DSSs.

4.2. Satisfaction survey

The goal of this section is to present the results of a survey conducted among students of Warsaw School of Economics after they participated in a on-campus course that was supported with distance education software – Combine!. The goal of the survey was to answer the following research hypothesises:
(H1) participation of an electronic education system increases attractiveness of a course
(H2) electronic learning should in the field of operations research should be used a support for a regular course (not standalone)

The tool for verification of the above hypothesis is a survey carried among students using the software framework presented in the paper. The research was carried out on a group of 234 students, who were studying fulltime (70,5%), part-time (10,3%) and MBA courses (23,2%). The research covered students from different courses in a two year period.

Firstly the respondents were asked about average time spend with the system. Subjects reported the average time of work supported by the system equal 162 min. (standard deviation 112 min). In order to compare real time of work with he declared one, subjects were asked to provide login for statistical purposes. 49% of population decided to provide their login. The average reported time was 7 minutes longer in the group who declared logins and 6 minutes shorter in the group who did not. However the analysis of their true activity revealed that time registered in Combine! logs is over half shorter and equals 71 minutes (standard deviation equals to 56 minutes).

This results (very high variance) show that users reveal extremely diversified attitude toward system assistance in learning. There are several possible reasons for difference in time given by respondents and real time spend with the software:

- respondents provided false information,
- some students worked in groups with the software and send only the final solution individually,
- some of the students could print out the educational materials, work with the homework off-line and only input into the software the final solution.

In the survey students were asked several questions regarding satisfaction with the educational software and the knowledge obtained in the learning process.
One question was about efficiency of the time spend with the software. Almost all respondents (92,6%) thought that working with a system was an efficient way of spending time (Table 1).

Table 1: *Question: "Was the time with software efficient"*

Type of studies	Definitely not	Not	Generally not	Generally yes	Yes	Definitely yes
Full-time	0%	3.1%	7.4%	32.7%	38.9%	17.9%
Part-time, extramural	0%	0.0%	0.0%	30.4%	56.5%	13.0%
MBA	0%	0.0%	0.0%	17.8%	60.0%	22.2%

Scale: Definitely not – 0, Not – 1, Generally not – 2, Generally yes – 3, Yes – 4, Definitely yes – 5

The students were asked about satisfaction from the course taken. Most of them decided that they would choose the classes again and they would be interested in using a similar software in a different course (Table 2). This means that introduction of an educational software platform increases attractiveness of a course.

Table 2: *Satisfaction from the course*

Type of studies	"Would you choose to attend these classes again" (average answer)	"Would you like to use a similar software at other courses" (average answer)
Full-time	3.74	3.80
Part-time, extramural	4.29	4.00
MBA	4.02	3.84

Scale: Definitely not – 0, Not – 1, Generally not – 2, Generally yes – 3, Yes – 4, Definitely yes – 5

On the other hand more students found lecture to be more useful in explaining educational material (Table 3). Therefore the best results can be obtained by combining distance learning with traditional lecture-based approach.

Table 3: *Question: "How these tools allowed you to understand the teaching material"*

Type of studies	Combine (average answer)	Lecture (average answer)
Full-time	3.61	4.07
Part-time, extramural	3.95	4.67
MBA	3.80	4.57

Scale: Definitely not – 0, Not – 1, Generally not – 2, Generally yes – 3, Yes – 4, Definitely yes – 5

The respondents were also asked about whether the knowledge obtained from using the software can be applied in practice. The most positive answers were obtained from MBA students (Table 4).

Table 4: *Question: “Does the abilities obtained by software usage can be applied in practice”*

Type of studies	Definitely not	Not	Generally not	Generally yes	Yes	Definitely yes
Full-time	0.6%	0.6%	28.8%	46.0%	19.0%	4.9%
Part-time, extramural	0.0%	0.0%	26.1%	43.5%	17.4%	13.0%
MBA	0.0%	0.0%	4.4%	20.0%	55.6%	20.0%

Scale: Definitely not – 0, Not – 1, Generally not – 2, Generally yes – 3, Yes – 4, Definitely yes – 5

The survey results show the usefulness of the presented approach. Most of the students were satisfied with the classes and decision support software implemented on the presented Combine! platform.

5. Conclusion

In the paper we have presented a new approach in supporting educational processes in the field of operations research and decision analysis.

We have identified technological barriers in construction of interactive educational decision support software (ability, complexity and flexibility). Next we proposed a modular design architecture allowing to overcame this barriers in a web-based educational decision support software. We proposed a guidelines on how to apply MVC design pattern to build a heterogeneous software platform for construction of educational software in the field of operations research. Designing the controller part as an XML file representing decision process model and code written in a computational language (e.g. Gnu Octave) allowed to define a framework for rapid creation of educational decision analysis software.

As a proof of concept we constructed a working software prototype named Combine! which was introduced into decision analysis classes at Warsaw School of Economics. The survey results show that software allowed to increase course’s attractiveness and allowed students to better understand decision theory.

Further work focuses on construction of new DSS software on the base of presented platform and integration with general e-learning systems (Moodle).

References

Bates T. (2004) The promise and myths of e-learning in post-secondary education, w: M. Castells (red.) The Network Society: A Cross-cultural Perspective, pp. 271-292, Edward Elgar Publishing

Bielińska E., D. Deręgowska, T. Szapiro, P. Szufel, M. Zagórski (2001). Multilayer decision analysis system (in Polish), post-conference materials – Proceedings of Conference “Preference modeling and Risk”, Ustroń, 2001

Brooks, Jr., F.P. (1995). The Mythical Man-Month: Essays on Software Engineering, Addison-Wesley, Reading

Crowder N.A. (1961) Intrisic and Extrinsic Programming, w: Programmed Learning and Computer-Based Instruction, New York, London

Felleisen M. (1990). On the expressive power of programming languages, Lecture Notes in Computer Science, 432, 134-151

Gammal E., R. Helm, R. Johnson, J. Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, New York

Gawroński P., V. Lubczenko, D. Pilawa, T. Szapiro, P.Szufel, M. Wujec (2004) Supporting Educational Process In Open Source Contribution Model (in polish) presented as internal research in Institute of Econometrics, Warsaw School of Economics

Horton W. (2000) Designing Web-Based Training, Wiley

Kay H., Dodd B., Sime M. (1968) Teaching Machines and Programmed Instruction, Suffolk

Koloch G., P. Szufel, T.Szapiro, P.Wojewnik (2007). Remote support of decision analysis support in heterogenic information systems (in Polish) Technical Report TR03/E/0003/07, Institute of Econometrics, Warsaw School of Economics

Kupisiewicz Cz. (1975) Teaching programming methods (in Polish), Warsaw

OECD (2004) E-Learning Case Studies in Post-secondary Education and Training

Pressey S.L., (1927) A machine for automatic teaching of drill material, School and Society 23(586)

Skinner B. F., Teaching Machines, Science 128(3330), 1958

Szufel P. (2004). Technological aspects of remote support of operational research education (in Polish) in: J. Hołubiec (ed.) System Analysis in Finance and Management, vol. 6, pp. 201 213, Exit, Warsaw

Szufel. P., P. Wojewnik (2008) Universal software platform for construction of web-based negotiation support systems, in: J. Climaco, G. Kersten, J. P. Costa (red.) Group Decision and Negotiation 2008, Proceedings – Extended abstracts, Coimbra