

RTU
ZINĀTNISKIE
RAKSTI

SCIENTIFIC
PROCEEDINGS
OF RIGA
TECHNICAL
UNIVERSITY



DATORZINĀTNE

COMPUTER SCIENCE

SĒRIJA 5
SĒJUMS 22

RĪGA 2005



ISSN 1407-7493

RĪGAS TEHNISKĀS UNIVERSITĀTES
ZINĀTNISKIE RAKSTI

SCIENTIFIC PROCEEDINGS
OF RIGA TECHNICAL UNIVERSITY

5. SĒRIJA

DATORZINĀTNE

COMPUTER SCIENCE

LIETIŠKĀS DATORSISTĒMAS
APPLIED COMPUTER SYSTEMS

22.SĒJUMS

IZDEVNIECĪBA "RTU", RĪGA – 2005

**Galvenais redaktors:
Editor in Chief:**

Profesors J.Grundspeņķis (Rīgas Tehniskā universitāte, Latvija)

**Redkolēģija:
Editorial Board:**

Profesors J.Bārzdiņš (Latvijas universitāte, Latvija)
Profesors J.Bubenko, jr. (Stokholmas universitāte, Zviedrija)
Profesors A.Čaplinskā (Matemātikas un Informātikas institūts, Lietuva)
Profesore H.-M.Haava (Konkordijas universitāte, Igaunija)
Profesors A. Kalniņš (Latvijas universitāte, Latvija)
Profesors H.Lingers (Monašas universitāte, Austrālija)
Profesors L.Novickis (Rīgas Tehniskā universitāte, Latvija)
Profesors J.Osis (Rīgas Tehniskā universitāte, Latvija)
Profesore V.Vojtkovska (Boises universitāte, ASV)

**Editor:
Redaktore:**

Dr. O. Ņikiforova (Rīgas Tehniskā universitāte, Latvija)

**Redkolēģijas adrese:
Editorial Board Address:**

Rīgas Tehniskā universitāte
Meža iela 1/3
LV 1048 Rīga
Latvia

Phone : +371 708 95 06

AN INTELLIGENT TUTORING SYSTEM FOR MINIMAX ALGORITHM

A.Anohina

Intelligent tutoring system, Minimax algorithm

1. Introduction

Nowadays knowledge acquisition plays a very important role because of the rapid growth of modern technologies that require new knowledge and skills. People choose computer-based learning, when they need knowledge, but do not have time to visit some educational institution. Learning via computer has trend to be more effective, if learning software demonstrates some intelligent capabilities as human-teacher does it. These capabilities include the presentation of tasks and learning materials corresponding to the learner's current state of knowledge, allowing eliminating learner's misconceptions and helping to evolve his/her state of mind, the explanation of the same concept by means of different words, the adaptation of the structure and the pace of the learning process to the psychological and cognitive needs of a learner, and etc. The research in this area led to the appearance of intelligent tutoring systems. Intelligent tutoring systems are computer software that is based on artificial intelligence techniques and simulates a human-teacher. Such systems like a human-teacher store and manipulate knowledge about domain, teaching strategies, and learner, and carry out learning process grounded on some learning theory. Research has been making in the field of intelligent tutoring systems for last three decades and during this time a variety of intelligent tutoring systems has been implemented for different areas and disciplines.

The development of an intelligent tutoring system is the application of theoretical results get by the author during her work on PhD. thesis at the Riga Technical University. The intelligent tutoring system is intended for the course "Foundations of artificial intelligence". The subject of artificial intelligence is not simple for understanding by itself. Furthermore, lectures on this subject at the Riga Technical University are given for the group, where individual attention to a learner cannot be given. As course consists of many topics and the department has not big experience in the development of intelligent tutoring systems, only one topic included in the exam on artificial intelligence was chosen for the implementation of the prototype. This topic is the Minimax algorithm.

The remainder of this paper is organized as follows. Section 2 presents some related work in the field of the development of intelligent tutoring systems. The main parts of architecture of an intelligent tutoring system, their relations and purposes are introduced in Section 3. Section 4 describes the Minimax algorithm in details, as well as some concepts about the implementation of two-player games are discussed. Section 5 presents some ideas about the architecture of the intelligent tutoring system for Minimax algorithm. Finally, Section 6 draws some conclusions and presents some directions for future work related to the further development of the system.

2. Related work

While developing the intelligent tutoring system for Minimax algorithm, many existent intelligent tutoring systems have been studied. Such systems have been developed for different areas and disciplines. Andes [1] is an intelligent tutoring system for classical Newtonian physics. Passive voice tutor [2] is an intelligent multimedia tutoring system for the passive voice of the English grammar.

Many systems have been developed for the area of medicine. Some examples are CIRCSIM-Tutor [3], which is a natural language dialogue based interactive intelligent tutoring system for the domain of cardiovascular physiology, or Ines [4], which is an agent-based intelligent tutoring system for the task of the nurse training or Web based intelligent tutoring system in dermatopathology [5].

The field of the computer science also has not been forgotten. ADIS system [6] is intended to help with teaching of data structures. The “Java™ Intelligent Tutoring System” (JITS) [7] provides help for students in their first programming course in Java™ at the College and University level. WADIES [8] is Web- and agent-based adaptive intelligent educational system for teaching compilers, but SQL-Tutor [9] is a computer based teaching system that has been developed to help the learners in learning of structured query language.

Regardless such rich variety of intelligent tutoring systems only one system related to the field of artificial intelligence was found during the investigation. This system has been developed at the Department of Artificial Intelligence of Kyushu Institute of Technology of Japan [10] and its main purpose is to help students to learn search algorithms- one of the basic topics of the course of artificial intelligence- through exercises. The developers’ main idea was not only to teach the characteristics of the search algorithms, but also provide deep understanding of each step of algorithms and its meanings. For this purpose three views are available to students: the data structure view showing the algorithm-operated data, the future view demonstrating the algorithm’s characteristics and the problem view presenting the problem under solving. Using these views students can understand the meaning of data operations and the relationship between the problem constraints and the execution process of the algorithm.

3. The architecture of intelligent tutoring system

The field of intelligent tutoring systems has been researching over 30 years. During this time the main parts of the architecture of such systems have been defined.

As the basic kinds of knowledge stored in an intelligent tutoring system are domain knowledge, knowledge about teaching strategies, and knowledge about learner the principal components of the system’s architecture are the expert module, the tutoring module, and the student diagnosis module accordingly. These modules are referred to as “Traditional Trinity” of intelligent tutoring systems in [11].

The student diagnosis module carries out the student diagnosis process that operates on the student model, allowing identifying the student’s current understanding of the subject. The student model serves as a basis for tailoring the learning process to the needs of the learner. It contains information about the current state of knowledge of a particular learner, that is, what he/she has learned, what mistakes he/she has done, what misconceptions he/she has, and etc., as well as this model additionally can store information about learner’s psychological features and his/her past experience.

The tutoring module holds teaching strategies and instructions needed to implement the learning process. This module can also be referred as curriculum and instruction module [11, 12] or the pedagogical module [5, 13].

The expert module stores the knowledge the system is teaching. It represents skills and expertise that an expert in a particular domain has and serves as standard for evaluating the student's performance. Some systems have the domain knowledge [2] or the domain model [4] instead of the expert module. However, these two components are distinguished in [13].

The intensive communication takes place between a learner and an intelligent tutoring system. The module controlling these interactions is called communication module [13] or user interface [2, 6] and it comprises one more part of the architecture.

The system's working cycle is organized as follows. The system selects or generates some problem or some task appropriated to the learner. For this purpose it uses the student model assessing what the student already knows and the teaching strategy provided by the tutoring module. When the system has received the student solution, it performs a diagnosis comparing the student's solution and the one generated by the expert module or prepared in advance. Basing on the diagnosis results the tutoring module provides some feedback and the system updates the student model. After that the working cycle is repeated.

All abovementioned components are only the basic parts of intelligent tutoring systems. Each system can have additional constituents that are required by peculiarity of certain area and technology used for the implementation of a system.

The general architecture of an intelligent tutoring system and the interaction between its components are shown in Figure 1.

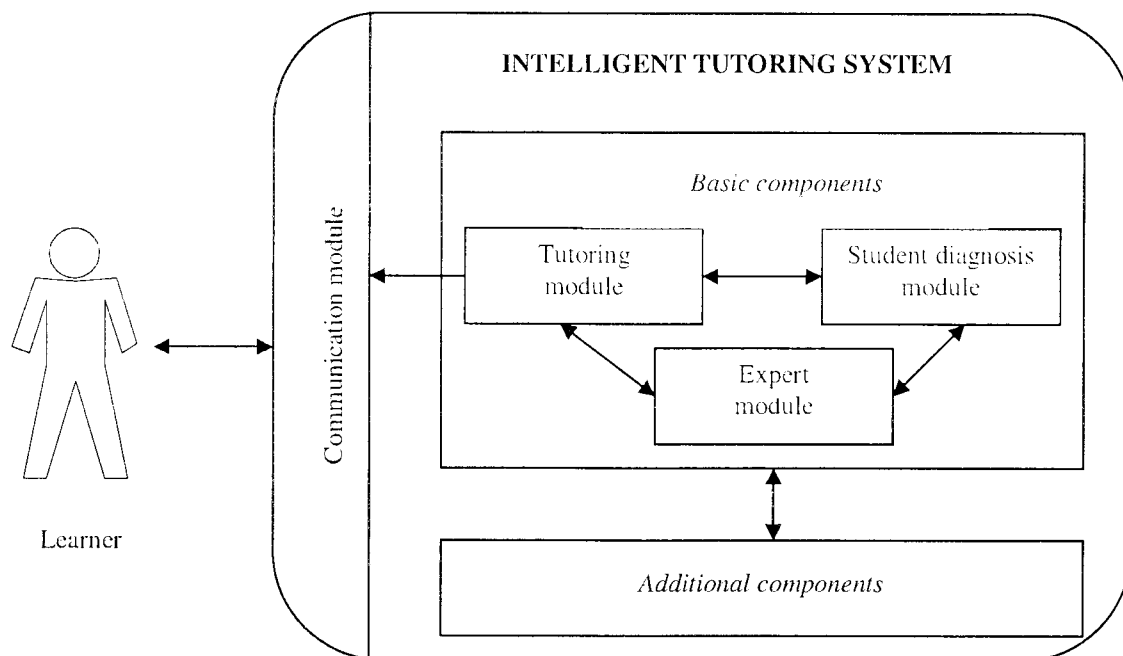


Figure 1. The general architecture of an intelligent tutoring system

4. The Minimax algorithm

Researchers in the field of artificial intelligence have tried to make a computer an opponent of equal value in two-player games since the first chess programs were introduced in 1950. Two

players that have alternate moves until the game ends characterize two-player games. One kind of such games is a game of perfect information, in which all relevant information about current state of a game is available to the opponents. Noughts-and-crosses, checkers, chess, game of go are examples of two-player games of perfect information.

For the purpose of the computer-based implementation of two-player games a game is represented by a game tree, which consists of nodes and arcs. The nodes correspond to the discrete states of a game, for example, the positions of the figures on the board. The arcs represent the legal moves from one state to another. One of the nodes is a root node. It reflects the initial state of the game. Some of the nodes represent terminal states, where the game is over. The structure of a game tree is limited by two rules:

- arcs between nodes on the same level are not allowed, and
- only arcs between a node at higher level and a node at the nearest lower level are allowed.

The players in a game are referred as the maximiser (Max) and the minimiser (Min). The maximiser tries to win, but the minimiser attempts to minimize maximiser's score.

Some algorithms have been developed for implementing two-player games of perfect information. The basic ones are the Minimax algorithm and the Alfa-beta pruning algorithm.

The Minimax algorithm allows choosing of the best next move in a given situation. It associates a value with each state of a game tree and this value indicates how good this state is for a player. The player then chooses the best move available for him/her.

The Alpha-beta pruning algorithm allows reducing the number of states evaluated by the Minimax algorithm, pruning out some branches of a game tree, that have not perspective for the win.

Both abovementioned algorithms are taught to students during lectures on foundations of artificial intelligence. However, the Minimax algorithm was chosen for the implementation of the prototype of the intelligent tutoring system, because of its simplicity.

The steps in the Minimax algorithm are the following:

- Create the game tree.
- Label each level in the game tree as Min or Max alternately.
- Give the value 1 for all the terminal states, where Max player wins, and 0- otherwise.
- Propagate the values of the terminal states up the tree in accordance with the rule:
 - a) if the parent node is at the Max level, it gets the maximum value of its children;
 - b) if the parent node is at the Min level, it gets the minimum value of its children.
- Determine the winning paths, when the propagation of the values is finished and the root node is reached. If maximiser wants to win, he should choose states with value 1. If minimiser wants to win, he should choose states with value 0.

Figure 2 shows the results of the application of the Minimax algorithm. The game is following. There are nine bonbons on the table. Each player can take one, two or three bonbons, but not the same number that his/her opponent has taken. The player who cannot make a move loses the game. The bold digits near the nodes (Figure 2) are propagated values. The bold arcs are the winning paths for the maximiser. The negative numbers on the arcs are the taken bonbons. As the root node has a value 1 the maximiser will always win the game, if he/she plays carefully.

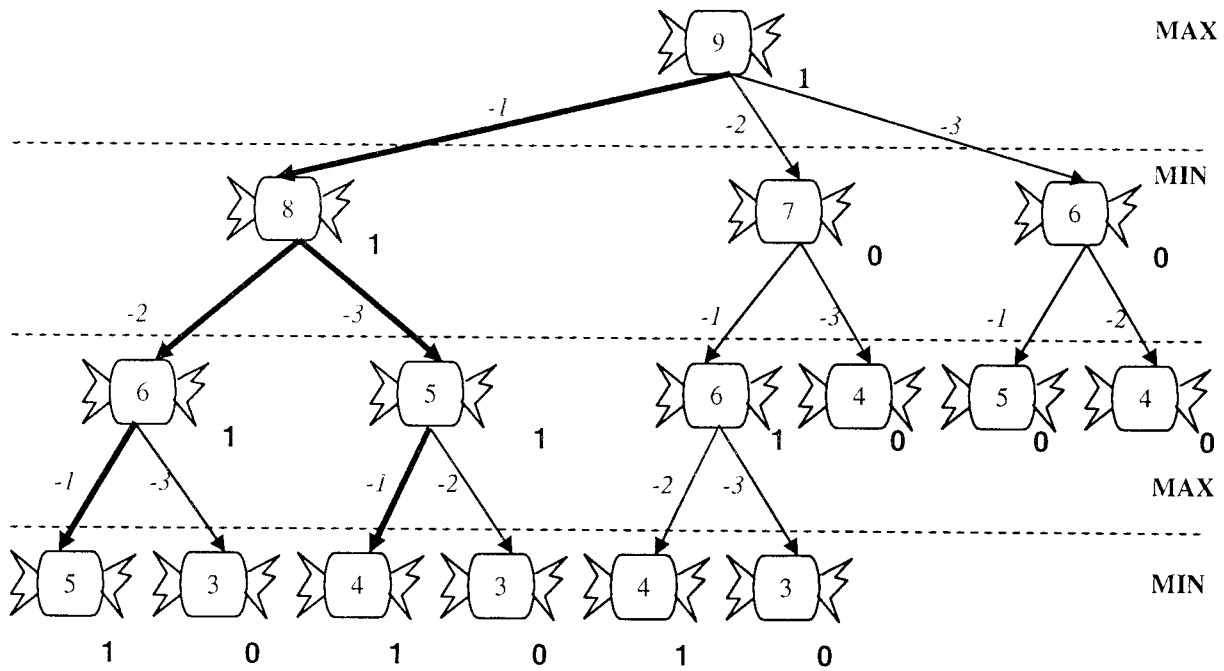


Figure 2. The game tree with applied Minimax algorithm (bold digits are propagated values; bold arcs are the winning paths for the maximiser; negative numbers on the arcs are the taken bonbons)

5. System architecture

Now, when the details of the Minimax algorithm has been discussed, it is possible to describe the architecture of the intelligent tutoring system for Minimax algorithm.

In order to learn the Minimax algorithm, a learner will be provided with three tasks:

- a) the task of the refining of a game tree;
- b) the task of the propagation of values;
- c) the task of the determining the winning paths.

The tasks are sequential, that is, the learner could be able to start the next task if he/she has performed the previous one successfully.

All tasks have been divided into subtasks. For example, the task of the refining a game tree consists of three subtasks (Figure 3):

- Subtask 1: a game tree holds the arcs between nodes on the same level;
- Subtask 2: a game tree holds the arcs between nodes over levels;
- Subtask 3: a game tree holds the arcs of both abovementioned kinds.

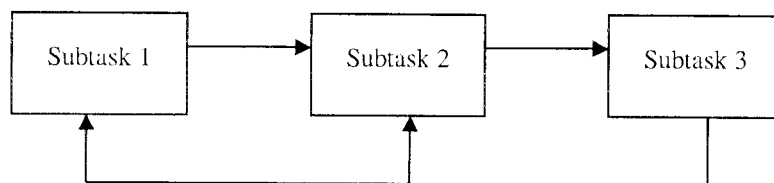


Figure 3. The subtasks of the task of the refining of a game tree and its relationships

The subtasks can be described as follows:

1.subtasks

Precondition:	None because it is the first subtask
Description:	The game tree that has arcs between nodes on the same level are presented to a learner
Purpose:	To teach the creating of correct game tree, deleting the wrong arcs between the nodes on the same level

2.subtasks

Precondition:	The first subtasks is solved correctly
Description:	The game tree that has arcs between nodes over levels are presented to a learner
Purpose:	To teach the creating of correct game tree, deleting the wrong arcs between nodes over levels

3.subtasks

Precondition:	The second subtask is solved correctly
Description:	The game tree that has both arcs between nodes over levels and arcs between nodes on the same level are presented to a learner
Purpose:	To teach previously learned skills to get the correct game tree

The hints, which a learner can get, if he/she acts incorrectly, have been defined for all tasks. For example, the learner will receive feedback from system after each his/her action during the task of the refining of a game tree. If he/she deletes the wrong arc, then he/she will be praised. Otherwise, one of the following hints will be presented to a learner:

- a) the system will point to the last arc that was deleted correctly;
- b) the system will point out one of the levels, where the wrong arcs exist;
- c) the system will point one of the wrong arcs;
- d) the systems will refine a game tree by itself.

The intelligent tutoring system for Minimax algorithm uses two kinds of student model. The short-term model is created for every learning session. It contains the information about the learner's action and the received hints in the current learning session. The long-term student model stores information about the task and the subtask that must be solved. This model is used at the beginning of the new learning session and finishing the current session. At the beginning of the new learning session the information from the long-term student model is read and in accordance with it the subtask of corresponding task is presented to the learner. Finishing the current learning session information from the short-term model is extracted and posted to this model, as well as the information about the last task and subtask is stored.

Because of praising the learner after each right action the storage of the praising phrases is necessary. The items have been extracted in the random way.

The following components comprise the interface of the system for the present:

- the field for system's messages, where the system's commentaries about learner's actions and its guidelines, as well as praising phrases is presented;
- the field for system's hints, where system's hints is presented forcedly, if the learner acts incorrectly;
- some tools that allow the learner to choose hint, if he/she cannot solve the task;
- the field for output of the text of the subtask;
- the work area and the toolbar.

The expert module does not store some information. It only generates the right solution to the given subtask and compares it with the learner's solution. Thus, this model contains the set of algorithms for solving of the subtask.

The domain model partly has been already implemented. It stores the theoretical learning material, which is presented to the learner, if he/she cannot solve the task. The tutorial mode is provided. It consists of slides of the theoretical learning material and provides navigation facilities, that is, the ability to move to the next slide and to the previous slide, as well as to choose the slide by its heading. This mode automatically starts after registering of new student in the system. In the case, if he/she just has some knowledge about the Minimax algorithm, he/she could be able to skip this mode. A student could be able to start solving exercises after he/she would be successfully performed a test about basic concepts after the tutorial mode. A student that has already registered in the system could be able to choose the tutorial mode starting new learning session, if he/she needs recall some concepts. The simple reactive agent [14] supports the tutorial mode. It perceives the learner's actions such as the choice of a particular slide by its heading or the click on the button responsible for the moving to the next slide and changes the content of the current slide.

The game trees have been generated dynamically instead of prepared in advance. They are abstract ones and are not related to the some particular game.

In order to provide the teacher the ability to manage the learning material, simultaneously with the implementation of the intelligent tutoring system the shell of the editor of the learning material is developed. The main purpose of the editor is to allow the teacher to add new slides, to modify or to remove existent slides, as well as to create new chapters of the learning course or create new learning course without programming. The editor provides a blank slide in which the teacher can put some text and images, as well as can define its heading. In future this editor will allow to load text for slides from some well-known text editors such as MS Word, MS Notepad and MS WordPad.

6. Conclusion and future work

The development of the prototype of the intelligent tutoring system for Minimax algorithm now is in progress. The general principles of the functioning of the system have been understood and the main parts of its architecture have been identified. However, the right teaching strategy has not been determined yet, as well as the information that will be stored in the student model has not been defined completely yet. Therefore the first direction of the future work is the development of the detailed architecture of the system, determining and specifying each component, its purpose and functions, as well as relations to other components.

The second direction is the transformation of the system architecture into agent-based architecture and the development of the animated pedagogical agent. As it is mentioned in [15] the agent-based technology is a very promising approach to the implementation of intelligent tutoring systems as it allows to pass from the monolithic structure of such systems to the structure of relatively independent components improving the expandability and reuse of intelligent tutoring systems as well as their maintainability. In order to define an agent-based architecture of intelligent tutoring system for Minimax algorithm it is necessary to determine what components of intelligent tutoring systems can be implemented as agents, what functions agents can perform in each component and which agent types can be used. The

animated pedagogical agent will be responsible for presenting of the feedback to a learner making the learning process more interesting and similar to entertainment.

The third direction of the future work can be related with the development of more general-purpose architecture, that is, the system that is not intended for a specific subject, but is able to operate correctly with different subjects. This requires the development of some shells allowing to manage the knowledge of the intelligent tutoring system about domain, teaching strategies and learner.

References

1. "The Andes Intelligent Tutoring System"// University of Pittsburgh.- <http://www.andes.pitt.edu/> (visited 05.06.2004)
2. Virvou M. , Maras D., Tziriga V. "Student Modelling in an Intelligent Tutoring System for the Passive Voice of English Language"// Educational Technology & Society 3(4) 2000.- http://ifets.icee.org/periodical/vol_4_2000/virvou.html (visited 01.05.2003)
3. Evens M.W., Brandle S., Chang R-C., Freedman R., Glass M., Lee Y.H., Shim L.S., Woo C.W., Zhang Y., Zhou Y., Michael J.A., Rovick A.A. "CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue"// 12th Midwest AI and Cognitive Science Conference, Oxford OH, 2001, pp. 16–23.- <http://www.csam.iit.edu/~ciresim/documents/mwemai01.pdf> (visited 02.10.2003)
4. Hospers M., Kroezen L., Nijholt A., Akker R., Heylen D. "An Agent-based Intelligent Tutoring System for Nurse Education", 2003.- <http://www.ub.utwente.nl/webdocs/ctit/1/000000c1.pdf> (visited 18.10.2003)
5. Crowley R.S., Medvedeva O., Jukic D. "Design and Early Development of an Intelligent Tutoring System in Dermatopathology".- <http://www.pathology.pitt.edu/apiii02/CrowleyAPIII2002.doc> (visited 06.06.2004)
6. Warendorf K. "ADIS - An Animated Data Structure Intelligent Tutoring System on the WWW"// Proceedings of ICICS 97, 9-12 September 1997, Singapore, Vol. 2. Pg. 944-947.- http://www.it.fht-esslingen.de/~warendorf/publications/ADIS_ICICS97.htm (visited 18.10.2003)
7. Sykes E.R., Franek F. "A Prototype for an Intelligent Tutoring System for Students Learning to Program in Java TM".-<http://www.cas.mcmaster.ca/~franek/submit/sykes.pdf> (visited 26.06.2004)
8. Georgouli K., Paraskakis I., Guerreiro P. "A Web Based Tutoring System for Compilers"// 14th EAEEIA conference, Gdansk, 2003.- http://www-ctp.di.fct.unl.pt/~pg/docs/Gdansk_2003_2.pdf (visited 02.10.2003)
9. Suraweera P. "An animated pedagogical agent for SQL-Tutor".- http://www.cosc.canterbury.ac.nz/research/reports/HonsReps/1999/hons_9908.pdf (visited 01.09.2004)
10. Kalayar M., Ikematsu H., Hirashima T., Takeuchi A. "Intelligent Tutoring System for Search Algorithms"//2001.- <http://www.icce2001.org/pdf/p12/JP073.pdf>(visited 21.06.2003)
11. Smith S. "Tutorial on"// School of Computing Science Middlesex University.- <http://www.cs.mdx.ac.uk/staffpages/serengul/table.of.contents.htm> (visited 27.03.2003)
12. Ziemer S. "Intelligent Tutoring Systems in general and Curricula in particular".- <http://www.sts.tu-harburg.de/~st.ziemer/its.pdf> (visited 02.10.2003)
13. Beck J., Stern M., Haugsjaa E. "Applications of AI in Education"// ACM Crossroads, The Student Journal of the Association for Computing Machinery.- www.acm.org/crossroads/xrds3-1/aied.html (visited 25.02.2003)
14. Russell S. J., Norvig P. "Artificial Intelligence: A Modern Approach"// Upper Saddle River, N.J: Prentice Hall, 2003.

15. Grundspenkis J., Anohina A. "Agents in intelligent tutoring systems: state of the art" // In: Scientific Proceedings of Riga Technical University, Computer Science, Applied Computer Systems, 5th thematic issue, Riga, 2004 (to appear).

Alla Anohina, Riga Technical University, Meza 14, Riga, LV-1048, Latvia, Mr. sc.ing., alleila@algs.lv

Anohina A. Intelektuālā mācību sistēma Minimaksa algoritmam

Pētījumi intelektuālu mācību sistēmu jomā ilgst jau vairāk nekā 30 gadus un šajā laikā daudzveidīgas intelektuālas mācību sistēmas tika izstrādātas dažādām sfērām. Šīs programmatūras sistēmas imitē cilvēkskolotāju, demonstrējot intelektuālas spējas, un tādējādi nodrošina efektīvu veidu kā iegūt zināšanas, kad cilvēkiem tās ir vajadzīgas, bet viņiem nav laika apmeklēt kādu mācību iestādi.

Dotais raksts apraksta rezultātus, kas tika iegūti intelektuālas mācību sistēmas, kas paredzēta mācību kursam "Mākslīgā intelekta pamati" Rīgas Tehniskajā universitātē, izstrādes sākotnējā stadijā. Sistēmas mērķis ir palīdzēt studentiem apgūt Minimaksa algoritmu, risinot uzdevumus. Idejas par sistēmas arhitektūras izstrādi ir sniegtas šajā rakstā. Uzdevumi un apakšuzdevumi, kuriem jābūt piedāvātiem studentam mācību sesiju laikā, ir aprakstīti un iespējama atgriezeniskā saite ir apspriesta. Sistēmas attīstības virzieni ir iezīmēti.

Līdzīgi darbi intelektuālu mācību sistēmu jomā, intelektuālas mācību sistēmas arhitektūras galvenās sastāvdaļas un to attiecības, kā arī daži jēdzieni par uz datoru balstīti divpersonu spēļu realizāciju un Minimaksa algoritms ir aprakstīti, lai lasītājs varētu dziļāk saprast sistēmas būtību.

Anohina A. An intelligent tutoring system for Minimax algorithm

The field of intelligent tutoring systems has been researching over 30 years and during this time a variety of intelligent tutoring systems has been implemented for different areas. Such software systems simulate a human-teacher, demonstrating some intelligent capabilities, and thus provide an effective way for acquiring of knowledge when people need them, but do not have time to visit some educational institution.

The paper describes the results of the initial stage of the development of an intelligent tutoring system for the course "Foundations of artificial intelligence" at the Riga Technical University. The objective of the system is to assist students in learning of the Minimax algorithm through exercises. Some ideas about the design of system's architecture are presented. The tasks and subtasks that should be presented to a student during tutoring sessions are described and the possible feedback is discussed. The some directions of the evolution of the system are outlined.

Related work in the field of intelligent tutoring systems, the main parts of architecture of intelligent tutoring system and their relations, as well as some concepts of the computer-based implementation of two-player games and the Minimax algorithm are described for the reader's deeper understanding of the essence of the system.

Анохина А. Интеллектуальная обучающая система для алгоритма минимакса

Исследования в области интеллектуальных обучающих систем ведутся уже более 30 лет и за это время множество интеллектуальных обучающих систем было разработано для различных областей. Эти системы программного обеспечения имитируют преподавателя-человека, демонстрируя некоторые интеллектуальные способности, и таким образом обеспечивают эффективный способ приобретения знаний, когда люди нуждаются в них, но не имеют времени, чтобы посетить некоторое учебное заведение.

Данная статья освещает результаты начальной стадии разработки интеллектуальной обучающей системы для учебного курса «Основы искусственного интеллекта» в Рижском Техническом Университете. Целью системы является помощь студентам в изучении алгоритма Минимакса посредством решения заданий. Некоторые идеи относительно разработки архитектуры системы представлены в статье. Задания и подзадания, которые должны быть предложены студенту во время учебных сессий, описываются, и возможные виды обратной связи обсуждаются. Направления развития системы намечены.

Схожие работы в области интеллектуальных обучающих систем, главные составляющие архитектуры интеллектуальной обучающей системы и их взаимосвязи, а также некоторые понятия основанной на компьютере реализации игр двух лиц и алгоритм Минимакса описаны в статье, чтобы читатель мог глубже понять сущность разрабатываемой системы.