Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
"Individualized learning in a course with a tight schedule"

"Jacek Marciniak" * Marcin Szczepański

"Adam Mickiewicz University in Poznań, ul. Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland"

Abstract

The article presents a solution supporting individualised learning in courses with a tight schedule. Such courses pose additional organisational challenges and require appropriate tools. The presented solution is based on an Intelligent Tutoring System immersed in repository of e-learning content, which enables selection of content immediately before its provision to the student instead of at the beginning of a course. Thanks to this, the system, having identified the student's needs, is able to make available the most suitable repository content at a given stage of education. The flexibility of the system is guaranteed by modularisation of content and its logical division using the UCTS taxonomy. The content has been described by means of concepts arranged according to the specificity of the domain to which the resources belong in order to ensure that the ITS is able to select relevant content. The proposed solution was used to set up an Applications of Fuzzy Logic course, which was part of an Artificial Intelligence class. The course was conducted within a very limited time frame resulting from the COVID-19 epidemic.

© 2020 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the KES International.

Keywords: Intelligent Tutoring Systems; individualized learning; content repositories

1. Introduction

Tight schedule teaching poses additional challenges for teachers and course support teams. When there is a group of students, their needs are always addressed within the time frame scheduled for the whole group, no matter how long the course lasts. The teacher performs a recurring number of tasks of the same nature over the same period of...
time, for example verifies knowledge corresponding to a given batch of material. If the schedule is tight, even a recurring number of activities will be a heavy burden for the teacher. Additionally, complexity arises when education is to be individualised. In such a case, the teacher is also forced to decide at what stage of learning a given student is, how his/her educational path should be individualised, what content he/she should receive, whether he/she can continue learning if they fall behind, etc. in addition to choosing the task to be done. In the traditional model (i.e. in a classroom) individualised learning can only take place in a small group of students and in classes of a specific nature. As far as education using e-learning systems (such as Moodle) are concerned, individualisation may be applied to the students' work pace. In such circumstances, resources distributed to students are, by default, the same for the entire group. Differentiation is only possible upon individual teacher intervention, which complicates the whole process of course management.

Intelligent Tutoring Systems (ITS) are the solution to the above issues. Such systems allow for adaptation of students' learning paths, adaptation to their learning styles, determination of the degree of difficulty of the material provided, etc. Nowadays, in order for ITS systems to be effective and respond to the real needs and habits of its users, it is essential that they operate on the basis of continuously supplied content repositories. This means that students will obtain updated resources, and not only those which were embedded in the ITS when the system was built. The need stems from the habits of the modern users, who, owing to a considerable flow of information, are experienced in working with huge data volumes and retrieving the most important pieces of them. Content repository systems should enable searching content by different parameters, in particular the subject matter and size of resources. Subject matter is the basic parameter that has to be taken into account in any educational process. Moreover, resources must be distinguished by their size, as large repositories may store resources on the same subject, but with different sizes and different levels of detail in terms of the covered subject. Where an ITS searches for resources in the repository immediately before they are given to the student, it must be able to select content with a level of detail and size that is consistent with the resources the student has previously worked with.

The article presents an Intelligent Tutoring System used in a course divided into seven stages, each ending in a sub-test giving the right to continue learning. The system individualises the learning path and provides for an individual work pace of each student. The proposed solution was used to set up an Applications of Fuzzy Logic course, which was part of an Artificial Intelligence class. The course was conducted within a very limited time frame imposed by changes in schedules resulting from the COVID-19 epidemic.

2. Other solutions

Research on Intelligent Tutoring Systems has been conducted since the 1980s [18], [21]. The purpose of such systems is to provide learners with content matching their individual needs, often without teacher intervention. Some systems feature fixed built in content and algorithms. These include mainly systems created at the initial stage of research: SOPHIE and GUIDON [4], [5]. Many similar systems were created later on, for example Cognitive Tutor [2], Logicando [10], Fractions Tutor [15] or ALS-KL [16]. As the systems with such architecture were built for a specific purpose, the content or operating algorithms did not permit change because there was no such need. In the case of applications such as Logicando (a system designed to teach basic logic in primary school), Fractions Tutor (an application designed to teach fractions in primary school) or ALS-KL (a system designed to teach English at a level defined by the result of an entry test), the aim was to teach a specific topic. Cognitive Tutor was created to enable study of human behavior during interaction with an intelligent system. There are also systems without the above mentioned limitation, i.e. which enable replacing content and operating algorithms. Examples of such systems include the KPP System [20] and the ADAPT System [6]. Another system of this type is AeLF [17], which enables full replacement of content at the level of learning objects in the xAPI standard.

ITS systems may also be used to identify learning styles and provide content in response to identified needs. This is true in the case of the system designed to prepare learners for the ICDL exam [1], in which the content is divided into learning objects and saved in SCORM. The course work style described in accordance with the Felder-Silverman model [7] is identified by the WEKA application on the basis of information about the student collected through a questionnaire [1]. Once the style is defined, the learner obtains the course in a form that is most suitable for him/her, for example an e-book, a video, a tutorial or an exercise.
Intelligent Tutoring Systems facilitate provision of content based on an analysis of the learner's current activity. ProTuS [9] and RiPPELE[8] are two examples of systems of this type. The first one, a software development tutoring system, recommends further content to the learner on the basis of a content tagging system within the application. The second one, on the other hand, is a general LCMS system in which learning resources are sorted to match the system's recommendation following previous progress of the student.

3. Intelligent Tutoring System used for individualised learning

The solution presented in the article is based on an Intelligent Tutoring System immersed in repository of e-learning content [11]. The prerequisites for the ability of a system with such an architecture to choose repository content that meets the needs of the student during individualised learning include:

- Knowledge about the size and structure of resources, and the level of detail in covering the subject matter.
- Knowledge about the subject matter and other parameters used when searching the repository.

The above knowledge is required owing to the expected nature of the individualisation of learning – the system is intended to select content from the repository supplied continuously with educational materials. Consequently, once the student's needs are identified at any stage of the course, he/she may be able to access resources which were not available in the repository when the course work began.

3.1. System architecture

The architecture of the solution incorporates an extension of the LMS (Learning Management System) including an autonomous ITS Agent controlling the distribution of resources depending on the student's state of progress, identified needs, and the adopted pedagogical strategy [11]. The ITS Agent has access to the LMS repository and can select any content from it. The essence of the adopted solution is that the teacher may upload content to the repository as SCORM packages at any time. Once uploaded, the content is defragmented in accordance with its internal technical structure (defined by the SCORM Content Aggregation Model) and a logical division established by the teacher (described using the UCTS). The learning individualisation process is controlled by an algorithm (pedagogical strategy) determined by the teacher. The process of providing content to students is shown in Fig. 1.

![Fig. 1. Providing content to students through an Intelligent Tutoring System immersed in repository of e-learning content.](image)

3.2. Content structure

Educational materials on which the ITS operates are saved in SCORM format, while the logical division is provided by the UCTS. SCORM allows to structure materials in a hierarchical way, in which SCOs (Sharable Content Objects) are the leaves. Technically, an SCO is an HTML file with any number of resources. In some
studies, SCOs are interpreted as the learning object [3]. The arrangement of resources built in SCORM is always up to the author. This means that resources of the same volume and comparable student engagement time may have different structures. For example, they may be divided into several large-volume SCOs or several dozen smaller SCOs. Therefore, to make it possible to decide on the volume of resources without dealing with their technical form, the logical interpretation must be conducted by means of a solution referring to their educational nature. One such solution is the UCTS (Universal Curricular Taxonomy System). The UCTS enables division of resources into four levels: Curriculum, Module, Unit, and atomic components such as Learning Object, Assessment or References [12]. The division is made without making a direct reference to the educational process (i.e. without using terms such as lesson, lecture or semester/term) and may be used regardless of the purpose of the resources. Logical division of hierarchical content using the UCTS is shown in Fig. 2.

Fig. 2. Logical division of content using the UCTS.

3.3. Content description

The adopted individualised learning model assumes that the system retrieves content from the repository before it is delivered to the student. This suggests that the algorithm responsible for individualised learning must refer to various content parameters. Content may be described using such metadata as author, language, difficulty, etc. The subject matter must also be defined. It may be described using indexing languages, for example thesauri, classification systems, taxonomies or domain ontologies. The languages are domain-oriented and are intended for the description of resources from one thematic area. If the repository is not of this nature and is to be treated as one coherent whole, the content must be described in an indexing language that allows for the integration of concepts from different domains. Wordnet based ontology, which allows for the integration of concepts from different domains categorised by the nature of conceptualisation (i.e. domain), is an example of such languages. This solution has been successfully used to describe e-learning content of various topics in e-learning content repositories and in ITS systems when searching for content [13], [14].

4. E-learning course with a tight schedule

The Applications of Fuzzy Logic course was created as a supplement to an Artificial Intelligence class. Its schedule has been greatly compressed for the purposes of teaching during the COVID-19 epidemic. The course, which was initially designed as a supplement to traditional classes, became a compulsory course in which students earned partial points influencing the final grade in the class.

4.1. Course structure

The main objective of the Applications of Fuzzy Logic course is to share well-structured knowledge on fundamentals of fuzzy logic and the theory of fuzzy sets, and their applications in IT. It is intended for IT students attending Artificial Intelligence or similar classes. The course comprises interactive educational materials and knowledge tests. Participation in the course is based on asynchronous learning and consists in students' own work
with distributed interactive resources within stages established in the course structure. Students' knowledge is verified after each stage.

The course has been structured using the UCTS. It consists of 53 SCOs, including components containing educational materials (learning objects), final tests and resit tests. The content has been divided into 7 UCTS Units compiled in 3 UCTS Modules. Course structure is shown in Table 1.

| Course parts                                | No. of SCOs | UCTS taxaons |
|---------------------------------------------|-------------|--------------|
| Basics of fuzzy inference                   |             | Module       |
| Introduction to Fuzzy Sets theory           | 5           | Unit         |
| Elements of fuzzy inference                 | 4           | Unit         |
| Tools of Fuzzy Sets theory                  |             | Module       |
| Linguistic variables                        | 3           | Unit         |
| Fuzzy numbers                              | 5           | Unit         |
| Introduction to aggregation of imprecise information | 2           | Unit         |
| Fuzzy Logic Applications                    |             | Module       |
| Bellman-Zadeh Model                         | 3           | Unit         |
| Fundamentals of recommendation systems      | 3           | Unit         |

This division enabled full granularity of content – each part is a logically complete whole. Separating UCTS Modules was necessary due to the need to establish a connection with traditional lectures. Thus, one module represents one 1.5 hour lecture. Introduction of UCTS Units enabled the distribution of smaller batches of resources to students and was supposed to enforce discipline. Limiting the structure to UCTS Modules only would produce larger batches of content, which, taking into account possible irregularities in performance on students' part, could lead to students failing comprehensive tests involving nontrivial batches. From the technical point of view, the content was saved as SCORM 2004. This format ensures that content from SCOs, the smallest components processed by the ITS, are in a hierarchical order and allow searching larger content structures labelled as UCTS Unit or UCTS Module in the repository.

Saving the content as SCORM enabled convenient content aggregation for use in the ITS, taking into account different needs of students working with the course. As a result, the content was uploaded to the system repository as 5 SCORM packages used at different stages of the course:

- course with tests – contains all 53 SCOs, serves as the basic set of resources used by the ITS Agent when providing content for students and individualising the learning path;
- course without tests – contains the entire content and is distributed to students who have completed the course;
- course with the first chapter for a group working with the ITS – contains the first 5 SCOs, and a final test; students start course work and subsequent components are added by the ITS taking into account the student's progress;
- course with the first chapter for a group working without the ITS – contains the first 5 SCOs, a final test, and a resit test; students start course work and subsequent components are added by the teacher taking into account the student's progress;
- course with errata – contains materials with corrections of substantive errors reported by students working with the course.

Moreover, a preliminary test (2 SCOs) has been prepared for all students, uploaded into the system as a separate SCORM package.
4.2. Content description

The content of each course chapter (UCTS Unit) has been described using words or expressions from a wordnet based ontology. The solution was used because it supports the selected content description method, i.e. tagging [19]. The process used tags consisting of words and expressions understandable for experts and non-experts. Ordering by domain (available to experts) was achieved by assigning words and expressions (organized as synsets) to domain categories. Currently, the ontology contains 137 concepts related to the basics of fuzzy logic and the theory of fuzzy sets assigned to 32 domain categories (Fig. 3). The ontology has ensured the homogeneity of the description of the content uploaded to the repository, and guarantees that the description of the content will remain homogeneous in future, when new courses on fuzzy logic appear, along with courses on other subjects as the repository expands. The ontology was created in an iterative process parallel to the building and tagging of content. It functions as a domain model from the Wenger's architecture model [21], underlying the ITS operation.

![Fig. 3. A part of the ontology with basic terms related to fuzzy logic and the theory of fuzzy sets](image)

All metadata used to describe the course is shown in Table 2.

| Parameter          | Description                                                                 | Metadata                  |
|--------------------|-----------------------------------------------------------------------------|---------------------------|
| Subject            | Assigned keywords (ontology concepts used as tags)                          | general/keywords          |
| Language           | Language of the resources                                                    | general/language          |
| Author             | Defines the author                                                           | lifecycle/contribute      |
| Component type     | Defines the type of the content (“lecture”/”exam”)                          | educational/learning resource type |
| Attempt type       | Type of test attempt (“first attempt”/”second attempt”)                      | classification/keyword    |

4.3. Edumatic ITS and algorithm of learning process individualization

The educational process was conducted using the Edumatic ITS system implementing the architecture of an Intelligent Tutoring System immersed in repository of e-learning content. Edumatic ITS is a Learning Management System with an embedded ITS Agent responsible for the individualisation of learning. The system employs a SCORM content repository. Edumatic ITS enables the ITS module to work in the following modes:

- automatic ITS mode – the system proposes content and places it in the individual student's course tree,
- semi-automatic ITS mode – the system proposes a set of content meeting the criteria set for a given stage of education and leaves it to the teacher to decide which of the suggestions will be placed in the individual student's course tree,
4.2. Content description

The content of each course chapter (UCTS Unit) has been described using words or expressions from a wordnet-based ontology. The solution was used because it supports the selected content description method, i.e. tagging [19]. The process used tags consisting of words and expressions understandable for experts and non-experts. Ordering by domain (available to experts) was achieved by assigning words and expressions (organized as synsets) to domain categories. Currently, the ontology contains 137 concepts related to the basics of fuzzy logic and the theory of fuzzy sets assigned to 32 domain categories (Fig. 3). The ontology has ensured the homogeneity of the description of the content uploaded to the repository, and guarantees that the description of the content will remain homogeneous in future, when new courses on fuzzy logic appear, along with courses on other subjects as the repository expands. The ontology was created in an iterative process parallel to the building and tagging of content. It functions as a domain model from the Wenger's architecture model [21], underlying the ITS operation.

![Fig. 3. A part of the ontology with basic terms related to fuzzy logic and the theory of fuzzy sets](image)

All metadata used to describe the course is shown in Table 2.

| Parameter | Description | Metadata |
|-----------|-------------|----------|
| Subject   | Assigned keywords (ontology concepts used as tags) | general/keywords |
| Language  | Language of the resources | general/language |
| Author    | Defines the author | lifecycle/contribute |
| Component type | Defines the type of the content (“lecture”/“exam”) | educational/learning resource type |
| Attempt type | Type of test attempt (“first attempt”/“second attempt”) | classification/keyword |

4.3. Edumatic ITS and algorithm of learning process individualization

The educational process was conducted using the Edumatic ITS system implementing the architecture of an Intelligent Tutoring System immersed in repository of e-learning content. Edumatic ITS is a Learning Management System with an embedded ITS Agent responsible for the individualisation of learning. The system employs a SCORM content repository. Edumatic ITS enables the ITS module to work in the following modes:

- **ITS mode off** – an individual course tree has been created for the student, however, the ITS module is not activated; the teacher is forced to independently search for content from the repository, assess which pieces are the most suitable for the student's progress and place them in the student's individual course tree.

The author of the course decides when the ITS module is to be launched and what should be the algorithm of its operation when individualising the learning process. In the case of the Applications of Fuzzy Logic course, the ITS module was launched each time the student left the final and resit test attached to each chapter (UCTS Unit). The system operated in the semi-automatic ITS mode to control what content would be proposed to the students. The system allows saving the algorithm of the ITS Agent's operation in such a way that the initial conditions are the student's progress in the course (scores gained, completion of individual components of the course), and the expected actions of the system are indicated for these conditions (searching for content from the repository meeting the set criteria). The rules specify priorities relating to the searched content (e.g. preferences as to the author of the content, language, level of difficulty, or size of the resources to be delivered to the student).

The algorithm of learning process individualisation in the Applications of Fuzzy Logic assumed that each student will take a final test after each chapter of the course (UCTS Unit). Students got a resit test after each failed final test. Once a chapter was successfully completed, a new UCTS Unit was added to the individual student's course tree. However, in case of failure, the students withdrew from the course and received access to all course resources without tests (the students could no longer collect partial points, and finished the course at a later date). The students could also take an additional preliminary test. A student who passed the test was released from the course and instantly received the entire course content. However, he or she had the right to improve the result achieved in the preliminary test by taking all the sub-tests in the course. The course work algorithm is presented in Fig. 4.

![Fig. 4. The algorithm of course work in the ITS mode](image)
4.4. Course schedule

Work with the Applications of Fuzzy Logic course was divided into 7 stages corresponding to 7 consecutive course chapters (UCTS Units). The students were separated into two groups: one group worked with the course in the semi-automatic ITS mode (37 students), and the second one with the ITS mode turned off (36 students). Each student working with the course in the ITS mode had the opportunity to work at his or her own pace, which means that he or she could spend as much time as was really needed to read the chapter and then take the final test. The remaining students, on the other hand, were given a schedule to follow in each chapter, specifying a certain period of time. This meant that the students had, for example, 2 days to learn and pass the first chapter. The course work preceded a laboratory on applications of fuzzy logic. However, passing the course was not a precondition of joining the laboratory. As in the case of traditional lectures, the course was supposed to introduce the necessary content, which was to be consolidated later in the laboratories. The opportunity to gain partial points was meant to activate students and was rewarded at final evaluation.

All students worked individually in the tight schedule course because they were required to perform in an organised manner in a short time (18 days). As intended, the course was divided into 7 UCTS Units and 3 UCTS Modules. Since it included individual work paths (different course variants depending on successes or failures in consecutive tests) and individual pace of course work, it was expected that the management of the educational process would be very absorbing for teachers without appropriate tools. Before launching the course, a less cumbersome option was considered, i.e. one in which students' knowledge would be verified less frequently, for example after each UCTS Module. However, it was decided that due to the need to activate students, partial marks should cover smaller batches of content and be more frequent, despite the expected higher teacher workload.

5. Evaluation of individualization process

As expected, the teacher of the group working with the course in the ITS mode had less work related to course management. It was not necessary to keep track of the schedule and the criteria for passing each chapter. There was no need to search the repository to find the batches of material that should be delivered to the students at the end of a given stage of the course taking into account their progress. In the automatic mode, the ITS system decides on its own what content will be appropriate, and in semi-automatic mode, the teacher only has to approve (or reject) the system's suggestion.

It should be noted that work with the group without the ITS was much more absorbing than anticipated. As expected, the teacher was obliged to manage the schedule and progress of each student (i.e. verify whether each student has completed the tasks required in a particular period of time), manage the requirements for passing individual batches of content (verify criteria applicable in individual chapters) and search for resources to be provided to the students from the repository. In addition, the teacher had to do minor auxiliary tasks, which had not been taken into consideration during assessment of the workload related to the Applications of Fuzzy Logic course conducted before the project started.

The amount of work was significant because after the first unit, the test was passed by 55% of students enrolled into the group without the ITS (others not decided to gain additional partial points), and 85% of them passed the entire course during 18 days that resulted from the course schedule. Moreover, on the date on which the test should be taken in this group, only 32.43% of the students in the second group (i.e. in the ITS mode) took the first test (others took it later; in total, 79% of all students from the ITS mode group taking the first test, passed the entire course). It shows that the work at student’s own pace would require a lot more attention from the teacher to mange the course if the ITS mode is turned off, due to the distribution of the same tasks over time.

Tasks performed during the individualisation of learning in the course are shown in Table 3. The overview demonstrates that work in any of the ITS modes significantly reduces the teacher's workload, in particular in the automatic mode, in which consecutive parts of the course are immediately added to the individual student's tree without teacher intervention. The semi-automatic mode requires certain actions, which are, however, supervisory in nature, i.e. the teacher can take a closer look at the progress made by a student and assess suggestions from the ITS. This enables modifying the system's operation in subsequent course editions. Such activity is natural in any educational process. Work without an ITS excludes individualised learning, even for smaller groups of students. The
number of actions to be done and the time they consume disqualify attempts of individualising learning paths in LMS systems without an ITS.

Table 3. Tasks performed by the teacher during individualised learning in the Applications of Fuzzy Logic course

| No. | ITS mode off | Semi-automatic ITS mode | Automatic ITS mode |
|-----|--------------|-------------------------|--------------------|
| 1   | (for each chapter) Download each student's results report for the chapter | Download student login information for the last 24 hours | Download each student's results report when the course ends |
| 2   | When a final (or resit) test is passed, perform steps 3-7 – otherwise perform steps 8-11 | Download a results report for the course for each student who has logged in within the last 24 hours | |
| 3   | Open the course tree of each student who has completed the chapter | Update the course tree of each student who has taken a final or resit test within the last 24 hours | |
| 4   | Set parameters (metadata and subject matter tags) describing the next chapter to be provided to each student | Find the ITS suggestion in the student's course tree and accepting (or rejecting) it | |
| 5   | Search for the consecutive chapter (UCTS Unit) in the repository based on the pre-determined parameters | Repeat steps 1-4 in time frames resulting from the course schedule (24 h) | |
| 6   | Find the appropriate place in the student's course tree | Download each student's results report when the course ends | |
| 7   | Place the chapter in the student's course tree | | |
| 8   | Find the student on the list | | |
| 9   | End the student's participation in the course | | |
| 10  | Find a course without tests | | |
| 11  | Enrol the student in the course | | |
| 12  | Repeat steps 1-11 in time frames resulting from the course | | |
| 13  | Download each student's results report when the course ends | | |

6. Conclusions

The article presents a solution which employs an Intelligent Tutoring System responsible for providing individualised learning by delivering content from a repository prepared for supporting courses on different subjects. To ensure that the system provides effective support for the teacher in the management of this type of learning the repository content must be structured and described to improve findability in situations where the repository content changes. The proposed learning individualisation algorithm imposes much workload on the teacher if no ITS is used. The workload is a considerable obstacle for individualised learning, even in small groups of students. Where the course schedule is extremely tight, the chances for carrying out the educational process without support from an ITS decrease.

The algorithm proposed in the article describes only one of the possible individualisation paths in the course. The available options include: no preliminary test, allowing more end of chapter test attempts, for example by providing additional supplementary content ending in a test classified as a third resit, a resit test covering the entire course after the course ends when a student fails the course. All these course implementation options are possible without any changes in the system architecture; the author simply has to modify the individualisation algorithm and upload new resources to the repository. Owing to the adopted system architecture, these tasks may be carried out even when the students are already working with the course, for example when the teacher has decided that the group requires a different approach than anticipated. This opportunity additionally facilitates the management of the entire learning individualisation process.
References

[1] Aziz, Nashwa A., Fathy Eassa, and Essam Hamed (2019) “Personalized Learning Style for Adaptive E-Learning System” International Journal of Advanced Trends in Computer Science and Engineering 8(1): 223-230.

[2] Alevin, Vincent, Bruce M. McLaren, Ido Roll, and Kenneth K. Koedinger (2006) “Toward Meta-cognitive Tutoring: A Model of Help-Seeking with a Cognitive Tutor” International Journal of Artificial Intelligence in Education 16: 101-130.

[3] Balatsoukas, Panos., Morris, Anne, O’Brien, Anne (2008) “Learning Objects Update: Review and Critical Approach to Content Aggregation”, Educational Technology & Society, vol. 11(2): 119-130.

[4] Brown, John S., Richard R. Burton, and Johan de Kleer (1982) “Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II and III”, in David Sleeman and John S. Brown (eds) Intelligent Tutoring Systems, New York, Academic Press

[5] Clancey, William J. (1986) “From GUIDON to NEOMYCIN and HERACLES in Twenty Short Lessons: ORN Final Report 1979-1985” AI Magazine 7(3): 40-60.

[6] Dias, Luis, Luiz Faria, Constantino Martins, Víriato Marques, and Eduardo Pratas (2014) “Adaptive learning management system to support an intelligent tutoring module” in Luis Gómez Chova, Antonio López Martínez, and Ignacio Candel Torres (eds) EDULEARN14 Proceedings 6th International Conference on Education and New Learning Technologies July 7-9, 2014 – Barcelona, Spain, IATED Academy: 598-607.

[7] Felder, Richard M., and Linda K. Silverman (1988) „Learning Styles and Teaching Styles in Engineering Education”, Engr. Education 78 (7): 674-681.

[8] Klašnja-Milićević, Aleksandra, Mirjana Ivanović, Boban Vesin, and Zoran Budimac (2018) “Enhancing e-learning systems with personalized recommendation based on collaborative tagging techniques” Appl. Intell. 48 (6): 1519-1535.

[9] Rau, Martina A., Vincent Aleven, Nikol Rummel, and Stacie Rohrbach (2012) “Sense Making Alone Doesn't Do It: Fluency Matters Too! ITS Support for Robust Learning with Multiple Representation”, in Stefano A. Cerri, William J. Clancey, Giorgos Papadourakis, and Kitty-Kiriaki Panourgia (eds) Intelligent Tutoring Systems, 11th International Conference ITS 2012, Chania, Crete, Greece Proceedings, Springer: 174-184 (Lecture Notes in Computer Science 7315).

[10] Sfenrianto, Sfenrianto, Yustinus Hartarto, Habibullah Akbar, Mukhneri Mukhtar, Efriadi Efriadi, Mochamad Wahyudi (2018) “An Adaptive Learning System based on Knowledge Level for English Learning” International Journal of Emerging Technologies in Learning 13 (12): 191-200.

[11] Wenger, Etienne (1987) “Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge” CA: Morgan Kaufmann Publishers.