*E-Learning and STEM Education*
Scientific Editor Eugenia Smyrnova-Trybulska
"E-Learning", 11, Katowice-Cieszyn 2019, pp. 295-320

# MODELS OF E-LEARNING SYSTEMS ARCHITECTURE USING AI COMPONENTS

**Jolanta Szulc**
Institute of Library and Information Science,
University of Silesia in Katowice
Bankowa 12, Katowice 40-007, Poland, jolanta.szulc@us.edu.pl

***Abstract:*** *The aim of the study is to present the current state of research on the models of e-learning systems architecture designed using Artificial Intelligence (AI) components. To achieve this goal, the literature of the subject selected from the databases: LISA, LISTA and ERIC and other internet sources were researched. The test results are presented in the following order: (1) Introduction. (2) Basic concepts. (3) Models of e-learning systems architecture. (4) Architecture of e-learning systems with AI elements. (5) Conclusions are presented at the end of the article.*

**Keywords:** Artificial Intelligence, Intelligent Tutoring Systems, Learning Technology Systems Architecture.

## INTRODUCTION

Intelligent Tutoring Systems (ITS) are the subject of research, often conducted from the point of view of their application and usefulness in modern learning concepts. This is evidenced by numerous studies and reviews of literature on this subject. The latest developments include work: (Soofi, Uddin 2019), (Mousavinasab, Zarifsanaiey, Kalhori, Rakhshan, Keikha, Saeedi 2018), (Alkhatlan, Kalita 2018), (Dašić, Dašić, Crvenković, Šerifi 2016), (Sharma, Ghorpade, Sahni, Saluja 2014), (Santhi, Priya, Nandhini 2013), (VanLehn 2011), (Graesser, VanLehn, Rosé, Jordan, Harter 2001), (Nwana 1990).

The reference model for Learning Technology Systems Architecture (LTSA) was developed by the Institute of Electrical and Electronics Engineers (IEEE) and published in the document (IEEE Std 1484.1-2003). LTSA proposes a conceptual architecture that facilitates the educational process using information technologies. According to LTSA, it is possible to identify processes such as:

Learner, Evaluation, System Coach and Delivery in the teaching system, which are units implementing educational processes. In addition, an important element of the system is the database storing historical data on student performance in the teaching/learning process (Records Database), as well as the repository for storing learning resources supporting the learning process (Learning Resources Repositories). The extension of this concept is a model called "Framework to Heritage Education" (Mendozaab, Baldirisab, Fabregat, 2015).

Models of architecture of e-learning systems are also the subject of research. Issues such as the architecture of learning technology systems, common structures in Learning and Teaching Services (LTS) and solutions for specific systems such as knowledge-based, distributed or adaptive e-learning applications are being addressed. In the literature of the subject, we can find, among others works: (Pattnayak, Pattnaik, Dash, 2017), (Armenski, Gusev, 2008), (Pahl, 2008), (Hoppe, Verdejo, Kay, 2003).

In the article, after explaining the necessary concepts, we will review contemporary models of e-learning systems architectures and present examples of e-learning systems architectures with AI components.

Artificial Intelligence is defined differently. Four AI approaches have been proposed in discipline research: Acting Humanly (the Turing Test approach), Thinking Humanly (the cognitive modeling approach). Thinking Rationally (the *laws of thought* approach), Acting Rationally (the rational agent approach) (Russell, Norvig, 2016, p. 2). In each approach, different definitions are formulated, e.g. in the Thinking Humanly category we can use the definition: AI is *[the automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ...* (Bellman, 1978, p. 3).

Depending on the definition, the AI may contain various components/ elements. The main research topics in AI include: automated reasoning, cognitive science (which connects computer models from AI and experimental techniques from psychology to construct precise and testable theories of the human mind), computer vision, knowledge representation, logic-based reasoning, machine learning, natural language processing, neural network, rational agents, robotics (Flasiński, 2016) (Russell, Norvig, 2016). It should be remembered that a number of disciplines form the basis of AI. These are Computer engineering, Control theory and cybernetics, Economics, Linguistics, Mathematics, Neuroscience, Philosophy, Psychology (Russell, Norvig, 2016). We will begin our considerations with the definition of the concept of the model of architecture of e-learning systems, which use AI elements.

# 1. BASIC DEFINITIONS AND CONCEPTS

## 1.1 Definition of the concept of the model

We will start our considerations with the definition of the model. This is an ambiguous concept. The model can be an abstract representation of the existing reality or reality that will be created. The model may refer to both material and non-material entities (Wolski, 2019). The model also means a certain pattern or a broadly understood way of acting (Findeisen, 1985, p. 339). The model understood in this way is often combined with the system. The system model means *presenting the important properties of the real (or created) system that we are interested in in a form convenient to us* (Findeisen, 1985, p. 339) and is usually a simplification of reality. There are conceptual or qualitative, physical, computer, mathematical models (Findeisen, 1985, p. 339).

As already mentioned, the concept of the model is ambiguous and has different meanings depending on the discipline. In IT projects, precisely in requirements engineering, the model represents the reality that interests us. Creating a model is cognitive and/or descriptive and in order for it to be possible, the model must reduce the given reality and present it at a certain level of abstraction. Each model is created for a specific purpose and in a specific context. The ideal model contains only information that is relevant from a particular perspective. When creating a model that will be understandable for its author and recipient, it is necessary to use the same language/notation. Modelling language is defined by: (1) syntax - which defines what elements can be used in this language and defines possible combinations of these elements, (2) semantics - which defines the meaning of individual elements and their linking, which allows for correct interpretation of models created in help of this language. The most commonly used language for modelling requirements is UML (Wolski, 2019).

In the field of requirements engineering, the needs of stakeholders are realized, which can be treated as goals to be achieved by the system being created. Goals can be documented using natural language as well as using goal models. Two types of AND/OR relationships/ compositions can be used to model such goals. The relationship/composition hierarchy can be modelled using a tree. In the case of an AND relationship, all sub-goals must be met to satisfy the parent goal. On the other hand, in the case of an OR relationship, it is enough to fulfil only one sub goal to fulfil the superior goal (Wolski, 2019).

In system analysis, we use the term model system, which should externally behave like a system, although it may have a different internal structure. The system model performs such tasks as: (1) describing the communication model and connections between system elements; (2) illustration of the course of all processes from the point of view of clients, specialists and users; (3) verification of facts in terms of completeness of consistency and correctness

(Plebaniak, 2013). In addition, the following are distinguished: experimental model, statistical model, context model, case comparison model, process stage model, negative case analysis model (Lofland, Snow, Anderson, Lofland, 2009, p. 222-223).

System analysis for modelling the system uses graphical tools or diagrams. At the stage of system analysis, it is important to develop: (1) a logical model of the designed information system, (2) a physical model of the system. A logical model is a set of information defining the behaviour of the system. The physical model is a proposal of a specific implementation of the logical model, i.e. a technical project of software and hardware solutions (Gryglewicz-Kacerka, Duraj, 2013).

In computer science we use the Database Physical Model. It is a model of reality slice expressed by means of classes, tables, files and structures enabling access to data. The physical model of the database contains all the detailed information about the organization of data in the database (tables, files, physical organization of sets) and is a mapping of the logical model of the designed and analysed system. The physical model is a concrete implementation of the logical model based on the selected data model (hierarchical, network, relational, object) (Gryglewicz-Kacerka, Duraj, 2013, p. 10).

Information systems use a system model that includes, among others, object class definitions, a database schema, and a description of the graphical user interface. These definitions are usually available in the basic information dictionary via a dedicated component, specifically for this purpose, e.g. system model server (Kołodziński, Betliński, 2007). The traditional model of the system life cycle is also analysed and modified (requirements analysis, design, coding, testing, installation, operation, withdrawal). These modifications result from the striving to adapt to the objectives and assumptions of a given, specific system or organization. Tadeusiewicz mentions: a cascade model, the model of Fry, a prototype model, a spiral model (Tadeusiewicz, 2019).

## 1.2 The concept of system architecture

System architecture is most often created for information and information systems. We are talking about IT system architecture, software architecture, application architecture, intelligent system architecture, service-oriented architecture (SOA).

In general, we can say that architecture is a high-level system presentation that can serve as a basis for discussion among various participants. System architecture is a compact and easy to learn description of system organization and cooperation of its components. It is a small, intelligible model of the structure of the system and the way in which its elements interact. Architecture can be passed on to other systems that have similar requirements. It can be represented using graphical system models (Sommerville, 2000).

The architecture of most large systems does not correspond to any chosen style, but there are architectural models specific to concrete fields of application. Such system architectures vary in detail, but you can repeatedly use a common architectural structure to build new systems. Such architectural models are called architectures characteristic of the field. There are two types of architectural models: general models and reference models.

General architectural models reflect the architecture of existing systems. Reference models are, however, the results of research in the field of application. They represent idealized architectures that include all the facilities that the system could offer. Reference architectures can serve as the basis for system implementation (Sommerville, 2000).

Large systems rarely correspond to one architectural model. They are diverse and include various models on various levels of abstraction. The models of the system division are, among others repository model, client-server model and abstract machine model. The repository is a model for sharing data in a shared store. In client-server models, data is usually dispersed. In layered models, each layer is implemented using the features of its base layer (Sommerville, 2000).

After designing the structural architecture of the system, the next step in the architectural design process is the division of subsystems into modules. There is no precise way to divide the system into modules. There are two models of sub-system division into modules: the object model and the data flow model. The object model divides the system into a set of communicating objects. The data flow model divides the system into functional modules that download input data and process it in some way into output (Sommerville, 2000).

Software architecture is defined as *the basic organization of the system together with its components, interrelationships, work environment and rules establishing the way of its construction and development* (ISO/IEC, 2007) (ISO/IEC/IEEE, 2011). Software architecture is an essential framework for structuring the system. Various architectural models can be developed during architectural design, for example, a structural model, a control model and a division model. Furthermore, Cloud Computing is the practice of implementing the component architecture of the system (Żeliński, 2019).

System architecture can consist of three basic layers: presentation layer, business logic layer, data layer (Dąbkowski, Kowalski, 2005). For example, *the Intelligent Transport Systems (ITS) architecture is a conceptual project that defines the structure and operation of an ITS system at the urban, regional, national or international level. It often covers not only technical aspects, but also related legal, business and organizational issues* (CUPT, 2019, author's trans.). Examples of architectures are as follows: computer system architecture, operating system architecture, von Neumann's architecture, database system architecture (client-server architecture, 3-layer architecture), application architecture, reference system architecture, reference workflow system architecture, integration architecture (model UML components) (Garcia-Molina, Ullman, Widom, 2014).

It is also vital to pay attention to the methodology of designing and implementing information systems functioning on the Internet and increasingly used to build a geographic information system (GIS). Desktop, client and server solutions (2-layer architecture) are replaced by systems with 3- and n-layer architecture. The functionality of the software developed in the 3-layer architecture is divided into: user interface, application server and database. The browser is used as the user interface (Geoforum.pl, 2019a).

There are also dynamically developing services enabling access to the Internet (in Poland: SDI (Spatial Data Infrastructures) (initially Fast Internet Access, later Fixed Internet Access). These services are growing and according to the Global Spatial Data Infrastructure (GSDI) (Gaździcki, 2003) they can be divided into six categories: human interaction services (including catalogue viewer, geographic browser), model management services, task management services, processing services (spatial, thematic, time, metadata), communication services, system management services (Geoforum.pl, 2019b). Defining services that meet user requirements are part of service-oriented architecture (SOA) (Erl, 2014).

## 1.3 System architecture model

Systems architecture models are created depending on the needs and specific applications in a given field. For example, the object-oriented model of the IT system architecture includes: identification of classes and objects, identification of class and object associations, identification of class attributes, identification and definition of methods and messages - object interfaces. In turn, the model of the real-time system architecture envisages associating the process with each class of detectors and effectors, and other coordination processes may be needed. This model enables fast receiving data from detectors (before next input data is ready), their subsequent processing and response by effectors (Sommerville, 2000).

In contrast, the service system architecture model (Enterprise Service Bus) is a system architecture model used to design and implement interactions and communication between various applications cooperating with each other having SOA architecture. As the system architecture model, this is the implementation of a more general model of client-server architecture, assuming communication and interaction between applications via data communications (IBM, 2019).

In the following chapters, we will discuss selected models of e-learning systems architecture and the architecture of e-learning systems with AI elements.

# 2. MODELS OF ARCHITECTURE OF E-LEARNING SYSTEMS

## 2.1 ITS system architecture model

ITS (Intelligent Tutoring System) is a complex, integrated software system that applies the principles and methods of artificial intelligence (AI) to the problems and needs of teaching and learning. They allow you to search the level of knowledge and learning strategies used to increase or improve students' knowledge. They are aimed at supporting and improving the teaching and learning process in the chosen area of knowledge, while respecting the individuality of the learner (Dašić, Dašić, Crvenković, Šerifi, 2016).

Traditional Intelligent Learning Systems (ITS) focus on rewarding and training, which is why their management mechanisms are often based on domains. More current ITS pay special attention to the familiar approaches to teaching, trying to separate architectural, methodological and manipulative problems from domain knowledge and real-life. The mainstream of current research in this discipline is dominated by problems such as collaborative studying, internet-established instructing and finding out pedagogical agents (Brusilovsky, Eklund, Schwarz, 1998) (Polson, Richardson, 1988).

ITS, based on knowledge, use: (1) knowledge about domain knowledge; (2) knowledge of the teaching principles; (3) methods by means of which these principles are applied and knowledge of methods and techniques of modelling the flow of students in order to acquire knowledge and skills. The traditional intelligent teaching system (ITS) was built based on four interlinked software modules and discussed in the work of Etienne Wegner. These modules were distinguished: (1) Domain Knowledge - Domain Module (DM) (knowledge), domain knowledge, with which the student will communicate during learning and teaching; (2) Student Model - Domain Module (SM) (student knowledge), dynamic model of acquiring knowledge and skills of students); (3) Pedagogical Knowledge - Teacher Module (TM) (tutoring skills), a unit that controls the process of acquiring knowledge and skills of students; (4) Interface - User Interface (UI) or Communication (CM module), system and environment of the student learning process, "knowledge-teacher-knowledge" interaction (Wenger, 1987).

Research on ITS continued in the 1980s and beyond. This is evidenced by the work of such authors as: (Polson, Richardson, 1988), (Psotka, Massey, 1988), (Murray, 1999), (Zhang, Ren, Chen, 2005), (Phobun, Vicheanpanya, 2010).

However, the first ITS systems as single-application adapting to the needs of the learner were based on the algorithms contained in the application and used the content embedded there. These include applications such as SOPHIE (Brown, Burton, Kleer, 1982) and GUIDON (Clancey, 1986). Similar systems are still being created, among them Cognitive Tutor, Help Tutor, Logicando, ASSISTments Platforms, and Franctions Tutor (Marciniak, 2015, p. 158).

Over time, different approaches have appeared that implement various architectures, e.g. IEEE LTSC CMI, IMS LIP (Learner Information Packadge), IMS Simple Sequencing, LMS (Learning Management System), SCORM (Sharable Content Object Reference Model), SCORM CAM (SCORM Content Aggregation Model) (Marciniak, 2015, p. 160). During the research, however, it was noticed that for ITS operating in the e-learning environment, there are no solutions in which the ITS system will work with the use of repositories of e-learning content, powered on a continuous basis. Although the approaches using IMS Simple Sequencing allow for adaptive matching of content for the learner, but due to the SCORM architecture, this can only be implemented within the SCORM CAM structure, i.e. within the SCORM package in which the e-learning course is transferred. Therefore, a solution was proposed where ITS is immersed in the e-learning content repository. This solution will be presented in the next chapter.

## 2.2 The architecture model of the intelligent e-learning system

In the architecture proposed by Jacek Marciniak, the ITS system has been implemented as an intelligent agent and is a module called Agent ITS embedded in the LMS system. The assumptions of this system are as follows: (1) the wordnet ontology is treated as a model about the world (domain model) and used for downloading content from the repository; (2) didactic strategies are built using IMS Simple Sequencing in SCORM CAM structures; (3) didactic content is created using the Extension Points and Triggers mechanisms, used to transfer to ITS the conditions for searching repositories and ITS operation strategies.

In addition, the system architecture is based on the assumption that the solution is built on the basis of the existing LMS remote teaching system, implementing SCORM 2004, which after the extension has ITS features. ITS provides content from the repository with the following rules: (1) didactic contents stored in the repository are in the form of training units; (2) the repository is powered by e-learning courses containing units of learning independently from each other and on various topics; (3) the courses are constructed in such a way that the components contained therein can function independently of the course; (4) new courses are constructed in a dynamic way and adapt to the needs of the learner identified during learning; (5) adaptation is carried out using all materials contained in the repository (Marciniak, 2015, p. 162).

Detailed discussions of the architecture of the intelligent e-learning system can be found in the works: (Marciniak, 2014), (Marciniak, 2015), (Marciniak, 2016).

According to the author of this solution, the whole project has all the components of the traditional ITS system architecture, i.e. Domain Module, Student Model, Pedagogical Model and Interface. At the same time, the Domain Module is built as a model of knowledge about the world using the wordnet ontology. Thanks to this structure, it is possible to map domain and expert conceptualizations of a general and local nature. Domain knowledge may come from thesauri, domain ontologies, classification systems, and it is possible to write relationships between concepts in the content of a given repository. The wordnet ontology is used in a normalized form, which means that it contains all information about scales determining the degree of connection between concepts.

The Student Model consists of data stored in the LMS system. They can be organized in various ways, e.g. according to the IMS ePortfolio specification.

The Pedagogical Model consists of didactic strategies, didactic patterns and the ITS Agent. Didactic strategies are patterns of behaviour implemented by the system during interaction with the learner. They are saved using the IMS Simple Sequencing in the organization of the course for selected content components. The didactic patterns are sets of predefined rules written in IMS Simple Sequencing, which after the introduction of a specific e-learning course to the organization will create a didactic strategy. An ITS agent is an IT module that extends the LMS architecture by signing in to the SCORM Navigation Model, which supports requests to provide further content to the learner.

The system interface is determined by the way the contents are presented in the SCO delivered to the learner. These contents can be saved as multimedia and interactive elements, text supplemented with graphic elements, video sequences (Marciniak, 2015, p. 167).

The presented intelligent e-learning system allows to conduct education in an adaptive way, thanks to which, when providing content, it is possible to take into account the specific needs of learners. Of course, also other models of e-learning systems architectures function today. An overview of such architectures will be presented in the next chapter.

## 3. OVERVIEW OF CONTEMPORARY MODELS OF ARCHITECTURE OF E-LEARNING SYSTEMS WITH ELEMENTS OF AI

### 3.1 Architecture for recommendation of courses in e-learning system (2017)

The goal of the system is to recommend e-learning courses to the student based on his/her profile. The student's profile is created by applying the k-means algorithm to the student's interaction in Moodle. To do this, follow the steps below: (1) To collect data from Moodle server. (2) to perform data preprocessing in order to make data suitable for data mining algorithms. (3) To build learners' profile by applying k-means algorithm.

Architecture is implemented in Moodle allows you to evaluate the work of students. The following attributes can be used: Number of quizzes completed, Messages sent to chat, Messages sent to teacher, Messages sent to forum, Messages read on forum, Time spent on quizzes, Time spent on forum, Total Marks obtained. The pre-processing of data includes several steps, including data cleaning, user identification, session identification, path completion, transaction identification, data transformation, data integration and data reduction. The Moodle platform provides login to each user and allows identifying the user and each session. However, the following tasks need to be performed: (1) Data selection. Select specific courses and parameters characterizing the student's work, e.g.: the time spent by students on quizzes. time spent by students on assignments, message sent by students to the teacher, messages sent by students on the forum, etc. (2) Create Summarization Table. The summary table contains a summary for each line of activities performed by each student during the course and the final grade obtained by each student on each course. (3) Data Discretization. Data discretization helps us to transform numerical data into categorical data. Different methods can be used. In the manual method, we assign attribute values to four categories EXCELLENT, GOOD, AVERAGE and POOR. (4) Transform the Data. The data initially saved in the Excel file must be converted to a CSV file. Then we convert them to the ARFF file format. Initial data processing includes several sub-steps such as data cleaning, identification, user identification, session identification and path completion. Further processing of files takes place in the WEKA (Waikato Environment for Knowledge Analysis) system. WEKA is data mining software that implements data mining algorithms (see: https://www.cs.waikato.ac.nz/ml/weka/).

The elements of the system are: User Authentication, Logging Check, Learner's interaction with Moodle, Data Stored in Moodle Database, Data Prepocessing, Applying K-Means Algorithm, View Module.

The research results show that advanced courses should not be recommended to inactive students who have poor grades.

Applications: as a research project intended for further research.

Developed by Bhupesh Rawat, Sanjay K. Dwivedi/Babasaheb Bhimrao Ambedkar University, India (Rawat, Dwivedi, 2017).

## 3.2 Architecture for e-learning system with intelligent component (2016)

Intelligence is built into the architecture of the e-learning system, thanks to which the system automatically responds to the user's requirements. The system responds to each individual user and is able to predict his preferences or interests.

The system architecture consists of the following components: User Interface, User profile, E-provider, Acquisition, Filtering, History database, Creator, Evaluator, Good rank, Bad rank and Selector. The component selector has the ability to analyse user feedback and create knowledge after user evaluation. It consists of four subsystems, which are: Evaluator, good rank, bad rank and intelligent component. The evaluator has the opportunity to receive feedback from the user based on what has been read, can identify a good and a bad resource. Good rank gathers electronic resources that are classified by the user as good electronic resources. Bad rank stores the lowest ranking for e-resources that are classified by the user as bad e-resources. A good rank stores a user profile and information history. This component can learn from the user's habit. The result from this component should be delivered to the Creator component. This automated assessment is based on the modelling of the activity history and its evaluation by the user.

Applications: as a research project intended for further research.

Developed by Mafawez Alharbi, Mahdi Jemmali/Majmaah University, Saudi Arabia (Alharbi, Jemmali, 2016).

### 3.3 E-learning system using machine learning and user activity analysis (2015)

The aim of the project is to present an autonomous and intelligent e-learning system in which machine learning and user activity analysis play the role of an automatic evaluator for the level of knowledge. The assessment of the level of knowledge is carried out in order to adapt the presented content to a realistic assessment of students in the online system.

The assumptions of the system are as follows: (1) User experience. E-learning users should have access to the educational system via various means with access to the Internet. Access should be flexible, tailored to users' lifestyles, global reach and / or independent collaboration between content providers. (2) Cloud-based environment. The cloud-based e-learning system offers a wide range of possibilities, for example competitive costs along with a high level of scalability. At the same time, it includes a subset of services (Intelligent Agents) dedicated to registering user activities (activity logs, exam results). (3) Incorporating concept maps. Maps are tools for understanding the knowledge for each system user. A simple map it consists of terms marked with circles, while the basic relationships are marked with links with annotations. It is possible to use conceptual maps in combination with multimedia. (4) Generating activity reports. The user activity summary contains several variable derivatives representing activity, assessment of conceptual maps, and exam results. The following aspects are analyzed: (a) The feature extraction (for example, the average time spent per study objective). (b) Categorizing the extracted features. (c) Concept map analysis. (d) Normalizing the results of the exam sets. (e) Attention level (for example, average time of focus). (5) Knowledge level assessment. The Machine Learning method for classification provides the desired results using the user's activity model. Before constructing classification models, in domain categories, two possible configurations depend on knowledge that is to be general or specific.

Elements of the system are as follows: Dynamic E-learning Environment, Extra Activity, Mobile, Web Based, Activity Monitor, User, Logs and Content, Pre-processing, Activity Report, Classification, Knowledge Level.

Applications: as a futuristic design for further testing.

Developed by Nazeeh Ghatasheh/The University of Jordan, Jordan (Ghatasheh, 2015).

## 3.4 E-learning Enterprise Architecture using SOA (2014)

The architecture of the e-learning system was designed using the Service Oriented Architecture (SOA) approach using cloud computing. The project also used mechanisms of related architectures, such as: Service Oriented Cloud Computing Architecture (SOCCA), Cloud-oriented e-Learning Model Architecture (COLMA).

The goal of the project is to provide e-learning services based on SOA architecture using cloud computing. Such architecture should include three components: (1) Technology or infrastructure architecture that can be gradually acquired and used to meet the need for modern learning applications; (2) a data architecture that can handle structured as well as unstructured data in a centralized manner and handle data and metadata needs of the future application; (3) Application architecture that should be open and expandable so that it can meet the ever-changing requirements of the education industry.

The elements of this project are: SOA Based Solution, Internal Assets, External Systems, users, such as: Students, Faculty, Administration, Other Stakeholders. The SOA Based Solution module includes: Application Architecture, Data Architecture, Infrastructure Architecture.

Applications: as a research project intended for further research.

Developed by Erick Fernando/University of Jambi, Indonesia (Fernando, 2014).

## 3.5 MVC (Model–View–Controller) based design pattern for context aware adaptive e-learning system (2013)

The innovative architectural model is based on the MVC (Model-View-Controller) design model, which is able to perform a personalized adaptive delivery of the course content in accordance with the contextual information of the student, such as the learning style and features of the learning device using the ontological approach.

The ontological approach ensures the management and organization of course materials based on their semantic relationships. Representation of the various topics of the proposed ontology of the course can be formally represented as G (T, P, R), where T = Set of topics of specific course or subject; P = is the property set such as ID, Name, Description, etc.; R = is the relation set indicating the semantic relationships between the pair of topics. However, thanks to the e-learning applications used based on device-independent courses, students can view their materials and can get additional help on difficult topics.

The main purpose of the proposed system is to improve the student's knowledge and to facilitate the learner's review of the course content at any time. Two strategies were used to achieve this goal: (1) Ontology based content organization and presentation of course material; (2) Device independent adaptive delivery of learning resources.

This system consists of: (1) Controller (context detection and adaptive mechanism): the controller is responsible for noticing what type of educational device has received the request, and then redirects to the appropriate view (web page). (2) Model (information storage and processing of queries): describes the student's context, the logic of adaptation and repository of the teaching content. (3) View (user interface): this is the output representation of the model data, it is a web browser with an internet connection.

In the context of the proposed contextual architecture of the adaptive e-learning system, the client device sends an HTTP request to the Web server (e.g. IIS). The context detection mechanism (Controller) implemented on the web server receives the request and identifies the device type based on the user agent's profile headers. The device context is saved along with the student's preferences within a given student context identifier as student context information (Model). When the student accesses the teaching content, the controller's Action class forwards the request to the adaptation logic that is responsible for providing the relevant content from the given database and in accordance with the presentation logic (View).

Applications: the system's prototype is implemented at the university.

Developed by Kalla Madhu Sudhana/St. Peter's University, India; V[ences] Cyril Raj/Dr.M.G.R University, India; T. Ravi/Srinivasa Institute of Engg & Tech, India (Sudhana, Cyril, Ravi, 2013).

## 3.6 Model of e-learning system with Extendable Open Source Architecture (2013)

Open Source extensions, developed and maintained by the open source community, increase the functionality of the e-learning system. In the appropriate system components, the server side extension manager checks the correctness of portability, e-learning standardization and security issues before approving the extension to the e-learning community. In contrast, the client side extension manager checks for new extensions/updates and allows these extensions/updates.

This system consists of: (1) core components: provide the main functionality of the e-learning system, (2) extensions: developed and maintained by the open source community, extend the functionality of the e-learning system, (3) server side extension manager: validates the portability, e-learning standardization and security (4) client side extension manager: checks for updates and update/disallow these extensions/updates.

According to the creators of this project, the expandable architecture of the open source e-learning system can better meet the expectations of the e-learners community.

Applications: as a research project intended for further research.

Developed by Murtaza Ali Khan, Faizan Ur Rehman / Umm Al-Qura University, Saudi Arabia (Khan, Rehman, 2014).

### 3.7 Adaptive e-learning systems (2010)

The aim of the project is to develop an agile e-learning system with flexible architecture, flexible use of resources and adapted content for learning in the university environment. It is true that the definition of the agile IS (information system) is still not developed, but some assumptions have been made in the architecture model. The definition was adopted that *The agile e-learning system is a system that has adaptable, reusable and easy changeable content* (Finke, Bicãns, 2010, p. 311).

According to the Agile Software Development Manifesto (access: https://agilemanifesto.org/), particular attention should be paid to: (1) Individuals and interactions over processes and tools; (2) Working software over comprehensive documentation; (3) Customer collaboration over contract negotiation; (4) Responding to change over following a plan. The concept of agility can be described as reliability, adaptability and flexibility. All these factors, through interaction, ensure system variability. These aspects can be implemented in e-learning by predicting algorithms that work with parameters defined for objects and by combining these objects together, as well as by using keywords or other element of metadata or combination of elements.

Elements of the system are as follows: E-learning system management, Services, Business process, University and Cloud. Cloud consists of four components: Storage (User profiles, Backups, LO Repository, Assessments), Social (Skype, Twitter, Blogs, Messenger), Multimedia (Audio, Video, Animations, Rich interactive applications, Still images), Load balancer (which, by purchasing some of its resources from the cloud, allows us to adjust available resources according to load indicators). Services include with Assessment, User management, Learning objects storing. Whereas, Learning objects development, Learning curriculum development, Feedback collection, Collaboration with industry, Research, Team work case studies, Scientific activities belong to Business process.

A system that changes architecture, acquiring part of its resources from the cloud, allows you to adjust the available resources in accordance with the load indicators. These changes mean partial assignment of business processes and storage of relevant information to the management of the system holder - university.

Applications: as a research project intended for further research.

Developed by Anita Finke, Janis Bicans/Riga Technical University, Latvia (Finke, Bicans, 2010).

### 3.8 A semantic web architecture to integrate competence management and learning paths (2008)

The aim of this project is to develop a prototype application based on competency ontology, which will be used to manage competences and learning paths in conjunction with the e-learning method. Modifying an employee or department/organization skills gap analysis with relevant learning subjects is critical to developing the right learning pathways, and consequently the appropriate competencies of employees or organizations.

The ontological approach ensures the management and organization of course materials based on their semantic relationships. Representation of the various topics of the proposed ontology of the course can be formally represented as G (T, P, R), where T = Set of topics of specific course or subject; P = is the property set such as ID, Name, Description, etc.; R = is the relation set indicating the semantic relationships between the pair of topics. However, thanks to the e-learning applications used based on device-independent courses, students can view their materials and can get additional help on difficult topics.

The technical architecture of the system consists of the following components: Web Browser, Web Server, Java (Servlet Container, JSP Pool (JavaServer Pages Pool Service), Ontology management, Ontology (RDF (Resource Description Framework)), Data Retrieval. Front-end has been designed as jsp pages and thanks to it, users can access various system functions, while some jsp pages contain JavaScript functions. Apache Tomcat was used as a servlet container. Back-end is implemented in Java, and access to the ontology is provided by the Jena and RDQL (RDF Data Query Language) APIs (application programming interfaces). Jena is an open Java API for RDF, available on the Internet (see http://jena.sourceforge.net/) and RDQL is the query language for RDF in Jena models.

Functional architecture of the system consists of the following components: Core System Function and Reporting Function. Core system functions include functions for inserting, updating and deleting ontology data. It also uses the functions of creating, updating and deleting the relationship between two competences, work assignment and the relationship between the subject and the competence. Reporting functions include providing the system user with a variety of view functions, such as *View Competency Model* and *View Jobs'*

*Infos* that generate a table with all the tasks of the organization. In addition, various reports are provided on the analysis of skills gaps, succession planning (including identification of successors in key positions, career path planning, university development), experts and projects.

Applications: the proposed application has been implemented in the Microsoft.NET version at Microsoft Hellas, a branch of the leading IT company Microsoft Corporation in Greece.

Developed by Fotis Draganidis, Paraskevi Chamopoulou, Gregoris Mentzas/National Technical University of Athens, Greece (Draganidis, Chamopoulou, Mentzas, 2008).

## CONCLUSION

Research and analysis of currently used and designed models of e-learning systems architectures are limited. The article presents only selected at random e-learning system architectures described in the registered documents selected in the databases. Despite these limitations, it can be seen that the architecture of e-learning systems is designed using new technologies, such as: agile Information System, Cloud Computing, Extendable Open Source, ontology based content organization, competency ontology, Service Oriented Architecture (SOA), semantic web architecture. In addition, some of these systems use (in assumptions/theory or in practice) elements of AI. These elements are concept maps, deep learning, intelligent agents, knowledge management, machine learning method, ontologies. Detailed comments on the advantages and disadvantages of these architectures and a list of AI components used are presented in Table 1.

**Table 1.**
**Advantages and disadvantages of selected models of architecture of e-learning systems with elements of AI**

| **Model of architecture** | **Advantages** | **Disadvantages** | **Methods, technologies, elements of AI** |
|---|---|---|---|
| Architecture for recommendation of courses in e-learning system (2017) | the option of recommending courses to students based on their profile | disadvantages arising from the operation of the WEKA platform (- data mining suites do not implement the newest techniques; - the documentation for the GUI is quite limited; - GUI does not implement | machine learning (algorithms for data mining tasks in WEKA) |

| | | | |
|---|---|---|---|
| | | all the possible options; - scaling; - the data preparation and visualization techniques offered might not be enough) | |
| | the ability to implant this architecture in Moodle | | |
| | the opportunity to find students who are less active on collaborative platforms | | |
| Architecture for e-learning system with intelligent component (2016) | it has ability to predicate e-recourse for the user | for implementing the intelligent component AI techniques must be used (not specified yet) | machine learning/auto evaluation<br>- is a multi-start (the intelligent component can make some knowledge after more evaluation by the user) |
| | auto evaluation based in history modeling and history evaluation make by user | is a multi-start local search method which requires iteratively solving a sequence of multi instances | |
| | the e-learning system is automatically responsive to the user requirements | | |
| E-learning system using machine learning and user activity analysis (2015) | flexible and adaptable learning environment to the users' lifestyle | tree-based and Naive-based algorithms show several variations among the different measures, which makes it more difficult to judge which one has an overall better | machine learning, Intelligent Agents, conceptual maps, cloud-based |

| | | | |
|---|---|---|---|
| | | error margin. | environment |
| | device-independent suitability (to offer a wide range of options to those unable to learn and work under strict time schedules, for example employees, housewives, travelers, and learners with high possibility of relocation) | future work needs to investigate the importance of input variables to the classification method | |
| | independent collaboration between content providers | | |
| | the ability to use concept maps that have different advantages over traditional approaches | | |
| E-learning Enterprise Architecture using SOA (2014) | the application can be continuously updated by the application provider | disadvantages resulting from the operation of cloud computing platforms, among others: high costs, a complicated pricing model for cloud services, data security | cloud computing platform, SOA, SOCCA, COLMA |
| | the application provider is offering a very scalable web application using a multi-tiered web architecture, implemented on a considerable infrastructure | | |
| | service-oriented cloud computing e-learning architecture include decentralized, cost effective, virtualized, flexible, personalized and scalability elements | | |
| IMVC (Model–View–Controller) based design pattern for | using server-side adaptation is the server usually has much more processing power than the | disadvantages of MVC operation: complexity, low flexibility of the model | adaptive learning, MVC |

| | | | |
|---|---|---|---|
| context aware adaptive e-learning system (2013) | client devices, so that server can dynamically perform adaptation according to learner device capabilities | | |
| | advantages of MVC operation: model independence (the application can have multiple independent views for the same model); high flexibility of views (views can be modified more often and at a lower cost) | | |
| Model of e-learning system with Extendable Open Source Architecture (2013) | the development and deployment of dynamic contents based on HTML or XML - these features of XML allows more efficient searching, intelligent data mining, querying (e.g. SQL) etc. | disadvantages of OpenSource software: openness of the code may mean a greater risk of attack, OpenSource is rarely compatible with future versions | extendable OpenSource Architecture of e-learning system |
| | open source architecture of e-Learning system that allows the user to enhance the functionality of existing e-Learning system through extensions | | |
| Adaptive e-learning systems (2010) | economy of time, because we do not have to define the link to the object (feature of Moodle) | disadvantages resulting from the use of Cloud Computing, ontology and Semantic Web technologies | Cloud Computing, ontology, Semantic Web (OWL language) |
| | structuring of the curriculum allows us to plan the topics of the course more effectively and therefore allows to provide comprehensive | | |

| | | | |
|---|---|---|---|
| | and detailed layout of the course | | |
| | users of the system do not have to perform manual linking by adding or deleting objects and their parameters | | |
| | the search system depends only on the entered parameters; search services indicate and combine queries with available objects | | |
| A semantic web architecture to integrate competence management and learning paths (2008) | possibility of extending the system with the semantic search function and the inference engine | unresolved issues that need further research in ontological systems, such as RDF cascading | Semantic Web, RDF, RDF Data Query Language |
| | the ability to assess the effectiveness of ontological systems in real environments | | |

*Source: Own work*

It is worth paying attention to interesting projects, e.g. using agile IS. The dynamic aspect of the development of agile IS structures is achieved mainly thanks to ontology and rules defined in the OWL (Web Ontology Language) language. These assumptions are in line with the latest trends in education, according to which educational systems must be agile enough to support future practices (Noskova, Pavlova, Yakovleva, 2018) (Smyrnova-Trybulska, 2018). Some researchers believe that Cloud Computing combined with rule-based systems is the future way of dynamic development of e-learning. In the future, it will also be necessary to examine the limits of the dynamics of the e-learning system, because at the moment it is only possible to determine the dynamics of the system within certain limits.

It seems that the development of e-learning system architectures will continue. This is a favourable tendency in the existence and development of these systems, especially from the point of view of their capabilities and usability.

## REFERENCES

Alharbi, M., Jemmali, M. (2016). Architecture of e-Learning. *International Journal of Knowledge Engineering*, *2(1)*, 20-25.

Alkhatlan, A., Kalita, J. (2018). Intelligent Tutoring Systems: A Comprehensive Historical Survey with Recent Developments. *arXiv:1812.09628*. Retrieved from https://arxiv.org/abs/1812.09628 (accessed 1 August 2019).

Armenski, G., Gusev, M. (2008). Architecture of modern e-learning systems. In: L. Antovski, M. Mihova (Eds.), *Proceedings of the Sixth Conference On Informatics And Information Technology CIIT 2008* (pp. 38-42). Retrieved from http://ciit.finki.ukim.mk/data/papers/6CiiT/6CiiT-09.pdf (accessed 1 August 2019).

Bellman, R.E. (1978). *An Introduction to Artificial Intelligence: Can Computers Think?* San Francisco: Boyd & Fraser Pub. Co.

Brown J. S., Burton R. R., Kleer, J. De (2008). Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II & III. In D. Sleeman & J. S. Brown (Eds.), *Intelligent Tutoring Systems* (pp. 227-251). New York: Academic Press.

Brusilovsky, P., Eklund, J., Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems*, *30(1-7)*, 291-300.

Clancey, W. J. (1986). From Guidon to Neomycin and Heracles in Twenty Short Lessons: ORN Final Report 1979-1985. *AI Magazine*, *7(3)*, 40-60.

CUPT. (2019). *Centrum Uniijnych Projektów Transportowych. Koncepcja architektury w procesie implementacji ITS*. Retrieved from https://www.cupt.gov.pl/wdrazanie-projektow/its/its-frame/architektura-frame/koncepcja-architektury-w-procesie-implementacji-its (accessed 1 August 2019).

Dašić, P., Dašić, J., Crvenković, B., Šerifi, V. (2016). A review of intelligent tutoring systems in e-learning. *Annals of the University of Oradea*, *3*, 85-89.

Dąbkowski, J., Kowalski, M. (2005). Technologia JSP w tworzeniu aplikacji rozproszonych na przykładzie systemu zarządzania nieruchomościami w gminie. *Inżynieria Rolnicza* [JSP *technology in network distributed applications applied to community real estate market. Agricultural Engineering*], *14*, 75-82.

Draganidis, F., Chamopoulou, P., Mentzas, G. (2008). A semantic web architecture for integrating competence management and learning paths. *Journal of Knowledge Management*, 12(6), 121-136.

Erl, T. (2014). *SOA : koncepcje, technologie i projektowanie*. (A. Grażyński, Trans.). Gliwice: Helion. (Original work published 2005).

Fernando, E. (2014, November). E-learning System Architecture based on Service orientation Architecture Cloud Computing. *International Conference on Education, Technology and sciences*. University of Jambi. Indonesia. doi: 10.13140/2.1.4700.3841

Findeisen, W. (Ed.). (1985). *Analiza systemowa. Podstawy i metodologia*. Warszawa: Państwowe Wydawnictwo Naukowe. [*System analysis. Fundamentals and methodology*. Warsaw: Polish Scientific Publisher]

Finke, A., Bicãns, J. (2010). E-learning system content and architecture evolution. In A. Targamadzé (Ed.) *Information Technilogies'2010: Proceedings of the 16th International Conference on Information and Software Technologies (IT 2010), Lithuania, Kaunas, 22-23 April, 2010* (pp. 311-315). Kaunas: Kaunas University of Technology.

Flasiński, M. (2010). *Introduction to artificial intelligence*. [Cham]: Springer International Publishing Switzerland.

Garcia-Molina, H., Ullman, J. D., Widom, J. (2014). *Database systems: the complete book*. Harlow, England: Pearson.

Gaździcki, J. (2003). Kompendium infrastruktur danych przestrzennych. Część II skróconej wersji polskiej podręcznika Developing Spatial Data Infrastructures: The SDI Cookbook GIS świat, s. 29, Douglas D. Nebert. [United States] : Global Spatial Data Infrastructure, 2001]. Red., oprac. wersji polskiej: J. Gaździcki. *Geodeta*, *3*, 29-32.

Geoforum.pl. (2019a). *Od GIS do SDI*. Retrieved from https://geoforum.pl/strona/46814,46843,46927/gis-krotki-wyklad-sdi-inaczej-od-gis-do-sdi (accessed 1 August 2019).

Geoforum.pl. (2019b). *Przyszłość SDI*. Retrieved

from https://geoforum.pl/page/46814,46843,46927/gis-krotki-wyklad-sdi-inaczej-od-gis-do-sdi (accessed 1 August 2019).

Ghatasheh, N. (2015). Knowledge Level Assessment in e-Learning Systems Using Machine Learning and User Activity Analysis. *International Journal of Advanced Computer Science and Applications*, *6(4)*, 107-113.

Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., Harter, D. (2001). Intelligent tutoring systems with conversational dialogue. *AI Magazine*, *22(4)*, 39-52.

Gryglewicz-Kacerka, W., Duraj, A. (2013). *Projektowanie obiektowe systemów informatycznych. Część II.* Włocławek: Państwowa Wyższa Szkoła Zawodowa we Włocławku. [*Object-oriented design of information systems. Part II Włocławek: State Higher Vocational School in Włocławek.*]

Hoppe, U., Verdejo, M. F., Kay, J. (2003). *Artificial intelligence in education: shaping the future of learning through intelligent technologies*. Amsterdam, Netherlands Washington, D.C.: IOS Press; Tokyo, Japan: Ohmsha.

IBM. (2019). *Design an SOA solution using a reference architecture*. Retrieved from https://www.ibm.com/developerworks/webservices/library/ar-archtemp/index.html (accessed 1 August 2019).

IEEE. (2003). *Std 1484.1-2003: IEEE Standard for Learning Technology -Learning Technology Systems Architecture (LTSA)*. [S.l.]: IEEE.

ISO/IEC 42010:2007 *Systems and software engineering - Recommended practice for architectural description of software-intensive systems* ISO/IEC/IEEE 42010:2011 *Systems and software engineering - Architecture Description*

Khan, M. A., Rehman, F. U. (2014, February 20). An Extendable Open Source Architecture of e-Learning System. In IEEE Computer Society; Institute of Electrical and Electronics Engineers, (ed.) *2013 Fourth International Conference on e-Learning Best Practices in Management, Design and Development of e-Courses: Standards of Excellence and Creativity*. Publisher: IEEE. doi: 10.1109/ECONF.2013.51

Kołodziński, E., Betliński, G. (2007). Modelowanie systemów informacji przestrzennej z zastosowaniem uogólnionej klasy obiektów. [*The modeling of spatial information systems with the use of Generalized Class of Objects*] *Roczniki Geomatyki*, [*Annals of Geomatics*] *5(2)*, 39-[53].

Lofland J., Snow, D., Anderson, L., Lofland, L. (2009). *Analiza układów społecznych*. [*Analyzing Social Settings*] (E. Hałas, A. Kordasiewicz, S. Urbańska, M. Żychlińska, Trans.). Warszawa: Wydawnictwo Naukowe SCHOLAR. [Warsaw: Scholar Publishing House]

Marciniak, J. (2014). Building E-Learning Content Repositories to Support Content Reusability. *International Journal of Emerging Technologies in Learning (iJET)*, *9(3)*, 45-52.

Marciniak, J. (2015). *Inteligentne systemy e-learningowe wykorzystujące ontologie typu wordnet*. [*Intelligent e-learning systems using wordnet based ontologies*]. Poznań: Wydawnictwo Naukowe UAM. [Poznan: UAM Scientific Publishing House]

Marciniak, J. (2016). Inteligentne systemy e-learningowe jako przykład wykorzystania sztucznej inteligencji w kształceniu na odległość. [*Intelligent E-Learning Systems as an Example of Application of Artificial Intelligence in*

*Distance Learning*] *EduAkcja. Magazyn edukacji elektronicznej*, [*EduAction. Electronic Education Magazine*] *2(12)*, 87-101.

Mendozaab, R., Baldirisab, S., Fabregat, R. (2015). Framework to Heritage Education Using Emerging Technologies. *Computer Science Procedures, 75*, 239-249.

Mousavinasab E., Zarifsanaiey, N., Kalhori, S. R. N., Rakhshan, M., Keikha, L., Saeedi, M. G. (2018, December 18). Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments.* doi: 10.1080/10494820.2018.1558257 [Published online: 18 Dec 2018]

Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education, 10*, 98-129.

Noskova, T., Pavlova, T.,Yakovleva, O. (2018). Study of students' educational activity strategies in the social media environment. In E. Smyrnova-Trybulska (Ed.). *E-learning and smart learning environment for the preparation of new generation specialist: monograph* “E-Learning”. 10. (pp. 113-123). Katowice - Cieszyn: Studio Noa; for University of Silesia in Katowice.

Nwana, H. S. (1990). Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, *4(4)*, 251–277.

Pahl, C. (2008). *Architecture Solutions for E-Learning Systems*. IGI Global.

Pattnayak, J., Pattnaik, S., Dash, P. (2017). Knowledge Management in E-Learning. A Critical Analysis. *International Journal of Engineering And Computer Science*, *6(5)*, 21528-21533.

Phobun, P., Vicheanpanya, J. (2010). Adaptive intelligent tutoring systems for e-learning systems. *Procedia - Social and Behavioral Sciences, 2(2)*, 4064-4069.

Plebaniak, R. (2019). *Modelowanie i analiza systemów informatycznych* [*Modeling and analysis of information systems*] *[Lecture]*. Retrieved from http://math.uni.lodz.pl/~robpleb/wyklad1_3.pdf (accessed 1 August 2019).

Polson, M. C., Richardson, J. J. (1988). *Foundations of intelligent tutoring systems*. Hillsdale (New Jersey-USA): Lawrence Erlbaum Associates Publishers.

Psotka, J.J., Massey, L.D., Mutter, S.A. (1988). *Intelligent tutoring systems:*

*Lesson learned*. Hillsdale (New Jersey-USA): Lawrence Erlbaum Associates Publishers, 1988.

Rawat, B., Dwivedi, S. K. (2017). Architecture for Recommendation of Courses in E-learning System. *Information Technology and Computer Science*, *4*, 39-47.

Russell, S.J., Norvig, P. (2016). *Artificial intelligence: a modern approach*. Boston [etc.]: Pearson Education Limited.

Santhi Dr. R., Priya B., Nadhini J. M.: Review of intelligent tutoring systems using bayesian approach. *arXiv:1302.7081*. Retrieved from https://arxiv.org/abs/1302.7081 (accessed 1 August 2019).

Sharma, S., Ghorpade, S., Sahni, A., Saluja, N. (2014). Survey of Intelligent Tutoring Systems: A Review on the Development of Expert/Intelligent Tutoring Systems, various Teaching Strategies and Expert Tutoring System Design Suggestions. *International Journal of Engineering Research & Technology (IJERT)*, *3(11)*, 37-42.

Smyrnova-Trybulska, E. (2018). Smart university in smart society – some trends. In E. Smyrnova-Trybulska (Ed.). *E-learning and smart learning environment for the preparation of new generation specialist : monograph* (pp. 65-80). Katowice: Studio Noa; Cieszyn: for University of Silesia in Katowice. Faculty of Ethnology and Sciences of Education.

Sommerville, I. (2019). *Inżynieria oprogramowania. Rozdział 10.* [*Software Ingineering. Chapter 10*] *[Lecture]*. Retrieved from www.mif.pg.gda.pl/kft/Rozdz10.ppt (accessed 1 August 2019).

Sommerville, I. (2000). *Software Engineering 6th Edition*. Addison Wesley.

Soofi, A. A., Uddin, M. (2019). A Systematic Review of Domains, Techniques, Delivery Modes and Validation Methods for Intelligent Tutoring Systems. *International Journal of Advanced Computer Science and Applications*, *10(3)*, 99-107.

Sudhana, K. M., Cyril, V., Ravi, T. (2013). An Architectural-model for Context aware Adaptive Delivery of Learning Material. *International Journal of Advanced Computer Science and Applications*, *4(10)*, 80-87.

Tadeusiewicz, R. (2019). *Analiza i projektowanie systemów informacyjnych* [*Analysis and design of information systems*] *[Lecture]*. Retrieved from https://www.uci.agh.edu.pl/uczelnia/tad/aipsi/index.php?id=psi (accessed 1 August 2019).

VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, *46(4)*, 197–221.

Wenger, É. (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Los Altos, CA: Morgan Kaufmann Publishers, Inc.

Wolski, M. (2019). Dokumentacja wymagań oparta na modelu. *[Model-based requirements documentation*] *Wolski*. Retrieved from https://www.michalwolski.pl/2015/05/ dokumentacja-wymagan-oparta-na-modelu/ (accessed 1 August 2019).

Zhang, J., Ren, F., Chen, T. (2006, February 27). Intelligent tutoring systems: Research status and its development in China. In IEEE (ed.) (2006). *2005 International Conference on Natural Language Processing and Knowledge Engineering*. Publisher: IEEE. doi: 10.1109/NLPKE.2005.1598823.

Żeliński, J. (2011). Cloud Computing to architektura system. *Jarosław Żeliński IT-Consulting*, blog post, 11 October. Retrieved from https://it-consulting.pl/autoinstalator/wordpress/?s=architektura+systemu (accessed 1 August 2019).

*Citation:* Szulc, J. (2019) Models of E-Learning Systems Architecture Using AI Components In E. Smyrnova-Trybulska (ed.) *E-Lerarning and STEM Education,* „E-learning", 11, (pp. 295-320) Katowice-Cieszyn: Studio Noa for University of Silesia.