Building Cloud Data Interchange Services for E-Learning Systems: Applications on the Moodle System

Alina Andreica[1] and Fernando Paulo Belfo[2]
[1]*Babes-Bolyai University, Cluj-Napoca, Romania*
[2]*Instituto Politécnico de Coimbra, Coimbra, Portugal*

Keywords: Learning Components, Learning Objects, Equivalence and Simplification Algorithms, Data Interchange, Cloud Services, Database Representation, Pattern Matching, Software Design.

Abstract: The cloud data interchange services among various information systems still have huge possibilities to evolve. We focus on applying data interchange principles that we have proposed for cloud environments in order to perform data exchange for e-learning systems. Our case study is based on the moodle system. Developing and improving data interchange standards for learning objects contributes to improving processing and exchanging techniques of digital content used for teaching, learning, or training among various e-learning systems. Equivalence algorithms and canonical representation are used in order to ensure the uniform representation in the cloud database. The solution we describe, designed for cloud architectures, has important advantages in e-learning systems communication and educational institutions cooperation, since different institutions, using different e-learning systems, do not have automatic means of exchanging learning information. Therefore, these techniques for representing and exchanging learning objects facilitate their sharing among different stakeholders.

1 INTRODUCTION AND WORKING FRAMEWORK

Developing standards for representing and exchanging learning resources among different stakeholders involved in educational processes bring them many practical advantages. Content authors may want their work to be easily spread and used, or they may want to collaborate with other co-authors on the same content (Andreica, Synasc 2017). Beneficiaries may want to be able to easily choose, render, or buy learning content, in order to learn more effectively. Standardizing the way in which learning objects are represented contributes to increasing their availability, accessibility, in a way they can be valued and, therefore, they can be easily transacted between potential beneficiaries and providers.

In order to exchange learning resources between different parties, the information systems have to be interoperable. Interoperability is the capability of different systems to share functionalities or data (Olmedilla et al, 2006). System interoperability has been approached on business models (Ziemann, 2010) and other various models (Morris et al, 2004).

Andreica, Covaci and Küng overviewed interoperability layers, principles and tools (Andreica et al, 2015), including: the Electronic Data Interchange (EDI) model (Adams et al, 2002), the XML standard (XML standard, 2015), RosettaNet (Rosetta, 2015), ebXML (OASIS, 2015) standards. Standards for e-learning objects and components are given in (SREB, 2007), (IMS, 2010). The Open Internet of Things standards (OpenIoT, 2015) may also be used as an efficient framework for data interchange (Andreica, Synasc 2017).

A widely used definition for the learning object (SREB, 2007) is the one proposed by the Institute of Electrical and Electronics Engineer's (IEEE): "any entity, digital or non-digital that can be used, re-used or referenced during technology supported learning" (IEEE, 2002). This definition is a very comprehensive one, anchored on the idea of "technology supported learning", yet a bit too large to be practically used for learning objects or components within e-learning system environments.

Andreica, A. and Belfo, F.
Building Cloud Data Interchange Services for E-Learning Systems: Applications on the Moodle System.
DOI: 10.5220/0006783905650572
In *Proceedings of the 8th International Conference on Cloud Computing and Services Science (CLOSER 2018)*, pages 565-572
ISBN: 978-989-758-295-0

More e-learning definitions are discussed in section 3.

Learning objects can have different formats and can either consist of simple assets as text, images or short videos, or can comprise more complex resources, with several components, combined in order to deliver meaningful learning experiences (Andreica, Synasc 2017). These components may consist of combinations of text, images, simulations, videos, assessment exercises or several other ways of achieving an educational objective (IMS, 2010).

In order to handle these composition aspects, the SREB-SCORE standard defines two levels of learning items (SREB, 2007): the first level is composed by assets and information objects and the second level is composed of learning objects and learning components. Assets are electronic representations of media, text, images, sounds, web pages and other reusable pieces of data. An information object is composed of assets that focus on a particular part of information, which can illustrate a principle, explain a concept or describe a process. The learning objects cover assets and information objects that teach a single concept or lesson (Andreica, Synasc 2017). The components of learning objects are their objectives, assessments, practices, concepts, but also their lessons, modules and courses (SREB, 2007).

We use equivalence algorithms in order to set the canonical representation of the exchanged data in the cloud database (Andreica, 2017).

The current paper applies the proposed framework for using simplification and equivalence algorithms in modelling data representations (Andreica, 2017) for learning objects and components within the case study performed on moodle system. We apply these techniques in designing data interchange service among various information systems, in particular e-learning systems, within cloud environments.

Within section 2 we overview the means we have proposed for implementing simplification and equivalence algorithms on various entities, including hierarchical structures and techniques for solving specific pattern matching problems using canonical representatives. Within section 3, we focus on e-learning objects and components which appear in the two moodle e-learning system instances that we use as case studies. Section 4 addresses principles of data interchange between information systems using a cloud database, which retains data in a canonical representation. We describe the structures for the entities and attributes to be used in the local and cloud databases and we discuss the results of transferring course related tables between different moodle databases. Conclusions reveal the most important topics presented in the paper and future research and development directions.

2 EQUIVALENCE ALGORITHMS AND PATTERN MATCHING PRINCIPLES

Within this section we overview the methods we propose (Andreica et al, 2010), (Andreica et al, 2012) for implementing equivalence algorithms at the database level for various entities and entity classes, including hierarchical structures, the entities being retained in database tables. This solution is very useful, since we often have to process large volumes of data retained in databases.

The equivalence algorithms we have implemented are based on the theoretical framework given in (Buchberger and Loos, 1982).

Principles for retaining (using ascendant / successor pointers) and processing hierarchical structures, postorder type n-ary tree parsing algorithms and a comparison of their processing techniques are presented in (Andreica et al, 2010). We prove that the function implementing the equivalence test for two entities based on the property that they have the same canonical representative is much more efficient than the one using the definition (Andreica et al, 2010).

In (Andreica et al, 2012) we discuss the case study of equivalent disciplines and processing modules of didactic activities. In (Andreica, 2016) we approach the case of an expert system for plant therapy. In (Andreica, 2017) we propose principles for designing uniform database structures in order to build cloud data interchange services between information systems. In (Andreica, Synasc 2017) we add the framework of e-learning systems.

The principles for implementing equivalence algorithms on categories of entities are also described in (Andreica et al, 2010), (Andreica et al, 2012). Intuitively, the canonical set of a category of entities may be obtained by "flattening" its category sub-tree and computing the union set of all canonical sets of its descendant leaf entities. In the case of categories of entities, the canonical representative is recursively computed (Andreica, 2017).

In many practical cases, entity equivalences or even canonical elements have to be mapped, sometime with user assistance. Therefore, in (Andreica, 2016) and (Andreica, 2017), we address mappings and pattern matching issues. We overview here the principles we proposed in this respect.

We designed pattern matching rules for equivalent entities by reducing the mapping between two elements, belonging to two equivalence classes that are to be mapped, to mapping their canonical representatives (Andreica, 2016) and (Andreica, 2017). The formalization is given in (Andreica, 2016), (Andreica, 2017).

Intuitively, instead of dealing with the equivalence of any elements between two classes, we reduce the problem to the simpler one of dealing with the equivalence of the two classes' representatives – see figure 1.

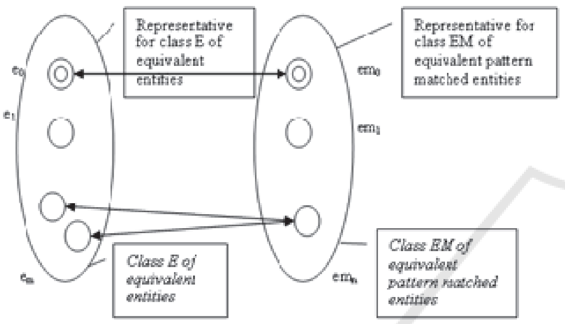

Figure 1: Pattern Matching scheme for Equivalence classes (Andreica, 2017).

For the case of equivalence classes with hierarchical representations, the canonical representatives of the classes are the roots of the corresponding trees (Andreica, 2017)

3 LEARNING RESOURCES, OBJECTS AND COMPONENTS. CASE STUDIES FOR MOODLE E-LEARNING SYSTEMS

E-learning is defined by Clark and Meyer as "instruction delivered on a digital device such as a computer or mobile device that is intended to support learning" (Clark & Mayer, 2016). The IEEE perspective on learning objects views them as entities, digital or not, which can be used during technology supported learning, this technology including or not computers or mobile devices.

We consider that a more digital and computer targeted definition for learning objects, would better support the e-learning field. In this respect, the definition proposed by David Wiley, as "any digital resource that can be reused to support learning" is more applicable for the e-learning (Ritzhaupt, 2010; Wiley, 2000), since it views the learning objects as digital ones, which and be delivered across any network.

In the e-learning field it is important how information and communication technologies are used to facilitate learning, especially over the Internet, the management of learning objects being therefore an important task.

We further present types of learning resources within two moodle case studies systems, the final goal being to discuss means of exchanging learning objects. Moodle, a "learning platform designed to provide educators, administrators and learners with a single robust, secure and integrated system to create personalized learning environments". Moodle is an open source software, provided freely under the GNU General Public License (Moodle, 2016).

The first case study system is used within the Institute of Accounting and Administration in Coimbra, Portugal, one of the schools of Polytechnic Institute of Coimbra. Moodle platforms manages different types of learning objects (see figure2), providing services for creating and uploading learning objects.

The types of objects displayed in the figures below are to be mapped using the mapping procedures discussed in section 4.

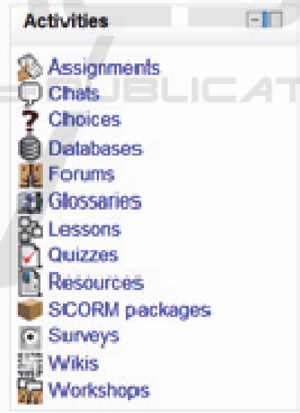

Figure 2: Types of learning objects in moodle.

Figure 3 and 4 present respectively features of the moodle platform regarding browsing repositories by course titles (figure3) and listing educational resources (figure 4).

Yet, as it is publically stated on the Moodle official site, there are several issues which are not yet solved and have to be discussed in the future, for example, metadata, tracing student learning and searchable repository services.

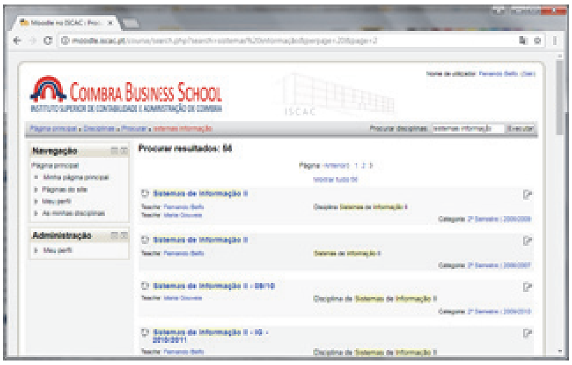

Figure 3: Browsing repositories by course title.

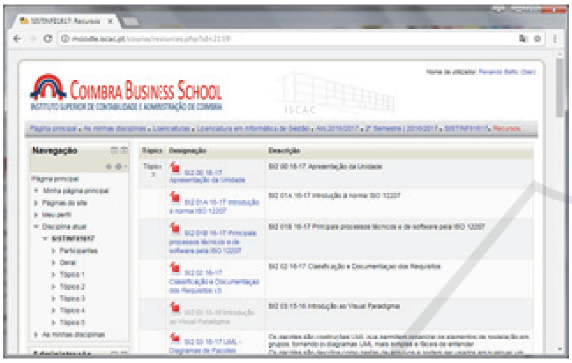

Figure 4: Educational resources listed in ISCAC Moodle.

The second system is a moodle test instance installed within Babes-Bolyai University in order to test data transfer between the two systems.

Within this system, we have created an IT strategies course both for educational purposes, and for data exchange ones. Figures 5-6 present the way in which educational resources have been added within this course. Unlike previous example, educational files have been added as links.

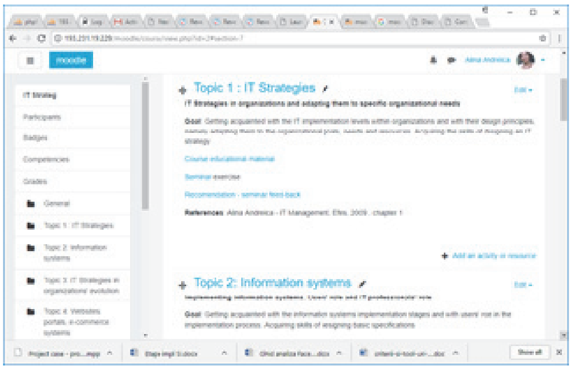

Figure 5: Course topics in design mode, Moodle instance - Babes-Bolyai University.

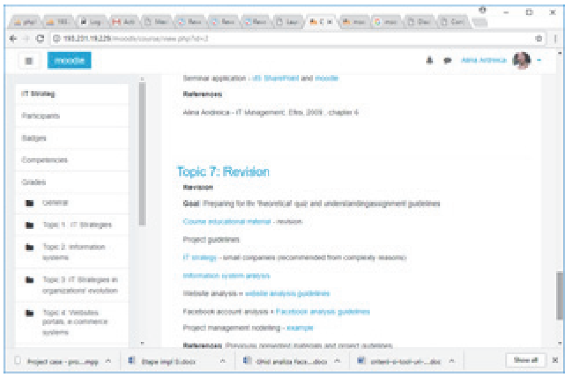

Figure 6: Course topics in reading mode, Moodle instance - Babes-Bolyai University.

4 PRINCIPLES FOR BUILDING THE CLOUD DATABASE REPRESENTATION AND FOR TRANSFERING DATA

We have previously addresses the topic of building the cloud database representation in (Andreica, 2016), (Andreica, 2017), (Andreica, Synasc 2017). Within this section, we overview the previously stated principles, including data structuring principles, and give further details on applying them on learning objects and components.

The cloud database retains the canonical representation of the exchanged data and is used as a communication proxy. The exchanged data consists of learning objects as the ones presented in section 3. We have designed the cloud database as a communication proxy in order to increase communication efficiency and to retain the canonical representation of the exchanged data. Therefore, if systems S_1 and S_2 have communicated and S_2 and S_3 have also communicated, then their mappings to the canonical representation have already been performed, consequently S_1 and S_3 can communicate straightforwardly, without any supplemental mappings (Andreica, Synasc 2017). The mapping process is user assisted, requiring human input.

The communication principle previously stated is also applied in the case of the systems presented in section 3 – the Moodle system of Coimbra Business School and the Moodle test instance installed within Babes-Bolyai University. Moreover, we have also created a blank moddle database in the test instance installed within Babes-Bolyai University in order to test table transfer.

The data exchange sequences between a source system Ss and a destination system Sd via the cloud database Scd pursue the following stages (Andreica, 2017): data to be exchanged is marked in the Ss database using dedicated tables and columns, mapped into the Scd database, and then sent and retained into Scd. For the destination system database Sd, which sends data requests, a data mapping is also performed to Scd and required data is sent from the Scd into the destination one Sd. The exchange framework is graphically represented in figure 7 (Andreica, 2017).

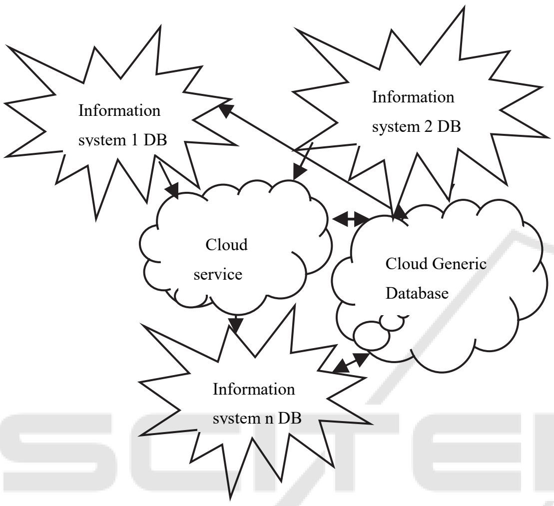

Figure 7: Data Interchange Model Using Cloud Services (Andreica, 2016).

The data exchange uses multi-criteria agents (Weiss, 1999) implemented in the cloud environment for performing necessary mappings and for handling communication (Andreica, 2016). The agent architecture includes self-adapting communicating objects, working on distributed datasets and supporting the exchange and analysis of distributed sources (Andreica et al, 2015).

In the proposed architecture (Andreica, 2017), when agents communicate, they may decide to cooperate on a given task. In this respect, they make a collective commitment to each other. Committing to another agent involves agreeing to pursue the given goal-data interchange (Andreica, 2017). The agent architecture is based on the multi-agent model proposed in (Faulkner et al, 2014)

In (Andreica, 2017) we propose a database model and structure for processing the entity equivalence from tables in various local databases, using the uniform canonical representation in the cloud database. The formalization of the local and cloud database structures and their mapping are given in (Andreica, 2017).

The database structures we propose in the local and cloud databases are described in details in (Andreica, 2017), (Andreica, Synasc 2017).

Learning objects are retained for each learning component, according to its characteristics. They may usually include: files, texts, images, links, documents (Andreica, Synasc 2017).

One of the most important learning component is the moodle lesson or learning module. In the examples given in section 3, lessons / topics in Moodle contain the following objects: *Title* – text; *Goal* – text; *Learning materials* – list of learning files or links to them; *References* – text (may include links*); Evaluation information* – text, learning file or link to a file. In the ISCAC moodle system, evaluation information is included in the learning materials.

In order to prepare the testing framework, we have installed a moodle test instance within Babeş-Bolyai University. In figure 8 we display the communication tables from the local database in this moodle test instance 193.231.19.229/moodle . Figure 9 presents the structure and content of the moodle block recent activity table.

We have studied the Moodle database schema (Moodle DBS, 2017) (Moodle Courses Schema, 2017). The course component of this schema is presented in figure 10. Further on, we have transferred the moodle block recent activity table into a blank moodle database, called moodletest, using an intermediate XML structure, namely the table conversion to XML and vice-versa – see figure 11. In (Prasad, 2017), Prasad proposes the SQL code sequence for transferring a table into the corresponding XML structure and from the XML structure into the corresponding table. In our example, into the new table,there were inserted 25 rows and the query took 0.0636 seconds – see figure 11.

5 CONCLUSION AND FUTURE WORK

The paper applies the data representation and design principles we have proposed for performing data interchange between various information systems databases, by means of cloud services, for e-learning systems, the case studies being implemented on moodle systems.

Simplification and equivalence algorithms are used in order to ensure canonical data representation in the cloud database and data mapping in the data exchange process. A specific structure for the entities / tables and attributes that are to be exchanged is used in the local and cloud databases. Pattern matching rules and canonical representatives are used in the cloud database.

We reveal the advantages of applying the algebraic equivalence algorithm and canonical representatives' properties in designing data interchange services for various information system cases, the present paper focusing on e-learning systems, with case studies on moodle system.

The data interchange model we have developed provides important practical advantages for increasing organizational competitiveness in the educational field, as well as between educational actors and other stakeholders, with a significant societal impact on institutional cooperation and improvement of learning processes, efficient information access, learning and communication for various stakeholders.

The solution we have designed is flexible and efficient, only relevant data being exchanged, minimizes resource usage and has significant security benefits.

Future work is related to further development and implementation of the above described techniques on e-learning systems and in other fields as well.

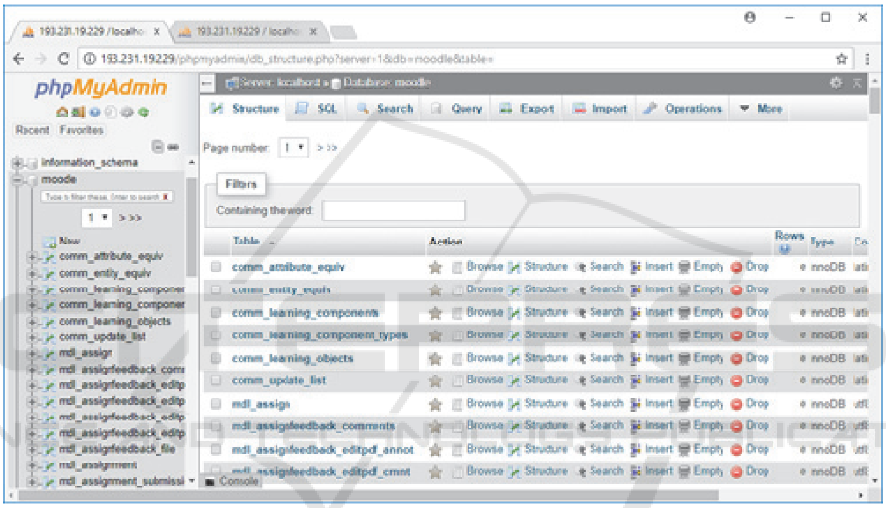

Figure 8: Communication tables in the local moodle database.

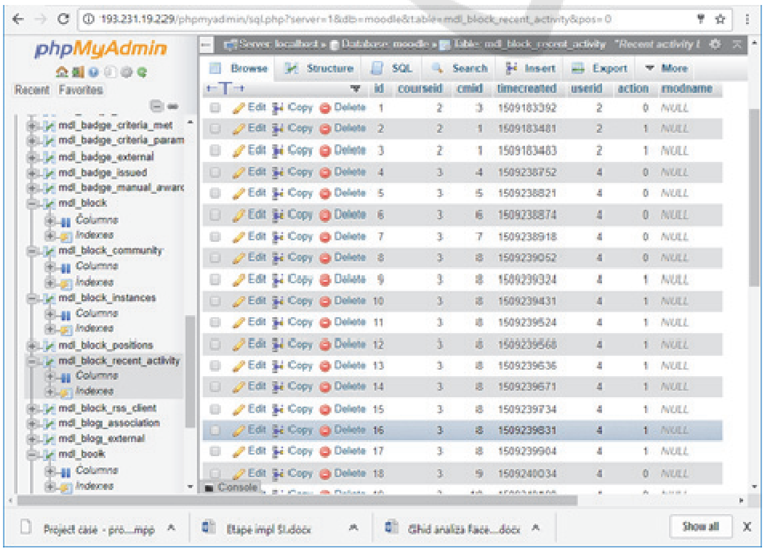

Figure 9: Moodle block recent activity table, test instance http://193.231.19.229/moodle.

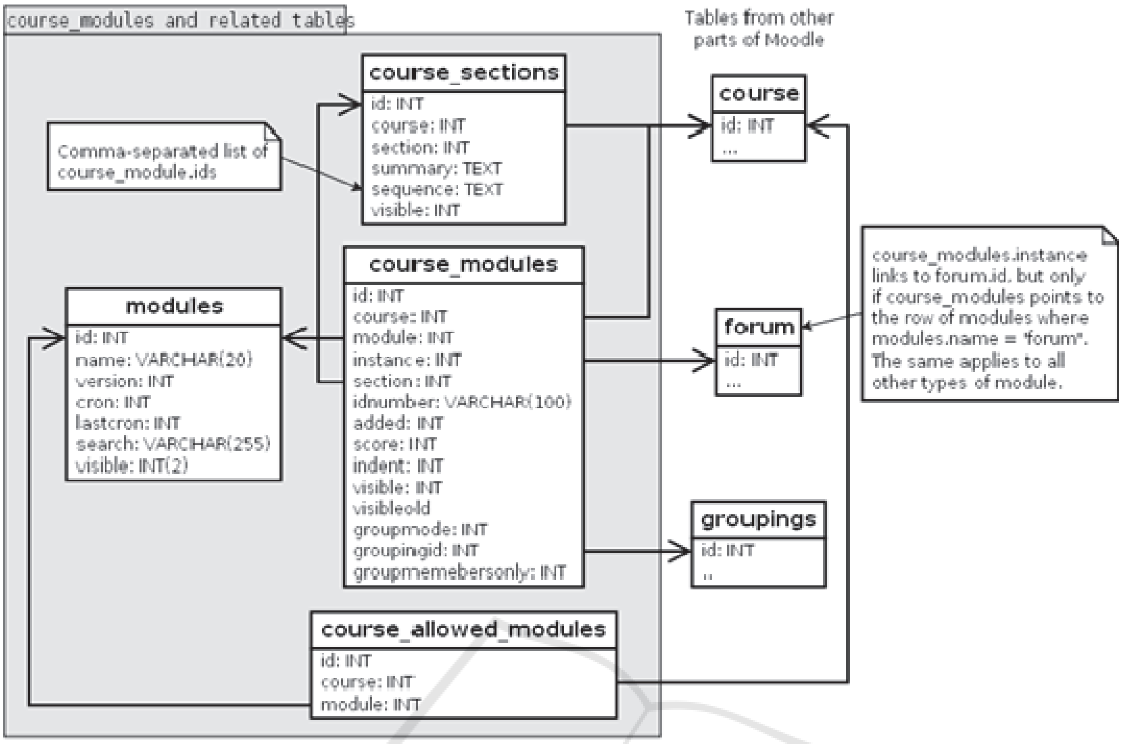

Figure 10: Moodle course tables; Source: https://docs.moodle.org/dev/Database_schema_introduction#Courses_and_their_organisation_into_categories

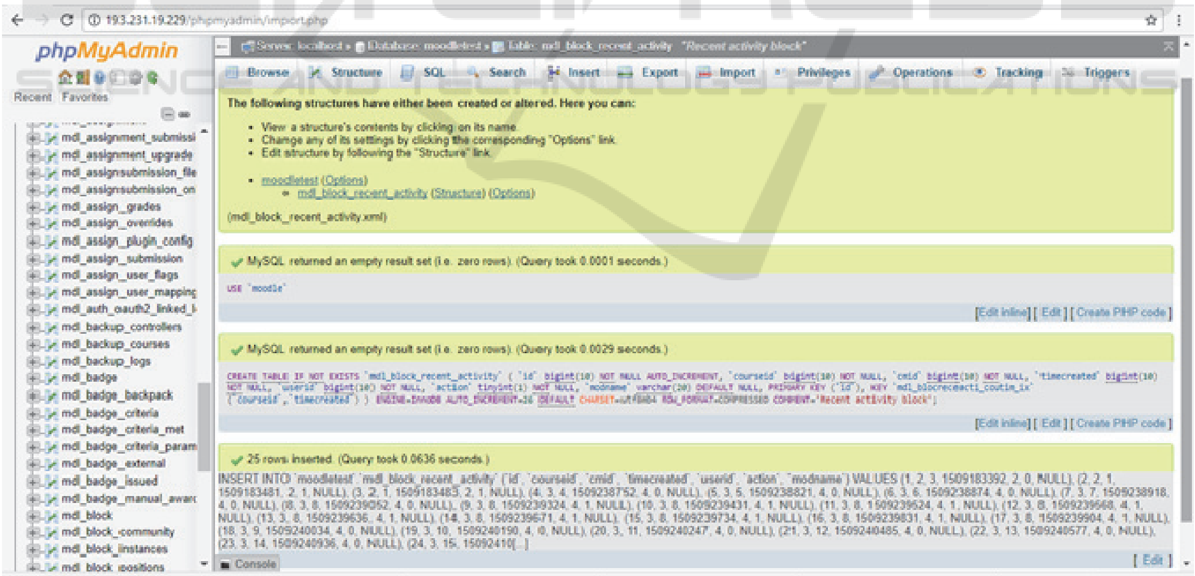

Figure 11: Table transfer for Moodle block recent activity table by means of an XMLstructure inti moodletest datadase.

REFERENCES

Andreica, A, Stuparu, D, and Miu, C., 2012. *Applying Mathematical Models in Software Design*, 2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, Proceedings of ICCP 2012, IEEE, Ed: Ioan Alfred Letia, p.87-90

Olmedilla, D., Saito, N., and Simon, B. eds. (2006). “Educational Technology & Society”, Special Issue on Interoperability of Educational Systems, vol 9

Andreica, A, Stuparu, D, and Miu, C., (2010). “Design Techniques in Processing Hierarchical Structures at Database Level”, *Proceedings of Iadis Information Systems 2010*, Porto, 18-20 March 2010 , IADIS Press, Ed: M Nunes, P Isaias, P Powell, p. 483-488

Andreica, A, Covaci, F. and Küng, J. (2015). *A Generic Model for Cloud Data Interchange*, Proceedings of 14h RoEduNet International Conference - IEEE, Craiova, 24-26 September 2015, IEEE Computer Society, p. 138-142

Andreica A **(**2017) *Designing Uniform Database Representations for Cloud Data Interchange Services*, 7th International Conference on Cloud Computing and Services Science, 24-26 April 2017, Porto, Portugal, Proceedings of CLOSER 2017, SCITEPRESS – Science and Technology Publications, Editor: Donald Ferguson, Víctor Méndez Muñoz, Jorge Cardoso, Markus Helfert and Claus Pahl, P. 554-559

Andreica, A, Belfo F P, Covaci F., **(**2017) Designing Data Interchange Services for Learning Components, paper presented at Symbolic and Numeric Algorithms for Scientific Computing Synasc 2017

Adams, S, Hardas, D, Iossein, A. and Kaiman,. C. (2002) *BizTalk Unleashed*. Indianapolis, Indiana: Sams Publishing. p. 966

XML standard, http://www.w3.org/TR/xml11/#charsets , retreived Nov 2015

Rosetta – Rosettanet Overview: Clusters, Segments, and PIPs (ver 02.13.00), 2011, retrieved December 2015 http://www.rosettanet.org/TheStandards/RosettaNetStandards/PIPOverview/tabid/3482/Default.aspx

OASIS – OASIS ebXML Messaging Services Version 3.0: Part 1, Core Features, 2007. http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/ebms_core-3.0-spec.pdf, retreived Dec 2015

Stumme, G., Wille, R. (2000). „*Begriffliche Wissensverarbeitung / Conceptual Knowledge Processing*“, Springer Verlag,

OpenIoT - Open Internet of Things architecture https://github.com/OpenIotOrg/openiot/wiki/OpenIoT-Architecture , retreived Dec 2015

Buchberger B. and Loos, R. (1982) *Algebraic Simplification*, Computing, Suppl. 4, Springer Verlag, p.11-43

Andreica, A (2016) *Applying Equivalence Algorithms in Solving Pattern Matching Problems. Case Study for Expert System Design*, Proceedings of International Conference on Theory and Practice in Modern Computing- TPMC, July 1-4, 2016, Portugal

Andreica, A, Stuparu, D. and Mantu, I. (2005). “Symbolic Modelling of Database Representations”, *International Symposium on Symbolic and Numeric Algorithms for Scientific Computing 2005*, IEEE Press, p 59-62

Morris, E., Levine, L, Meyers, C., Place, P. and Plakosh. D. (2004) “*System of Systems Interoperability (SOSI): Final Report*”, Carnegie Mellon Univ., Software Engineering Institute, http://www.sei.cmu.edu/reports/04tr004.pdf , accessed May 2016

Weiss, G. E, (1999). “*Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*”, MIT,

Faulkner, S, Kolp, M, Nguyen, Tai, Coyette, A, Do, T., 2014. “*Information Integration Architecture Development: A Multi-Agent Approach*”, retr Nov 2016

Clark, R. C., & Mayer, R. E., 2016. E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning (Third ed.). San Francisco: Pfeiffer.

Harman, K., & Koohang, A., 2007. Learning objects: Applications, implications, & future directions. Santa Rosa, California: Informing Science Press.

IEEE, 2002. 1484.12.1 Learning Object Metadata Standard.

IMS, 2010. IMS Learning Object Discovery & Exchange (LODE). IMS Global Learning Consortium. 2017, from https://www.imsglobal.org/lode/index.html

Moodle, 2017. *Moodle Documentation*. Retrieved July 16, 2017, from https://docs.moodle.org/33/en/About_Moodle, https://docs.moodle.org/23/en/Activities

Moodle Courses Schema, 2017. *Moodle Courses Database Organization*, retreived October 2017 from https://docs.moodle.org/dev/Database_schema_introduction#Courses_and_their_organisation_into_categories

Moodle Database Schema, 2017. Retrived October 2017 from http://www.examulator.com/er/

Prasad, D, 2017. *How to convert table data to XML and vice-versa*, retreived October 2017 from http://sqlserverlearner.com/2013/how-to-convert-table-data-to-xml-and-vice-versa

Ritzhaupt, A, 2010. Learning Object Systems and Strategy: A description and discussion. Interdisciplinary Journal of e-learning and Learning Objects, 6(1), 217-238.

SREB, 2007. Learning Object Metadata - SREB-SCORE Initiative. Atlanta: Southern Regional Education Board.

Wiley, D. A, 2000. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In David A. Wiley (Ed.), The instructional use of learning objects: Online Version.

Ziemann, J., 2010. “*Architecture of Interoperable Information Systems - An Enterprise Model-Based for Describing and Enacting Collaborative Business Processes*”, Logos Verlag, Berlin