



International Conference on Communication, Management and Information Technology (ICCMIT 2015)

## Developing a Computer Programming MOOC

Natalia Spyropoulou<sup>a\*</sup>, Gerasimoula Demopoulou<sup>b</sup>, Christos Pierrakeas<sup>a,b</sup>, Ioannis Koutsonikos<sup>a,b</sup>, Achilles Kameas<sup>a</sup>

<sup>a</sup>*Hellenic Open University (HOU), Patras 26225, Greece.*

<sup>b</sup>*Dept. of Business Administration, Technological Educational Institute (TEI) of Western Greece, Patras 26334, Greece.*

---

### Abstract

Open World Learning movement opens up opportunities for the collaboration between institutions, educators and learners globally, and for enabling more meaningful engagement in teaching and learning. Massive Open Online Courses (MOOCs), one of the latest internet revolutions, boasting tens or hundreds of thousands of participants worldwide, and, hence, present an intense interest by the research and education community. The purpose of this research is to explore the perspectives of a team-based methodology for the implementation of MOOCs. The paper focuses on the development of a computer programming MOOC, illustrating the development process.

© 2015 The Authors. Published by Elsevier B.V.  
Peer-review under responsibility of Universal Society for Applied Research.

*Keywords:* MOOCs, Open World Learning, Online Learning, Computer Science, Computer Programming, team-based methodology,

---

### 1. Introduction

Open education brings new opportunities for innovation in education that will not only support institutions to implement the fundamental values of university based education but it will also shift the focus from traditional lecturing to more learner-centered learning<sup>28</sup>. The implications of the mainstream use of open learning challenge existing models of research and teaching. Open learning brought the validity of existing educational systems into question. One important concern is due to the consensus that traditional ways of providing learning opportunities are

---

\* Corresponding author. Tel.: 0030-2610-367963.  
*E-mail address:* [nspirop@eap.gr](mailto:nspirop@eap.gr)

no longer adequate to equip teachers, students and workers with the competences required to participate successfully in the emerging knowledge-based society<sup>4</sup>.

Massive Open Online Courses (MOOCs) is one of the most innovative movements within distance and open education. Several empirical evidence and results has been demonstrated extensively in concerning the effect in higher education and MOOC pedagogy<sup>27</sup>. On the other hand, there is not much research literature available on MOOCs regarding the aspect of creator/institutions or other technological aspects<sup>14</sup>, therefore, the quality of MOOC design should be investigated in more detail. The purpose of this research is to explore the strengths, perspectives and opportunities of a collaborative team-based methodology for the implementation of MOOC. This paper emphasizes the instructional design of the MOOC by providing a case study of a computer programming MOOC.

The rest of the paper is structured as follows; in Section 2 we discuss the movement of open world learning, Computer Programming education and MOOCs. In Section 3 we describe the proposed methodology. In Section 4 we present the realization of the development of the MOOC entitled “Programming fundamentals using ANSI - C”. Finally, in Section 5 we discuss initial results about the evaluation of the proposed methodology.

## 2. Background

### 2.1. Open World Learning (OWL)

In the knowledge and information society we are living in, open access to education is highlighted as a key factor in addressing the multiple challenges of the global continuous changes. The information technologies have transformed to a significant extent the nature of work and the organization of production, causing new forms of learning and new ways of institutionalizing education focusing on open learning, according to which education is a right for all, throughout lifetime (Universal Declaration of human rights). Thus, the interest has grown steadily over the past years in making a university-level education openly available to students around the world who otherwise would not have this opportunity<sup>10</sup>. Towards to Open Educational Recourses (OER) recommendations for how to do this are well documented (e.g., UNESCO<sup>22</sup>, OECD<sup>24</sup>). Open World Learning movement opens up opportunities for exchanging ideas, collaborating between institutions, educators and learners worldwide, and for enabling more meaningful engagement in teaching and learning.

Yet, a number of the main aspects of openness are emerging in different areas, such as<sup>28</sup>

- Open Curriculum: learners mix educational resources, activities for different disciplines to meet their needs. Learners in charge of their own learning and they will choose what they need to learn to meet their personal objectives and requirements.
- Open Learning: instructors, learners, teaching assistants and/or peers will collaborate and engage in various activities, sharing their ideas and new findings during the learning process. This provides learners with opportunities for discussing, questioning and answering while making the learning process more interesting and effective.
- Open Assessment: the assessment of what learners have learned is carried out by their instructors, peers and others during the learning process via peer to peer or crowd-sourced assessment
- Open Platform: This provides a learning platform open for all, supporting a dynamic and interactive open education community, where any one get engaged and share their ideas.

Generally, debates and research around open education are increasingly extended focusing on the institutional, cultural, and pedagogical implications of adopting the open model rather than supporting focus on the resources themselves<sup>4</sup>. As Deimann and Farrow<sup>4</sup> indicate it is becoming increasingly evident that social frameworks are changing at a pace that is not satisfied by what most face-to-face educational institutions offer today as learning opportunities.

## 2.2. Computer Programming Education

Over the past years, computer science and information technology graduates have been in high demand. In the era of globalization, rapid technology development and knowledge-based economy, relevant jobs occupations have become an area of substantial interest. However, learning programming is generally considered a difficult achievement and courses with this subject area often present a high dropout rate and failure of learners.<sup>11, 13</sup>

A new trend of interest in online education appears on the computing community and is also a much stronger focus on teaching programming in the latest round of online educational initiatives<sup>23</sup>. The immense popularity of MOOCs alongside the collaboration between organizations (eg. Coursera, FutureLearn) and prestigious universities (e.g. MIT, Harvard, Berkeley) have brought a new action to a long-standing educational approach in programming. What the recent initiatives arguably contribute is large scale access to students, a new and highly interesting development in online education.<sup>23</sup>

## 2.3. Massive Open Online Courses (MOOCs)

According to the commonwealth of learning web<sup>2</sup>, MOOCs have been the most reported technology development in education over the past year<sup>1</sup>. European Commission defines a MOOC as “an online course open to anyone without restrictions, usually structured around a set of learning goals in an area of study, which often runs over a specific period of time on an online platform which allows interactive possibilities that facilitate the creation of a learning community. As it is the case for any online course, it provides some course materials and (self) assessment tools for independent studying”<sup>5</sup>.

Despite the critical debate which is brewing on their pedagogical effectiveness due to the high drop down rates of learners<sup>15, 20, 6</sup>, MOOCs have gained momentum attracting an immense number of learners from all over the world. MOOCs in general are offered to anyone willing to participate, enabling both open access and the participation of large numbers of students in online courses, hence creating additional challenges on making support resources available to participants. As McAuley<sup>16</sup> claims, this technological trend integrates the connectivity of social networking, the facilitation of an acknowledged expert in a field of study, and a collection of freely accessible online resources. Because of its openness, it can bring a diverse group of learners together regardless of their social and cultural background, while it enables geographically dispersed groups to collaborate and to create new forms of communication and collaboration for students and educators<sup>8, 12</sup>.

The growth of free access education and the increasing demand for higher education occurs specially in Science, Technology, Engineering and Math (STEM). MOOCs in computer programming can help to attract learners into this field. They are particularly relevant to software professionals in academia and industry because future researchers and practitioners will likely receive much of their education through MOOCs and associated digital learning methods and tools. The ability of MOOCs to connect experienced software professionals with motivated novices is a potent pedagogical combination. MOOCs also have the potential to help students personalize their learning experiences at a reasonable cost<sup>19</sup>.

## 3. Methodology

The proposed methodology describes a collaborative development of MOOC by a multidisciplinary team of professionals, as Cross<sup>3</sup> and McCallum et al.<sup>17</sup> propose. It adopts basic elements of the ADDIE model, in which process is iterative and self-correcting. Based on the collaborative development approach, in each phase, specific teams are collaborating. The team structure was based on Hixson<sup>9</sup> and McCallum et al<sup>17</sup>, whose proposals define a set of basic roles of members required for collaborative course development. The teams were formed from professionals and non-professionals (e.g. graduating students as teaching assistants), helping create MOOCs accessible to learners with different skills and abilities. The teams include:

- Course Manager team (Academic staff and teaching assistants): Responsible for the course design and implementation.

- Educational Support team (Scientific staff): Supports the Course Manager Team in all phases
- Technical Support team (Technical staff): Responsible for the operation, maintenance and technical support of the MOOC platform during the educational processes.
- Additional staff (eg. special multimedia developer) Responsible for several topics such as the media publications and support in the development of the educational material.

With the aim to assist the creation and implementation process, guidelines based on best practices were created<sup>21</sup>, in three major categories: the educational material, the course curriculum, and course implementation. The practices used by popular MOOCs such as Coursera, FutureLearn, Udemy, Udacity, edX and Iversity were investigated and emulated.

### 3.1. Methodology Steps

Regarding the content of the aforementioned methodology, we adopted the following five phases:

1. Analysis Phase: During the analysis phase, the educational topic of the course is analyzed, in order to specify the purpose of learning, the knowledge domain and the main learning goals. The teams involved in the analysis phase are the Course Manager Team, which is responsible to generate the phase outcomes and the Educational Support Team which assist and support the first team.
2. Design phase: During the design phase, learning activities and learning objects (LOs) are designed to demonstrate the way that the knowledge will be offered to the learners. The instructional design is not intended to create a standard format, but each course will be designed based on the needs that were uncovered in the analysis phase. In this phase, the Course Manager Team is involved with the collaboration of the Support Team.
3. Development Phase: In this phase, the learning objects are developed. The Course Manager Team collaborated with special multimedia developers (additional staff). The Educational Support Team assists the process while the Technical Support Team configured the MOOC Platform and integrated the educational material into the platform.
4. Implementation Phase: The educational process is implemented using the defined timeframe. A pilot course should run in order uncover matters that require improvement. During this phase, the Course Manager Team collaborates with the educational support team during the educational process. The Technical Support Team provides technical support throughout the educational process.

### 3.2. Evaluation

The evaluation of the proposed methodology is carried out in two levels-directions. The methodology process is evaluated in each step and the final evaluation take place at the end of the all phases. The purpose of the evaluation is to review the data that require improvement, through which the achievement of the success of the process are valued.

- Formative Assessment: The formative evaluation is presented in each stage of the process and includes collecting information in order to identify problems. Therefore, the formative evaluation includes (a) focus groups and interviews of the teams and (b) the pilot implementation. Furthermore, during the pilot implementation, questionnaires and/or interviews is given in order to consider any problems and improvements.
- Final Assessment: The final assessment aims to provide opportunities for feedback from users. Specifically, during the final evaluation questionnaires were given to registered users (either completed the courses or not). Interviews were held with the team members. The results of questionnaires and interviews coupled with useful information, from platform is analysed to provide useful feedback for information.

## 4. Developing the MOOC: Programming fundamentals using ANSI - C.

This section focuses on the implementation of the proposed methodology for the creation of a MOOC entitled “Programming fundamentals using ANSI – C”. The Course Manager Team was formed by two academic members of staff with engineering and programming background and one graduate student having background in pedagogy. The Educational Support Team was composed of two scientific collaborators, an e-learning services expert and a scientific collaborator with professional pedagogy background.

### 4.1. Analysis Phase

The Course Manager and Educational Support teams collaborate in the analysis phase in order to define the course description. Guidelines for MOOC development were given to the Course Manager Team, including the methodology procedure. The teams communicate mainly through emails, discussing ideas and issues concerning the process of implementation. The main course description follows.

The course trains in basic concepts of computer programming and examines their practical application through the development of programs in the programming language C. It also provides the necessary tools to manage the development process of structured programs both in theory by the presentation and analysis of the structured programming technique, and in practice by developing programs using an application development environment. A key element of procedural or structured programming is to build the program by repeatedly using subprograms, which either perform generic tasks or are addressing a part of the overall problem. The aim is to understand the fundamentals of programming and the consolidation of its philosophy, so that the learners will be able without difficulty to move to other programming approaches, such as object-oriented programming.

The C language is the most common procedural programming language, thus it can be the best starting point for quick and thorough understanding of basics in programming languages.

### 4.2. Design phase

The same teams collaborated during the design phase. Firstly, the course is built around six basic sections:

- Section 1: “Programming environments-language C”, describes the role of programming languages in the software development process and makes a brief reference to the development of programming languages and also to the most important categories in which are classified. There is also a brief reference to the alternative programming forms (programming paradigms) and, simultaneously, a throwback to the procedural programming language C, justifying at the same time the choice of this language as a part of this course. Finally introduces the basic concepts of the programming languages syntax and also the main restrictions for the process of programs writing.
- Section 2: “Variables, constants, data types”, presents the basic elements of the programming language C, and deals with the data types supported by the language, the types of variables, the way the expressions are calculated and the structure every program should follow. The basic commands of the programming language C, the assignment command of values to variables and the input-output commands, through which the program communicates with the user, will be also presented in this section.
- Section 3: “Operators, expressions, instructions”, will be presented the basic concept of the operator and the way how is used to form expressions and to calculate the value of these expressions. C has a rich set of operators, richer than other programming languages, for which, moreover, allows the mixing of its types almost with no limitations. Additionally the instruction as basic building unit for the procedural programming is introduced.
- Section 4: “Arrays, pointers” This section introduces the basic concepts of arrays and pointers in the C programming language and the way in which these two types are used to declare variables. The array is one of the most useful elements of C, supporting, among other things, the management of alphanumeric. Particular

emphasis is given to the handling of alphanumeric, as an array of characters. Finally the pointer provides us with an alternative way of referring to the information stored in the computer's main memory.

- Section 5: “Flow control instructions“, the programs execution flow configuration is the most important work of programmers in procedural programming. The familiarity with the available structures of the language facilitates the selection of the suitable structure and leads to writing a structured, readable and efficient code. A set of rules applies in the use of flow control instructions. Therefore, the aim of the section is to introduce the concept of flow control in a program, while giving the basic categories of instructions that allow the programmer to configure the execution flow depending on the requirements of each application.
- Section 6: “Abstraction in processes-Advanced Procedures issues“, refers to abstraction, one of the main techniques which face complexity. The abstraction is a simplified description or documentation, which highlights a number of features, while suspends others. The concept is fundamental for philosophy and mathematics, and is widely applied in the systems analysis phase. There are two forms of abstraction: a) abstraction in procedures (procedural abstraction) and b) data abstraction. This section, consisting of three operations, describes the procedural abstraction. The data abstraction will be the subject of another section that may be added in this course. The first operation of this section refers to the concept of abstraction and how it is used in programming to support the distinction between what a code part does and the way it does it. The second operation makes a very brief reference to the issue of modular design and the section is completed by the third operation which introduces the linguistic structure of the function.

Having the detail course description, the learning activities are designed on a weekly basis to the learners, based on the sections mentioned above. Each section begins with a presentation, which guides learners to the subject of the section. Each section contains at least two activities. In the following table we present an example of a learning activity description during the design process. The second course section entitled "Variables, constants, data types" consists of three educational activities related to basic data types in C, focusing on the way of handling of the variables of each type, the concept of constant and variable and the input and output functions of the language. Table 1 describes the third activity.

Table 1. Learning Activity Description.

Learning Activity 3	
<b>Title</b>	Input-output commands
<b>Section</b>	Variables, constants, data types
<b>Educational Strategy</b>	Cycle theory and practical application using matching, drag and drop exercises and project.
<b>Description</b>	The purpose of this section is to present the input and output functions used in C programming language. Firstly printf function and how it is used to print various types of data variables are described. Secondly, scanf function is described by the general formula and its arguments. The use of specifiers in scanf function is also being discussed. Finally, the basic instructions for the formation of a program in the programming language C with a description of the structure and comments are given.
<b>Language</b>	Greek
<b>Learning Outcomes</b>	Learners will be able to: <ul style="list-style-type: none"> <li>• recognize the use of Input / Output commands in C programming language</li> <li>• identify the basic determinants used in the Input / Output functions in C programming language</li> <li>• describe at least two examples of Input / Output functions with different determinants.</li> </ul>
<b>Learning Objects</b>	LA3LO1 - Input commands - printf function video LA3LO2 - Input commands - printf function exercise LA3LO3 - Input commands - printf function exercise LA3LO5 - Output commands - scanf function video LA3LO6 - Output commands - scanf function exercise LA3LO7 - Output commands - scanf function exercise

LA3LO8 - Output commands - scanf function exercise

LA3LO9 - The first program

**Assessment**

self-assessment exercises, project

**Keywords**

output, print, printf function, determinants, scanf function, input

The final step of this phase was the design of learning objects, which were defined in the learning activity description. Each activity consisting of one or more learning objects. Depending on the learning object nature, the learning time required and the difficulty degree of the exercise vary. On this basis we will present in detail the design of learning objects for the learning activity, which was described before. The learning objects LA3LO6, LA3LO7 and LA3LO8 focusing on the printf function and the way it is used to print various types of data variables. The language used is Greek; the theoretical aspects were already given to learners by educational video.

Thus, two self-assessment exercises and one project are used in designed to achieve the following learning outcomes:

- To enable students to describe the printf function of the C programming language with at least two examples
- To report key determinants used in the function
- To explain the use of determinants
- To construct an output example of one integer on the computer screen.

In the first self-assessment exercise a two-column table is given. The first column shows several of the determinants of the language and the second shows the data types. The learner is asked to match one or more determinants to the corresponding data table.

In the second self-assessment exercise a two-column table is given, with its first column filled with commands of the language and the second column empty. The student must complete the second column of the table with the results of the corresponding commands of the language by choosing from the answers given at the end of exercise (drag and drop). Every answer must be dropped to the correct position of the table.

Finally, the project gives two programs. The first uses this function several times and asks the learner to determine the output of the function every time it is used in the program. Specifically, the program of exercise 1 introduces an integer and a real number and displays them in 13 different ways using different determinants. The learner can confirm his/her answers, running this program with one of the C programming environments freely available on the internet. The project is completed by another program using the function, as shown in the second column of the table, for which the learner will follow the same procedure.

#### 4.3. Development Phase

To this end, a total of 62 Learning Objects, which had been designed during the previous phase, were developed. These Learning Objects consisted of educational videos, hypertexts, self-assessment exercises (eg. quizzes, multiple choice questions, fill in the blanks, drag and drops) and programming projects are also developed. Concerning the educational videos, an experienced multimedia developer supported the Course Manager team in the development of rich educational videos. Self-assessment exercises are also developed by the Course Manager team.

Following this, the Technical Support team in collaboration with the Educational Support team integrated the educational material into the platform. The platform used is the EAP-MOOC, a MOOC platform developed by HOU based on the edX Open platform.

On this basis in Fig.2 we present of the development of the learning objects LA3LO7, LA3LO7 which was described in the design phase.

Fig. 1. Learning objects in MOOC

#### 4.4. Implementation Phase

This MOOC is about to be launched at EAP-MOOC. Before delivering the course, we ran a pilot course in order to troubleshoot and evaluate the course and the platform. The participants in the testing phase were a small number

of students and experienced scientific staff. The disclosure of the course can be done via social networks, advertising, communities and email newsletters. During this phase, the Course Manager Team collaborates with the Educational Support Team during the educational process. The Technical Support Team provides technical support throughout the educational process.

## 5. Discussion

In this paper, we presented a collaborative team-based methodology focusing on the implementation of a computer programming MOOC. It appears that MOOCs, because of the scale, inherently require a team-based approach to course development. One focus group and interviews were held during the phases on the members of the teams.

During the analysis phase, the defined schedule was not exceeded, while the members of the Course Manager team indicated that the methodology instructions were understandable and facilitate the process in order to outcomes of the analysis phases were implemented without major adjustments. However, during the design phase timeframe was exceeded by far, due to the unexpectedly large amount of learning objects (62) and the development complexity induced by their varying types and formats. Yet the method of design and the standardization of design sections, educational activities and educational objects in tables as the example of the learning activity description (table 1) were characterized helpful and practical. A way to reduce the design time of learning objects could be the reuse of existing learning objects. Although, the MOOC guidelines were commented as extremely detailed, some suggested that we should add more material regarding the instructional design. This issue came up also during the focus group as there were concerns and questions about the design of learning objects.

After the MOOC has been delivered, data about learners will be gathered and analyzed in a top-down approach, driven by the research objectives. The findings are expected to provide useful insight for the establishment of MOOCs in our University. Moreover, with adjustments and adaptations, this model could help institutions create a process for organizing online course development using a team-based and cooperative approach.

## Acknowledgements

This research described in this paper has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) (Funding Program: "HOU").

## References

1. Blake, C., and Scanlon, E. Online Teaching: Applying Distance Learning. Handbook of research on emerging priorities and trends in distance education: Communication, pedagogy, and technology, 2014, 186.
2. Commonwealth of Learning (COL), Commonwealth of Learning, Learning for Development. 2014. Retrieved, October 2014 from <http://www.col.org/resources/Pages/EmergingIssues.aspx>
3. Cross, S. *Evaluation of the OLDS MOOC curriculum design, course: participant perspectives, expectations and experiences*. OLDS MOOC Project, Milton Keynes. 2014.
4. Deimann, M., and R. Farrow. Rethinking OER and their use: Open education as Bildung. *The International Review of Research in Open and Distributed Learning* 14.3 2013; 344-360.
5. European Commission. Report on Web Skills Survey: Support Services to Foster Web Talent in Europe by Encouraging the use of MOOCs Focused on web Talent—First Interim Report. 2014. Retrieved, February 2015 from: <http://openeducationeuropa.eu/sites/default/files/MOOCs-for-web-skillssurvey-report.pdf>
6. Fyle, C. O., Teacher Education MOOCs for Developing World Contexts: Issues and Design Considerations. In *Proceedings of the Sixth Conference of MIT's Learning International Networks Consortium (LINC)*, June 16th - 19th, Cambridge, Massachusetts, USA. 2013.
7. Geser, G. *Open educational practices and resources: OLCOS Roadmap 2012*. 2007. Salzburg: Salzburg Research.
8. Gütl, C., Rizzardini, R. H., Chang, V., & Morales, M. Attrition in MOOC: Lessons Learned from Drop-Out Students. In *Learning Technology for Education in Cloud. MOOC and Big Data* (pp. 37-48). 2014. Springer International Publishing.
9. Hixon, E. Team-based online course development: A case study of collaboration models. *Online Journal of Distance Learning Administration*. 2008; 11(4).
10. Ives, C., and M. M. Pringle. Moving to open educational resources at Athabasca University: A case study. *The International Review of*

- Research in Open and Distributed Learning* 14.2, 2013; 1-13.
11. Katai, Z., and L. Toth. Technologically and artistically enhanced multi-sensory computer-programming education. *Teaching and teacher education*, 2010; 26.2, 244-251.
  12. Kop, R., Fournier, H., and Mak, J. S.F. A pedagogy of abundance or a pedagogy to support human beings? Participant support on massive open online courses. *International Review of Research in Open and Distance Learning, Special Issue - Emergent Learning, Connections, Design for Learning*, 2011; 12(7), 74-93. Retrieved May, 2014, from <http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=19040607>
  13. Law, K.MY, Victor CS L., and Yuen-Tak Yu. Learning motivation in e-learning facilitated computer programming courses. *Computers & Education* 55.1, 2010; 218-228.
  14. Liyanagunawardena, T. R., Adams, A. A., and Williams, S. A. MOOCs: A Systematic Study of the Published Literature 2008-2012. *International Review of Research in Open & Distance Learning*, 2013; 14(3).
  15. Mackness, J., Waite, M., Roberts, G., & Lovegrove, E. Learning in a Small, Task--Oriented, Connectivist MOOC: Pedagogical Issues and Implications for Higher Education. *International Review of Research in Open & Distance Learning*, 2013; 14(4).
  16. McAuley, A., Stewart, B., Siemens, G., and Cormier, D. The MOOC model for digital practice, 2010.
  17. McCallum, C., Thomas S., and Libarkin J. C. The AlphaMOOC: Building a Massive Open Online Course one graduate student at a time. *Open Education Europa. eLearning papers*, 2013.
  18. Nguyen, A., Piech, C., Huang, J., and Guibas, L. Codewebs: scalable homework search for massive open online programming courses. *Proceedings of the 23rd international conference on World wide web*. International World Wide Web Conferences Steering Committee, 2014.
  19. Schmidt, D. C., and Z. McCormick. Producing and delivering a coursera MOOC on pattern-oriented software architecture for concurrent and networked software. *Proceedings of the 2013 companion publication for conference on Systems, programming, & applications: software for humanity*. ACM, 2013.
  20. Sonwalkar, N. The First Adaptive MOOC: A Case Study on Pedagogy Framework and Scalable Cloud Architecture-Part I. In *MOOCs Forum* (Vol. 1, No. P, pp. 22-29). 140 Huguenot Street, 3rd Floor New Rochelle, 2013. NY 10801 USA: Mary Ann Liebert, Inc.
  21. Spyropoulou, N., Pierrakeas, C., & Kameas, A. Creating MOOC Guidelines based on best practices. In *Proc. of the 6th Annual International Conference on Education and New Learning Technologies (Edulearn14)*, (6981-6990), Barcelona, Spain, 2014.
  22. UNESCO. *Forum on the impact of open courseware for higher education in developing countries* (Final Report), 2014 Retrieved February 2015 from <http://unesdoc.unesco.org/images/0012/001285/128515e.pdf>
  23. Vihavainen, A., M. Luukkainen, and J. Kurhila. Multi-faceted support for MOOC in programming. *Proceedings of the 13th annual conference on Information technology education*. ACM, 2012.
  24. Warren, Joe, Rixner, S., Greiner, J., and Wong, S. "Facilitating human interaction in an online programming course." *Proceedings of the 45th ACM technical symposium on Computer science education*. ACM, 2014.
  25. Wiley, D. *On the sustainability of open educational resource initiatives in higher education* (Paper commissioned by the OECD's Centre for Educational Research and Innovation (CERI) for the project on Open Educational Resources), 2007. Retrieved February 2015, from [www.oecd.org/edu/oer](http://www.oecd.org/edu/oer)
  26. Willis, B. Instructional development for distance education, 2002. Retrieved, October 2014 from <http://www.uidaho.edu/evo/dist3.html>.
  27. Yousef, A. M. F., Chatti, M. A., Schroeder, U., and Wosnitza, M. What Drives a Successful MOOC? An Empirical Examination of Criteria to Assure Design Quality of MOOCs. In *Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on* (pp. 44-48), 2014. IEEE.
  28. Yuan, Li, S. Powell. MOOCs and open education: Implications for higher education. *Cetis White Paper*, 2013.