

The 10th International Scientific Conference
eLearning and software for Education
Bucharest, April 24-25, 2014
10.12753/2066-026X-14-000

Educational Resources as Web Game Frameworks for Primary and Middle School Students

Ciprian-Bogdan Chirila

Department of Computer Science and Software Engineering,
University Politehnica Timișoara, V. Pârvan Blv. no. 2, Timișoara, Romania
chirila@cs.upt.ro

Abstract: *The massive open online course is a development in distance education. The existing online courses for primary and middle school students are based on several classic open educational resources like images, sounds, animations, but they could also be based on competence achieving web games. Writing web games for each discipline competence is a hard task, achievable only by programmers. Using frameworks we can offer teachers and other content editors the possibility of creating their own web games for the students. In this sense we designed and implemented two frameworks. One framework is based on a configurable dialog between two players. The dialog can be configured with mathematical expressions based on formulas, fields for questions with expected answers, loops for repetition in case of wrong answers, mark giving algorithms. The second framework is based on a configurable board and dice. The board has to be designed as an image with several ordered cells. On each cell we can configure events like: showing simple messages, asking questions, reading surprises. All events can be configured with actions like: moving backward or forward on the table stepping a certain number of cells, rolling the dice again. Both frameworks are using the HTML5 web technology and a simple communication server side logic. Dialog based games are suitable to disciplines which have mathematical formulas like arithmetic, physics, chemistry. The board games are suitable to disciplines where concepts must be learned and actions must be performed. Thus, the games implemented with these frameworks represent a more attractive way to perform distance education for primary and middle school students.*

Keywords: *MOOC; open educational resources; two player online games; game frameworks; dialog games; board games*

I. INTRODUCTION

In this paper we will present two educational resources as web game frameworks which can be played on a MOOC (Massive Open Online Courses) platform by primary and middle school students. The platform is named "The Little Prince" like the story with the same name written by Antoine Saint-Exupery French writer and pilot [Sai1943] and was developed in the context of a European project spread in three Romanian counties.

The main characteristics of the project involve: i) to develop competences; ii) to provide a natural and a free interaction between teachers and students; iii) to create a close and sensitive education system; iv) to provide help for the traditional education system; v) to ameliorate Romanian language and mathematics disciplines; vi) to visit the portal and not to oblige the student to do some specific action [Vla2013b].

The concept of competence is related to a quality participation of an individual in a real life situation or to the capability of an individual to use its acquired knowledge for solving practical problems or overcoming a certain situation [Vla2013a].

The classical “magister dixit” teaching systems do not always motivate the actual students since they force them to memorize knowledge under the pressure of oral or written exams. An e-learning system where a student can choose its own competences for development is more attractive and constructive. The traditional learning system is based on knowledge learning and does not focus on competence development. Students study better when they can choose their own learning path and really feel how knowledge helps them in practical or real-life situations.

In the center of the MOOC platforms advantages imaginary graph is the student. The student can choose freely the time to study since the MOOC platform is available 24 hours per day. The student can choose the place to study (home, school); the only necessary thing is the Internet connection. The student can choose freely the competence he wants to study. The student can have its own pace in the competence development. The student is evaluated online in more objective manner since the MOOC platform is not subjective like human evaluators.

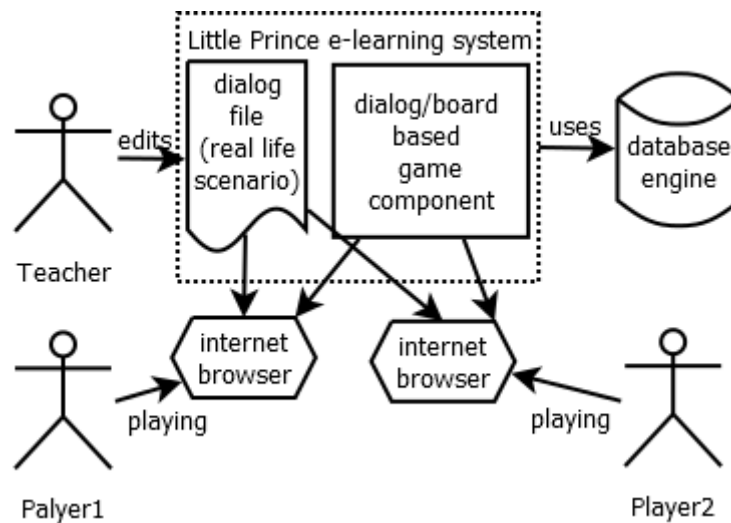


Figure 1. The approach

In Figure 1 we present our approach of how the educational resources can be used. The process starts with the teacher which edits the dialog/board game configuration files in XML format. The teacher must embed all his competencies under the form of questions and answers in the case of dialog games and in the form of quizzes in the case of board games. Then, the students can access the game components which interpret the configuration files and make the game play a competence training and assessment activity. During game play the assessment data is transmitted from the player browser to the learning record store. Each assessed piece of information is linked to one or more competence items hierarchically organized in the learning record store.

The paper is structured as follows. In chapter II we present the platform front-end, while in chapter III we will present the platform backend. In chapters IV and V we discuss about the two e-learning resources proposed in this paper. Chapter VI will present a case study on using those resources. Chapter VII will present related works, while chapter VIII will conclude and set the perspectives.

II. THE STUDENT FRONT END WEB INTERFACE

The platform has two ends: a front-end where the students achieve their competencies and a back-end dedicated to teachers and tutors for editing the MOOC materials. In this section we will present the front end elements the students will interact with. Like on every e-learning website there is an e-mail based subscription system in order to record learning evolution. The access is granted also without authentication but without progress recording and evaluation history. The main graphical

theme of the website is inspired from the story of Little Prince written by Antoine Saint-Exupery. The website is designed for students from grades 3 to 8. The interface represents a universe having 6 planets one for each grade.

The planets have different names and colors in order to be easily identified. The first planet is the Liberty planet which was designed for the 3-rd grade students. The credo of this planet is about the freedom of movement, thinking without any restrictions. The second planet is the planet of Mobility designed for the 4-th grade. The motto of this planet is about all the things around us which are in motion and ever changing. The third planet for the 5-th grade is the planet of Relations. The relations can be made only with our neighbors and these relations reflect who we really are. The fourth planet is the planet of Collaboration created for the 6-th grade. Collaboration is about building things together with other people. Without collaboration any group activity is impossible to fulfill. The fifth planet is the planet of Culture created for the 7-th grade. The culture is the result of living together of the way we behave, we relate with each other. Finally, the sixth planet is the planet of Competence for the 8-th grade. All planets can be seen in Figure 2. The competence gives us the power people have when they are united. Planets are fixed elements in the universe that we describe. Around planets we can find satellites spinning.

The satellites model the disciplines which are studied in each grade. In our implementation for each planet we will have the same number of satellites. There is one satellite for the communication in native language. A second satellite is created for the instruments of mathematics. A third satellite is created for the communication in other languages. The fourth we have is for science and technology. The fifth satellite was made for learning and adaptation. The sixth was created for community participation. The seventh was created for initiative and management. The eighth was created for cultural sensibility. The goals of our project included the creation of competence development materials for only the first two satellites. Unlike planets satellites can be aligned on their planets meaning that the student owns the respective competencies or they can have random positions in space meaning a work in progress for the current student.



Figure 2. Planets as grades

After choosing a satellite the list of islands is shown. Islands correspond to general competences and they have several shapes: a tree garden, a bunch of houses, an area filled of blocks, a playground etc. Inside the islands we have several zones corresponding to specific competences. Each zone has several points of competences where the competence action lists can be launched by the student.

III. THE LITTLE PRINCE APPLICATION MODEL

In this section we will present the web application model. We will present: i) the structure of the model; ii) the actions that teachers can implement for the students; iii) the functionality of the model; iv) the adaptation mechanism; v) the rewarding part; vi) the users.

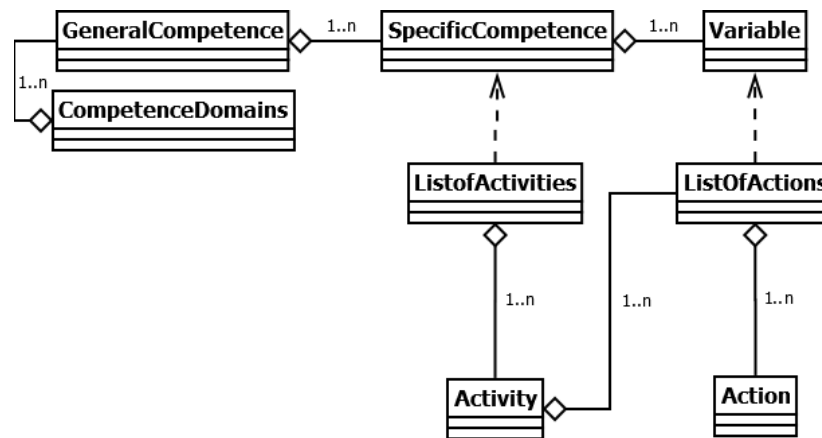


Figure 3. Application model

The structure of the platform e-learning content is hierarchical. On the top we have the competence domains which correspond to the disciplines described in the previous section and presented as planets in the platform front-end. Domains are split into general competencies for each domain. Next, general competencies are split into fine grained units called specific competencies. Usually these two levels of this taxonomy are set by laws issued by the ministry of education. Finally, specific competencies are divided into competence variables.

The content refers the previously described taxonomy but has its own internal structure. The action is the smallest e-learning project entity created by a tutor and presented by the system to the student in order to develop its competences. An action has the following attributes: type, difficulty level, abstraction level, user types, characteristic, name, description, weight, content. The type can be i) single selection; ii) multiple selection; iii) short answer; iv) ordered list; v) dialog; vi) associated list; vii) classification; viii) game; ix) simple drag and drop; x) image drag and drop; xi) multiple answer; xii) free answer. The difficulty can be set to three levels: beginner, intermediate and advanced. The abstraction level of an action can be: i) concrete; ii) representation; iii) abstract; iv) relational. The action characteristic can be: visual, kinesthetic, sequential, sensitive. The name and description attributes are set according to the action context. The weight attribute was designed for giving different weights to actions in a list when computing the weighted evaluation mark. The content has different formats depending on the action type all expressed in XML format in the persistence level. Still each type of action has its own editing interface to simplify the teacher task. Some actions which depend on complex resources like one or two player games or framework based game will be edited offline in a text file, uploaded in the backend interface and referenced from the action editing interface. The currently available frameworks are: i) the general framework where tutors can write any e-learning application in JavaScript; ii) the dialog game framework and iii) the board game framework.

Actions are grouped further in lists of actions which are designed for a competence variable. The attributes of an action list are: type, difficulty, abstraction, order, name, description and parent activity. The action list type is: learning, training and evaluation. The order can be: in-order or random. In order that an action list to include actions all common set of attributes must have equal values. Action lists with a predefined set of attribute values will accept only actions with the same attribute values.

The actions lists are grouped into an activity at a higher level. The activity has the same attributes as the action list and it is linked to a specific competence. Next, activities are grouped in activity lists which are designed also for specific competences.

The adaptation mechanism works inside a specific competence. It is based on jumping between different levels of abstraction and difficulty. The jumps were created in order to disallow a student to be obliged to stay too much on the current level. The levels of abstractions are in growing

complexity order: i) concrete; ii) representation; iii) abstract; iv) relational. The difficulty levels are: i) beginner; ii) intermediate and iii) advanced. The variables of a specific competence are ordered and they respect the pedagogical learning order. At variable levels we designed containers with actions namely lists of actions which have to be executed by the student. We remind the reader that a student can choose directly his action list he wants to fulfill from the whole hierarchy. Depending on the students learning capabilities we have to deal with three situations: i) the student is unable to perform a minimum amount of actions from the current list, so he or she shouldn't be on it, requiring a jump to a lower level; ii) the student is unable to finish a sufficient numbers of actions in his current list in order to be able to jump to the next list in the hierarchy, but he or she fulfilled the critical actions allowing him to stay on the current list; iii) the student finished a certain number of actions with good results and he is ready to jump to a higher level list of actions.

To jump to a lower level from the current list means jumping on a list with a lower level of difficulty on the current abstraction level. When this is not possible in the case of the beginner difficulty level the jump is made on a lower level of abstraction. If the levels of abstraction are all depleted it means that our e-learning system is not able to form the competence of that certain student. In such a situation there can be two explanations: i) either the content is not well structured and has no appropriate difficulty and abstraction levels; or ii) the student has severe knowledge gaps and he or she should choose the competences of a different grade. The parameters to be considered when deciding to make a jump are: i) a minimum and a maximum weight percentage of the fulfilled actions; ii) a minimum and a maximum number of performed actions; iii) a minimum and a maximum number of obtained points.

The rewarding system is based on offering medals and trophies to the students according with their accomplishments. For finishing a certain level of abstraction of a variable a medal of bronze is awarded to the student. When he or she finishes a full specific competence then a silver medal is awarded. When a general competence is finished entirely then a gold medal is given to the student. All these medals of bronze, silver and gold may have several colors depending on the situation they are given. Assisted evaluations on different levels are awarded with medals on colored ribbons. Partially assisted evaluations are awarded with medals with white ribbon. Unassisted evaluations are awarded with medals with no ribbon at all. Unevaluated but started levels are awarded with grey medals without ribbon at all. For competences and grades trophies are awarded. The trophies awarded for domain competence accomplishment have a balloon like shape and the trophies awarded for finishing the content of a grade have a Little Prince in amber of different colors. For competence domains finished by assisted evaluations the awarded trophies have four legs. The ones partially assisted are awarded with three legged trophies while the ones unassisted are awarded with two legs trophies. Unevaluated competence domains are awarded with grey two legged trophies. A similar strategy was developed for grades accomplishments. The idea behind these virtual prizes is that they should be visible on the history page for each student and they are available for online sharing on Facebook social network. Thus, student accomplishments can be made public at their will for the friends and acquaintances.

The users of the web application can have of several roles. The majority of the users are the students for whom this web application is dedicated. For them we created the student role which is the implicit role for any newly created user. Another role is the teacher role which has rights to manage the student users. Their rights are: creation, update, delete and also they can manage student class membership. Thus, teachers can create their working groups selecting the students they want to train and evaluate. Teacher users can be administered by users who have school administrator roles. This role should be given to persons subordinated to the school principal or to the school principal himself if he or she is an IT person. School administrators are managed by users who possess the county administrator role. This role was designed to persons from the management of the county schools inspectorates. Finally, the county administrators are managed by users (usually one or two) having the role of national administrator. All the role validations between institutions are meant to be done based on the applicant scanned work institution certificate. Further verifications can be done in more direct manner: face to face, by phone or by official e-mail exchanges.

IV. THE DIALOG GAME E-LEARNING RESOURCE

The dialog game is a complex e-learning resource modeling a dialog between two online players. The dialog game consists in a dialog data model and a dialog game framework interpreting the data model [Chi2013]. The dialog data model is designed as a state machine driven by players or by the framework itself depending on the correctness of the answers given in the context. The dialog data model is also equipped with variables and related semantic actions allowing the author to formulate questions and to compute and verify answers.

The dialog data model is expressed in XML and has the following structure: table of roles, table of symbols, table of conditional verifications, table of sentences and variant answers. The table of roles is used in order to define roles for the conversation. Role identifiers will tag the dialog sentences. For example role pairs can be: seller and buyer, boss and employee, manager and driver etc.

```
<tabRoles>
  <role szGradeSymbol="nPointsBuyer">idBuyer</role>
  <role szGradeSymbol="nPointsSeller">idSeller</role>
</tabRoles>
```

Figure 4. Role table example

In figure 4 we present only two roles one for the buyer and one for the seller with their attached identifiers. We mention that these identifiers will be used in the dialog sentences. We also need to mention that the assessment grade symbol is referred in the definition of the role in order to be used later by the grading algorithm.

The table of symbols (variables) is used to define JavaScript like variables in order to store variable values during the dialog. Symbols have types (integer, float, string, array) and can be initialized with random values in order to create always new dialog contexts. Symbol values can be computed in several states of the dialog discussion reflecting its history. For example, we can have symbols for product prices, shopping money, change given by a seller.

```
<tabSymbols>
  <symbol szName="nPretMere" szType="int" nMinValue="1" nMaxValue="8"> 3 </symbol>
  <symbol szName="nPretPere" szType="int" nMinValue="2" nMaxValue="10"> 4 </symbol>
  <symbol szName="nPretBanane" szType="int" nMinValue="3" nMaxValue="12"> 5 </symbol>

  <symbol szName="nCantitateMere" szType="int" nMinValue="1" nMaxValue="4"> 0 </symbol>
  <symbol szName="nCantitatePere" szType="int" nMinValue="1" nMaxValue="4"> 0 </symbol>
  <symbol szName="nCantitateBanane" szType="int" nMinValue="1" nMaxValue="4"> 0</symbol>

  <symbol szName="nPret" szType="int">10</symbol>
  <symbol szName="nSuma" szType="int">100</symbol>
  <symbol szName="nRest" szType="int">90</symbol>

  <symbol szName="nPunctajCumparator" szType="int">0</symbol>
  <symbol szName="nPunctajVanzator" szType="int">0</symbol>
</tabSymbols>
```

Figure 5. Symbol table example

In figure 5 we present an example of a symbol table for a three product shopping dialog. In this case the buyer wants to get apples, pears and bananas so we had to consider variables for their prices and also for their quantities. A different set of variables are necessary to be defined for the total price, the paid sum and the change. Also two symbols must be defined for the grades and attached to the roles. A symbol definition has a name, a type and possible value range limits. The type and the range limits are used to check the user inputs. A symbol definition has also a default initial value.

The table of conditional verifications is used to define boolean expressions based on symbols which will be referenced and evaluated in order to evaluate the players. Each conditional verification is mapped to a competence variable so at the end of the dialog we will be able to know exactly which competences are weak and which are strong for the players, thus enabling a personalized rehearsal plan.

```
<tabVerificationConditions>
```

```

<verificationCondition name="vc1">
  v("nPretMere") * v("nCantitateMere") +
  v("nPretPere") * v("nCantitatePere") +
  v("nPretBanane") * v("nCantitateBanane") == v("nPret")
</verificationCondition>

<verificationCondition name="vc1n">
  v("nPretMere") * v("nCantitateMere") +
  v("nPretPere") * v("nCantitatePere") +
  v("nPretBanane") * v("nCantitateBanane") != v("nPret")
</verificationCondition>

<verificationCondition name="vc2">
  v("nSuma") == v("nPret") + v("nRest")
</verificationCondition>

<verificationCondition name="vc2n">
  v("nSuma") != v("nPret") + v("nRest")
</verificationCondition>
</tabVerificationConditions>

```

Figure 6. Conditional verification table example

In Figure 6 we present an example of conditional verification table example for the seller-buyer scenario in the context of a groswery store. In our design we need to define direct and negated boolean expressions in order to be able to perform competency assessment. Conditional verifications have identifiers attached are they are stored in a table in order to be referred from several places and thus to be locally reused. Verifications conditions are evaluated during the game play on the client side using the JavaScript “eval” function. Alternative approaches deal with manual parsing and expression evaluation. The access to the symbols is performed through the use of value() or v() JavaScript user defined function which performs a lookup in the symbol table, takes into account the symbol type and provides the appropriate value.

The table of sentences contains all the sentences that can be used during the game. The content of those sentences is filled with pre-computed values from variables and with fields in order to get player input into variables.

The dialog game framework is implemented in JavaScript on the client side and in Microsoft C# on the server side. The communication between the two components is governed by a simple communication protocol having two main functions sendMessage(...) and receiveMessage(...) with JSON encoded arguments. The protocol is used for the initiation of the dialog, for sending and receiving the answers and for the finalization part where the marks are stored on the server.



Figure 7. Dialog game screenshot

In figure 7 we can see both players screens in the middle of a game. The player from the left as a buyer ordered 1 kg of apples and 1 kg of bananas, apples cost 1 lei/kg and bananas cost 3 lei/kg,

and the player on the right as seller must compute the price for the products and communicate it to its peer. This is an example of a moment where conditional verifications make silent evaluation about the correctness of the answer.

V. BUILDING A DIALOG GAME SCENARIO

In this section we will present the steps to be made in order to build a dialog game scenario starting from the competency demands. The proposed general competence is called operating correctly with fractions and rational numbers. The specific competence deals with operating correctly with fractions and the variable is getting the common denominator to fractions.

The proposed scenario deals with a virtual competition between two farmers. The dialog is designed having three assessed questions with an introductory and an ending part. The dialog is focused on adding fractions representing infield surfaces having different purposes like: pasture, orchard, swamp.

As a diversity aspect with no impact on the assessment each infield is attributed with some random characteristics selected from a list: i) the pasture can have fat and juicy grass or clover or lucerne (alfalfa) which are stored in a symbol named "tabTypes1"; ii) the orchard can have plum trees or apple trees or pear trees or cherry trees or apricot trees or peach trees stored in a symbol named "tabTypes2"; iii) the swamp can have pigs or oxen or ducks or geese stored in a symbol named "tabTypes3".

In order to be able to assess such a dialog a few symbols must be prepared for this matter. Firstly, two index symbols are defined for the random selection of pasture types, one for each player "nIndex11" and "nIndex12". As the options list has a length of 3 the indexes are between 0 and 2. Using the values of the indexes two string typed variables are initialized with random values from "tabTypes1".

Next, we create two pairs of numerator and denominator symbols: i) "nNumerator11" initialized with random integer values between 1 and 3; ii) "nDenominator11" initialized with random integer values between 4 and 6; iii) "nNumerator12" initialized with random integer values between 4 and 6; iv) "nDenominator12" initialized with random integer values between 7 and 12.

Further on, we need to compute the correct results the players will have to answer. For this reason we define an anonymous JavaScript function with two integer parameters which will be applied to the two denominators "nDenominator11" and "nDenominator12". In this function we define a named function for computing recursively the highest common factor using the algorithm of Euclid. Then, we compute the lowest common multiple of the two denominators using the product divided by the highest common factor. Finally, we compute the two amplified numerators which will be checked to as a correct result.

Another two sets of variables are defined and initialized in the very same manner for the second and the third assessed questions. As verification conditions we need to check if the entered numerator is equal with the computed one and entered denominator is equal with the computed denominator. Also the negated verification condition is defined also in order to test negated dialog answers.

Regarding the sentences we use a pattern formed out of a small introduction, followed by the three assessed questions and ending with saying goodbye. In the introduction sentences the players greet each other. Usually there are available several greeting forms and the player can choose the one he wants to use. The available greetings have different degrees of kindness or politeness.

This is yet another subtle form of education. The first assessed question structure contains a statement where variable values are printed to the screens of the two players. In this particular scenario the first player declares the fraction of pasture owned by him and the fraction of pasture owned by his opponent. The next sentence is issued for the second player where he must fill in the amplified fractions such that they have the same denominator in order to continue the discussion. This sentence is assessed with the verification condition designed earlier.

Next, the first player must confirm or deny the values entered by the first player, an action which is also assessed by conditional verifications. This is why we need two versions positive and negated of the conditional verification.

VI. THE BOARD GAME E-LEARNING RESOURCE

The board game is a complex e-learning resource modeling a board game with two online players which embeds learning and testing actions. The board game data model consists in an image

file name, a table of spaces, a table of surprises and a table of questions all expressed in JSON format. The image filename must refer a picture containing the game board with all cells the online players will have to step.

The table of spaces contains a set of attributes for all cells like: index, description, an action tag and a pair of integers representing their Cartesian coordinates. There are several types of actions tagged in the board game such as: “startGame”, “rollAgain”, “readQuestion”, “readSurprise”, “stepForward”, “stepBackward”, “restartGame”, “skipATurn”, “jumpTo”, “null”. The “startGame” action is used for states initialization. The “rollAgain” action is used to allow the player to roll the dice two times thus to advance faster. In the case of “readQuestion” and “readSurprise” actions the player will be given randomly a question or a surprise to answer or to execute. Question answering is the way players will get their marks. Surprises contain actions which makes the play more dynamic and interesting. The “restartGame” action is used in case of surprises when the player has to start the game from the beginning. The “skipATurn” action is used for the opponent player to roll the dices twice. The “jumpTo” action is used for making the players change their current position to a predefined space. The “null” action means taking no action at all. In the case of stepping or jumping actions a new attribute must be present naming “noOfSpaces” in order to specify the extent of the backward or forward leap.

The table of questions contains a set of question-answer pair. When a player hits a space with a “readQuestion” type of action he has to choose the correct answer from a list of mixed answers from other questions. This choice of single answer was taken in the current implementation but other type of testing could be also used like in the previously described dialog games.

The table of surprises contains context defined actions which must be executed when a space with “readSurprise” tag was hit.

The board game data model is implemented in JSON and the interpreting framework in JavaScript. On the server side we use the same implementation as in the dialog framework and its associated chat protocol.



Figure 8. Board game screen shot

In figure 8 we present the two player screens of the board game. Each player will see only one screen in its web browser. In the upper part there is an automatic scrolling board and in the lower part there is the control panel where messages and actions are taken by the players.

VII. CASE STUDIES

In this section we present some of the ideas behind the dialog games that we designed for developing competences for the general competence 118 which is about executing operations with rational numbers. The general competence 118 has 6 specific competences. Next, we will present each specific competence with the dialog games contexts that were imagined in this sense.

Specific competence 223 is used to evaluate correctly rational numbers. In this context we chose a dialog game about a triathlon athletic competition where the players have to identify the integer part and the rational part of the trials time expressed as a random rational number. Another dialog game was created for the interpretation of the integer and rational part of a number in a context of a pizza diner where clients order whole pizzas representing integers and pizza slices representing rational fractions. In order to compare rational numbers we imagined a dialog in an electronic supermarket where the two players act as buyers and compare random prices having rational fractions for laptops, computer screens, tablets.

Specific competence 224 is used for competences to execute correctly additions with rational numbers. For this specific competence we chose a scenario of a funny mechanic which changes the oil and other fluids for a customer combining different rational number expressed quantities of liquids present in several recipients.

The specific competence 225 is used for competences to compute correctly the subtraction between rational numbers. For specific competence 225 we imagined a situation where a cook needs to buy some food ingredients from a salesman who already reserved some of its stocks expressed with rational numbers thus enabling the players to do subtractions.

The specific competence 226 is created for competences to use multiplication of rational numbers in order to solve exercises and problems. Specific competence 227 is used to divide correctly rational numbers. Specific competence 345 is designed to compute correctly the arithmetic mean. In total we have developed 40 specific competences and 300 variables for two disciplines.

For the board game framework we built an example about avoiding natural disasters. The spaces and the surprises contain messages and questions including: to reduce disasters, to avoid building houses near volcanoes, to avoid cutting trees, to help people when they are flooded, to be prepared for earthquakes and their replicas, to plant trees avoiding landslides etc.

VIII. RELATED WORKS

Most of the related works tend to focus on learning objects (LO) reuse like in the works of [Arm2006],[Ben2008],[Mat2011],[Soo2011]. In this sense our e-learning resources can be highly reused because of the big number of learning scenarios that can be imagined and the random generated variables for each scenario.

Our educational resources tend to be more complex than a common LO, but we consider that they can be covered by the LO term semantics.

A different set of related works is dedicated to LO composition and sequencing like in [Ben2008],[Kno2003],[Por2004] based on: metadata, query languages, sets, algebras, decision models and databases.

Generative learning objects (GLO) [todo]...

The Experience API or Tin Can API [todo] is used for storing “I did this” student actions in order to be verified later by different parties for assessment purposes. In our approach the “I did this” three concept model is implemented somehow different. The “this” concept is related to the competence variables from the competence hierarchy tree organized on four levels. The “did” concept has in correspondence a lot of actions on different levels of abstraction, difficulty and characteristic. The “I” concept refers to the student who is the main objective of the whole our work.

In [Kno2003] composition methods are presented for creating courses out of library learning objects. The approach is based on special computations which determine the direct and indirect prerequisites for certain learning objects taking into account their metadata. In our approach we offer a structure on four levels of lists to sequence LOs.

LOs adaptability is another major point discussed in works like: [Mat2011] where LOs are selected by a learning style assessment engine or [Got2003] where the LOs platform tracks users and adapts its content to the user needs. In our approach we have such an adaptive mechanism which is based on both the learning history of the student and on the competence dependency graph.

The work of [Gor2013] is very similar to ours since they created an authoring tool for building LO that are called Virtual Excursion based on digital resources thus creating virtual labs or virtual visits which are close to small virtual universes. Our approach is a bit different since we enable competence development through online collaborative games which can be considered small learning universes also.

In [Gru2005] a pricing model is developed for LOs based on XML metadata extension. In [Mor2005] a knowledge management is presented in order to achieve LOs quality. In our work LOs are focused on developing competences.

IX. CONCLUSIONS AND PERSPECTIVES

In this paper we presented two e-learning software components for gaming in the context of a MOOC platform created for primary and middle school students. The e-learning components are more complex than classic learning actions found on other e-learning platforms. One conclusion we can draw is that in order to make e-learning attractive you have to create more complex actions types which can emulate a small universe with images creating virtual spaces and sentences creating roles to be played by the young students.

The two game frameworks are quite different but they still can be compared. Regarding their extent we expect dialog games to be shorter and board games to be longer comparing the number of states with the number of spaces. Long dialog games probably can get tedious. Short board games may not be very interesting. In the current implementation dialog games seem to be more useful for engineering disciplines like arithmetic, algebra, geometry, physics, logic etc. On the other hand we proved that is possible to create MOOC for primary and middle school students with actual results.

For the student there are several advantages our e-learning platform promotes: developing competences and choosing freely the competence development path enabling a better future job based on native skills. For the teacher we offer structures based on the national competence plan for adding new learning materials.

A very near future work would be the integration of all types of actions into the board game questions and surprises. An interesting idea of a future work would be to make the games composable in the sense that when playing with the board game and hitting a surprise to take the two players to a short dialog on the same topic. The reverse combination is probably not very interesting because of their extent.

Another interesting perspective would be to use also other real life games to develop competences, like card games for example. Such a task is quite difficult because it needs finding the right principles in order to make it interesting for the students and not to dilute its educational content.

One future work involves researching the possibility of porting the gaming frameworks to other e-learning platforms. The estimated difficulties are found around the server side which could be implemented as a platform plugin. Regarding the implementation aspect we have to think also about making the MOOC and its components more available on mobile devices.

Acknowledgements

We would like to thank to all members of the Little Prince project www.miculprint.eu developed under the POSDRU contract number 85/1.1/S/64320 and namely to: Ioan Vlaşin the general manager of the project, Sorin Andraş the software team manager, Iulian Necea the head of the MOOC system development team, Samuel Bogdan the graphic interface designer.

References

- [Arm 2006] Nick Armitage - Design of e-Learning Objects for Call in Syntactics. World Automation Congress (WAC 2006), Budapest, July 2006.
- [Ben2008] Edgard Benitez-Guerrero, Carmen Mezura-Godoy - A Declarative Approach for Managing and Querying e-Learning Objects. In proceedings of 2008 Mexican International Conference on Computer Science, Mexico, 2008.

[Chi2013] Ciprian-Bogdan Chirila. A Dialog Based Game Component for a Competencies Based E-Learning Framework, 2013 8th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), pp. 055--060, Timisoara, Romania, May, 2013.

[Gor2013] Aldo Gordillo, Enrique Barra, Daniel Gallego and Juan Quemada - An online e-Learning authoring tool to create interactive multi-device learning objects using e-Infrastructure resources. In proceedings of 2013 IEEE Frontiers in Education Conference, Oklahoma City, OK, October 2013.

[Got2003] Gottfried Vossen - Learning Objects as a Uniform Foundation for E-Learning Platforms, 2003.

[Gru2005] Markus Gruene - Pricing of Learning Objects in a Workflow-Based E-Learning Scenario. In proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 1 - Volume 01, Hawaii, 2005.

[Kno2003] Gerhard F. Knolmayer - Decision Support Models for Composing and Navigating through e-Learning Objects. In HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track1, 2003.

[Mat2011] Nasim Matar - Adaptive Learning Objects Repository Structure Towards Unified E-Learning. In proceedings of International Conference on Information Society (i-Society), London, June, 2011.

[Mor2005] Erla Morales, Dr. Francisco Garcia, Hugo Rego, Tiago Moreira, Hector Barbosa - Knowledge Management for E-learning based on Learning Objects. ITHET 6th Annual International Conference F4B-12, Juan Dolio, Dominican Republic, July 7 – 9, 2005.

[Por2004] Fabio Porto, Ana Maria, C. Moura, Fabio José, Coutinho Silva - ROSA a Repository of Objects with Semantic Access for e-Learning. In Proceedings of 8-th International Database Engineering & Applications Symposium, 2004.

[Sai1943] Antoine de Saint-Exupery – The Little Prince, 1943.

[Soo2011] Lian Kei Soo, Eng Thiam Yeoh & Sin Ban Ho - Searching & Ranking Learning Objects in a Service Oriented Architecture for E-learning. Proceedings of the 5th International Conference on IT & Multimedia at Uniten (ICIMU 2011), Malaysia, 2011.

[Tes2004] Lee S. Tesdell - Converting Technical Communication Courses to Online Delivery. In proceedings of International Professional Communication Conference (IPCC 2004), 2004.

[Vla2013a] Ioan Vlaşin. Competența - O participare de calitate, Editura Unirea, Alba Iulia, Romania, 2013.

[Vla2013b] Ioan Vlaşin. Ameliorarea dezvoltării competențelor și a interacțiunilor în educație cu ajutorul platformei online. Studiu de caz portalul www.miculprint.eu, A XI-a Conferință Națională de Învățământ Virtual Virtual Learning – Virtual Reality Tehnologii Moderne în Educație și Cercetare Models & Methodologies, Technologies, Software Solutions, ISSN 1842-4708, October, 2013.