


System-Based Ontology for Assessing Learner's Programming Practical Works Activities (S_Onto_ALPPWA)

Karima Boussaha, Department of Computer Science, Larbi Ben M'hidi University Oum El Bouaghi, Algeria

 <https://orcid.org/0000-0003-4622-7625>

Farid Mokhati, Department of Computer Science, Larbi Ben M'hidi University Oum El Bouaghi, Algeria

Amira Hanneche, Department of Computer Science, Larbi Ben M'hidi University Oum El Bouaghi, Algeria

ABSTRACT

This article introduces a new learner's self-assessment environment as CEHL that allows comparison of learners' programs with those elaborated by the teacher. The subjacent idea is to indirectly compare programs through their graphical representations described by ontologies. So, CEHL developed so-called S_Onto_ALPPWA which allows comparing learners' productions with those elaborated by the teacher. The tool allows essentially (1) generating two ontologies from the learner's program and the teacher's one, (2) applies some matching algorithms for measuring degrees of similarity and dissimilarity between learner's program and teacher's one, and (3) assessing the learners by giving them a list of semantic and syntactic errors detected in their programs. The present work is an extension of the authors' previous work, which did not take into account semantics errors. In the present work, they have managed to detect syntactic and semantic errors by using ontologies. To demonstrate the effectiveness of the system, two prospective experiments were conducted. The obtained results were very encouraging.

KEYWORDS

CEHL, CSCL, Didactic of the Programming, Learners' Assessment, Matching Algorithm, Multi-Agent System, Object-Oriented Programming, Ontologies, Ontology Matching Algorithm, Practical Works

INTRODUCTION

The teaching of practical works is fundamental in scientific and technical learning disciplines, in in-class as well as in distance, and meets the learners' needs. Unfortunately, learners are often deprived of this essential instructional teaching opportunity. This is, in fact, due to several problems. The

DOI: 10.4018/IJWLTT.20210901.oa5

This article, published as an Open Access article on July 9th, 2021 in the gold Open Access journal, the International Journal of Web-Based Learning and Teaching Technologies (converted to gold Open Access January 1st, 2021), is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

researchers cite, among others, the unavailability of the assistants, the obstruction of learners, and the material is expensive and cannot be duplicated. To minimize these problems thus teaching must answer to these needs (Guillaume, 2006).

Due to the previously mentioned problems, the practical work in general and especially the practical work of programming languages in computer science in the introductory courses in the first university cycle in the university years is usually accomplished by a group of learners as a result of the lack of adequate devices. And therefore when the assessment is given, it is one mark for all the group members and this makes the assessment subjective and does not reflect the true level of each learner belonging to the group, because there are elements of the group that do not work and rely on the active elements. In addition to this problem, the authors cite the problem of failure to learn these types of activities as studies have proven. Knowing that the studies confirm that the rate of failure or abandonment of the programming in the introductory courses in the first university cycle varies from 25 to 80% of the share in the world (Aiouni et al., 2018). This problem of programming failures does not only concern our institution. Several studies on algorithmic / programming learning conducted by different institutions in other countries have converged towards the same conclusion (Boussaha et al., 2015a). Learning algorithmic / programming was always a source of difficulty not only for students but also for teachers too. This is due to the intrinsic characteristics of the discipline and the classic methods of teaching (classic practical works in the classes rooms).

To overcome these problems, several problem-based learning systems are developed (Tadjer et al., 2018) these systems did solve the hardware problem, but the assessment problem was still not resolved.

We think that the self-assessment, in its formative function, is in the middle of the training considering its regulating function, which is paramount. The construct of self-assessment refers to the degree to which students can regulate aspects of their thinking, motivation, and behavior during learning (Tadger et al., 2018)(Tadger, et al. 2020). When the authors consider the topic of the self-assessment, some key generic factors should inform, the self-assessment has several different purposes for all those involved in the process these include (Hadadi & Bouaarab-Dahmani, 2019):

- Measurement of learning and achievement (Smith et al., 2013)
- Institutional promotion and marketing
- Diagnosis of learning (Sadler, 1989)
- Feedback and feedforward for learners (Nicoland and Macfarlane, 2006)

Feedback for teachers (Nicoland and Macfarlane, 2006)

- Certification of learning
- Development of learning outcomes for a course and program(Biggs and Tang, 2011)
- Development of knowledge, skills, and dispositions for the long term, including judgment (Boud and Falchikov, 2007)

The present work concerns more particularly, the learners' self-assessment in the CEHL (Computing Environment for Human Learning) environments of remote practical works in programming. Our goal is to suggest a self-assessment CEHL environment for thinking about measures of cognitive knowledge, a self-assessment CEHL that will help generate feedbacks, guide future research, and develop learners' efforts.

Self-assessment in the CEHL environment remains a poor relation because it goes very often away in the formalisms suggested. However, self-assessment, which is a big part of teaching activity, deserves to also have it off the methods, techniques, and tools which make a general manner to evaluate the context of CEHL (Computing Environment for Human Learning).

Currently, some researchers have examined issues related to the learner' assessment the authors can cite among them (Tadger,2018,2020)(Hadadi & Bouaarab-dahmani,2019)),(Pang et al.,2019) (Indira et al.,2019)(Seman et al.,2018)and others. But they have omitted a more precise learners' self-assessment in the practical works activities. Nevertheless, environments dedicated to learning practical works have been developed in the few last years.

To cope with this problem, in this paper, the authors aim to prototype a new learner's self-assessment environment as a CEHL system. The self-assessment environment developed so-called S_Onto_ALPPWA(System Based Ontology for Assessing Learner's Programming Practical Works Activities) allowing an individual assessment for each learner by comparing learners' productions with those elaborated by the teacher. The authors' contribution seeks to explore the impact of web technology on the design, implementation, and evaluation of the distance practical works learning process as well as the development of new activities, relationships, skills, and competencies for different levels of learners, implicit in such processes. The developed system consider like a result of the investment in answering the research questions presented in the second paragraph (see research questions paragraph).

In speaking to this issue of construct validity, the authors organize this research paper into nine main sections. In the second section, the researchers give some research questions. The authors review the various related works about self-assessment that will guide the discussion in subsequent, for each of these works, the researchers try to illustrate its strengths and limitations in section three. In the fourth section, the authors describe the methodology of the self-assessment proposed approach. Five, the researchers discuss in some detail several components of the general architecture of the self-assessment proposed approach, also the proposed hybrid matching-agent algorithm, and the modeling of the practical work with the ontology. In section six the researchers illustrate the developed environment using a concrete case study with some screenshots. Section seven details the benefits and the evaluation of the developed system with an experimental study. In the eighth section, the authors discuss the results and the findings. The researchers draw some conclusions about the self-assessment proposed approach, suggest some directions for future research, and raise some issues that merit consideration in the development and evaluation of the self-assessment CEHL.

RESEARCH QUESTIONS

The previously mentioned problems that learners face in studying practical works with the classical ways, let us reformulate our proposed contribution by investing in answering the following research questions:

- Does the self-assessment CHEL for learning practical works activities based on a hybrid matching-agent algorithm and ontologies reduce the rate of failure or abandonment of the programming in the introductory courses in the first university cycle?
- Can the self-assessment CHEL for learning practical works activities based on a hybrid matching-agent algorithm and ontologies show better achievement for students who learned with self-assessment CHEL versions than those who learned with the conventional system with no self-assessment versions?
- Can the self-assessment CHEL for learning practical works activities based on a hybrid matching-agent algorithm and ontologies ameliorate the learners' results (Author et al.,2015a)?

RELATED WORKS

The construct of self-assessment refers to the degree to which students can regulate aspects of their thinking, motivation, and behavior during learning(Pintrich, and Zusho,2002;Tadger et al.,2018;

Tadger, et al.2020). According to Sadler (1989), he refers to “assessment that is specifically intended to generate feedback on performance to improve and accelerate learning”. Such feedback helps students to restructure their understanding/skills and build more powerful ideas and capabilities (Nicoland and Macfarlane,2006; Zimmerman,2008; Pintrich, and Zusho,2002). This kind of self-assessment has been a constant demand in higher education for graduating students to be equipped with the capacity for lifelong learning that helps learners to continually upgrade their skills and knowledge through their self-motivation and learning(Wigfield et al,2011; Zimmerman,2004; Tadger et al.,2020; Hadadi&Boaarab-dahmani et al.,2018). Its principal goal is to help students take greater responsibility when it comes to managing their learning; this is done by helping them become more strategic learners through self-assessment learning(De Corte et al.,2011). Being equipped with skills such as self-assessment is important because a major function of education is the development of lifelong learning skills(Zimmerman,2004; Tadger et al.,2020; Deep et al.,2019).

So several works focused on the self-assessment in the broad sense, the authors can cite among others Black and Wiliam(1998), Sadler(2005), Zimmerman(2008,2004), Pintrich(2002), Boud and Falchikov (2007), Tadger et al.,(2018,2020), Pang et al.,(2019),Indira et al.(2019), Seman et al.(2018) and others.

In what follows, the authors give the literature about the self-assessment in a specific activity, it is the learners' self-assessment in practical works.

The teaching of practical works is fundamental in scientific and technical learning disciplines, in in-class as well as in distance, and meets the learners' needs. Unfortunately, learners are often deprived of this essential instructional teaching opportunity. This is, in fact, due to several problems. The authors cite, among others, the unavailability of the assistants, the obstruction of learners, and the material is expensive and cannot be duplicated. To minimize these problems thus teaching must answer to these needs(Guillaume,2006).

Although learner's self-assessment in practical works represents an important activity during learners' training, it has not been addressed enough. In this context, the authors find some works which focus on practical works that are presented in Bennouna et al. (2008), and Choquet et al. (2011). Other works that deal with learners' assessment in the modeling process(Alonso et al.,2008) and (Mitrovic &Suraweera,2004) have also been proposed.

The authors in Bennouna et al. (2008), proposed a formative assessment method of learners' skills using algorithms of supervised classification. Of course, this work has considerably forwarded the domain by proposing novel strategies for learners' assessment in the practical works. However, it does not deal with learning programming languages.

The authors in Choquet et al. (2011), describe a way to identify and model indicators that can be calculated and provided in real-time to teachers when they are involved in synchronous tutoring of practical work. The author based on his work on the collection and the storage of tracks of the learning session participants (learners and teachers).

Alonso(2008), developed a CEHL called Diagram dedicated to the training of OOM (Object-Oriented Modelling) concepts starting from a textual specification. It proposes an automatic method for analyzing learners' diagrams by comparing and matching class diagram components. The proposed method is implemented as an integrated component of the diagram called ACDC (Automatic Class Diagram Comparator). Furthermore, the author only uses matching methods, which are based on simple matchers (i.e., matchers that use only one kind of data). This work treated the problem of learners' assessment in online courses and classrooms, but it does not treat the learners' assessment in the practical works.

The authors in Mitrovic and Suraweera(2004), presented a problem-solving environment for the university level students in which they can practice conceptual database design using the entity-relationship data model. This work presents an intelligent tutor called KERMIT (Knowledge-based Entity-Relationship Modeling IntelligentTutor). It uses CBM(Constraint-Based-Modeling) to model the domain knowledge and generate student models. This work treated the problem of learners'

assessment in online courses and classrooms, but it does not treat the learners' assessment in the practical works.

The authors in Hadadi et al.(2019), proposed learner modeling and automated assessment in MOOCs,based on the ODALA approach.

The proposed learner model includes different dimensions:general information, Cognitive state,learning style,preference , and behavior dimensions.The authors in this paper deal with the cognitive state dimension that is updated based on the results of the proposed activities.Inparticular,they proposed an assessment planner where the acquisition of knowledge and skills takes different levels. This planner follows the learner's progress with the transition from a level to another in the pyramid of pedagogical activities. Of course, this work has considerably forwarded the domain by proposing novel strategies for learners' assessment. However, it does not deal with the practical works activities.

The authors in Tadjer et al. (2020) proposed a method that aims to improve the soft skills as well as the students' cognitive skills in a problem-based learning environment by taking into account the traces of the students who contribute to the development of a software project. So, the authors in this research paper proposed a method that is based on traces in order to help students to improve their skills. This work treated the problem of learners' soft skills assessment in online courses and classrooms, but it does not treat the learners' assessment in the practical works.

The authors present, in this research paper, a new environment for learner's self-assessment which extends our published works introduced in previous papers (Author et al.,2015a; Author et al.,2012). In the previous works, the researchers have used the graphs as a modeling tool. So, the researchers could only detect syntactic errors in the learner's programs. In the present work, the authors have managed to detect syntactic and semantic errors in the learner's programs by using ontologies as modeling tools.

The main objective of the developed environment is to evaluate learners' practical works about the development of Java object-oriented programs. Among the approaches quoted above, the approach proposed in Alonso(2008), Mitrovic, and Suraweera(2004) consider only the structural aspect of the application (i.e. the dynamic aspect is omitted). Furthermore, only the approach proposed in Alonso(2008) is based on matching techniques. It uses simple matchers during the comparison process between class diagrams. Our approach considers both structural and dynamic aspects of object-oriented programs. It uses a new hybridization strategy between two kinds of ontology matching (static and dynamic matching) and agents to compare, learners' programs, and teacher's one.

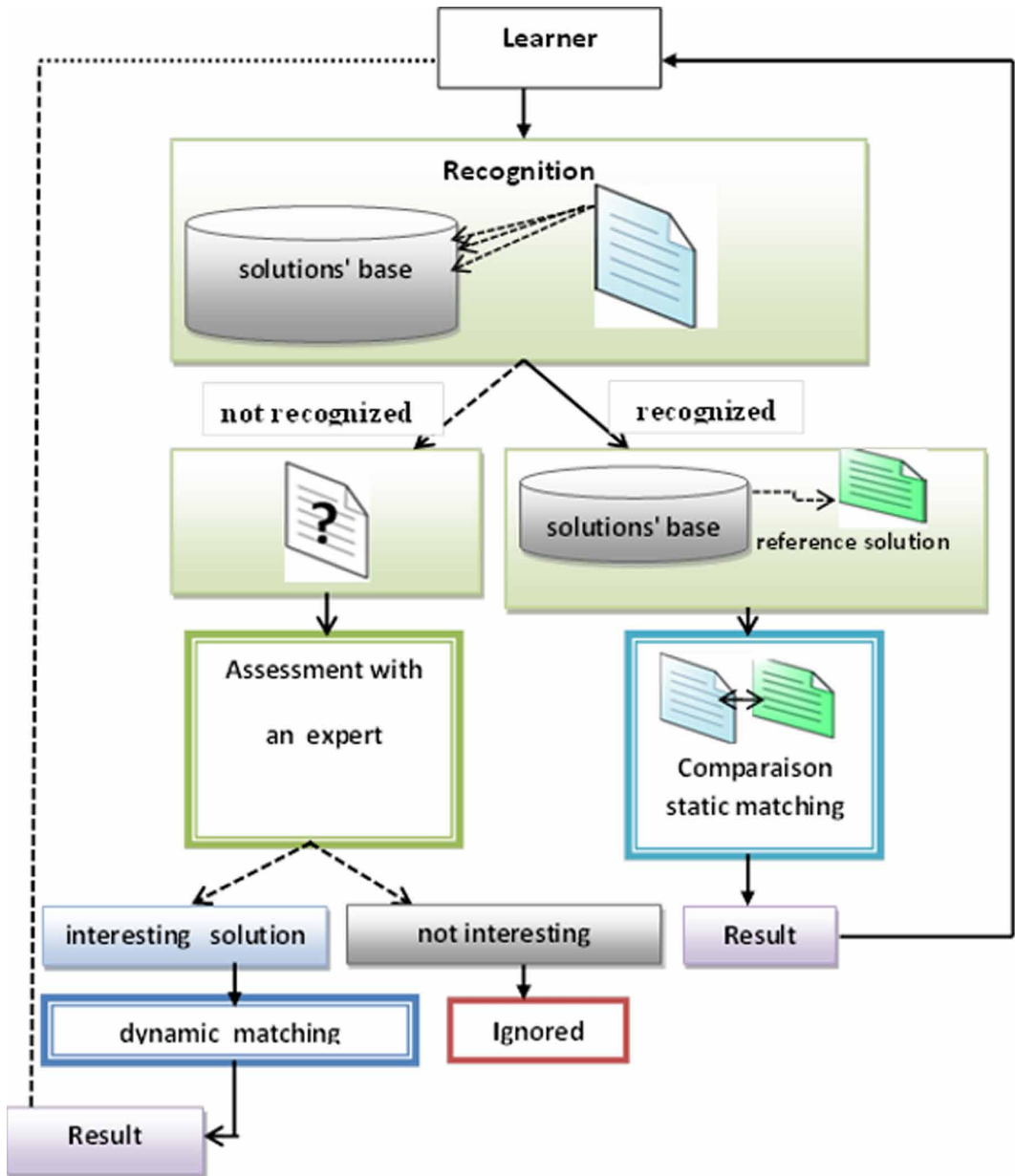
METHODOLOGY OF THE SELF- ASSESSMENT PROPOSED APPROACH

The process of learners' self-assessment the authors proposed consists of two main phases (Figure 1). During the two phases 1 and 2, the teacher's program and the learners' program are treated after they are transformed into two ontologies. The researchers detail in what follows the two phases.

Phase One: During this phase, the learner and the teacher have to develop two Java object-oriented programs that undergo an analysis process for generating two ontologies. The first one is generated from the teacher program, it is named reference ontology(RO)and the second one is generated from the learner program it is named learner ontology(LO).

Phase Two: This phase consists of applying the matching technique(Euzenat and Shvaiko,2008; Euzenat et al,2010; Giuseppe and Talia,2010; Zghal,2010), which the authors adopted. It offers two kinds of matching: static matching and dynamic matching. .The second step of the assessment process consists of recognizing the reference solution in the solution's base: to compare the proposed solution with the solutions' base, the authors must measure the degree of similarity

Figure 1. The methodology of the assessment proposed approach



and retain the solution closest to the solution proposed by the learner. Here the researchers propose an agent named decision agent this agent gives us two possible decisions according to a decision threshold (DT):

- reference ontology found: a found ontology can be presented in two forms: identical or similar.
 - identical ontologies: ($LO = RO$) here the similarity (Sim) calculated by the decision agent is equal to 1.

Table 1. Examples of the obtained results during the assessment of learner's program in the system

Concept 1 (teacher)	Concept 2 (learner)	Results of the calculated similarity between concept 1 and concept 2
Public class person	Public private class person	0.8
Public int end_date ;	Public int end date ;	0.72
public int id_card;	Id_card (without type)	0.64
Void veif(){hotelbooking.exist();}	void verif(){}	0.52

- Similar ontologies: ($LO \approx RO$) here the similarity (Sim) calculated by the decision agent is less than 1 and greater than the decision threshold (DT). ($DT < Sim < 1$).

In this case(reference ontology found), the authors apply the static matching between the two ontologies to make the comparison between the learner's program and the teacher's one.

- reference ontology not found: here the similarity (Sim) calculated by the decision agent is less than the decision threshold. ($Sim < DT$) in which case the intervention of a human expert is necessary. The human expert will begin to assess the learner's solution:
 - if he decides that it is interesting: in this case, it is proposed to generate a dynamic matching between the learner ontology and the ontologies that exist in the solution's base. This matching has for goal to find the correspondences (the semantic relations) between the learner's ontology and the ontologies in the solution's base to diminish the addition of a new ontology in the base.
 - if he decides that it is not interesting: in this case, the solution is rejected.

Let us note that to calculate the similarity between the two ontologies the authors re-use the matching algorithm of Wu-Palmer and Dice(1994), This algorithm is based on ontologies and the similarity calculation between ontology concepts.

$$\text{Wu-Palmer : } SimwPalmer(C1, C2) = \frac{2 * D}{D1 + D2 + 2 * D} \text{ (wu-palmer and Dice,1994)}$$

$$\text{Dice : } Simc(x, y) = \frac{2xy}{\|x\|_2^2 + \|y\|_2^2} \text{ (Palmer\&Wu,1994)}$$

$$\text{The Wu-Palmer measurement: } Sim wPalmer(C1, C2) = \frac{2 * D}{D1 + D2 + 2 * D} \text{ (Wu-Palmer and}$$

Dice,1994)The principle for calculating the Wu-Palmer similarity is based on the distances D1 and D2 which separate the concepts C1 and C2 from the concept (the most specific father) (MSF) and the distance D that separates MSF from the root concept.

Distance_WPalmer Calculated

```
Distance_WPalmer (concept1, concept2)
D←0; D1←0; D2←0 ;
MSF←seek for father (concept1, concept2) ;
D ←distance (root,MSF) ;
```

D1 ←distance (concept1, MSF) ;

D2 ←distance (concept2, MSF)

In table 1 below the authors present the results of the similarity calculated:

Explanation

The syntax error value is 0.9.

The semantic error value is 0.8.

Example

The similarity between C1 (Void verif () {hotelbooking.exist ();}) and C2 (void verif () {}) is given by:

D: the distance from the root concept.

D1: the distance of the concept of the learner's program.

D2: the distance from the concept of the teacher's program.

$$\text{SimwPalmer}(C1, C2) = (2 * 1) / (0.8 + 1 + 2 * 1) = 0.52$$

ARCHITECTURE OF THE SELF-ASSESSMENT PROPOSED APPROACH

The architecture of the proposed approach has four levels:

- The resource level: contains resources to be evaluated (learners' programs).
- The ontology level: presents a modeling of resources in the form of concepts linked by hierarchical relationships. Each resource is associated with several concepts.
- The interface level: where we find the learner and the three agents (decision agent, ontology agent, and matching agent).
- The expert level: helps the matching and ontology agents. After his intervention for assessing the proposed solution: if he decides that it is interesting he sends a positive message to the matching agent and the ontology agent, if he decides that it is not interesting he sends an ignorance message of the solution to the matching and ontology agent.

The Role of Each Agent in the Self-assessment Proposed Architecture

Decision Agent (DA)

it is responsible for looking for the reference solution of the solution proposed by the learner if it found, it sends a message to the matching agent (MA) to launch the static matching. In this case, the ontology agent does not work. In the case where the decision agent does not find the reference solution. It sends a message to the matching agent to launch a dynamic matching, the matching agent itself sends a message to the ontology agent to create relations (correspondences) between the concepts of the proposed solution ontology and the concepts of the ontologies in the solution's base.

Matching(Alignment) Agent (MA)

it is responsible for launching the two kinds of matching: static and dynamic matching according to the message sent either by the decision agent or by the expert.

Ontology Agent (OA)

it is responsible for creating relations (correspondences) between the concepts of the proposed solution ontology and the concepts of the ontologies in the solution's base in the case where the proposed

solution does not find in the solution's base and the human expert he decides that the proposed solution is interesting after his assessment intervention.

Let us note that, this paper is a part of a global project. For this reason, the authors are focusing in this research paper on the design and the development of the ontologies.

The Modeling of the Practical Work With the Ontology

According to Boaarab-Dahmani et al.(2017) an ontology has been defined as a formal representation of knowledge. An ontology is made up of four main elements: concept, instance, relation, and axiom.

- Concepts: are the fundamental elements of an ontology. They represent generic classes or a group of objects, in a domain, which share common properties. The concepts are organized hierarchically or each of them can have several sub-concepts and have one or more parent concepts(Hadadi & Bouaarab-Dahmani, 2019)
- An instance: is an occurrence object of concept.
- A relation: used to express the non-taxonomic semantic relation between two concepts in a given domain. Indeed, the so-called "semantic" relations are above all defined between instances. The authors derive from its relations between concepts, with a more or less well-defined semantics (Hadadi & Bouaarab-Dahmani, 2019).
- The role of an axiom in an ontology is to impose constraints on concepts, their instances, and their relationships.

The researchers explain the interest of using ontologies in the field of distance learning of practical work, particularly in the learners' assessment in this type of activities.

The Benefits of Using Ontologies in Interactive Environments for Learning Practical Work Activities

- The formal representation of knowledge: ontology provides the basis for a formal encoding of entities, attributes, their relationships.
- Reuse and sharing of educational objects and this is relevant in the case of systems using educational resources that are already built because building them again can waste time. This through reusable ontology libraries.
- Identification of educational objects: an ontology can be used as a meta-descriptor to describe the semantic content of educational objects;
- Knowledge acquisition: The use of ontologies increases the speed and reliability of the knowledge acquisition process when building a practical work.
- It provides annotation markers that might facilitate the interoperability and exchange of learning resources,

The General Structure of the Practical Work Ontology

In this work, the authors need an ontology to fully understand the structure of practical work, as well as to ensure the correct assessment of learners. To build an ontology, there are several effective and correct methodologies. The designer will carefully choose the most suitable methodology with his objectives so that he does not find a major divergence that sometimes leads to a contradiction. For the construction of our ontology, the researchers relied on the use of the Stanford methodology: it is suitable with the work on E-Learning. The authors used this method because Stanford University itself, which is developing the latter(ontology), is developing an editor called "protégé 2000" to properly show the practical side of ontology. This method goes through the following steps to build an ontology (Konys, 2018).

- Step 1: Determine the domain and scope of the ontology

The field of use designed in our ontology is E-Learning. The purpose of using our ontology is to properly structure the practical work offered by the teacher, as well as implicitly ensuring an automated task which is the self-assessment task which is the key point in distance training. The authors' ontology will be used by two actors: the learner and the teacher.

- Step 2: Reuse of existing ontologies

The authors don't need to reuse an existing ontology, the authors have to build their ontology because the domain is restricted.

- Step 3: List the important terms of the ontology

The important terms in our work are: practical work, assessment, resource, editor, correction type, learner answer...etc.

- Step 4, 5, and 6: Description of the ontology classes, their properties, and the class hierarchy

The authors have summarized these steps as follows: table 2 details each class with its attributes and its designation.

To create our ontology the authors used the "protégé 2000" editor. "Protégé 2000" is an authoring system for creating ontologies. It was created at Stanford University and very popular in the field of the Semantic Web and computer science research. "Protégé 2000" is developed in Java. "Protégé 2000" can read and save ontologies in most ontology formats: RDF, RDFS, OWL, etc.

In figure 2, the authors present an OntoGraf generated from "Protégé 2000" editor to clearly understand the hierarchy of the practical work ontology. Consequently figure 3 shows the creation of the practical work ontology with the "protégé2000" editor.

In Table 3, the researchers represent the different links between some classes of practical work ontology, in our ontology: there are different types of links which are:

- Generalization/specialization type links: this is "is a" type links: this is known in inheritance links, they are defined in the strict hierarchy of the model.
- The links "is connected of": this type of link defines the semantics between two classes. In our ontology, the authors used generalization/specification type links as well as a set of semantic links between classes.

The Proposed Hybrid Matching-Agent (Alignment) Algorithm Applied in the Self-Assessment Proposed Architecture

Below is the algorithm that describes the general assessment method

```
[1] :Input : S : solution ; RS: reference solution ; recognized :  
boolean ;  
[2] :Output: assessment solution S, Sim, a set of syntactic and  
semantic errors, observations ;  
[3] : Begin  
[4] :S-modeling the solution S; // with ontology: two ontologies  
are generated: learner's ontology from the learner's program and
```

Table 2. The set of classes, attributes of the Practical work ontology

Class	Data Properties	Designation
PW(practical work)		This is the main class of our ontology.
date	Start-date End-date	Every PW has a start date and an end date.
Language	None-language	This class shows the language used by the learner to write his code.
Notion	Content-notion	Each PW has a notion that shows the main goal to be achieved and the work to be done in that practical work.
Example	Content –exple	Each PW presents examples to help learner to understand.
Observation	Observation	This class represents well the observation assigned to a learner after he has passed his assessment (the result of the test).
Question	Num-qst Ennonce-qst Type-qst Num-open-qst- Content-open -qst Num-closed-qst Content-closed-qst	As the precise class for assessing learners, it is presented as an open or closed question. Open question: represents questions for which the answer is open, a learner constructs his practical work freely. Closed question: represents questions whose answer is true or false or by a choice between propositions well defined in advance.
the standard answer	Ennonc-qst Content-Stdans	It represents the answer key to an open question, but it is not a final version as the learner will respond in their style.
Answer	Content -Answ	It shows the answer proposed by the learner concerning the practical work.
Error	Content-err Num-er Type-er	This is the class that defines the set of errors that the learner makes when responding to an open-ended question. And it contains subclasses: Syntax and lexicon errors. Semantic error which represents all the errors of meaning.
Goal	Goal	The objective of the PW proposed by the teacher.
Author	First name Last name Grade	This class represents the author who writes the resource (reference) provided to the learner.
Editor	First name Last name	This class represents the publisher name of the resource.
Format	Type-format	This class represents the format of resource.
Language	Language	This class represents the language of the resource.
Type	Type	It shows whether the resource is of type site web, article,....
Title	Title	The title of each PW suggested by the teacher.

Figure 2. The OntoGraf of the practical work ontology

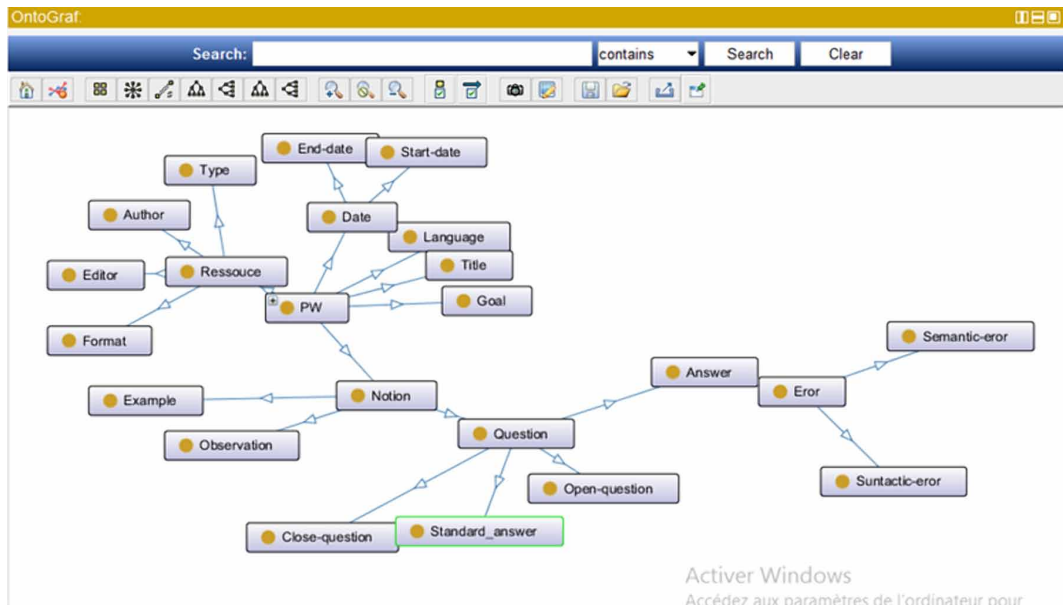


Figure 3. The hierarchy of concepts in the practical work ontology

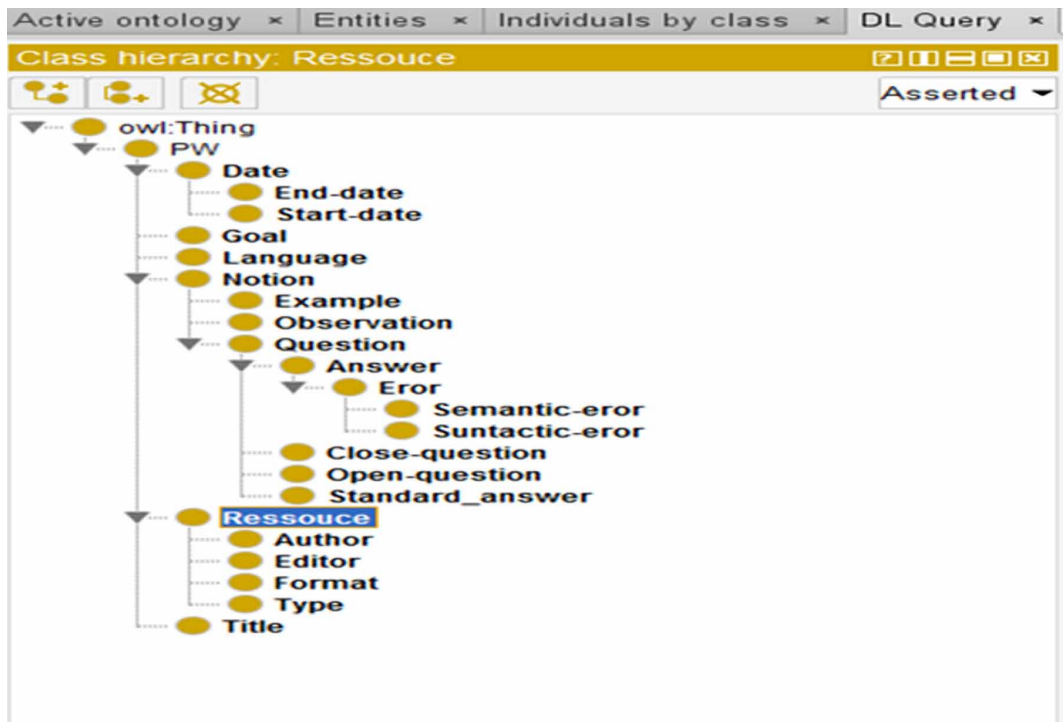


Table 3. The description of the links between the classes of the practical work ontology

Object Properties	Domains	Ranges	Inverse Of	Comment
attribute	Answer	Observation	is attributed to	Each completed PW has an observation.
Compare with	Answer	Standard _answer	Is compared to	The standard_answer is compared by the learner's answer to get an assessment.
Contain	Notion	Question	Is Expressed A	Each notion of PW contains questions.
Is a	Error	Syntactic_eror Semantique_eror	/	Each PW has errors and these can be syntactic or semantic.
May be	Question	Close_question Open_question	/	Each PW has closed or open questions.
Publish in	PW	Start-date	Is Date In	a PW has a publication date.
Must have	Question	Standar _answer	Contains	Each question in the PW has a typical standard_anwser.

teacher's ontology from the teacher's program

```

[5]:Recognizedrecognized(S, RS) ;
[6]: if recognized then // the decision is offered by the decision
Agent (DA)
[7]: Generated static Matching(S, RS); applying the matching
algorithm of Wu-Palmer and Dice(1994).
[8]:Return(résultats : feedback(syntactic and semantic errors),
observation) ;
[9]:else
[10]: Solution S assessed by a human expert
[11]:if S is interesting then//the decision is offered by a human
expert
[12]:Generated dynamic Matching;// To remove the problem of the
scalable base's solution
[13]:Return(results : feedback(syntactic and semantic errors),
observation) ;
[14]else
[15]:S is rejected
[16]:end
[17]:end
[18]:end

```

Algorithm Explanation

line [1]: represents *the inputs* of the algorithm:

S: represents the proposed learner's solution

RS: represents the reference solution proposed by the teacher

Recognized: this function has two values: *true or false*.

line [2]: represents *the outputs* of the algorithm

Sim: represents the calculated similarity between the learner's proposed solution and the teacher's solution.

a set of syntactic and semantic errors founded in the learner's solution.

Observation: this function has five possibilities (Excellent; Very good, Average Below, Average, Failing) according to the value of the similarity calculated, the system give the observation for the learner:

- if the *sim* \in [0.8; 1] \Rightarrow the observation will be "Excellent".
- if the *sim* \in [0.6; 0.8] \Rightarrow the observation will be "Very good".
- if the *sim* \in [0.5; 0.6] \Rightarrow the observation will be "Average".
- if the *sim* \in [0.3; 0.5] \Rightarrow the observation will be "Below Average".
- if the *sim* \in [0; 0.3] \Rightarrow the observation will be "Failing".

line [4]: the function **modeling** this function consists in modeling the solution proposed by the learner and the solution proposed by the teacher with an ontology. So two ontologies are generated: learner's ontology from the learner's program and teacher's ontology from the teacher's program.

line [5]: the function **recognized** consists in checking in the base solution for the ontologies the ontology similar to the ontology proposed by the learner: if the ontology founded in the base solution the function **recognized** will be **true**, but if the ontology is not founded, the function **recognized** will be **false**.

line [6]: if the function **recognized** will be **true**, the decision is offered by the decision Agent(DA) like it explain in the paragraph of the decision agent (AD) previously.

line [7]: the function **generate** consists of generating a static Matching between the learner's solution and the teacher's one (S, RS) by applying the matching algorithm of Wu-Palmer and Dice(1994).

line [8]: the function **Return** consists of showing the results for the learner, these results are feedback with syntactic and semantic error and observation. The observation has one of the five values possibles (Excellent; Very good, Average, Below Average, Failing).

line [9][10]: if the learner's proposed ontology is not founded in the base solution the function **recognized** will be **false**, in this case, the Solution "S" **assessed** by a **human expert**. If the human expert decides that the proposed solution is **interesting**, so the system will be generating a dynamic Matching **line [12]**. This dynamic matching is to found the **nearest solution** in the base solution for the solution proposed by the learner. The authors propose this type of matching to remove the problem of the scalable base's solution. In this case, the system will give the results **line[13]** (with feedback and observation) like it explained previously(line [8]).

line [14]: when the human expert decides that the learner's proposed solution is not interesting the system will reject it (**line[15]**).

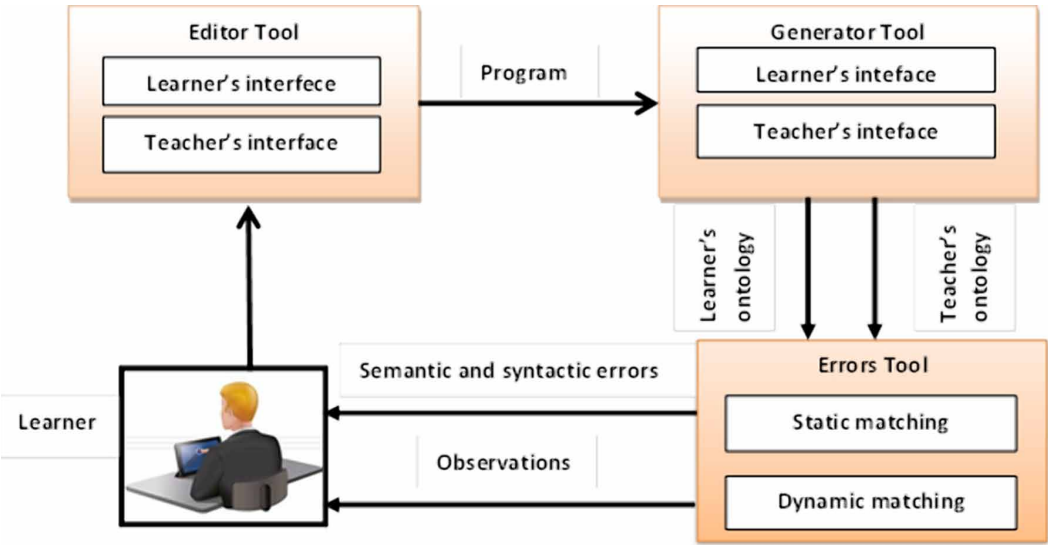
ENVIRONMENT PRESENTATION

General Architecture of the Assessment Tool

This work introduces a new learner self-assessment environment called S_Onto_ALPPWA : (System based Ontology for Assessing Learners' Programming Practical Works Activities), for allowing comparing learners' productions (Java object-oriented programs) with those elaborated by the teacher. The subjacent idea is to proceed to an indirect comparison of two object-oriented programs through their graphic representations described by two ontologies using the matching algorithms as a comparison strategy. S_Onto_ALPPWA is composing of three main components: the Editor Tool, the Generator Tool, and the Errors Tool. The most important component is the Errors tool, which is implemented using the matching developed algorithm that evaluates the learner's solution in terms of similarity values and observations. Its architecture is presented in Figure 4.

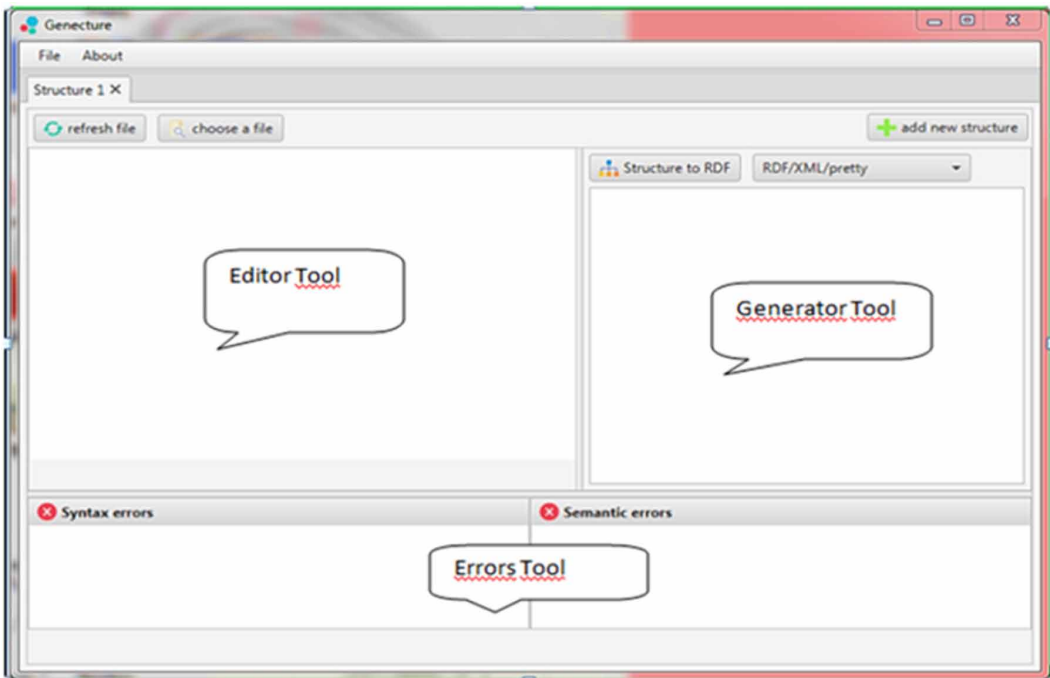
- **Editor Tool:** allows learners and teachers to edit codes and programs.
- **Generator Tool:** allows learners and teachers to generate different ontologies from their codes.

Figure 4. General architecture of the assessment tool



- **Errors Tool:** is The most important component. It is implemented using the developed matching algorithm that evaluates learner's solution in terms of similarity values to compare between learners' programs and the teacher's one and gives an observation to the learner according to the value of similarity calculated like flowing:

Figure 5. S_Onto_ALPPWA's different components



- $[0.8; 1] \Rightarrow$ the observation will be "Excellent".
- $[0.6; 0.8] \Rightarrow$ the observation will be "Very good".
- $[0.5; 0.6] \Rightarrow$ the observation will be "Average".
- $[0.3; 0.5] \Rightarrow$ the observation will be "Below Average".
- $[0; 0.3] \Rightarrow$ the observation will be "Failing".

It offers also to the learners a set of syntactic and semantic errors detected in their programs. Figure 5 presents the screenshot of the three different components of the assessment tool developed (S_Onto_ALPPWA).

Case Study

The Practical Work Statements

The authors present in what follows the environment using a concrete case study. It is about a simple hotel booking process. When a client arrives at the hotel, he presents his ID card. The hotel receptionist checks the availability of rooms. He proposes to the client the types and the price list of rooms. The client chooses a room and informs the date of departure to the hotel receptionist. The latter gives him the number of the reserved room.

Teacher's Java Object-Oriented Program Practical Work (Reference Solution)

in what follows a teacher's reference solution java object-oriented program

```
1. public class person{
2.     public String first_name;
3.     public String last_name;
4.     public String adress;
5. }
6. public class room_booking{
7.     public int id_card;
8.     public int room_number;
9.     public int start_date;
10.    public int end_date;
11.    public boolean reserve;
12.    public boolean smoking;
13. }
14. public class hotel_booking{
15.    room_booking rb = new room_booking();
16.    int nb_room;
17.    public boolean existe(){};
18.    public int get_nbfreeroom(){};
19.    public void reserve(int nb,int nb_ch){};
20. }
21. public class the_hotel_receptionist extends person{
22.    public int id;
23.    hotel_booking hotelbooking = new hotel_booking() ;
24.    public void verif(){hotelbooking.existe();}
25.    public void set(int id){
26.        int nb = hotelbooking.get_nbfreeroom();
27.        hotelbooking.reserve(id,nb);
28.    }}
29. public class client extends person{
```



```

30. the_hotel_receptionist p = new the_hotel_receptionist() ;
32. public int id_card;
33. public void want_reserve(){
34. p.verif();
35. p.set();
36. }}
37. public class Main{
38. public static void main(String[] args){
39. client client1;
40. client1.want_reserve();
41. }}

```

Generated Ontology from Java Object-oriented Program Practical Work

In the realized system, the researchers can generate for each practical work its ontology in the form

Figure 6. Ontology generation in the form of an RDF code



of an RDF / XML file. Figure 6 shows part of the RDF code generated from the Java Object-oriented Program practical work.

Program Fragment of Learner's Solution With Some Errors

Semantic Errors

- Define the same variables twice in different classes, for example:

```

public class client_a extends pers{
the_hotel_receptionist_a p;
int id_card;
String id_card;

```

```
void want_reserve() {  
p.verif();  
p.set();  
}  
}
```

- Define a method twice in the practical work , for example:

```
public class the_hotel_receptionist_a extends pers{  
int id;  
hotel_booking_a hotelbooking;  
void verif() {hotelbooking.existe();}  
void verif() {}  
void set(int id) {int nb = hotelbooking.get_nbfreeroom();  
hotelbooking.reserve(id,nb);  
}  
}
```

- The combination of two modifiers in the practical work, for example:

```
public private class person_a{  
final String first_name;  
String last_name;  
String adress;  
}
```

Syntactic Errors

- named an attribute or a class with a name reserved for a keyword in the language used for programming, for example:

```
public class person_a{  
}  
public class{  
}  
public class hotel_booking_a{  
protected room_booking_a rb;  
int nb_room;  
int if;  
boolean existe(){}  
int get_nbfreeroom(){}  
void reserve(public int nb,int nb_ch){}  
}
```

- Forget to put the comma, semicolon(, ;)...

```
public class room_booking_a{  
public int id_card,room_number,start_date;  
public int end date; // wait for a semicolon .
```

```
public boolean reserve;  
public boolean smoking;  
}
```

Figure 7. The learner chooses his programming practical work

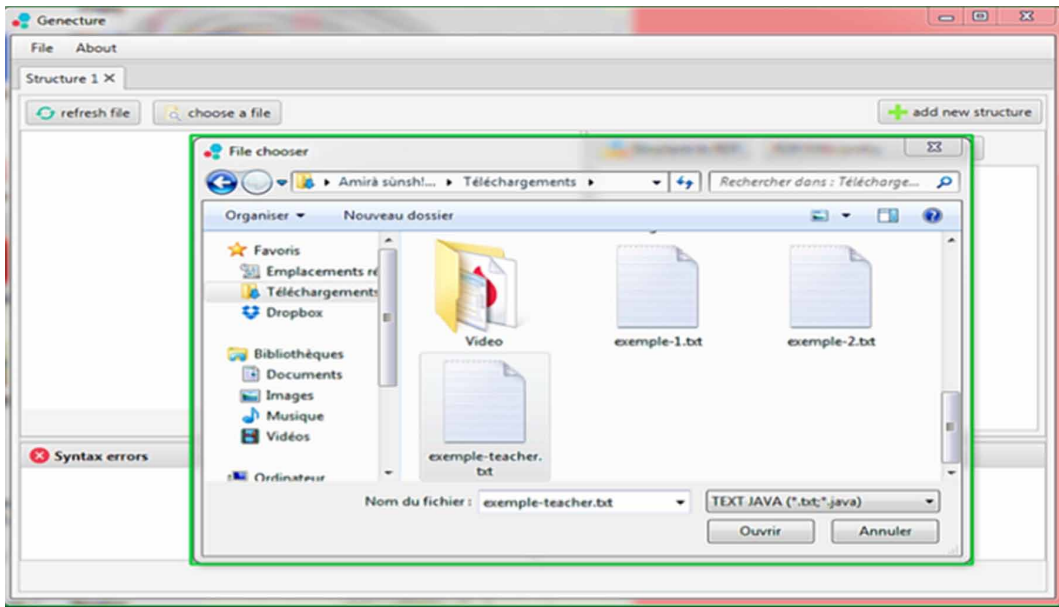


Figure 8. The learner writes his programming practical work

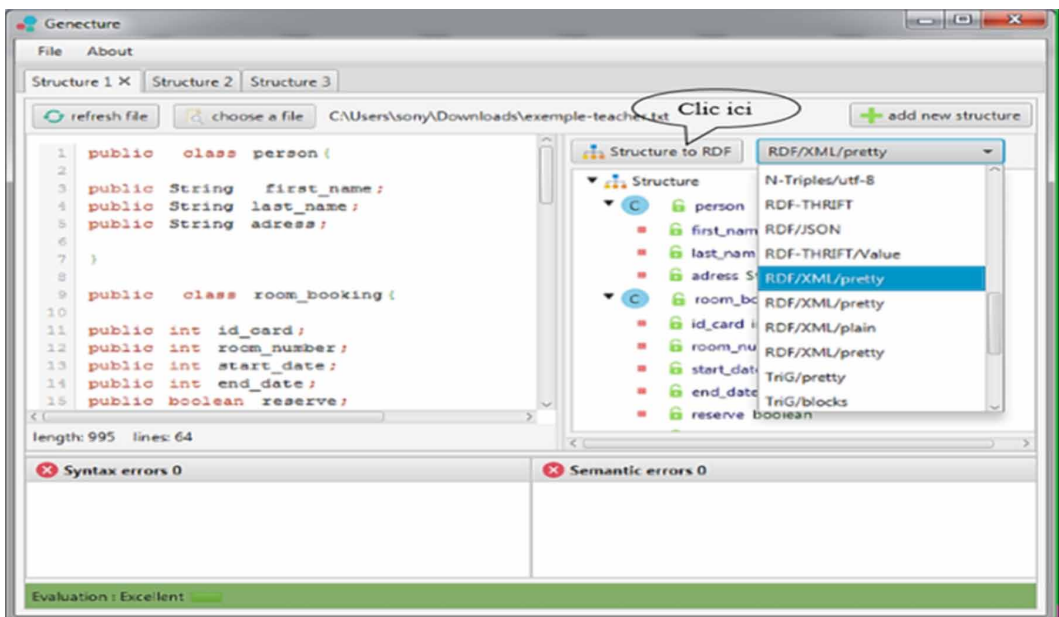
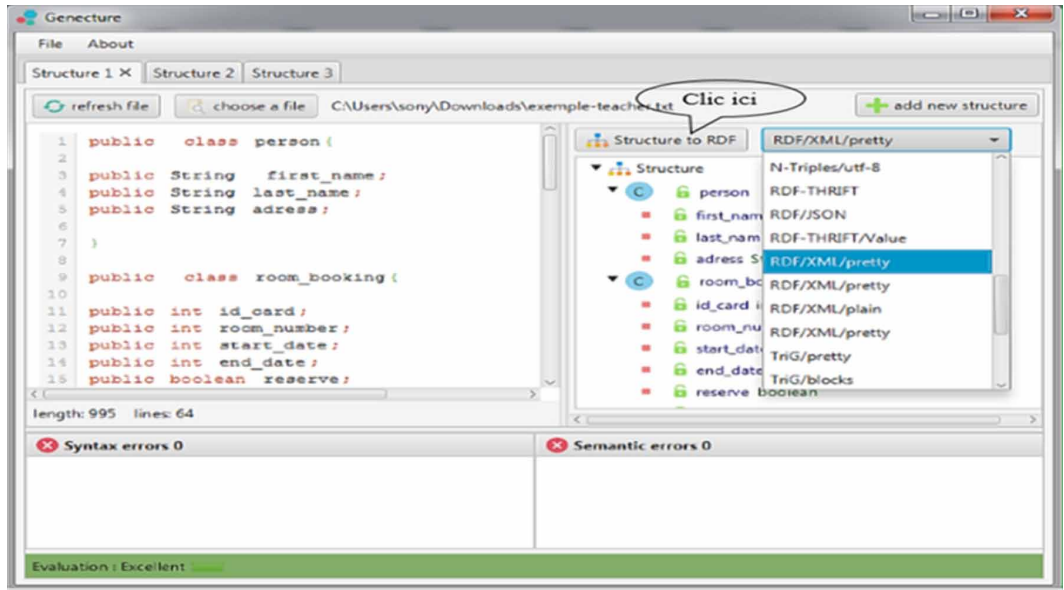


Figure 9. The generated ontology from the learner's program



Some Screenshots

From the Editor Tool

Using the developed environment, teachers and learners can write or choose their java object-oriented programs for implementing the hotel booking process. Figures 7 and 8 illustrate how to write or choose a practical work by the learner. Let us note that the learners cannot use the environment before the teacher's program is edited and the corresponding ontology is generated.

From the Generator Tool

The comparison of the two edited programs is accomplished by comparing their ontologies representations. Firstly, the teacher uses the generator tool to generate the ontology from his Java program. This ontology should be stored for later use. After that, the learner must generate the ontology from his Java program (Figure 9).

From the Errors Tool

The main objective of the developed environment is to allow learners to auto-evaluate their skills in java object-oriented programming. The rest of the steps are only done by learners (i.e., the teacher is not involved in the rest of the self-assessment process). Figure 10 illustrates the different errors detected and the observations. During the comparison process the authors proposed, the researchers use the teacher's ontologies as references (i.e., they are supposed correct because they are generated from the teacher's correct programs). So, the list of detected errors is essentially different concepts that are not found in the learners' program. Also, several kinds of errors may be detected in learners' program (see Figures 10,11).

Our tool displays to learners the list of the detected errors in their programs. Figure 11 illustrates the errors detected in learners' programs.

Figure 10. List of detected errors in the case of learner's correct programming practical work

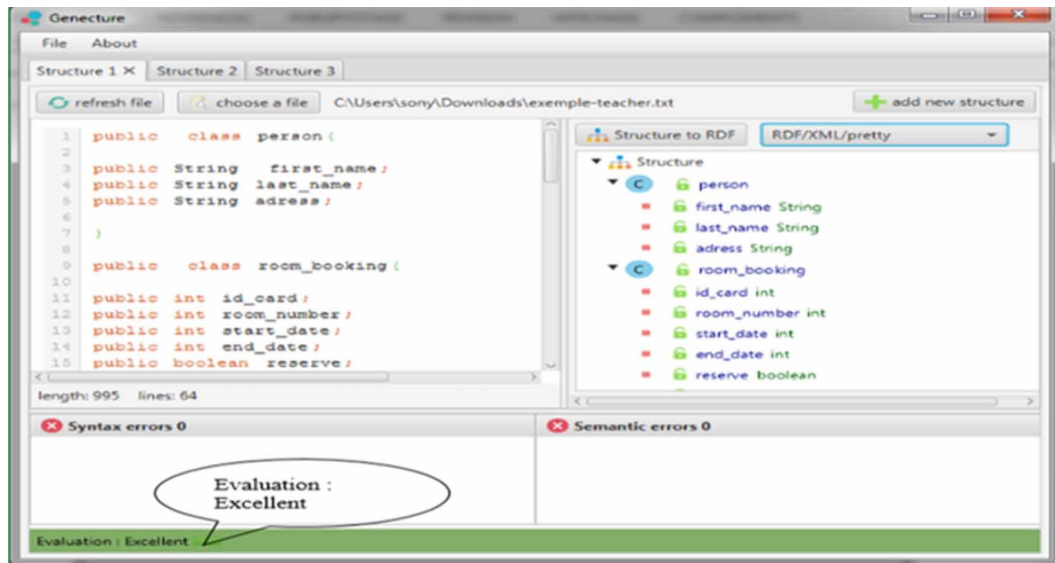
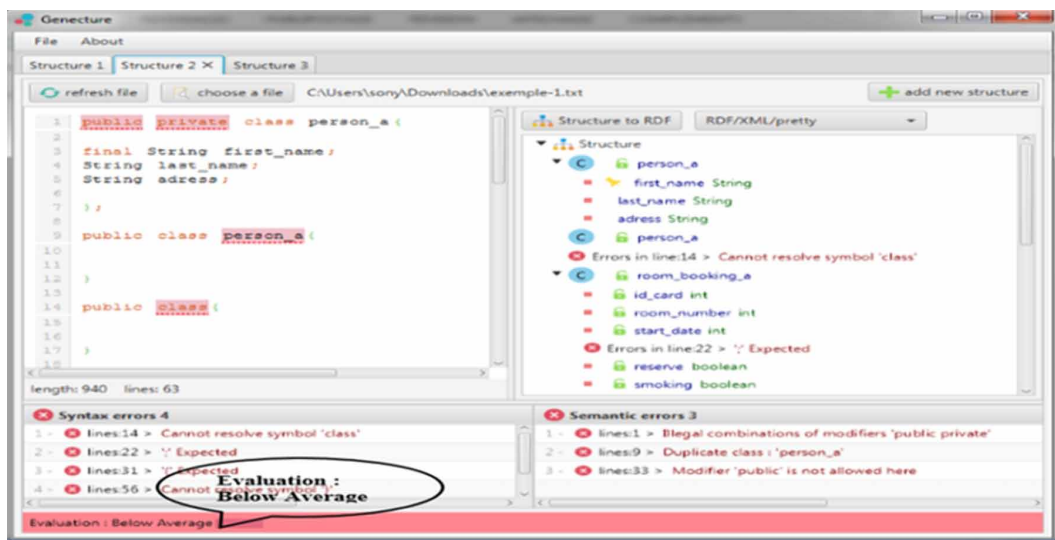


Figure 11. List of detected errors in the case of learner's incorrect programming practical work



BENEFITS AND EVALUATION OF THE DEVELOPED SYSTEM

Experiment Study

Methodology

To validate our contribution and evaluate our developed system, two experiments are conducted at the technique science department at Oum El Bouaghi University (Algeria). The authors ask to participate in the experiments sixty (60) students were randomly selected to take part in these experiments. The researcher adopted in the analysis of data and sample selection the student t-test methodology. The student test is a mechanism that makes it possible to decide between two hypotheses given the

results of a sample. H_0 and H_1 be two hypotheses (H_0 is called the null hypothesis, H_1 the alternative hypothesis), one and only one of which is true. The decision is to choose H_0 or H_1 . To confirm or reject a hypothesis the authors must compare the calculated P-value with the significance level $\alpha = 0.05$ to decide if there is a significant difference or no. So the two hypothesis reformulated from the research questions are:

Null hypothesis H_0 : “ the use of self-assessment CHEL for learning practical works activities based on a hybrid matching-agent algorithm and ontologies cannot ameliorate the learners’ results, and cannot reduce the rate of failure or abandonment of the programming in the introductory courses in first university cycle”?”.

Alternative hypothesis H_1 : “ the use of self-assessment CHEL for learning practical works activities based on a hybrid matching-agent algorithm and ontologies ameliorates the learners’ results, and reduces the rate of failure or abandonment of the programming in the introductory courses in first university cycle”?”.

To confirm or reject a hypothesis, the authors use the student test with the following detailed experience:

Participants

before discussing our online experiment, sixty (60) students were randomly selected to take part in this experiment. All were second-year students at the technique science department at Oum El Bouaghi University (Algeria). The students are novices in the field of object-oriented programming language. Therefore, they had no pre-requisites of the oriented programming language practical work. The participants were randomly divided into two groups that are group 2 ($n=30$) – in which the assessment tool was used in the learning process–, and group1 – in which the assessment tool was not used in the learning process.

The authors study the effects of the assessment tool on one hand, for the mastery of the concepts of the object-oriented programming language and the resolving of the blocking situation according to the learners. On the other hand, we study the effects of this tool for the number of intervention of the teacher assistant of the practical work. The researchers want to improve if the students in group 2 were better able to solve their problems than the students in group 1.

Experimental Procedures

In the first stage, the students were instructed in the basic knowledge of Java oriented programming language with practical work after receiving this fundamental knowledge, the students were asked to take a pre-test, which aimed to evaluate their basic knowledge before participating in the learning activity. In the second stage, the students in group 2 were arranged to learn with our system where the assessment tool is used. On the other hand, the students in group1 learned with another version of the system without the assessment tool. After conducting the learning activity, the students took a post-test.

RESULTS AND DISCUSSION

This part is concerned with demonstrating the effectiveness of the proposed system for both the learner or the teacher.

Results Specific to the Learners

Results Specific to Pre-Test (Before Using the Assessment Tool)

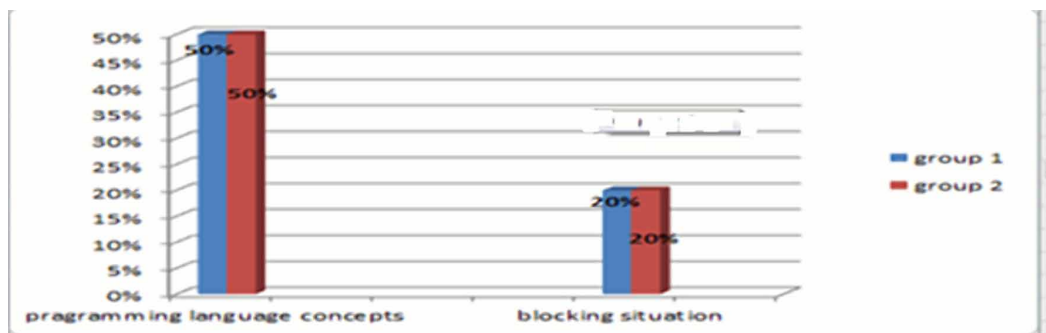
The independent sample t-test was performed first to determine whether group 1 and group2 had the same prior knowledge on the studied domain. As can be seen in Table 4 and figure 12, there were no significant differences between group1 and group 2 in their prior knowledge ($P\text{-value} = 0.9051$

Table 4. Independent t-test before using of assessment tool

	Group 1	Group2
Mean	10,025	09,96
SD(standard deviation)	2,1298	2,1825
T-test	0,1197	
df	58	
p-value	0,9051	

SD: standard deviation, t-test: student test value, df: degree of freedom, P-value: probability value.

Figure 12. Results before integration of assessment tool



>0.05). This result implying that these two groups did not significantly differ before the experiment. That is, the two groups of students had statistically equivalent abilities before taking the subject unit.

As shown in Figure 12, for both group 1 and group 2 only 50% for learners who can mastery the concepts of oriented programming language and only 20% for them who can resolve their blocking situation. So, there were no significant differences between group1 and group 2 in their prior knowledge.

Results Specific to Post-Test (After Using the Assessment Tool)

Let us remember that: the Null hypothesis reformulated from the research questions is:

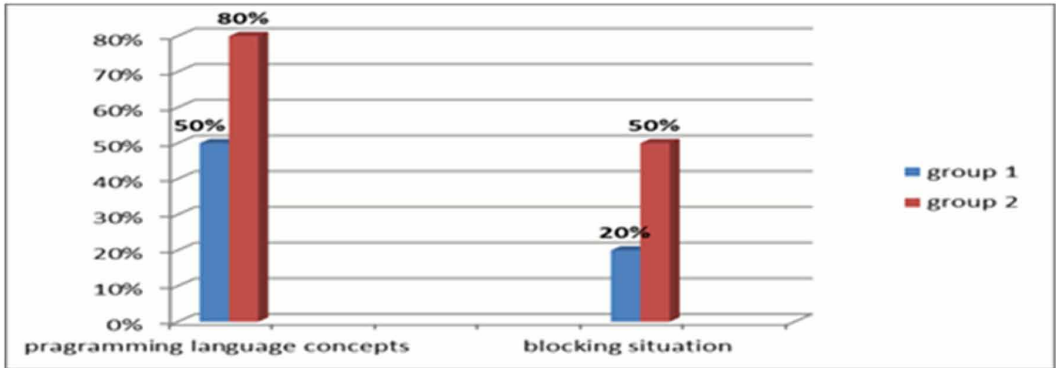
Null hypothesis H0: "the use of self-assessment CHEL for learning practical works activities based on a hybrid matching-agent algorithm and ontologies cannot ameliorate the learners' results,

Table 5. Independent t-test after using the assessment tool

	Group 1	Group2
Mean	10,0833	11,25
SD(standard deviation)	2,0734	2,3184
T-test	-2,0545	
df	58	
p-value	0,0444*	

*Significant difference, SD: standard deviation, t-test: student test value, df: degree of freedom, P-value: probability value.

Figure 13. Effects of assessment tool on the mastery of the concepts and the blocking situations



and cannot reduce the rate of failure or abandonment of the programming in the introductory courses in first university cycle''.

The independent sample t-test was performed to compare the mean scores for the two groups in post-test results. The t-test determined that the differences measured between the means of group 1 and group 2 were significantly different and could be attributed to the learning through the assessment tool given to the group2. Results show the group2 performed significantly better than group 1 in the post-test ($t = -2.0545$, $df = 58$, $p\text{-value} = 0.0444 < 0.05$). Table 5 shows the comparison results, therefore the null hypothesis H_0 is rejected. So, the alternative hypothesis H_1 is proved and we can affirm that ''the use of self-assessment CHEL for learning practical works activities based on a hybrid matching-agent algorithm and ontologies ameliorates the learners' results, and reduces the rate of failure or abandonment of the programming in the introductory courses in first university cycle''.

The findings were highly representative as indicated in figure 13, for group 1 only 50% for learners who can mastery the concepts of oriented programming language and only 20% for those who can resolve their blocking situation. In contrast to group 2, 80% for learners who can mastery the concepts of oriented programming language and 50% of them who can resolve their blocking situations. *The authors notice that the group in which the assessment tool is used was better able to solve their practical work than the one in which the assessment tool is not used.*

Figure 14. Effects of assessment tool on teachers' number of intervention



Results Specific to the Teacher

In Figure 14, the authors observe for group 1 the number of interventions of the teacher's assistant of practical work is 80 interventions in the session of 2 hours (3 interventions per learner) but for group 2 the number of interventions becomes 40 (2 interventions per learner). ***We notice that the assessment tool makes free the teachers assistant of practical works.***

CONCLUSION AND FUTURE WORK

This paper deals with the learners' cognitive state dimension that is updated based on the results of the proposed practical works activities. The researchers propose a self-assessment CEHL (Computing Environment for Human Learning) environment. This kind of self-assessment has been a constant demand in higher education for graduating students to be equipped with the capacity for lifelong learning that helps learners to continually upgrade their skills and knowledge through their self-motivation and learning. The proposed system allows the learner to accomplish his work alone and to assume his responsibility because through self-assessment and learning by the problem, it improves the acquisition of information and the self-correction of his mistakes. The developed environment allows comparing indirectly the learners' programs with the teacher's one through the comparison (syntactic and semantic comparison) of their ontologies descriptions using two kinds of matching (static and dynamic matching). The authors deduct that using ontologies plays an important role in instructional design and the development of course content. It helps learners access to content in a knowledge-guided way.

So the researchers can say that the proposed system is a promising and ambitious tool for helping in the assessment of students' computer programming practical works activities and it can free teachers from all the disadvantages of traditional students' assessment by shortening time, reducing effort, and granting accurate and objective assessment. The authors believe that the efficiency of the proposed system can help teachers in scoring large classes with the same precision. The major limitation of the proposed methodology especially the hybrid matching-agent algorithm, that the authors developed only the part of the ontologies but the part of the agent not developed yet.

This study opened new areas of research, to provide new suggestions and additions in the current work to improve the services provided in the future. For future work, we still need to assess how much this system can improve students' achievement in programming but in the Moocs context. Also, the authors plan to extend their developed system by :

- Complete the development of the agent part
- Assessing learners' practical works activities in different domains not just in computer science.
- Designing other ontologies and combine them with ours to enrich the semantics especially in the assessment task.

REFERENCES

- Aiouni, R., Bey, A., & Bensebaa, T. (2018). eALGO: An automated assessment tool of flowchart programs for novices. *International Journal of Innovation and Learning*, 23(2), 5. doi:10.1504/IJIL.2018.088785
- Alonso, M., Auxepaules, L., Lemeunier, T., & Py, D. (2008). Design of Pedagogical Feedbacks in a Learning Environment for Object-Oriented Modeling. *Promoting Software Modeling through Active Education. Educators Symposium of the ACM/IEEE 11th International Conference on Model Driven Engineering Languages and Systems (MoDELS'08)*, 39-50.
- Andrade, H., & Du, Y. (2007). Student responses to criteria-referenced self-Assessment. *Assessment and Evaluation. Int. Journal of Higher Education (Columbus, Ohio)*, 32(2), 159–181.
- Arnous, S. (2014). *Conception générique d'un outil de configuration de e-TP* [Ph.D. dissertation]. INSA, Lyon, France.
- Bennouna, M., Delestre, N., Pécuchet, J.-P., & Tanana, M. (2008). Évaluation du savoir-faire en électronique numérique à l'aide d'un algorithme de classification [paper presentation]. The conférence Technologies de l'Information et de la Communication pour l'Éducation (TICE'08), Paris, France.
- Bennouna, M., Delestre, N., Pécuchet, J.-P., & Tanana, M. (2009). Génération d'exemples pour l'évaluation de l'apprenant en électronique numérique à l'aide d'un algorithme de classification [paper presentation]. The Conférence EIAH, France.
- Biggs, J., & Tang, C. (2011). *Teaching for quality learning at university*. Open University Press.
- Black, P., & Wiliam, D. (1998). Assessment and classroom learning. *Assessment in Education: Principles, Policy & Practice*, 5(1), 7–74.
- Bouarab-Dahmani, F., Comparot, C., Si-Mohammed, M., & Charrel, P. J. (2017). Ontology-Based Teaching Domain Knowledge Management for E-Learning by Doing Systems. *Electronic Journal of Knowledge Management*, 13(2), 156-171.
- Boud, D., & Falchikov, N. (2007). Assessment for the longer term. In D. Boud & N. Falchikov (Eds.), *Rethinking assessment in higher education*. London: Routledge.
- Boussaha, K. (2011). Modélisation d'une situation d'évaluation de l'apprenant avec UML: CAS d'application pour l'apprentissage des langages de programmation [paper presentation]. The 8eme colloque sur l'optimisation et les systèmes d'information, COSI'2011, Guelma, Algeria.
- Boussaha, K. (2016). *l'Évaluation de l'apprenant dans les environnements d'apprentissage de Télé-TPs* [Ph.D. dissertation]. Badji Mokhtar Univ, Annaba, Algeria.
- Boussaha, K., & Bensebaa, T. (2009). *Design of an environment of Remote practical works between realities and prospects* [paper presentation]. The International Conference of Novel Digital Technology, Republic Check.
- Boussaha, K., Mokhati, F., & Chaoua, Z. (2015a). Architecture of a specific platform for training practical works: Integration of learners assessment component. *Int. J. of Technology Enhanced Learning*, 7(3). doi:10.1504/IJTEL.2015.072809
- Boussaha, K., Mokhati, F., & Chaoua, Z. (2015b). *Using a matching approach to assess learners in their Practical works activity with a specific CEHL* [paper presentation]. The 2nd International Conference on Multimedia Information processing, CITIM'2015, Mustapha Stambouli University-Faculty of Science and Technology, Mascara, Algeria.
- Boussaha, K., Mokhati, F., & Taleb, N. (2012). *A novel learner self-assessment approach – application to practical work* [paper presentation]. The 4th international conference on computer supported Education, the CSEDU 2012, Porto, Portugal.
- Chilowicz, M. (2010). *Recherche de similarité dans du code source* [Ph.D.dissertation]. Paris –Est Univ, France.
- Choquet, C., Després, C., Iksal, S., Jacobi, P., Lekira, A., Py, D., Pham, T., Ngoc, T., Pham, & Ngoc, D. (2011). Using indicators during synchronous tutoring of practical work [paper presentation]. The 11th IEEE International Conference on Advanced Learning Technologies, ICALT'11.

Cross, V., & Xinran, Y. (2011). *Investigating Ontological Similarity Theoretically with Fuzzy Set Theory, Information Content, and Tversky Similarity and Empirically with the Gene Ontology* [paper presentation]. The 5th International Conference on Scalable Uncertainty Management, Dayton, OH.

De Corte, E., Mason, L., Depaepe, F., & Verschaffel, L. (2011). Self-regulation of mathematical knowledge and skills. In B. J. Zimmerman & D. H. Schunk (Eds.), *Handbook of self-regulation of learning and performance*. New York: Routledge.

Deep, S., Salleh, B. M., & Othman, H. (2019). Study on problem-based learning towards improving soft skills of students in effective communication class. *International Journal of Innovation and Learning*, 25(1), 17–34. doi:10.1504/IJIL.2019.096512

Euzenat, J., DjoufakKengue, J. F., & Valtchev. (2010). The Results of the Ontology Alignment Evaluation Initiative. *Ontology Matching Workshop, International Semantic Web Conference*.

Euzenat, J., & Shvaiko, P. (2008). *Ten Challenges for Ontology Matching* [paper presentation]. Confederated International Conferences, OTM 2008, Monterrey, Mexico.

Giuseppe, P., & Talia, D. (2010). UFOme: An Ontology Mapping System with Strategy Prediction Capabilities. *Data Knowledge*, 69(5), 444–471.

Guibert, N. (2006). *Validation d'une approche basée sur l'exemple pour l'initiation à la programmation* [Ph.D,dissertation]. Poitiers Univ, France.

Guillaume, D. (2006). *Vers une scénarisation de l'évaluation en EIAH* [paper presentation]. EIAH.

Guittet, L., Guibert, N., & Girard, P. (2005). A study of the efficiency of an alternative programming paradigm to teach the basics of programming. LISI/ENSMA, 86961 Futuroscope Chasseneuil Cedex, France.

Hadadi, L., & Bouaarab-Dahmani, F. (2018). Multi-level computer-aided learner assessment in massive open online courses. *International Journal of Knowledge and Learning*, 12(4), 2018.

Hadadi, L., & Bouaarab-Dahmani, F. (2019). Gradual Learners' Assessment in Massive Open Online Courses Based on ODALA Approach. *Journal of Information Technology Research*, 12(July-September), 2019.

Indira, B., Valarmathi, K., & Devaraj, D. (2019). An approach to enhance packet classification performance of software-defined network using deep learning. *Soft Computing*, 23(18), 8609–8619. 03975-810.1007/s00500-019-

Konys, A. (2018). An Ontology-Based Knowledge Modelling for a Sustainability Assessment Domain. *Sustainability*, 10, 300. doi:10.3390/su10020300

Mitrovic, A., & Suraweera, P. (2004). An Intelligent Tutoring System for Entity Relationship Modelling. *International Journal of Artificial Intelligence in Education*, 14(3), 375–417.

Nicoland, D., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. in *Studies Int. Journal of Higher Education (Columbus, Ohio)*, 31(2), 199–218.

Palmer, M. S., & Wu, Z. (1994). Verb Semantics And Lexical Selection. *The 32nd Annual Meeting of the Association for Computational Linguistics*, 133-138.

Pang, E., Wong, M., Leung, C. H., & Coombes, J. (2019). Competencies for fresh graduates' success at work: Perspectives of employers. *Industry and Higher Education*, 33(1), 55–65. <https://doi.org/10.1177/0950422218792333>

Pintrich, P. R., & Zusho, A. (2002). The development of academic self-regulation: The role of cognitive and motivational factors. In A. Wigfield & J. Eccles (Eds.), *Development of achievement motivation*. San Diego, CA: Academic Press.

Sadler, D. R. (1989). Formative assessment and the design of instructional systems. *Australia in Instructional Science*, 18(11), 119–144.

Sambell, K. (2016). Assessment and feedback in higher education: Considerable room for improvement? *Student Engagement Int. Journal of Higher Education (Columbus, Ohio)*, 1(1), 2016.

- Seman, L. O., Hausmann, R., & Bezerra, E. A. (2018). On the students' perceptions of the knowledge formation when submitted to a project-based learning environment using web applications. *Computers & Education*, 117, 16–30. <https://doi.org/10.1016/j.compedu.2017.10.001>
- Smith, C. D., Worsfold, K., Davies, L., Fisher, R., & McPhail, R. (2013). Assessment literacy and student learning: The case for deliberately Developing students assessment literacy. *Assessment and Evaluation Int. Journal of Higher Education (Columbus, Ohio)*, 38(1), 44–60.
- Tadjer, H., Lafifi, Y., & Seridi-Bouchelaghem, H. (2018). A new approach for assessing Learners in an Online problem-based learning environment. *International Journal of Information and Communication Technology Education*, 14(4), 18–33. <https://doi.org/10.4018/IJICTE.2018100102>
- Tadjer, H., Lafifi, Y., Seridi-Bouchelaghem, H., & Gülseçen, S. (2020). Improving soft skills based on students' traces in problem-based learning environments. *Interactive Learning Environments*. Advance online publication. doi:10.1080/10494820.2020.1753215
- Tchounikine, P. (2009). *Précis de recherche en ingénierie des eiah*. Available <http://membresliglab.imag.fr/tchounikine/Precis.html>
- Wigfield, A., Klauda, S. L., & Cambria, J. (2011). *Influences on the development of academic self-regulatory processes*. Academic Press.
- Zghal, S. (2010). *Contributions à l'alignement d'ontologies OWL par agrégation de similarités* [Ph.D.dissertation]. Tunis Univ, Tunisia.
- Zimmerman, B. J. (2004). Sociocultural influence and students' development of academic self-regulation: A social-cognitive perspective. In D. M. McInerney & S. Van Etten (Eds.), *Big theories revisited*. Greenwich, CT: Academic Press.
- Zimmerman, B. J. (2008). Investigating self-regulation and motivation: Historical background, methodological developments, and future prospects. *American Educational Research Journal*, 45(1), 166–183.
- Zimmerman, B. J., & Schunk, D. H. (Eds.). (2011). *Handbook of self-regulation of learning and performance*. Routledge.

Karima Boussaha is an Associate Professor in the Department of Mathematics and Computer Science at Oum El Bouaghi University in Algeria. She is obtained her Ph.D. at Annaba University in Algeria since 2016. She prepares her HU (Habilitation Universitaire) since 2016. She is a Member of the Distributed-Intelligent Systems Engineering (DISE) in the Research Laboratory on Computer Science's Complex Systems (ReLa(CS)2) at Oum El Bouaghi University. Her PhD research is based on computer-assisted assessment of learners, application in the programming language field. Her main areas of interest include computer-based/assisted assessment, assessment of procedural knowledge, and assessment in collaborative learning, computer-based/assisted collaborative tutoring computer-based/assisted collaborative learning, Ontologies, collaboration.

Farid Mokhati is a Full Professor at the Department of Mathematics and Computer Science at Oum El Bouaghi University in Algeria. He is the Head of the Research Team Distributed-Intelligent Systems Engineering (DISE) in the Research Laboratory on Computer Science's Complex Systems (ReLa(CS)2) at Oum el Bouaghi University. He holds an HU (Habilitation Universitaire) in Computer Science (distributed artificial intelligence) in 2010 from Annaba University. His main areas of interest include agent-oriented software engineering, embedded systems quality control and assurance and formal methods.

Amira Hanneche is a Software Development Engineer. She completed her Masters in 2017 from Oum el Bouaghi University. Her main areas of interest include CEHL (Computing Environment for Human Learning), learner's assessment, practical works activities, computer-based/assisted collaborative learning, tutoring, computer-based/assisted collaborative tutoring.