A new multi-agent approach for generating feedbacks base on Multiple Choice Questions

Imane Lmati
IR2M Laboratory, Faculty of Sciences and Techniques, Hassan First University of Settat Morocco
lmati2010@gmail.com

Abstract

Most intelligent tutorial systems promote the learning and resolution of exercises based on feedback in the form of advice, remarks, explanation…To always remain in self-assessment, our approach offers a new type of feedback in the form of multiple-choice questions applied in the field of algorithms (language c, java ...) dedicated for beginners in programming.
The approach is based on the multi-agent model to have interaction between learners without the help of the teacher. There are three types of agents in our work, there is the learner agent who represents the learner, there is also the feedback agent who sends the MCQs to the learners who made mistakes and finally, the controller agent feeds the base of multiple-choice questions based on feedback from learners.
The controlling agent compares the instructions of the learners with the instructions of the correct model based on the AST abstract syntax tree and detects errors which will be proposed as erroneous items (distractors) for the learners. We can say that this type of feedback is not direct like other work (advice, comments, explanation ...) but we always remain in the evaluation: MCQ, exercise ...
As perspectives, we will focus mainly on the classification of LO learning objects in the form of an ontology to facilitate the use of data and the generation of multiple-choice questions at different levels. And also, we aim to develop a suitable platform which allows to define the agents and the messages to be exchanged between them to set up our system.

Keywords: Evaluation; Feedback; MCQ; Exercise; Agent; Multi Agent Systems.

1. State of art

Feedback is considered one of the pillars of formative assessment (EL HAGE et al., 2020), It is also one of the most important factors influencing learning in different contexts (Hattie & Gan, 2011). Feedback helps the student to regulate his learning process (NARCISS, 2008). Several studies aim to identify the different types of feedback formulated by the teacher, whether given synchronously or asynchronously (SHUTE, 2008), whether retrospective or prospective (HATTIE & TIMPERLEY, 2007), whether they are given orally in a more or less formal manner or in writing in the form of various annotations. In this work, two types of feedback seem useful to us: that of the teacher and that of other students, the "peers".
In the literature, there are many software solutions that gradually support learners during the resolution of the exercises or problems and that either by the teacher or by the peers. These intelligent systems can:

- Telling a student whether or not a solution to an exercise is correct,
- Explain the most likely mistake when the answer is wrong,
- Use this student model to give specialized feedback to a student,
- Generate an infinite amount of exercises,

For example, we find the project of El Hage (EL HAGE et al., 2020) which offers a Digital and Interactive Support Platform. It is a dedicated platform for formative assessment. It is based on the notion of Learning by Exploitation of Error and at a distance. It aims to support students through feedback, constructive, deferred and personalized. Example: Closed-response exercises (interactive MCQ with constructive feedback: each chosen false answer is justified, allowing students to analyze and understand their mistakes).

ActiveMath (MELIS et al., 2004) is also a helpful tool for solving arithmetic problems. It offers feedback in the form of remarks about the equivalence of expressions given by the learner (Deduce if the expression is equivalent to the expert's goal). These feedbacks are generated without guiding the learner during the solution of the exercise.

Heeren (HEEREN et al., 2008) has developed a language for specifying strategies for solving exercises. This language helps to improve feedback in e-learning systems or intelligent tutorial systems. The language makes it easier to automatically deduce feedback when users make wrong measurements in a calculation.

These strategies make it possible to automatically calculate all types of desirable feedback (Syntax error, Rewrite error, Error in the strategy to solve the exercise ...).

The "Note as feedback" approach (SILVESTRE et al., 2014; SILVESTRE et al., 2015) allows the teacher to automatically generate a Moodle-compatible export file recycling the interactive questions asked by the teacher during the course,

and the notes taken by the students on the different questions. The notes taken by the students on a given question constitute the feedback presented to the student during his revision test.

Most intelligent ITS tutorial systems promote learning and problem-solving or exercise-based feedback in the form of advice, remarks, explanations… So, to always remain in the self-assessment and to facilitate the resolution of the exercises, we offer feedback in the form of another type of assessment, which is multiple choice questions. In addition, several studies have shown that collaborative or peer feedback improves learner engagement in the activity and gives good results (GAVOTA et al., 2010; ORTOLEVA et al., 2012). This is why we thought of exploiting multi-agent systems in our work for the generation of collaborative feedback without the help of the teacher. Subsequently, we propose the Multi agent SMA model which we used for the generation of multiple-choice questions.

2. The multi agent model for the generation of multiple-choice questions

Several studies have shown that collaborative or peer feedback improves learner engagement in the activity and gives good results (GAVOTA et al., 2010; ORTOLEVA et al., 2012). It is for this reason that we thought of exploiting multi-agent systems in our work for the generation of collaborative feedbacks without the help to the teacher. We will identify a set of agents for the development of our approach. These agents work together in order to do an automatic analysis of the learners' feedback, and by returning multiple choice questions and this in order to fill their gaps which it encounters at the time of the self-assessment.

The multi-agent system consists of three agents; there is the learning agent, the controlling agent and the feedback agent.

Learning agent: represents the feedback of learners. Each time a new session is opened for a new learner, a learning agent is assigned to him so that he communicates with other agents through ACL (Agent Communication Language) messages.

Controlling agent CA: compare each line (instruction) of learners' coding with the instructions of the correct model of the algorithm. In order to detect the line corresponding to the correct model, the agent controller calculates the similarity rate between the AST abstract syntax tree of each instruction of the learner and that of the correct model. If the rate is average: the agent calculates the steps (declaration, initialization, if, loop ...) before and after the requested instruction. If they are equal in this case, the controlling agent sends the wrong instruction to the feedback agent so that it transforms it into a distractor item of MCQ. If not, the instructions are not the same, the agent must look for another suitable instruction. If the rate is high, the controlling agent returns the correct instruction to the feedback agent for the creation of the correct items.

feedback agent FA: After the creation of all correct and erroneous items and their storage by the controlling agent, the feedback agent can create MCQ and sends them to the learner who has errors in the form of feedback and this is based on the work of all connected learners and also old work. If the learner fails to detect the correct item in relation to level i of the LO learning object model (See figure 4), in this case, the agent proposes multiple choice questions for level i-1 le of the model and so on. We can say that the feedback agent allows items to be classified by level.

I give in figure 1 a diagram concerning the communication of the agents for the generation of multiple choice questions in the form of feedback.

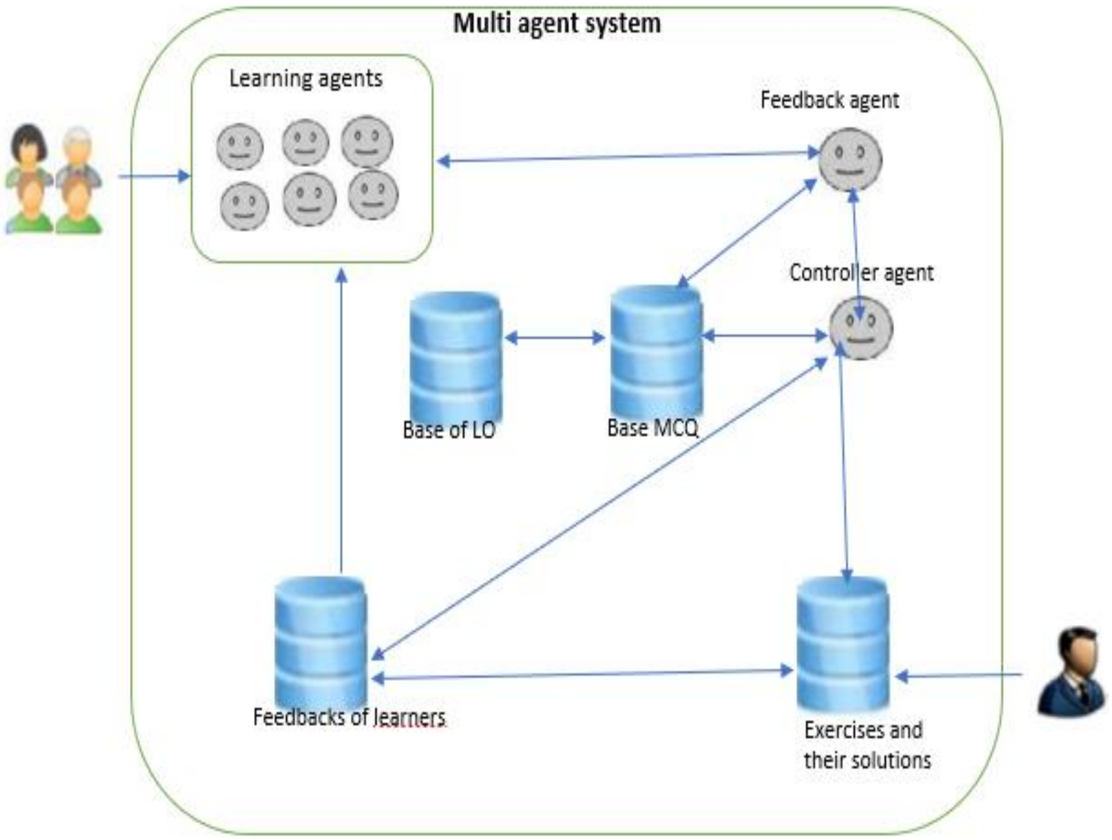

Figure 1. Multi-agent system model for generating multiple choice questions in the form of feedback.

Subsequently, I will detail the role of each agent in the process of creating feedback in the form of multiple choice questions, composed of three parts:

1- Creation of MCQ items based on multi-agent systems
2- Classification of items by level (LO learning object model) by feedback agent.
3- Generation of MCQs by feedback agent.

2.1 Creation of MCQ items based on multi-agent systems

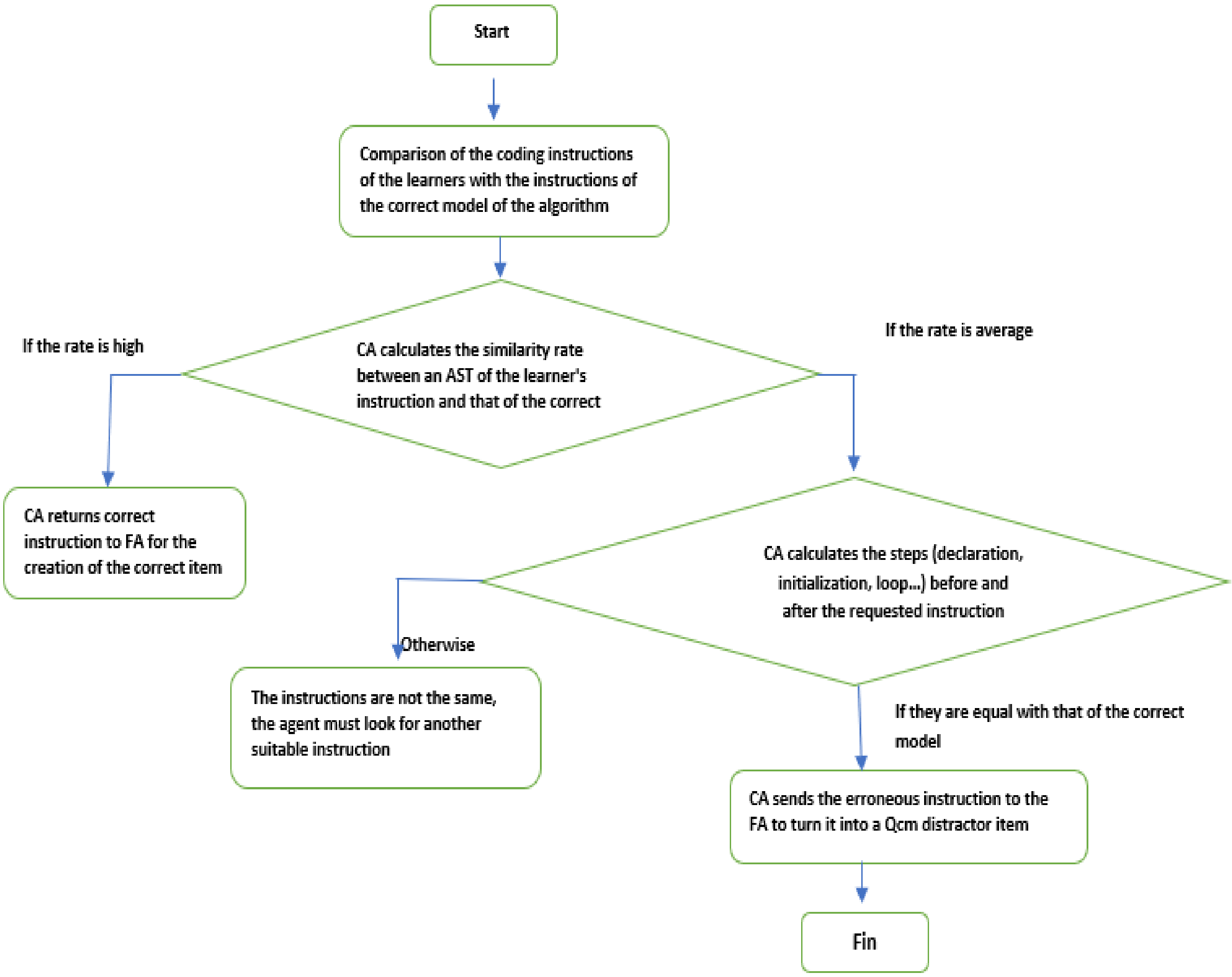

Figure2. Process of creating multiple choice question items based on multi-agent systems

After authenticating the learners, learning agents are created to monitor each learner during the self-assessment.

To complete the course learning, the teacher can ask an online algorithmic exercise for his students. When students finish their works the learning agent send it to the controlling agent. The latter compares each line (instruction) of coding learners with the teacher's correct algorithm instructions.

In order to detect the line corresponding to the correct model, the CA controller agent calculates the similarity rate between the AST of the learner's instruction and that of the correct model. If the similarity rate is average, the agent calculates the steps (declaration, initialization, if, loop ...) before and after the instruction of learners and compares with that of the correct model.

If they are equal, in this case the controlling agent sends the erroneous instruction to the FA feedback agent so that it transforms it into a MCQ distractor item. Otherwise, the instructions are not the same (case of two loops in the same algorithm), in this case the agent should look for another relevant instruction.

If the rate is high, the AC controller agent returns the correct instruction to AF for item creation.

2.2 Comparison of AST for algorithmic instructions

If we take the following program, we can represent its AST syntax tree by line (see figure 3).

Example:

```
int compute_sum_integer(int start, int end)
{
int result = 0;
int i = start;
while(i <= end)
{
result = result + i;
i++;
}
return result;
}
```

Its AST presentation is based on program lines, such that each line represents a branch in AST.

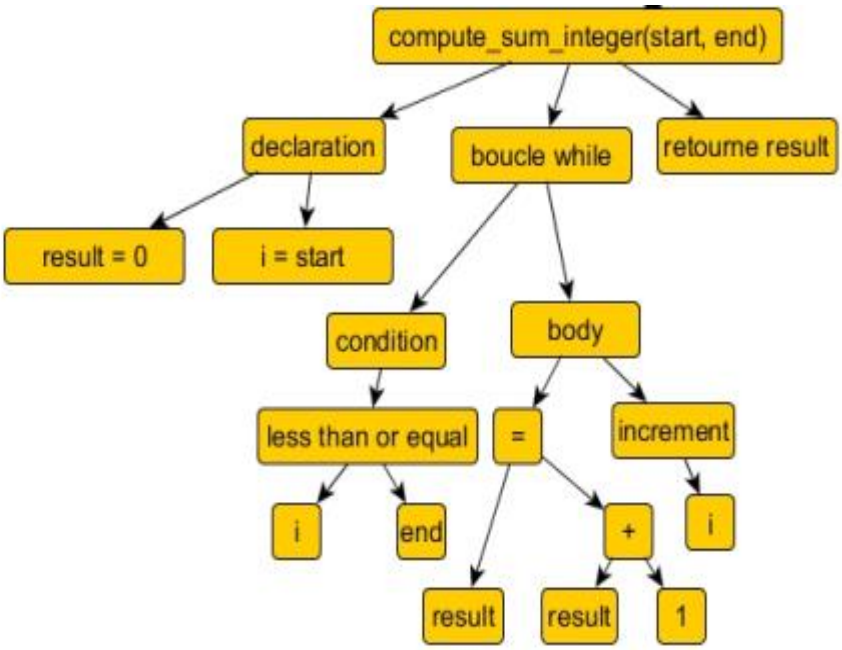

Figure 3. AST presentation of the algorithm

2.3 Classification of items by level of education

Based on the work of Sabeima (SABEIMA ,2021), we managed to classify the different algorithm instructions by level based on the LO learning objects.

For example, if the learner does not manage to answer the MCQ of the “for” loop level (knowing that it made errors in the syntax of the loop for) he is offered multiple choice questions for the descending level of instruction.

Example of instruction in c language: for (i=0 ; i<10 ; i++)

In the example of the loop “for”, there are three other instructions: assignment (i = 0), comparison (i <10) and increment (i ++), from this observation, you can divide the LO learning objects into several levels (figure 4).

Level (i) : Loop for
Level(i- 1)(1) : assignment
Level (i- 1)(2) : comparison
Level (i- 1)(3) : increment

In this case, if the learner cannot answer the MCQ from the "for" loop level (knowing that he made errors in the syntax of the loop for) the system offers the learner multiple choice questions for the decreasing level of LO model: assignment, comparison or increment.

Table 1. Presentation of the prerequisites for learning objects

Learning Object	**Title**	**Prerequisites**
LO1	Declaration of variables	
LO2	Operators: comparison	LO1
LO3	Operators: assignment	LO1
LO4	Operators: Calculation (+, -…)	LO1
LO5	Operators: increment/ decrement	LO3, LO4
LO6	Inputs	LO1, LO2, LO3, LO4
LO7	Output	LO1, LO2, LO3, LO4
LO8	Loop for	LO2, LO3, LO5

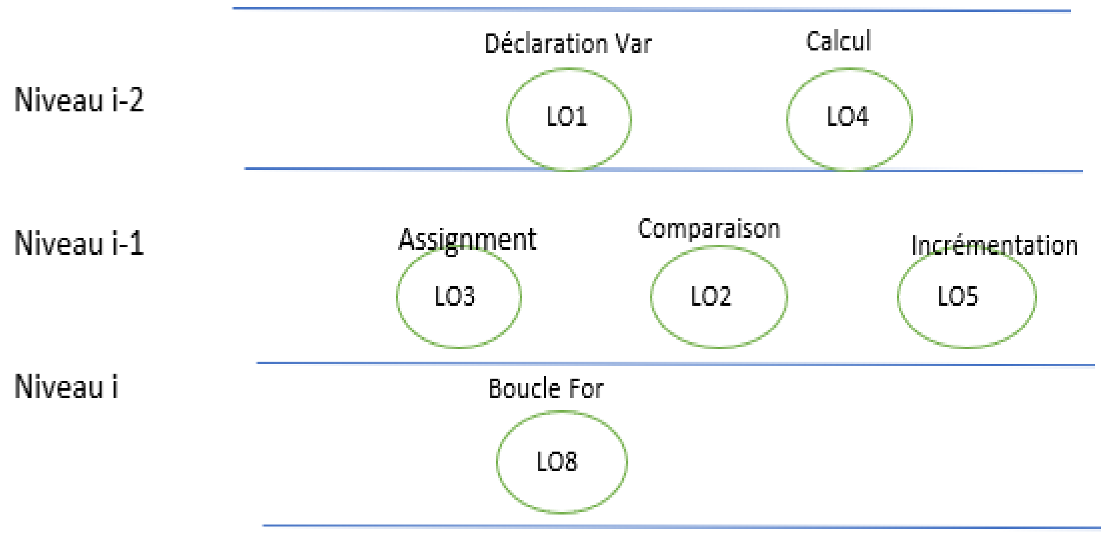

Figure 4. Classification of learning objects by level

According to figure 4, each LO has a prerequisite, which makes it possible to create levels, for example if the learner has not been able to answer a MCQ concerning the loop “for”, the system sends him another MCQ, but this time, it offers him MCQ of the descending level: assignment, comparison or even incrementation.

2.4 MCQ generation process by the feedback agent

After the creation of all correct and incorrect items and their storage by the controlling agent,
the feedback agent can create multiple choice questions at the end and send them back to learners who have errors in the form of feedback and this based on feedback from all connected learners and also old feedback. If the learner cannot detect the correct MCQ item for a given instruction (Ex: declaration of the loop), in this case, the agent proposes multiple choice questions of the descending level of the error from the base of learning objects LO (descending level of declaration of the loop can be an assignment i = 0 or comparison i <x or increment i).

In the following figure, we suggest the process of generating multiple choice questions by the Feedback agent.

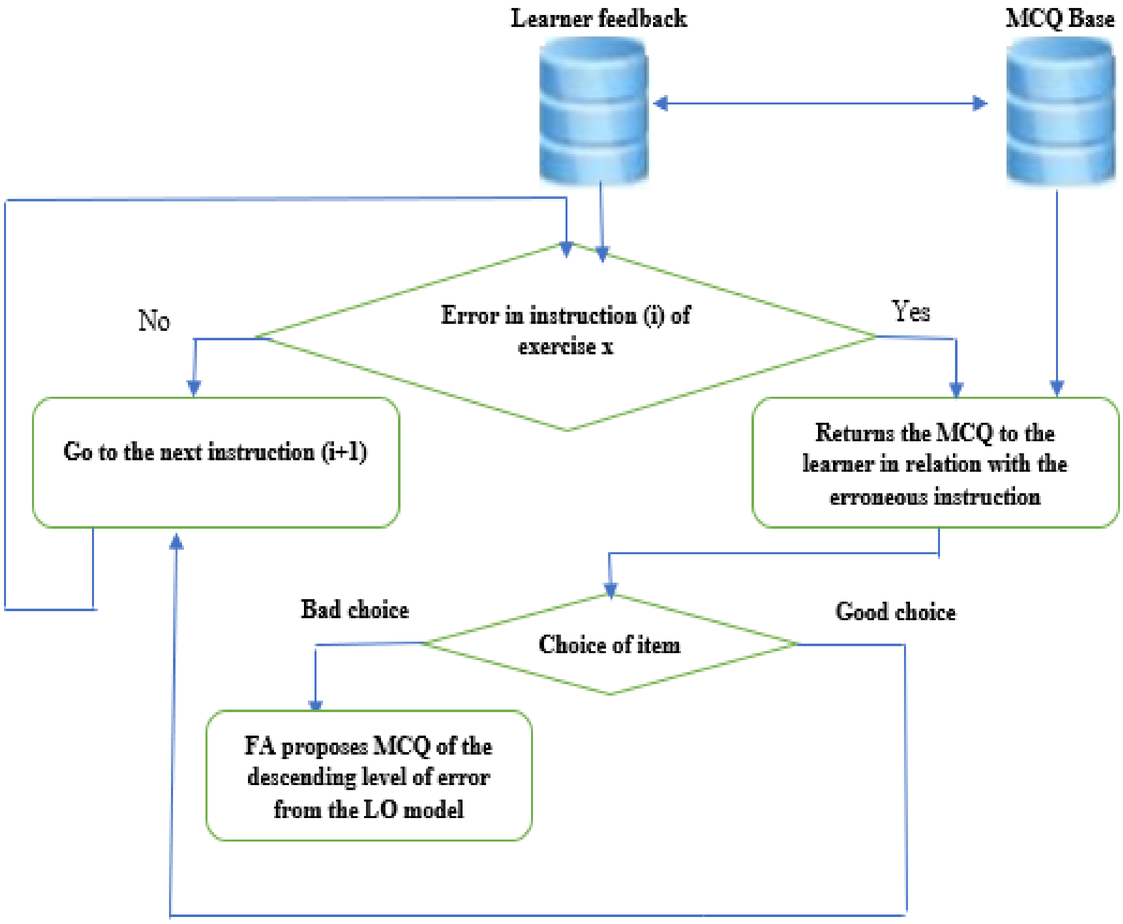

Figure 5. MCQ generation process by the Feedback agent

3. Example of application

In this section, we give some examples of MCQ items according to levels extracted from students' answers (correct and incorrect) in C programming.

.

Consider the following exercise offered to students:

Exercise: Give the algorithm that calculates the sum of the numbers between the values: start and end.

Example:

```
int compute_sum_integer(int start, int end)
{
int i ;
int result = 0;
for ( i = start; i <= end ; i++)
{
result = result + i;
}
return result;
```

}

I show you in the following table the different correct and incorrect items based on the answers of the students.

Table 2. Presentation of correct and incorrect items for the generation of multiple-choice questions

Line	Error type	Wrong answer	Correct answer
1	Declaration	- i ; - int i - i ;	- int i ;
2	Assignment	- 0 = result; - result != 0; - int result == 0 - ...	- int result = 0;
3	Loop	- for (i ++) - for (i <= end) - for ; - ...	- for (i = start; i <= end ; i++)
	descending level: Assignment Comparison Increment	- start= i ; - ... - i <> end ; - ... - i+1=i; - ...	- i = start ; - i <= end ; - i++ ;
...	...	...	...

From the table above, we can generate several types of multiple-choice questions per level. For example, if the student cannot answer the MCQ concerning line 3 (for loop), he is offered descending level MCQ: assignment, comparison or even increment.

4. Conclusion

As part of a formative evaluation, feedbacks constituting a powerful tool to help students overcome certain learning obstacles. This support is also relevant when it is done remotely, because it has several advantages over traditional remediation. Among these advantages, the possibility of having collaborative interactions without resorting to an external resource. Our approach converges in this direction, it is based on the multi-agent model to have an interaction between the learners, it offers a new type of feedback in the form of multiple-choice questions. The Approach can be used in platforms for learning algorithms as it can use for other languages (c, j ava…).

As perspectives, we will mainly focus on the classification of LO learning objects in the form of ontology to facilitate the use of data and the generation of multiple-choice questions at different levels. And also, we aim to develop a suitable platform which makes it possible to define the agents and the messages to be exchanged between them to set up our system.

5. References

El Hage, F., & Nahed, R., Apprentissage par Exploitation de l'Erreur et à Distance (AEED). Pour une évaluation formative et un feedback interactif et digital. Évaluer. Journal international de recherche en éducation et formation, Numéro Hors-série, 1, 2020, pp. 131-142.

Hattie, J.A., & Gan, M., Instruction based on Feedback. In Mayer, R & Alexander, P. (Eds). Handbook of Research on Learning and Instruction, New York: Routledge, 2011. pp. 249-271.

Narciss, S., Feedback strategies for interactive learning tasks. In J.M. Spector, M.D. Merrill, J.J.G. van Merrienboer, & M.P. Driscoll (Eds.), Handbook of Research on Educational Communications, and Technology (3rd). Mahaw, NJ: Lawrence Erlbaum Associates. 2008, pp. 125-144.

Shute, V.J. Focus on formative feedback. Review of Educational Research, 78, 2008, 153-189.

Hattie, J., & Timperley, H. The power of feedback. Review of Educational Research, 77 (1), 2007, 81-112.

Melis, E., &Siekmann, J. : ActiveMath: Un système de tutorat intelligent pour les mathématiques. En Intelligence Artificielle et Soft Computing-ICAISC. Springer Berlin Heidelberg, 2004.

Heeren, B., Jeuring, J., Van Leeuwen, A., &Gerdes, A. : Specifying strategies for exercises. In Intelligent Computer Mathematics. Springer Berlin Heidelberg, 2008.

Gavota, M., Bétrancourt, M. & Schneider, D. (2010). Writing and commenting on professional procedures. In K. Gomez, L. Lyons, & J. Radinsky (Eds.) Learning in the Disciplines: Proceedings of the 9th International Conference of the Learning Sciences (ICLS 2010) - Volume 1, Full Papers, pp 683-689. International Society of the Learning Sciences: Chicago IL.

Ortoleva, G., Bétrancourt, M., et Morand, S., Entre personnalisation et contraintes collectives: Une démarche centrée utilisateur pour la mise en place d'un livret numérique de Suivi Pédagogique. Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation, 2012, vol. 19, no 1, p. 233-251.

F. Silvestre, P. Vidal, J. Broisin. Qualification semi-automatique de contributions d'apprenants pour l'intégration de feedbacks de qualité dans les tests en ligne. 7ème Conférence sur les Environnements Informatiques pour l'Apprentissage Humain (EIAH 2015), Agadir, Maroc, Jun 2015, pp.210-221.

Silvestre, F., Vidal, P., and Broisin, J.: Génération semi-automatique de tests d'autoévaluation pourvus de feedback résultant de la prise de notes collaborative, conférence des Technologies de l'Information et de la Communication pour l'Enseignement, 2014.

Sabeima, Massra, LAMOLLE, Myriam, et NANNE, Mohamedade. Tour d'horizon des systèmes d'apprentissage adaptatifs et collaboratifs. 2021.