IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 13, 2018, accepted June 6, 2018, date of publication June 11, 2018, date of current version June 29, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2846398

OntoPeFeGe: Ontology-Based Personalized Feedback Generator

MONA NABIL DEMAIDI[1], MOHAMED MEDHAT GABER[1], AND NICK FILER[2]
[1]School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7BD, U.K.
[2]School of Computer Science, The University of Manchester, Manchester M13 9PL, U.K.

Corresponding author: Mona Nabil Demaidi (mona.demaidi@gmail.com)

ABSTRACT Virtual Learning Environments provide teachers with a web-based platform to create different types of feedback. These environments usually follow the 'one size fits all' approach and provide students with the same feedback. Several personalized feedback frameworks have been proposed which adapt the different types of feedback based on the student characteristics and/or the assessment question characteristics. The frameworks are intradisciplinary, neglect the characteristics of the assessment question, and either hard-code or auto-generate the types of feedback from a restricted set of solutions created by a domain expert. This paper contributes to research carried out on personalized feedback frameworks by proposing a generic novel system which is called the Ontology-based Personalized Feedback Generator (OntoPeFeGe). OntoPeFeGe addressed the aforementioned drawbacks using an ontology—a knowledge representation of the educational domain. It integrated several generation strategies and templates to traverse the ontology and auto-generate the questions and feedback. The questions have different characteristics, in particular, aiming to assess students at different levels in Bloom's taxonomy. Each question is associated with different types of feedback that range from verifying student's answers to giving the student more details related to the answer. The feedback auto-generated in OntoPeFeGe is personalized using a rule-based algorithm which takes into account the student characteristics and the assessment question characteristics. The personalized feedback in OntoPeFeGe was quantitatively evaluated on 88 undergraduate students. The results revealed that the personalized feedback significantly improved the performance of students with low background knowledge. In addition, the feedback was evaluated qualitatively using questionnaires provided to teachers and students. The results showed that teachers and students were satisfied with the feedback.

INDEX TERMS Ontology, formative feedback, Bloom's taxonomy, question generation, feedback generation, personalized.

I. INTRODUCTION

Personalized learning environments are Virtual Learning Environments (VLEs) which tailor the learning content and generate feedback to meet student's knowledge and goals [1]–[3]. The research presented in this paper focuses on one important aspect in personalized learning environments, which is providing students with formative feedback while they are working on assessment questions [4]. Formative feedback is the feedback provided to students after answering an assessment question and it is a key element in formative assessment systems [5]. It provides students with the information required to close the gap between their current performance and the desired performance [6]. In addition to the importance of formative feedback content, Price *et al.* [7] specified that feedback can only be effective when the learner understands the feedback and is willing and able to act on it. Formative feedback can be delivered to students immediately or after some delay [8]. This research focuses on the immediate formative feedback which students receive after answering an assessment question.

Formative feedback provided to students in learning environments can be classified into several types, where each type provides students with different pedagogical content at a different level of detail [1], [8]–[10]. Fig. 1 illustrates an example of two types of formative feedback and the pedagogical content associated with each type. Verification feedback which is also called Knowledge Of Results (KOR) feedback. It is the simplest type of feedback as it only verifies whether a student's answer is right or wrong. Response Contingent (RC) feedback is the type of feedback which provides students

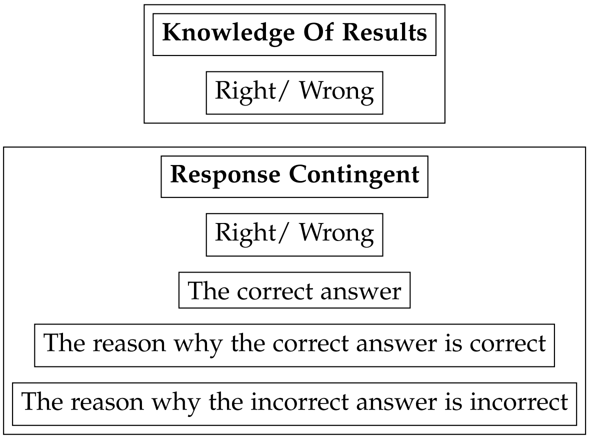

FIGURE 1. Types of formative feedback and their pedagogical content.

with the information required to understand an educational concept. It verifies the student's answer, provides him or her with the correct answer, and explains to the student the reason why the correct answer is correct and vice-versa.

Providing students with personalized feedback has been identified as a powerful method that helps them understand the gaps in their knowledge, monitor their progress and improve their overall performance [11]. Personalized feedback is defined as adapting the type of feedback provided to a student based on the student's characteristics (e.g., the background knowledge and the current level of knowledge) and/or the question's characteristics (e.g., the level of the question in Bloom's taxonomy) [1], [12]–[14].

Several frameworks had been developed to provide students with personalized feedback by either hard-coding the feedback in the system or auto-generating the feedback from a restricted set of solutions created by the teacher or a domain expert [1], [12], [15]–[20]. This has two main disadvantages: it is a time consuming process, and the frameworks are intradiscipline and cannot be used to auto-generate feedback across different educational domains(e.g., computer science and medicine) [21]–[23].

The research presented in this paper aims to address the drawbacks mentioned above by proposing an interdisciplinary framework which auto-generate personalized feedback across different educational domains using a broad knowledge base. The framework in called an Ontology-based Personalized Feedback Generator (OntoPeFeGe). OntoPeFeGe generates personalized feedback using the ontology which is a conceptualization of the domain knowledge in terms of concepts and properties and it captures the concepts in an educational course [24]. Ontology has been used in the past by several feedback generators to generate different types of feedback [21], [25]–[30]. However, the feedback generators are interdisciplinary as in addition to the ontology the generators use an expert knowledge base which captures the experts' solutions to the problem scenarios or human intervention (e.g., domain experts and teachers). The generators also follow the 'one size fits all approach' and ignores the student and the assessment question characteristics. Therefore, OntoPeFeGe was designed to only use the ontology in the auto-generation process and provide students with personalized feedback which takes into account the student and the assessment question characteristics.

This paper contributes to the research carried out in personalized feedback frameworks, the ontology-based question generators, and the ontology-based formative feedback generators by achieving the following:

Contribution 1: Formative feedback generator. The generator is interdisciplinary and generates different types of feedback using pre-existing domain ontology. No expert knowledge base which captures the experts' solutions to the problem scenario or human intervention (teacher or domain expert) is needed. The generator also associates the different types of feedback to questions auto-generated from the ontology.

Contribution 2: Personalized feedback algorithm. A formative feedback algorithm had been implemented in Moodle VLE to provide students with the appropriate type of feedback immediately after answering an assessment question. The algorithm adopts Mason and Bruning's personalized feedback framework [9]. The algorithm adapts the type of feedback provided to students based on student's characteristics: background knowledge about a specific educational topic, current level of knowledge while answering one question after another, and the question's characteristics which is the level of the question in Bloom's taxonomy. This allowed the relationship between student's characteristics, question's characteristics, and the personalized feedback to be studied for the first time.

Contribution 3: Analyze the effect of personalized feedback on students' performance. This paper presents the experiment carried out in Moodle VLE to evaluate the personalized feedback. Both the personalized feedback algorithm and the auto-generated feedback were evaluated.

The paper is organized as follows: section II presents related work, section III illustrates the OntoPeFeGe framework and explains it in details, section IV illustrates the OntoPeFeGe framework evaluation, and section V concludes the paper with a discussion of the main results and directions for future research.

II. RELATED WORK

This section reviews the existing personalized feedback frameworks and generators, and highlights the need for a new personalized feedback generator.

A. PERSONALISED FEEDBACK FRAMEWORKS

Researchers such as Gouli *et al.* [31], Mason and Bruning [9] proposed guidelines to develop personalized feedback frameworks. Gouli *et al.* framework adapted the type of feedback based on students' current level of knowledge. While Mason and Bruning's framework considered students' background knowledge and current level of knowledge as well as the question's difficulty. Both frameworks are theoretical and have never been evaluated on students. Other researchers such as Narciss *et al.* [1], Arroyo *et al.* [16]–[18], and Woolf *et al.* [19] focused on providing students with personalized feedback based on the student's current level of knowledge. Their frameworks were evaluated on students and the results revealed that the personalized feedback improved students' performance. However, their evaluations had contradictory results regarding the impact of personalized feedback on the performance of male and female students. Narciss *et al.* [1] showed that female students had higher performance than male students. Arroyo *et al.* [16], [17], had similar results in one study, however, in another study they carried out no difference in performance was found between male and female students [18], [19]. The personalized feedback evaluation studies mentioned above suggest that there is still no clear understanding regarding the relationship between the student's characteristics, the question's characteristics and the personalized feedback [1], [8], [9], [18]. Moreover, none of the personalized feedback frameworks which were evaluated on students considered the question's characteristics in the feedback adaptation process. This issue has been addressed by Narciss *et al.* [1] who suggested considering the question's difficulty while providing students with personalized feedback. Therefore, this paper aims to evaluate Mason and Bruning's [9] personalized feedback framework which adapts the different types of feedback based on the student and the question's characteristics.

B. ONTOLOGY-BASED FEEDBACK GENERATORS

The personalized feedback frameworks explained in Section II-A provide students with different types of feedback by either hard-coding the feedback in the system [12], [15]–[19] or auto-generating the feedback from a restricted set of solutions created by the teacher or a domain expert [1], [20]. This has two main disadvantages: it is a time consuming process, and the frameworks are intradiscipline and cannot be used to auto-generate feedback across different educational domains (e.g., computer science and medicine) [21]–[23].

To address the time consumption drawback, researchers used a broad knowledge base called ontology to auto-generate different types of feedback. Table. 1 presents the surveyed ontology-based feedback generators. Kazi *et al.* [21], [25], and [26] generated hint feedback which provides the student with information on what to do next to guide him or her towards the right solution. Frutos-Morales *et al.* [28], del Mar Sánchez-Vera *et al.* [29], and Castellanos-Nieves *et al.* [30] generated Knowledge Of Results (KOR) and Knowledge of Correct Response (KCR) feedback which verifies a student's answer and also provides him or her with the correct answer. Duboc *et al.* [27], [32] generated three types of feedback: KCR, Bugs-Related (BR)

TABLE 1. Ontology-based feedback generators.

Feedback generator	Types of feedback	Domain dependency
Kazi et al.	Hint	Dependent (medicine)
Sanchez-Vera et al.	KOR KCR	Dependent (Design and Production of Educational Materials)
Duboc et al.	KCR BR TC	Dependent (medicine)

feedback which verifies the student's answer and provides him or her with the reason why an incorrect answer is incorrect without giving the student the correct answer, and Topic Contingent (TC) feedback which verifies the student's answer, provides him or her with the correct answer, and explains to the student the reason why the correct answer is correct.

The ontology-based feedback generators addressed the time consumption drawback. However, they have the following drawbacks: 1) The auto-generated feedback is domain dependent. This means that in addition to the ontology, the generators either use an expert knowledge base which captures the experts' solutions to the problem scenario or human intervention (e.g., domain experts and teachers) to auto-generate the different types of feedback. 2) The auto-generated feedback is not personalized to meet the student or the question characteristics.

Providing students with personalized feedback after auto-generating different types of feedback requires information about the assessment question characteristics. The feedback generators illustrated in Table. 1 hard-coded the assessment questions, which means that the questions are only valid in the educational domain they are created in. In addition, the feedback generators did not specify the question characteristics [21], [25]–[30], [32]. Both drawbacks hinder providing students with personalized feedback in a generic framework. To address this issue, this research investigated several domain independent question generators [33]–[39], which use an ontology to auto-generate several types of questions (true or false, multiple choice, and short answer questions) with different characteristics. In particular, questions aimed to assess students' cognition at different levels in Bloom's taxonomy (knowledge, comprehension, application, and analysis) [40], [41]. To generate questions designed to assess students at different levels in Bloom's taxonomy, the ontology-based question generators use several stem templates, which are the text stating the question. In addition, the generators use several strategies to traverse the domain ontology and generate the question. For example, a strategy is used to generate the question's key which is the correct answer and the question's distractors which are the incorrect answers in a multiple choice question.

To address the drawbacks mentioned above. This paper presents an Ontology-based Personalized Feedback Generator (OntoPeFeGe). OntoPeFeGe integrates the stem templates and strategies to auto-generate various types of

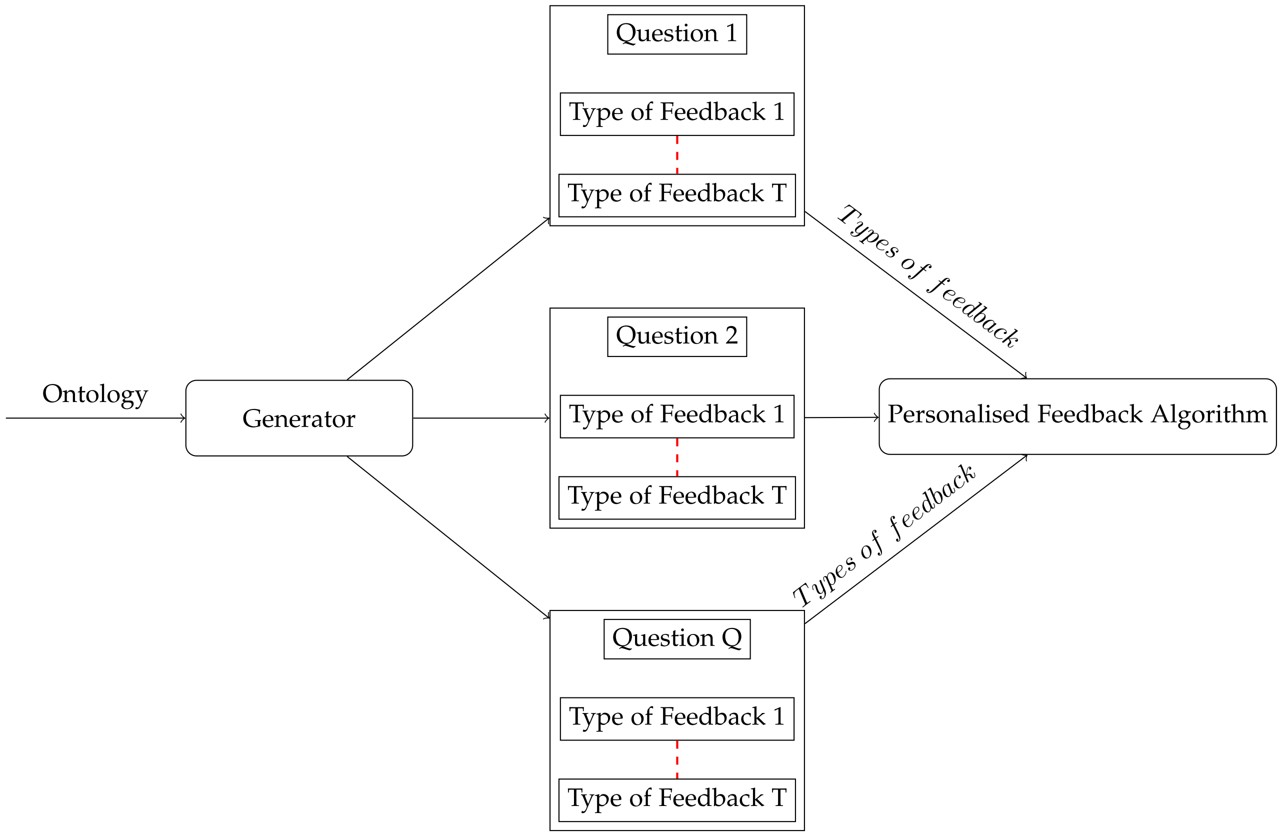

FIGURE 2. Ontology-based Personalized Feedback Generator framework.

assessment questions and associates each question with different types of feedback auto-generated from ontology. Moreover, OntoPeFeGe provides students with personalized feedback immediately after answering an assessment question by adopting Mason and Bruning's personalized feedback framework.

III. OntoPeFeGe FRAMEWORK

This section introduces the OntoPeFeGe framework which is shown in Fig.2. The framework consists of two main components: 1) The generator which auto-generates questions and associates each question with different types of formative feedback. 2) The personalized feedback algorithm which provide students with the appropriate type of feedback. The components are explained in detail in the following sections.

A. GENERATOR

The generator takes the domain ontology which captures the concepts in an educational course as an input and outputs Q questions associated with T types of feedback. This is shown in Fig. 2. The generator generates different types of questions (true and false, multiple choice, and short-answer) and different types of feedback using the ontology-based generation strategies defined by Papasalouros *et al.* [33], [34], Grubišić *et al.* [36], [37], Al-Yahya [38], [39], Cubric and Tosic [35].

1) QUESTION GENERATION

The ontology-based generation strategies traverse the domain ontology to instantiate a set of stem templates which are designed to assess student's cognition at different levels in Bloom's taxonomy. Table. 2 illustrates part of the stem templates for true and false questions (e.g., question 3 in Table. 2), multiple choice questions (e.g., question 4 in Table. 2), and short answer questions (e.g., question 8 in Table. 2). Grubisic's stem templates aimed to assess students' cognition at the following levels in Bloom's taxonomy: 1) Knowledge level: Questions at this level focus on assessing if the students are aware of the subclasses and superclasses properties between concepts in the domain ontology. 2) Comprehension level: Questions at this level focus on asking the students to identify the educational concept's subclasses and superclasses. 3) Application level: Questions at this level assume that the students are more familiar with the domain ontology being tested, as students are asked to list subclasses and superclasses in the domain ontology. 4) Analysis level: Questions at this level focus on assessing the concept's annotation properties and the concept's datatype and object properties with other concepts in the domain ontology.

Cubric and Tosic followed a different approach in forming the stem templates. They used words that define each level in Bloom's taxonomy such as demonstrate, define, relate, and analyses [42], [43]. See questions 1, 2, and 5 in Table. 2.

TABLE 2. Part of the stem templates integrated into OntoPeFeGe.

Question Number	Stem template	Bloom's level	Type of question	Generation strategy	Literature
1	Which of the following definitions describes the concept Class A?	Knowledge	Multiple choice	Property-based	Cubric and Tosic [35]
2	Read the paragraph and decide which one of the following concepts it defines?	Knowledge	Multiple choice	Property-based	Cubric and Tosic [35]
3	Are Class A and Class B directly connected?	Knowledge	True and false	Terminology-based	Grubisic [37]
4	What directly connects Class A and Class B?	Knowledge	Multiple choice	Property-based	Grubisic [37]
5	Which one of the following response pairs relates in the same way as: Class A Property Class B	Comprehension	Multiple choice	Property-based	Cubric and Tosic [35]
6	Are Class A and Class B indirectly connected?	Comprehension	True and false	Terminology-based	Grubisic [37]
7	Which one of the following demonstrates the concept Class A?	Application	Multiple choice	Class-based	Cubric and Tosic [35]
8	How many concepts is Class A connected with?	Application	Short answer	Property-based	Grubisic [37]
9	Analyse the following text and decide which one of the following words is a correct replacement for the blank space in the text?	Analysis	Multiple choice	Property-based	Cubric and Tosic [35]

The stem templates are instantiated during the generation process by the ontology-based generation strategies defined by Papasalouros *et al.* [33], [34], Grubišić *et al.* [36], [37], Al-Yahya [38], [39], and Cubric and Tosic [35]. The generation strategies could be categorized into the following: (a) the class-based strategies, which use the relationship between the ontology classes and individuals, (b) the terminology-based strategies, which use the relationship between the class and sub-class in ontologies, and (c) the property-based strategies, which use the object, datatype, and annotation properties in the ontologies.

The class-based strategies in the current OntoPeFeGe framework traverse the input domain ontology to auto-generate multiple choice questions which assess students' cognition at the application level in Bloom's taxonomy (see question 7 in Table. 2). The true and false and short answer stem templates defined by Grubišić *et al.* [36], [37], Cubric and Tosic [35] were not designed to use the class-based generation strategies. Instead, these questions were generated using the terminology-based and the property-based strategies. The class-based strategies exploit the property between the individuals and the class in the input domain ontology (e.g., Sakathi's Computer Networks ontology [44]) to generate the question's Key and Distractor individuals using the five class-based generation strategies shown in Fig. 3 [33], [34].

The terminology-based generation strategies used in OntoPeFeGe are Strategy 6 and Strategy 7 which are shown in Fig. 4. The strategies are used to generate true and false questions which assess students' cognition at the knowledge, comprehension, and application levels in Bloom's taxonomy. The terminology-based strategies exploit the *subClass* property which relates the subject resource to the object resource in the domain ontology as follows:

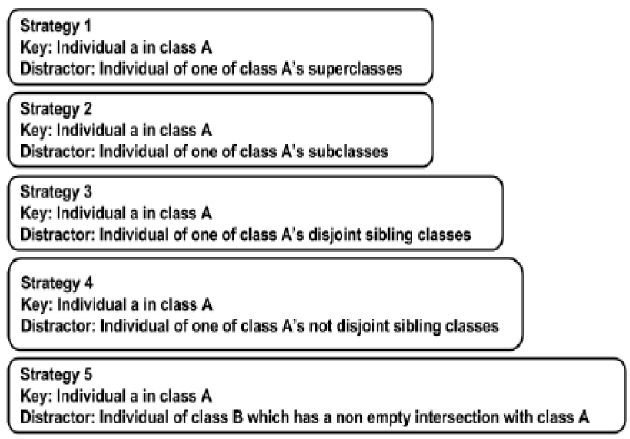

FIGURE 3. Class-based strategies.

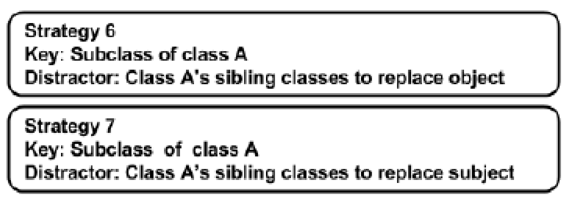

FIGURE 4. Terminology-based strategies.

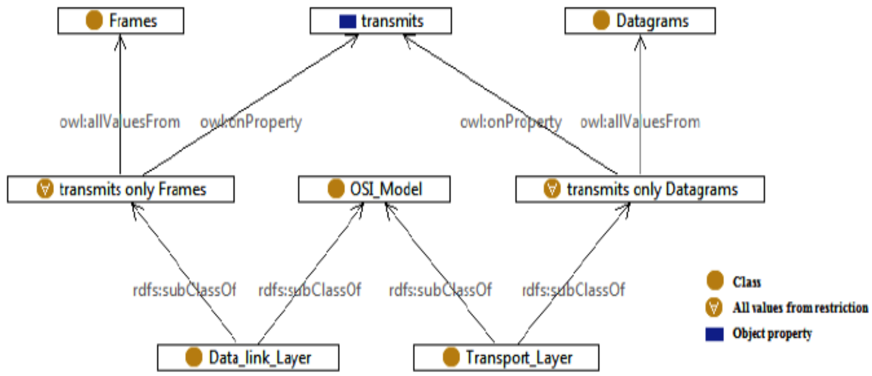

FIGURE 5. Transport Layer and Data link Layer concepts in Sakathi's ontology [44].

Subject subClass Object

The subject is a class in the domain ontology (e.g., *Transport Layer* class shown in Fig. 5) and the object could be either a class or a restriction (a restriction in OWL is a class defined by describing the individuals it contains [45]) such as the *transmits only frames* and the *transmits only datagrams* restriction classes shown in Fig. 5.

The property-based generation strategies are used in OntoPeFeGe to generate true and false, multiple choice, and short answer questions from the domain ontologies. The questions generated using the property-based strategies assess the students' cognition at the knowledge, comprehension, application and analysis levels in Bloom's taxonomy. The property-based strategies are categorized into:

1) Object-based strategies, which exploit the object properties in the domain ontology. Object properties are used to connect two resources together where the subject resource and the object resource are classes in the domain ontology.
2) Datatype-based strategies, which exploit the datatype properties in the domain ontology. Datatype properties are used to connect a resource to an RDFS:Literal or to an XML schema built-in datatype value [46].
3) Annotation-based strategies, which exploit the *rdfs:comment* (a property which provides human readable descriptions to concepts in the domain ontology), and the *rdfs:label* (a *property* which is used to provide a name for the class or the property in the domain ontology) properties.

The object-based strategies are used to auto-generate true and false, multiple choice, and short answer questions which assess students on the knowledge, comprehension, application and analysis levels in Bloom's taxonomy. Fig 6 shows the nine object-based generation strategies which are used in the current OntoPeFeGe framework to generate questions.

The datatype-based strategies are used in OntoPeFeGe to generate true and false, multiple choice, and short answer questions by exploiting the datatype properties in the domain ontology. Fig. 7 shows strategy 17 [36], which generates the question's *Key* and the question's *Distractors*. The *Key* is the object of the datatype property and it is a numerical value while the *Distractors* are the multiples or submultiples of the numerical value.

Strategy 8
Key: Individuals of Class A related to individuals of Class B using the Property P
Distractor: Individuals of Class A's subclasses

Strategy 9
Key: Individuals of Class A related to individuals of Class B using the Property P
Distractor: Individuals of Class A's sibling classes

Strategy 10
Key: Individuals of Class A related to Individuals of Class B using the Property P
Distractor: Individuals of Class B's subclasses

Strategy 11
Key: Individuals of Class A related to individuals of Class B using the Property P
Distractor: Individuals of class B's sibling classes

Strategy 12
Key: Property P has an individual in the domain Class A and an individual in the range Class B
Distractor: Property S has an individual in the domain Class C which is a subclass of Class A and an individual in the range Class D which is a subclass of Class B

Strategy 13
Key: Property P has an individual in the domain Class A and an individual in the range Class B
Distractor: Property S has an individual in the domain Class C which is a sibling of Class A and individual in the range Class D which is a sibling of Class B

Strategy 14
Key: Class A related to Class B using the property P
Distractor: Class A's sibling classes

Strategy 15
Key: Class A related to Class B using the Property P
Distractor: Class B's sibling classes

Strategy 16
Key: Property P has the domain Class A and the range Class B
Distractor: Property S has the domain Class C which is a sibling of Class A and the range Class D which is a sibling of Class B

FIGURE 6. Object-based strategies.

Strategy 17
Key: numerical value
Distractor: The value multiples or submultiples

FIGURE 7. Datatype-based strategy.

Strategy 18
Key: Annotation X in class A
Distractor: Annotation Y in class B, X and Y have high similarity measure

Strategy 19
Key: Class A which is described in annotation X, X(A)
Distractor: Class B, A:Z and B:Y have high similarity measure

Strategy 20
Key: Class A
Distractor: Class B, X and Y have high similarity measure

FIGURE 8. Annotation-based strategies.

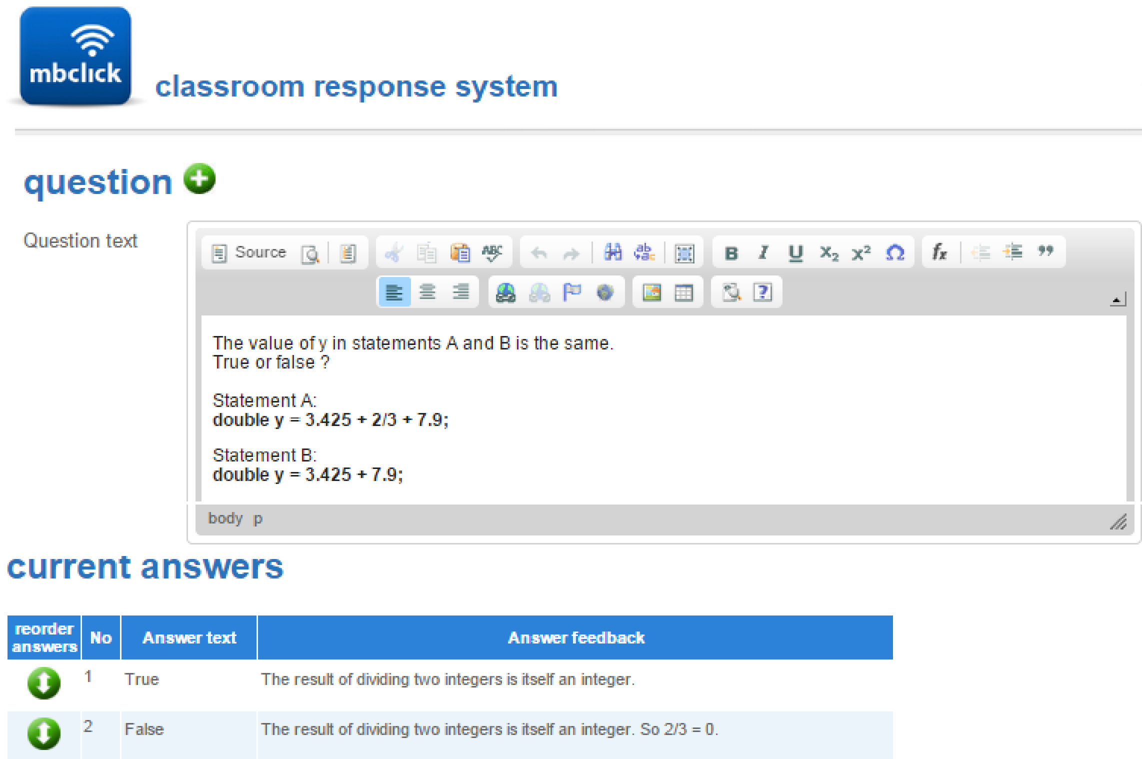

FIGURE 9. A true and false question in mbclick and the associated feedback comments.

The annotation-based strategies exploit the rdfs:comment and the rdfs:label associated with the ontology classes and individuals in the domain ontology. Fig. 8 shows the annotation-based strategies, which were used to generate the multiple choice questions illustrated in Table. 2. The true and false and short answer stem templates defined by Grubišić *et al.* [36], [37], and Cubric and Tosic [35] were not designed to use the annotation-based strategies. Instead, they focused on assessing the students on the object properties in the educational domain.

Integrating the stem templates and the different ontology-based generation strategies described above into OntoPeFeGe allowed the quality of tests and questions auto-generated to be quantitatively analyzed for the first time in [47]. The experiment was carried out on three different auto-generated tests which were performed by 126 students, 88 students and 89 students respectively. The results revealed that the three assessment tests formed from the auto-generated questions had medium difficulty values, which are very close to the value (0.5) that the test authors are advised to achieve when constructing tests. In addition, the results revealed that the questions and tests had satisfactory positive discrimination values, which indicate that the questions and tests could effectively discriminate between high ability and low ability students. The results obtained from the experimental study encouraged associating the questions generated with different types of feedback. To specify the types of feedback which teachers usually provide to students in VLEs, and OntoPeFeGe should focus on, a preliminary study is carried out in Section III-A.2. In addition, a detailed description on feedback generation is provided.

2) FEEDBACK GENERATION

This section presents a preliminary study which aims to specify the types of formative feedback teachers usually use in VLEs. In addition, it explains the feedback generation process in detail.

The study focused on analyzing the content of formative feedback which teachers provide to students in VLEs immediately after answering an assessment question. To achieve this, Brown and Glover qualitative coding system was used [1], [8]–[10], [13], [48]. The coding system categorizes the types of feedback according to the depth of detail provided in each type into the following three main categories [49], [50]: 1) Indication feedback which notifies students if the provided answer is correct or incorrect. This category contains the Knowledge Of Result (KOR) feedback. 2) Correction feedback which provides students with the correct answer. This category contains the Knowledge of Correct Response (KCR) feedback. 3) Explanation feedback, which provides students with explanation relevant to their answers. For example, students who fail to answer the assessment question receive feedback which explains to them the reason why their answer is incorrect. The explanation feedback defined by Brown and Glover contains

TABLE 3. Number of questions and feedback comments analyzed in mbclick.

Discipline	Course	Teacher ID	Number of students	Year of study	Number of questions	Number of feedback comments
EEE	Java Programming [153]	A	109	Second year undergraduates	25	88
EEE	Data Networking [154]	A	64	Third year undergraduates	12	47
Social Sciences	Introductory Mathematics [155]	B	218	First year undergraduates	4	17
Chemistry	Introductory Chemistry [156]	C	225	First year undergraduates	9	36

the Bugs-related (BR), the Topic Contingent (TC) and the Response Contingent (RC) types of feedback.

Three teachers volunteered to take part in the experiment from the following schools at the University of Manchester: the School of Electrical and Electronic Engineering (EEE), the School of Social Science, and the School of Chemistry. The teachers used the mbclick [51], [52] assessment system which is an electronic voting system developed by the University of Manchester to assess students during a lecture session [51]. The system provides teachers with a web-based VLE to create true and false, multiple choice, and short answer questions. It also provides teachers with the facilities to associate hard-coded feedback, which is called the feedback comment to each question's option. Fig. 9 is a screen shot of a true and false question created in the mbclick system. It shows the two formative feedback comments created by the teacher for the question's true and false options.

Students used their mobile phones to access the mbclick web-based environment and answer the questions. After students have submitted their answers, mbclick provides them with immediate feedback related to their selected option [51].

In this study, the feedback comments the three teachers provided to students using mbclick were analyzed. Table. 3 shows the educational courses, the number of students, the level of students, the number of questions and the number of feedback comments analyzed in this study.

Brown and Glover's feedback coding system [149, 150] was used to analyze the KOR, KCR and explanation feedback comments teachers provided to students in each of the courses presented in Table. 3. Each question consisted from two to five options and each option was associated with a feedback comment. After that, further analysis was carried out to investigate the percentage of feedback comments in each of the explanation feedback categories: BR, TC, and RC types of feedback. The study also investigated other types of feedback, which teachers could provide to students in VLEs. These types of feedback are the hint feedback which provides a student with information on what to do next to guide him or her towards the right solution, and the and Answer Until Correct (AUC) feedback which provides the student with KOR feedback until he/she answers the question correctly.

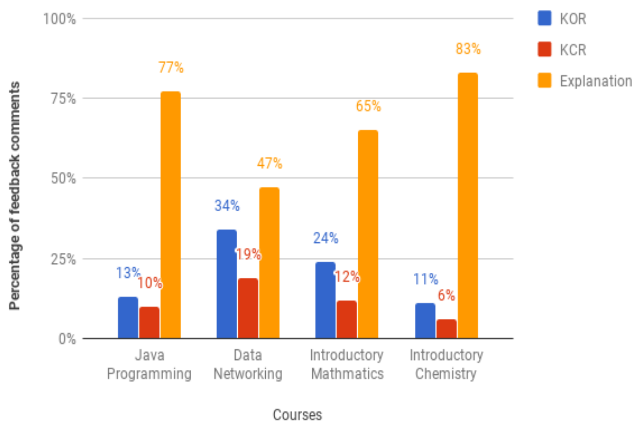

FIGURE 10. Percentage of types of feedback provided to students in learning environments.

Fig. 10 shows that the percentages of explanation feedback comments were much higher in the four educational courses [47%-83%] compared to the percentage of KOR feedback comments which ranged between [11%-34%] and the percentages of KCR feedback comments which ranged between [6%-19%]. More detailed analysis was carried out to investigate the percentages of feedback comments in:

1) The explanation feedback categories: BR, TC, and RC.
2) The hint feedback.
3) The AUC feedback.

Fig. 11 shows that teachers used the BR, TC, and RC feedback comments in the four educational courses. However, hint and AUC feedback were not used.

Based on the preliminary study results discussed above, five different types of formative feedback (KOR, KCR, BR, TC, RC) were generated in OntoPeFeGe using the domain ontologies. These types of feedback were either neglected (Kazi *et al.* [21], [25], and [26] focused on auto-generating hint feedback) or partially supported (Frutos-Morales *et al.* [28], del Mar Sánchez-Vera *et al.* [29], and Castellanos-Nieves *et al.* [30] focused on auto-generating KOR and KCR feedback) by the feedback generators introduced in Section II-B. The feedback generated in OntoPeFeGe is domain independent feedback where no

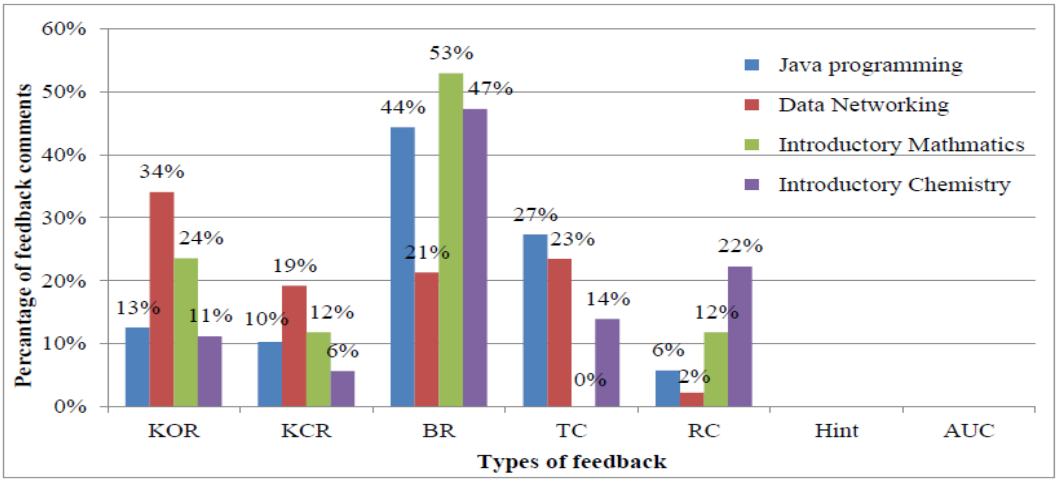

FIGURE 11. Percentage of types of feedback provided to students in mbclick.

expert knowledge base or human intervention (teachers or domain experts) is needed.

The generator associates the questions with different types of feedback. Fig. 12 shows that the different types of feedback are formed from one or more of the following four pedagogical contents:

1) Right/wrong.
2) The correct answer.
3) The reason why the correct answer is correct.
4) The reason why an incorrect answer is incorrect.

The feedback pedagogical contents are auto-generated by traversing the domain ontology and filling the pedagogical content templates, which may change according to the ontology-based generation strategies (class, terminology, and property-based strategies) used during the generation process.

The *right/wrong* pedagogical content is specified in Algorithm 1 and it is used to auto-generate the KOR feedback. The algorithm does not depend on the ontology-based generation strategies. It only depends on the auto-generated question's *Key* and *Distractor* individuals. Each *Key* individual is associated with *your answer is right feedback* (line 5), and each *Distractor* individual is associated with *your answer is wrong feedback* (line 7).

Algorithm 1 Right/ Wrong Pedagogical Content

1 *op* ← options which consist of a key and distractors;
2 *K* ← key;
3 *KOR* ← Knowledge Of Results feedback;
4 **if** *op* == *K* **then**
5 | *KOR*=GenerateRight();
6 **else**
7 | *KOR*=GenerateWrong();

Similarly, *the correct answer pedagogical content* does not depend on the ontology-based generation strategies. It only requires the auto-generated question's *Key*, which represent the correct answer. The *Key* could be an individual, class, or property in the domain ontology. This depends on the ontology-based generation strategy used during the

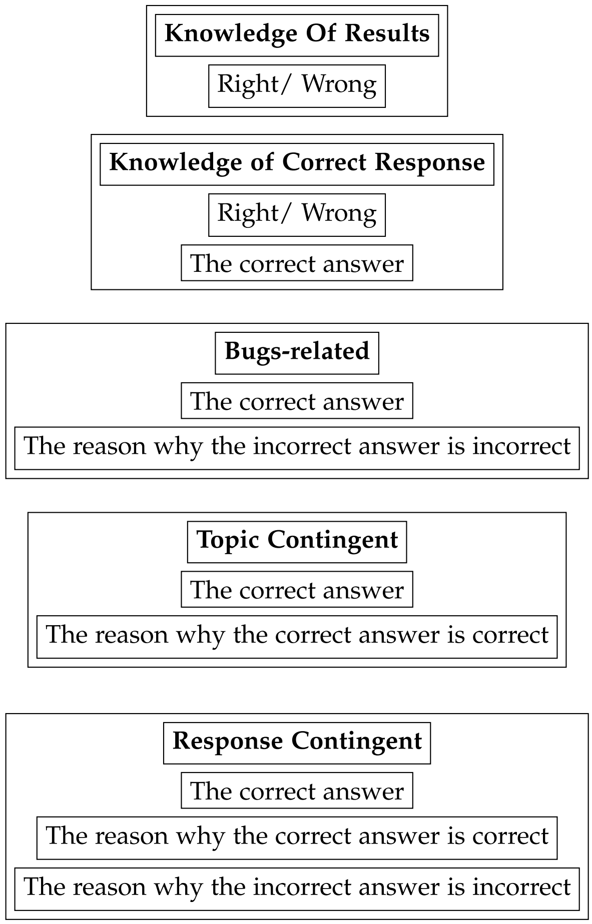

FIGURE 12. Types of formative feedback and their pedagogical content.

generation process. For example, in a class-based strategy the question's *Key* will be an individual in the domain ontology while in a terminology-based strategy the question's *Key* will be a class in the domain ontology. The correct answer pedagogical content is generated using Algorithm 2 which uses the Key *label* (line 4).

The KCR feedback is formed by calling Algorithm 1 and Algorithm 2 for the auto-generated question's *Key* and *Distractors*.

Algorithm 2 Correct Answer Pedagogical Content

```
1 Function CorrectAnswer (Key)
2     CA ← Correct Answer;
3     CA.append("The correct answer is");
4     CA.append(key → label);
5     return CA;
```

As explained above neither the *right/wrong pedagogical content* nor the *correct answer pedagogical content* depend on the ontology-based generation strategies used in the generation process. Whereas *the reason why the correct answer is correct* and *the reason why an incorrect answer is incorrect* pedagogical contents depend on the ontology-based generation strategies as shown in Algorithms 3 and 4. This means that the Bugs-Related (BR), Topic Contingent (TC), and Response Contingent (RC) feedback pedagogical content will change based on the ontology-based generation strategy used in the generation process. The following sections illustrate the algorithms used to generate BR, TC and RC types of feedback in OntoPeFeGe. The algorithms are presented according to the ontology-based generation strategies. The ontologies used in the following examples are OpenCyc [53] and Sakathi's Computer network ontology [44].

Algorithm 3 Reason Why Correct Pedagogical Content

```
Function ReasonCorrect (Strategy)
    if Strategy.isClass() then
        Call Function ClassBasedReasonCorrect;
    else if Strategy.isTerminology() then
        Call Function
        TerminologyBasedReasonCorrect;
    else if Strategy.isObjectProperty() then
        Call Function
        ObjectPropertyBasedReasonCorrect;
    else if Strategy.isDatatypeProperty() then
        Call Function
        DatatypePropertyBasedReasonCorrect;
    else if Strategy.isAnnotationProperty() then
        Call Function AnnotationReasonCorrect;
```

Algorithm 4 Reason Why Incorrect Pedagogical Content

```
Function ReasonIncorrect (Strategy, Key)
    if Strategy.isClass() then
        Call Function ClassBasedReasonIncorrect;
    else if Strategy.isTerminology() then
        Call Function
        TerminologyBasedReasonIncorrect;
    else if Strategy.isObjectProperty() then
        Call Function
        ObjectPropertyBasedReasonIncorrect;
    else if Strategy.isDatatypeProperty() then
        Call Function
        DatatypePropertyBasedReasonIncorrect;
    else if Strategy.isAnnotationProperty() then
        Call Function AnnotationReasonIncorrect;
```

CLASS-BASED STRATEGIES

The five class-based generation strategies shown in Fig. 3 [33], [34] are used in OntoPeFeGe to associate the question's Key and Distractor individuals with different types of formative feedback. The types of feedback are formed from the four pedagogical contents shown in Fig. 12.

For a concrete example, Fig. 13 shows the Transport Layer Protocol class in the OpenCyc ontology [53] which has six individuals. Applying a class-based strategy (strategy 3 in Fig. 3) to the ontology will generate the multiple choice question shown in Table. 4, which assess students at the application level.

Table. 4 shows that the question's *Key* is the *Transmission Control Protocol* which is an individual in the *Transport Layer Protocol* class, while the *Distractors* are generated from sibling classes such as the *Domain Name System Protocol* which is an individual in the *Application Layer Protocol* class.

When a student chooses the *Domain Name System Protocol* as an answer, he or she will be provided with the auto-generated formative feedback shown in Table. 4. For example, in Table. 4 the feedback pedagogical contents *your answer is wrong* and *the correct answer is Transmission Control Protocol* are generated using Algorithm 1 and Algorithm 2 respectively. OntoPeFeGe also auto-generates *the reason why the correct answer is correct* and *the reason why the incorrect answer is incorrect* pedagogical contents using Algorithm 5 and Algorithm 6 respectively. Algorithm 5 takes the question's Key individual (line 1) as a parameter (e.g., *Transmission Control Protocol*) and provides students with the ontology class (e.g., *Transport Layer Protocol*) which the *Key* individual belongs to (line 7). See *the reason why the correct answer is correct* pedagogical content auto-generated for the question example in Table. 4.

On the other hand, Algorithm 6 takes the question's *Distractor* individual as a parameter (e.g., *Domain Name System Protocol*) and provides students with information about the *Distractor* class in which the individual they

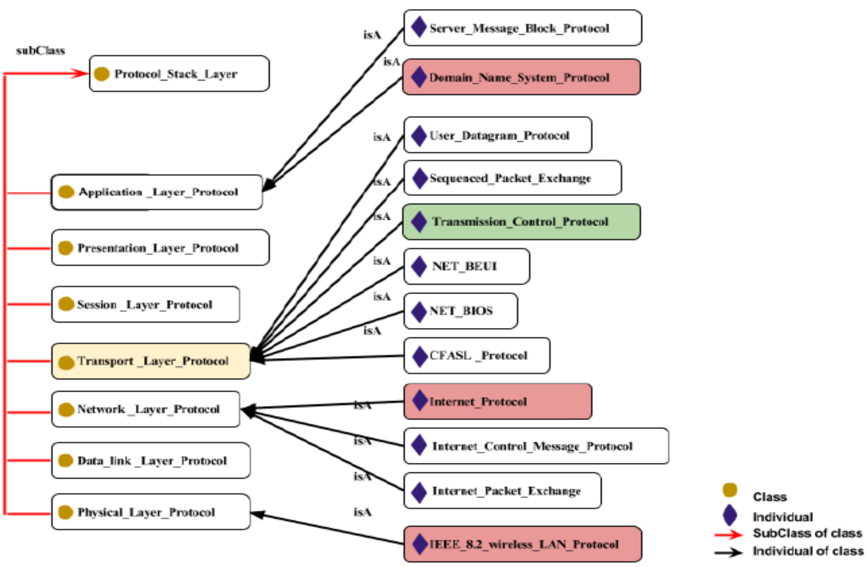

FIGURE 13. Transport Layer Protocol class and individuals in OpenCyc ontology.

TABLE 4. Question and feedback generated using a class-based strategy.

Ontology-based generation strategy	Class-based generation strategy (Strategy 3)
Stem template	Which one of the following demonstrates the concept *Class A*?
Stem individual	Which one of the following demonstrates the concept **Transport Layer Protocol**?
Key	Transmission Control Protocol
Distractors	IEEE 8.2 wireless LAN protocol Domain Name System Protocol Internet Protocol
Generated feedback pedagogicalcontent when a student selects the **Domain Name System Protocol**.	1. Your answer is wrong. 2. The correct answer is **Transmission Control Protocol**. 3. The reason why **Transmission Control Protocol** is the correct answer is due to the following: Transmission Control Protocol is a Transport Layer Protocol. 4. The reason why **Domain Name System Protocol** is the incorrect answer is due to the following: **Domain Name System Protocol** is an **Application Layer Protocol**.

Algorithm 5 Reason Why Correct (Class-Based Strategies)

```
Function ClassBasedReasonCorrect (Key)
    KR ← Reason why the Key option is correct;
    KR.append("The reason why");
    KR.append(Key → label);
    KR.append("is the correct answer is due to the
    following:");
    KR.append(Key → label);
    KR.append(Key → class);
    return KR;
```

Algorithm 6 Reason Why Incorrect (Class-Based Strategies)

```
Function ClassBasedReasonIncorrect (Distractor)
    DI ← Reason why the distractor option is incorrect;
    DI.append("The reason why");
    DI.append(Distractor → label);
    DI.append("is the incorrect answer is due to the
    following:");
    DI.append(Distractor → label);
    DI.append(Distractor → class);
    return DI;
```

selected belongs to (e.g., *Application Layer Protocol*). See *the reason why the incorrect answer is incorrect* pedagogical content auto-generated for the question example in Table. 4.

TERMINOLOGY-BASED STRATEGIES

Two terminology-based generation strategies (Strategy 6 and Strategy 7) shown in Fig. 4 are used in the current OntoPeFeGe framework to generate true and false questions which assess students' cognition at the knowledge, comprehension, and application levels in Bloom's taxonomy.

OntoPeFeGe auto-generates *the reason why the correct answer is correct* pedagogical content for the true and false questions using Algorithm 7. The algorithm uses the question's *Key*, and the subject parameters (line 1). The subject of the *subClass* property is used as a parameter because the

Algorithm 7 Reason Why Correct (Terminology-Based Strategies)

```
1  Function TerminologyBasedReasonCorrect (Key, Subject)
2      KR ← Reason why the Key option is correct;
3      KR.append("The reason why");
4      KR.append(Key → label);
5      KR.append ("is the correct answer is due to the following:");
6      foreach class ∊ Subject.listSuperclasses do
7          if class.isRestriction() == false then
8              KR.append(Subject → label);
9              KR.append("is");
10             KR.append(class → label);
11         else
12             KR.append(Subject → label);
13             Restriction = class → asRestriction();
14             Type = Restriction → type ;
15             KR.append(Type → getPropertyLabel);
16             KR.append(Type → getValuesFromLabel);
17     return KR;
```

Key in true and false questions is either a *yes* or *no* individual. Algorithm 7 retrieves the superclasses for the subject to help the student relate the subject to the correct object (line 6). For each superclass (Object) the algorithm checks if it is a class (line 7) or a restriction. If the superclass is a class, then the algorithm retrieves the superclass label (line 10). On the other hand, if the superclass is a restriction (line 11) the algorithm retrieves the type of the restriction (line 14) which could be *owl:allValuesFrom*, *owl:someValuesFrom*, or *owl:hasValue* (see Section 2.3.1 in Chapter 2, page 38), and then retrieves the property label (line 15) and the class label (line 16) which the restriction is applied on.

For example, Table. 5 shows a true and false knowledge level question auto-generated using the terminology-based strategy 6 shown in Fig. 4. The question is auto-generated after traversing the domain ontology shown in Fig. 5. The ontology shows that the *Transport Layer* is a subclass of *transmits only datagrams* restriction class, and the *Data link Layer* is a subclass of *transmits only frames* restriction class. The question is auto-generated by replacing the object in the following statement from *transmits only datagrams* to *transmits only frames*:

$$\overbrace{Transport\ \ Layer'}^{Subject}\ \overbrace{subclass}^{Property}\ \overbrace{transmits\ \ only\ \ datagram'}^{Object}$$

Table. 5 also shows an example of *the reason why the correct answer is correct* pedagogical content, which explained to students that the *Transport layer* transmits datagrams and not frames. OntoPeFeGe also auto-generates *the reason why the incorrect answer is incorrect* pedagogical content using Algorithm 8. The algorithm uses the question's *Distractor* and the object parameter (line 1) which is used to retrieve the object subclasses (line 6). To auto-generate the pedagogical content the algorithm uses the subclass label (line 7) and checks if the object parameter is a class (line 8) or a restriction. The object in the example shown in Table. 5 is *transmits only frames*, which is a restriction and the subclass of the object is the *Data link Layer* (see Fig. 5). The pedagogical content is auto-generated to explain to students that the *Data link Layer* transmits frames.

Algorithm 8 Reason Why Incorrect (Terminology-Based Strategies)

```
1  Function TerminologyBasedReasonIncorrect (Distractor, Object)
2      DI ← Reason why the distractor option is incorrect;
3      DI.append("The reason why");
4      DI.append(Distractor → label);
5      DI.append("is the incorrect answer is due to the following:");
6      foreach class ∊ Object.listSubClasses do
7          DI.append(class → label);
8          if Object.isRestriction() == false then
9              DI.append(Object → label);
10         else
11             Restriction = Object → asRestriction();
12             Type = Restriction → type ;
13             DI.append(Type → getPropertyLabel);
14             DI.append(Type → getValuesFromLabel);
15     return DI;
```

PROPERTY-BASED STRATEGIES

The following three categories of the property-based strategies: object-based strategies, datatype-based strategies, and Annotation-based strategies are used to generate the different types of feedback.

The object-based strategies generate *the reason why the correct answer is correct* pedagogical content using Algorithm 9. The algorithm uses the question's *Key*, which could be an individual, class, property, or *yes/no* (if a true and false question is generated). The algorithm also takes the *Key* object property, the *Key* subject, and the *Key* object parameters to capture the statement associated with the correct answer (subject property object). OntoPeFeGe also auto-generates *the reason why the incorrect answer is incorrect* using Algorithm 10 which takes the *Distractor* object property, the *Distractor* subject, and the *Distractor* object parameters which capture the statement associated with the incorrect answer (subject property object). *The reason why the correct answer is correct* pedagogical content auto-generated using the object-based strategies provides students with the statement associated with the correct answer, while *the reason why the incorrect answer is incorrect* pedagogical content

TABLE 5. Question and feedback generated using a terminology-based strategy.

Ontology-based generation strategy	Terminology-based generation strategy (Strategy 6)	
Stem template	Is *Class A* subclass of *Class B*?	
Stem individual	Is Transport layer transmits frames?	
Key	No	Transport layer transmits datagrams
Distractors	Yes	Transport layer transmits frames
Generated feedback pedagogical content when a student selects the Yes option.	1. Your answer is wrong. 2. The correct answer is No. 3. The reason why No is the correct answer is due to the following: **Transport layer transmits datagrams**. 4. The reason why Yes is the incorrect answer is due to the following: **Data link layer transmits frames**.	

TABLE 6. Question and feedback generated using an object-based strategy.

Ontology-based generation strategy	Property-based generation strategy (Strategy 14)
Stem template	Which superclass is directly connected by *Property* with *Class A*?
Stem individual	Which one of the following is a function of the **Transport Layer**?
Key	Connection Control
Distractors	Synchronisation Logical Addressing Physical Addressing
Generated feedback pedagogical content when a student selects the **Logical Addressing**.	1. Your answer is wrong. 2. The correct answer is **Connection Control**. 3. The reason why **Connection Control** is the correct answer is due to the following: **Transport Layer functions Connection Control**. 4. The reason why **Logical Addressing** is the incorrect answer is due to the following: **Network Layer functions Logical Addressing**.

Algorithm 9 Reason Why Correct (Object-Based Strategies)

```
Function ObjectPropertyBasedReasonCorrect (Key, keyObjProperty, keySubject, keyObject)
    KR ← Reason why the Key option is correct;
    KR.append("The reason why");
    KR.append(Key → label);
    KR.append("is the correct answer is due to the following:");
    KR.append(keySubject → label);
    KR.append(keyObjProperty → label);
    KR.append(keyObject → label);
    return KR;
```

Algorithm 10 Reason Why Incorrect (Object-Based Strategies)

```
Function ObjectPropertyBasedReasonIncorrect (Distractor, distractorObjProperty, distractorSubject,distractorObject)
    DI ← Reason why the distractor option is incorrect;
    DI.append("The reason why");
    DI.append(Distractor → label);
    DI.append("is the incorrect answer is due to the following:");
    DI.append(distractorSubject → label);
    DI.append(distractorObjProperty → label);
    DI.append(distractorObject → label);
    return DI;
```

provides students with the statement associated with the incorrect answer.

Table. 6 shows an example of a comprehension level question auto-generated using the object-based strategy 14 shown in Fig. 6. The table shows *the reason why the correct answer is correct* pedagogical which explains to the student that the Connection Control (*Key object*) is a function (*Key object property*) of the Transport Layer (*Key subject*). While *the reason why the incorrect answer is incorrect* explains to the student that the Logical Addressing (*Distractor object*) is a function (*Distractor object property*) of the Network Layer (*Distractor subject*).

The data-type based strategies generate true and false, multiple choice, and short answer questions by exploiting the datatype properties in the domain ontology. Fig. 7 shows strategy 17 [36], which generates the question's *Key* and the question's *Distractors*. The *Key* is the object of the datatype property and it is a numerical value while the *Distractors* are the multiples or submultiples of the numerical value. OntoPeFeGe uses Algorithm 11 to auto-generate *the reason why the correct answer is correct* pedagogical content. The algorithm takes the question's *Key* and the datatype property associated with the *Key* (keyDatatypeProperty) as parameters (line 1) and then retrieves the object of the key datatype property (line 9). Table. 7 illustrates a true and false analysis level question auto-generated using Strategy 17. The question aims to assess if the students know the *number of layers* in the Transmission Control Protocol/ Internet Protocol model (*TCP/IP model*). *TCP/IP model* is a class in the domain ontology which has the *number of layers* datatype property. Table. 7 shows that *the reason why the*

TABLE 7. Question and feedback generated using a datatype-based strategy.

Ontology-based generation strategy	Property-based generation strategy (Strategy 17)
Stem template	Is *Property* of *Subject Object*?
Stem individual	Is number of layers of TCP/IP model 8?
Key	No
Distractors	Yes
Generated feedback pedagogical content when a student selects the **Yes** option.	1. Your answer is wrong. 2. The correct answer is **4**. 3. The reason why **4** is the correct answer is due to the following: The number of layers of TCP/IP model is 4. 4. The reason why **8** is the incorrect answer is due to the following: **8** is double the number of layers of TCP/IP model and number of layers of TCP/IP model is 4.

Algorithm 11 Reason Why Correct (Datatype-Based Strategies)

```
Function DatatypePropertyBasedReasonCorrect (Key, keyDatatypeProperty)
    KR ← Reason why the Key option is correct;
    KR.append("The reason why");
    KR.append(Key → label);
    KR.append("is the correct answer is due to the following:");
    KR.append(keyDatatypeProperty → label);
    KR.append("of");
    KR.append(Key → label KR.append("is");
    object = keyDatatypeProperty → Object;
    KR.append(object → label);
    return KR;
```

correct answer is correct pedagogical content explained to students that the *number of layers* in the *TCP/IP* model is *4*.

In addition to *the reason why the correct answer is correct* pedagogical content, OntoPeFeGe auto-generates *the reason why the incorrect answer is incorrect* using Algorithm 12. The algorithm uses the question's *Distractor* (line 1). It starts by providing students with information about their selected answer (line 4), and then explains that the selected answer is double, triple, or quadruple the correct answer (line 8). After that, the algorithm provides the students with more details about the correct answer. Table. 7 shows an example of *the reason why the incorrect answer is incorrect* pedagogical content which is auto-generated in OntoPeFeGe. The pedagogical content explained to students that 8 is double the *number of layers* in the TCP/IP model. It also provided the students with information about the number of layers in the TCP/IP model. The annotation-based strategies auto-generate *the reason why the correct answer is correct* pedagogical content using Algorithm 13. The Algorithm takes the following parameters: the *Key* in the auto-generated question, the name of the annotation-based strategy (e.g., Strategy 18), and the ontology class having the annotation property used to auto-generate the question's *Key (ClassAnnot)*. The *ClassAnnot* parameter is used when questions are generated using strategy 19 (see Fig. 8). Strategy 19 shows that the question's *Key* is a class in the domain ontology, which is described

Algorithm 12 Reason Why Incorrect (Datatype-Based Strategies)

```
1  Function ObjectPropertyBasedReasonIncorrect (Distractor, Key, keyDatatypeProperty)
2      DI ← Reason why the distractor option is incorrect;
3      DI.append("The reason why");
4      DI.append(Distractor → label);
5      DI.append("is the incorrect answer is due to the following:");
6      DI.append(Distractor → label);
7      DI.append("is");
8      DI.append(multiplierValue → label);
9      DI.append(keyDatatypeProperty → label);
10     DI.append(Key → label);
11     DI.append("and");
12     DI.append(keyDatatypeProperty → label);
13     DI.append("of");
14     DI.append(Key → label);
15     object = keyDatatypeProperty → Object;
16     DI.append(object → label);
17     return DI;
```

in the annotation property of another class in the same domain ontology.

The algorithm shows that the annotation-based strategies auto-generate different pedagogical contents for *the reason why the correct answer is correct*. When strategy 18 [35] is used in the generation process, students are provided with questions to assess if they could provide a definition of the educational concepts (class or individual) in the domain ontology (see question 1 in Table. 2). The options (*Key* and *Distractors*) in the auto-generated question are definitions retrieved from several classes or individuals in the domain ontology. OntoPeFeGe auto-generates *the reason why the correct answer is correct* pedagogical content to provide the students with the *Key* class (the correct educational concept) which the definition belongs to (line 9).

On the other hand, when strategy 19 [35] is used to auto-generate the multiple choice questions, the question's *Key* is auto-generated from an ontology class having an annotation property containing the *Key*. Therefore, the pedagogical content is auto-generated by querying the class

Algorithm 13 Reason Why Correct (Annotation-Based Strategies)

```
1  Function AnnotationReasonCorrect (Key, strategyName, ClassAnnot)
2      KR ← Reason why the Key option is correct;
3      if strategyName == Strategy18 then
4          KR.append("The reason why");
5          KR.append(Key → comment);
6          KR.append("is the correct answer is due to the following:");
7          KR.append(Key → comment);
8          KR.append("is the definition for");
9          KR.append(Key → label);
10     else if strategyName == Strategy19 then
11         KR.append("The reason why");
12         KR.append(Key → label);
13         KR.append("is the correct answer is due to the following:");
14         KR.append(ClassAnnot → comment);
15     else if strategyName == Strategy20 then
16         KR.append("The reason why");
17         KR.append(Key → label);
18         KR.append("is the correct answer is due to the following:");
19         KR.append(Key → label);
20         KR.append("is defined as");
21         KR.append(Key → comment);
22     return KR;
```

Algorithm 14 Reason Why Incorrect (Annotation-Based Strategies)

```
1  Function AnnotationReasonIncorrect (Distractor, strategyName)
2      DI ← Reason why the distractor option is incorrect;
3      if strategyName == Strategy18 then
4          DI.append("The reason why");
5          DI.append(Distractor → comment);
6          DI.append("is the incorrect answer is due to the following:");
7          DI.append(Distractor → comment);
8          DI.append("is the definition for");
9          DI.append(Distractor → label);
10     else if strategyName == Strategy19 then
11         DI.append("The reason why");
12         DI.append(Distractor → label);
13         DI.append("is the incorrect answer is due to the following:");
14         DI.append(Distractor → comment);
15     else if strategyName == Strategy20 then
16         DI.append("The reason why");
17         DI.append(Distractor → label);
18         DI.append("is the incorrect answer is due to the following:");
19         DI.append(Distractor → label);
20         DI.append("is defined as");
21         DI.append(Distractor → comment);
22     return DI;
```

annotation property (line 14). For example, Table. 8 illustrates an analysis level question generated using strategy 19. The question's *Key* is the *Application layer protocol*, which is contained in the *Presentation Layer Protocol* annotation property (rdfs:comment). The table shows *the reason why the correct answer is correct* pedagogical content, which provides the students with the *rdfs:comment* of the *Presentation Layer Protocol*.

In addition to strategies 18 and 19, strategy 20 [35] is used to auto-generate questions which assess if the students could relate a specific definition to a concept in the domain ontology. Algorithm 13 shows that *the reason why the correct answer is correct* pedagogical content is generated to provide the student with the correct definition that is related to the question's *Key* (see line 22 in Algorithm 13).

OntoPeFeGe also auto-generates *the reason why the incorrect answer is incorrect* using Algorithm 14. The generation process is similar to Algorithm 13. However, instead of using the *Key* parameter the function used the *Distractor* parameter. For example, Table. 8 shows *the reason why the incorrect answer is incorrect* pedagogical content auto-generated in OntoPeFeGe when strategy 19 is used. The table shows that when a student chose the *Session Layer Protocol* he or she was provided with the annotation property associated with the chosen *Distractor* (*Session Layer Protocol*).

B. PERSONALISED FEEDBACK ALGORITHM

The previous section introduced the generator, which auto-generates KOR, KCR, BR, TC, and RC types of feedback from a domain ontology. The generator associated the different types of feedback with questions aimed to assess the students at different levels in Bloom's taxonomy. This section explains the personalized feedback algorithm which provides the appropriate type of formative feedback to the students immediately after answering an assessment question. The algorithm is rule-based which adopts and implements the theoretical personalized feedback framework proposed by Mason and Bruning [9]. The algorithm starts by fetching the first question in a test. Students with low background knowledge receive Response Contingent feedback regardless of the correctness of their answer or the level of the question in Bloom's taxonomy. On the other hand, students with high background knowledge are provided with different types of feedback based on their current level of knowledge and the level of the question in Bloom's taxonomy. Students who answer the knowledge level questions correctly are provided with Bugs-Related feedback, and the students who answer the knowledge level questions incorrectly are provided with Topic Contingent feedback. The algorithm also considers students with high background knowledge and provides them with Topic Contingent feedback after

TABLE 8. Question and feedback generated using an annotation-based strategy.

Ontology-based generation strategy	Property-based generation strategy (Strategy 19)
Stem template	Analyse the following text and decide which one of the following words is a correct replacement for the blank space in the text: *Note: the text is Class B's annotationproperty (comment) and the blank spaceshown below is Class A, which is contained in the comment.*
Stem individual	Analyse the following text and decide which one of the following words is a correct replacement for the blank space in the text: 'A presentation layer protocol takes the responsibility for routine tasks from an ——, such as converting between character sets.'
Key	Application Layer Protocol
Distractors	Presentation Layer Protocol Transport Layer Protocol Session Layer Protocol
Generated feedback pedagogical content when a student selects the **Session Layer Protocol** option.	1. Your answer is wrong. 2. The correct answer is **Application Layer Protocol**. 3. The reason why **Application Layer Protocol** is the correct answer is due to the following: A presentation layer protocol takes the responsibility for routine tasks from an **Application Layer Protocol**, such as converting between character sets. 4. The reason why **Session Layer Protocol** is the incorrect answer is due to the following: **Session Layer Protocol** allows sessions to be established between two machines. A session facilitates processes that involve intensive data transfer between two computers, such as transferring a large file.

answering comprehension, application and analysis level questions regardless of the correctness of their answer.

IV. OntoPeFeGe FRAMEWORK EVALUATION

This section presents the experiment carried out to evaluate the ontology-based personalized feedback generator and contributes to the research carried out in the personalized feedback frameworks [1], [9], [16]–[19] and the ontology-based formative feedback generators [21], [25]–[30] by achieving the following: 1) Examine the effect of personalized feedback on students' performance. 2) Study the relationship between student's characteristics (background knowledge), the question's characteristics (the level of each question in Bloom's taxonomy [40]) and the personalized feedback, and how they affect students' performance. 3) Observe students and teachers' satisfaction regarding the auto-generated feedback.

In 2013/2014, eighty-eight (69 males, 19 females) second and third year undergraduate students registered in the Data Networking course [54] and the Computer Networks course [55] at the University of Manchester volunteered to take part in the experiment. Students' identities were kept anonymous. Several ontologies which capture the educational concepts in the Data Networking and Computer Networks courses exists. To select the best candidate ontology which could be used in OntoPeFeGe, a method for Terminological Ontology Evaluation (TONE) was developed and used. TONE uses a textual corpus (e.g., textbooks) to evaluate the conceptual coverage of the underlying ontology, and the level of details an ontology captures about each concept (semantic richness). TONE combined the individual features introduced in existing methods by extracting terms form the corpus using several term extraction and recognition tools including noun phrase extractor and term frequency algorithm. Then it measures the ontology coverage and semantic richness metrics using the following equation:

$$Score = w_c \times \frac{F(O,T)}{max(F(O,T))} + w_s \times \frac{SR(O,T)}{max(SR(O,T))} \quad (1)$$

Where:

O: Set of concepts in the candidate domain ontology.

T: Is a set of terms extracted from the corpus and their synonyms obtained using WordNet.

$F(O,T)$: Is the F-measure Score of the candidate domain ontology.

$SR(O,T)$: Is the Semantic Richness Score of the candidate domain ontology.

w_c: Is the weight assigned by the teacher to the F-measure coverage score and it has a value between 0 and 1.

w_s: Is the weight assigned by the teacher to the Semantic Richness Score. It is (1-w_c) and it has a value between 0 and 1.

The ontology coverage was measured using the F-measure metric, while the semantic richness was measured for each concept in the candidate domain ontology which matches a term in the list of terms extracted from the corpus using the following formula:

$$Semantic\ Richness = \frac{\sum_{i=1}^{mt}(R_i + A_i + S_i)}{mt} \quad (2)$$

Where:

R_i: The summation of the number of concept' superclasses, subclasses and sibling classes.

A_i: The summation of the number of object properties, datatype properties, and annotation properties associated with the concept i.

S_i: The number of concepts in the candidate domain ontology that have the same meaning (synonymous) or have a name that contains concept $i's$ name.

TABLE 9. Distribution of the ontology-based generated questions.

Tests	# of questions	Ontology-based generation strategies			level of the question in Bloom's taxonomy				Types of question		
		Class	Terminology	Property	Knowl-edge	Comp-rehension	App-lication	Ana-lysis	True/False	Multiple choice	Short answer
1	14	1	4	9	4	4	4	2	4	10	0
2	16	1	4	11	4	4	4	4	4	11	1
3	14	1	4	9	4	4	4	2	4	9	1

mt: Is the total number of matched terms between the concepts in the candidate domain ontology and the list of extracted terms.

Values of F-measure and semantic richness are normalized to be in the range [0, 1] by dividing them by the maximum value of the measure for all candidate ontologies. Finally, ontologies are ranked according to their score. TONE was used to select the best candidate domain ontologies from a set of four ontologies using Eq. 1: the Computer Networks ontology which was intentionally developed to capture concepts in the computer networks domain, OpenCyc ontology which captures concepts related to the computer networks domain, Pizza ontology which describes the domain of pizza including pizza types, and C-Programming Language ontology which captures general concepts in C programming. The w_c and w_s were assigned a 0.5 value. TONE selected two ontologies which had the highest scores: the Computer Networks ontology which had the 0.531 score and the OpenCyc ontology which had 0.522 score. The Pizza ontology and C-Programming Language ontology had lower scores with 0.062 and 0.065 values respectively. TONE is an essential preface to OntoPeFeGe as it helps teachers select the most suitable candidate domain ontology for the generation process.

The experiment was carried out in the Moodle Virtual Learning Environment (VLE) [56] in a course called 'Computer Networks', which was created for the purpose of this experiment. The course included three tests which consist of assessment questions generated in OntoPeFeGe using the Computer Networks and OpenCyc ontologies. The tests aimed to assess students' knowledge of the transport layer topic. The tests shown in Table. 9 are not identical (the assessment questions in each test were different) but have similar structure; i.e., the tests consisted of questions assessing students' cognition at four levels in Bloom's taxonomy (knowledge, comprehension, application, and analysis). In addition, the tests used in the experiment were evaluated in [47] and proved to have approximately similar difficulty and discrimination values.

This study used the pre-test/treatment/post-test design. Students were asked first to answer the pre-test, and the test scores were used to allocate them randomly to the experimental group (40 males, 8 females) and the control group (29 males, 11 females) using the matched pairs design approach [57]. The basis for allocation is matching each member of the experimental group to a member of the control group based on their pre-test scores (background knowledge). This prevents having an unbalanced assignment of students with similar background knowledge in the same group. In the treatment phase, students in the experimental group received personalized feedback after answering each question, while students in the control group received KOR feedback. The KOR feedback was chosen because it is the default type of feedback auto-generated to students after answering true and false, multiple choice and short answer questions in VLEs (e.g., Moodle). Moreover, KOR provides students with the lowest level of information (correct or incorrect) compared with other types of feedback. After receiving personalized and KOR feedback in the treatment phase students were asked to answer the post-test.

A. EFFECT OF PERSONALISED FEEDBACK ON STUDENTS' PERFORMANCE

The results shown in Fig. 14 revealed that the personalized feedback significantly improved the performance of students with low background knowledge ($Z = -1.989$, P-value $= 0.047$, P-value < 0.05). On the contrary, students with low background knowledge in the control group who received KOR feedback had no difference in their performance ($Z = -1.574$, P-value $= 0.116$, P-value > 0.05). Moreover, the results revealed that high background knowledge students in the experimental and control had no statistically significant difference between the pre-test performance and post-test performance. This suggests

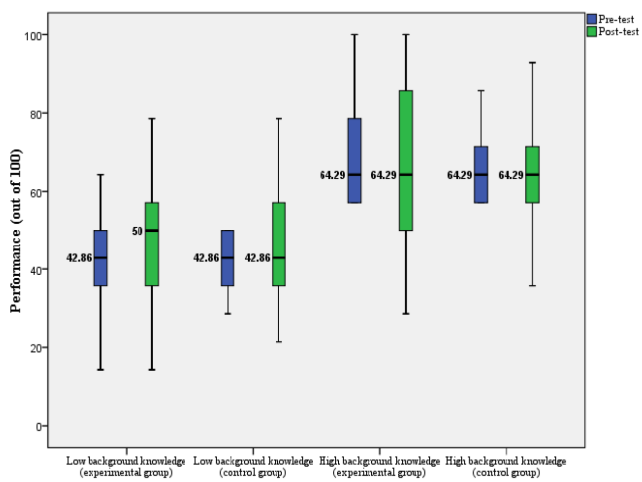

FIGURE 14. Performance of students with different background knowledge.

that students with low background knowledge benefit more from the personalized feedback compared to KOR feedback. The results are consistent with the results obtained by Arroyo *et al.* [18] and Woolf *et al.* [19]. Moreover, the results comply with Black and William's findings in [58].

B. RELATIONSHIP BETWEEN STUDENTS AND QUESTIONS' CHARACTERISTICS

The OntoPeFeGe adopts Mason and Bruning's personalized feedback theoretical framework [9] which considered both student's characteristics (background knowledge, current level of knowledge) and the question's characteristics (level of assessment question in Bloom's taxonomy). None of the personalized feedback frameworks in section II-A adapted the different types of feedback based on the question's characteristics or studied the relationship between the personalized feedback and the question's characteristics. Therefore, this experiment aims to examine students' performance after receiving the personalized feedback associated with questions designed to assess students at each level of Bloom's taxonomy. Moreover, the effect of the personalized feedback is compared to KOR feedback. The results revealed that both the personalized feedback and KOR feedback have the same effect on students' performance when provided to students after they answered questions designed to assess them at the knowledge and comprehension levels in Bloom's taxonomy. However, the effect of personalized feedback and KOR feedback on students' performance differed for questions designed to assess students at the application and analysis levels in Bloom's taxonomy. While the personalized feedback had no statistically significant effect on students' performance for questions designed to assess students at the application level, KOR feedback improved students' performance significantly ($Z = -2.495$, P-value $=$ 0.013, P-value < 0.05). On the other hand, the personalized feedback improved students' performance significantly compared to students who received KOR feedback at questions assessing the analysis level, as 50% of students had learning gain (post-test - pre-test) above zero in the experimental group compared to 25% of students in the control group. This result suggests that students benefited more from the personalized feedback at questions assessing the analysis level in Bloom's taxonomy.

C. QUALITY OF AUTO-GENERATED FEEDBACK

The quality of auto-generated feedback were evaluated by observing students' satisfaction and teachers' satisfaction. Students in the experimental group (48 students) answered a questionnaire which aimed to assess if the students understand the formative feedback and are willing and able to act on it [7]. The questionnaire assessed students' satisfaction regarding the feedback's usefulness, clarity, and whether the feedback helped them answer other questions in the test. The questionnaire had three questions scored on a 3-point Likert scale (agree, neutral, disagree). Fig. 15 shows that 72.92% of the students in the experimental group agreed that the feedback is useful, 70.83% agreed that the generated feedback was easy to read, and 68.75% agreed that the formative feedback provided in Moodle VLE helped them in answering some of the following questions in the assessment tests. The results are consistent with the evaluation results obtained by the ontology-based formative feedback generators in Section II-B where students accepted the auto-generated feedback and agreed that it was useful [21], [25], [27]–[29].

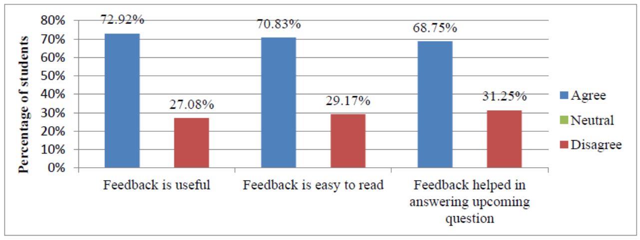

FIGURE 15. OntoPeFeGe feedback evaluated by students (Experimental group).

Fig. 15 also shows that approximately one-third of students in the experimental group were not satisfied with the formative feedback provided. Further investigation was carried out to investigate the correlation between students' responses to each question in the questionnaire and their background knowledge, post-test performance and the change in their performance from the pre-test to the post-test (increase, decrease, no change). The results revealed no correlation between students responses to the first question 'feedback is useful' and their background knowledge, post-test performance, and the change in their performance from the pre-test to the post-test. The results also revealed that students responses to the 'feedback is easy to read' question had no correlation with their background knowledge and post-test performance. However, the percentage of students in the experimental group who had an improvement in their performance (from the pre-test to the post-test) and agreed that the formative feedback was easy to read (64.7%) was higher than the percentage of student who agreed that the feedback was easy to read and had no improvement (5.9%) or decrease (29.4%) in their performance (Spearman's $R = 0.378$, P-value $= 0.008$, P-value < 0.01). Moreover, the percentage of students in the experimental group who had a decrease in their performance and disagreed that the feedback was easy to read (64.3%) was high compared to students who had no effect (14.3%) or improvement in their performance (21.4%). The results also revealed that 93.8% of students in the experimental group with low background knowledge (pre-test performance < 50) agreed that the formative feedback helped them answer some of the upcoming questions in the tests, compared to 56.3% of students with high background knowledge (Spearman's $R = 0.381$, P-value $=$ 0.007, P-value < 0.01). Moreover, Students in the experimental group with post-test performance below 50 agreed that the formative feedback helped them answersome of the upcoming question in the test while students with post-test performance above 50 disagreed (Spearman's R $=$ 0.358, P-value $=$ 0.013, P-value $<$ 0.05). These results are consistent with the results obtained by Bedford and Price's [59] which showed that students with high performance scores disagreed that the feedback was helpful.

The ontology-based auto-generated feedback was also evaluated by three domain experts (teachers). One domain expert was a computer networks lecturer at the School of Electrical and Electronic Engineering, University of Manchester and the other two domain experts were specialists in Virtual Learning Environments, however, they do not teach courses related to computer networks. The three experts accessed the ontology-based auto-generated tests in Moodle VLE in order to evaluate the auto-generated questions and formative feedback by answering a 5-point Likert scale (1: strongly disagree, to 5: strongly agree) questionnaire. The teachers (three domain experts) were satisfied with the ontology-based auto-generated feedback as they agreed that the feedback was easy to read (the average ranking score is 4.0), useful (the average ranking score is 3.67), and that the OntoPeFeGe provides students with different types of feedback (the average ranking score is 3.67). Moreover, they agreed that the feedback's pedagogical content is reasonable and related to the auto-generated question (the average ranking score is 4.34).

V. CONCLUSION

The work presented in this paper is motivated by the existence of several personalized feedback frameworks which are intradisciplinary, i.e., the different types of feedback are either hard-coded or auto-generated from a restricted set of solutions defined by the teacher or the domain expert [1], [20]. Furthermore, Mason and Bruning's personalized feedback framework [9], which adapts the different types of feedback based on the student and the question characteristics was never evaluated on students even though the question characteristics were considered as important factors in the process of personalizing feedback [1]. Therefore, the primary aim of this paper was to propose a novel, interdisciplinary, generic framework which addresses the aforementioned drawbacks. The framework is called the Ontology-based Personalized Feedback Generator (OntoPeFeGe) and consist of two main components. The first component is the generator which auto-generates questions with different characteristics and associates each question with different types of feedback. The generated questions were evaluated in [47] and proved to have efficient difficulty and discrimination values. The generator presented in this paper associated the auto-generated true/false, multiple choice and short answer questions with five different types of feedback which teachers usually use when providing students with immediate feedback. The five different types of feedback were associated with each question's option. The second component is the personalized feedback algorithm which provide students with appropriate type of feedback after answering an assessment question.

The generated personalized feedback were evaluated by 88 undergraduate students and three domain experts. The results revealed that the personalized feedback improved students' performance significantly. In addition, the results revealed that the students and the domain experts found the ontology-based auto-generated feedback easy to read, useful, and related to the auto-generated questions.

Future work includes applying OntoPeFeGe across several educational fields such as; medicine and engineering. This will help in evaluating the quality of auto-generated feedback and the effect of personalized feedback in several fields. In addition, OntoPeFeGe could be enhanced by integrating additional types of feedback such as hint feedback.

ACKNOWLEDGMENTS

The authors would like to thank students at the University of Manchester who participated in the study, and the domain experts who helped in evaluating the framework.

REFERENCES

[1] S. Narciss *et al.*, "Exploring feedback and student characteristics relevant for personalizing feedback strategies," *Comput. Educ.*, vol. 71, pp. 56–76, Feb. 2014.
[2] J. R. Anderson, A. T. Corbett, K. R. Koedinger, and R. Pelletier, "Cognitive tutors: Lessons learned," *J. Learn. Sci.*, vol. 4, no. 2, pp. 167–207, 1995.
[3] A. Mitrovic, "Fifteen years of constraint-based tutors: What we have achieved and where we are going," *User Model. User-Adapt. Interact.*, vol. 22, nos. 1–2, pp. 39–72, 2012.
[4] M. Demaidi and N. Filer, "Wireless ray tracing educational land," in *Proc. INTED Conf., Int. Assoc. Technol., Educ. Develop. (IATED)*, 2012, pp. 0402–0411.
[5] P. Black and D. Wiliam, "Developing the theory of formative assessment," *Educ. Assessment, Eval. Accountability*, vol. 21, p. 5, Feb. 2009.
[6] E. Brown and C. Glover, "Evaluating written feedback," in *Innovative Assessment in Higher Education*. Oxfordshire, U.K.: Routledge Taylor & Francis Group plc., 2006, pp. 81–91.
[7] M. Price, K. Handley, J. Millar, and B. O'Donovan, "Feedback: All that effort, but what is the effect?" *Assessment Eval. Higher Educ.*, vol. 35, no. 3, pp. 277–289, 2010.
[8] V. J. Shute, "Focus on formative feedback," *Rev. Educ. Res.*, vol. 78, no. 1, pp. 153–189, Mar. 2008.
[9] (2001). *Providing Feedback in Computer-based Instruction: What the Research Tells Us*. [Online]. Available: http://dwb.unl.edu/Edit/MB/MasonBruning.html
[10] E. H. Mory, "Feedback research revisited," in *Handbook of Research on Educational Communications and Technology*, vol. 2. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 2004, pp. 745–783.
[11] K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educ. Psychol.*, vol. 46, no. 4, pp. 197–221, 2011.
[12] S. Narciss and K. Huth, "How to design informative tutoring feedback for multimedia learning," in *Instructional Design for Multimedia Learning*. Münster, Germany: Waxmann, 2004, pp. 181–195.
[13] S. Narciss, "Designing and evaluating tutoring feedback strategies for digital learning environments on the basis of the interactive tutoring feedback model," in *Digital Education Review*, vol. 23. Barcelona, Spain: Univ. Barcelona, 2013, pp. 7–26.
[14] E. Vasilyeva, P. De Bra, M. Pechenizkiy, and S. Puuronen, "Tailoring feedback in online assessment: Influence of learning styles on the feedback preferences and elaborated feedback effectiveness," in *Proc. 8th IEEE Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2008, pp. 834–838.
[15] G. S. Ana and D.-S. J. Macario, "Designing feedback to support language acquisition using the ingenio authoring tool," *Procedia-Social Behav. Sci.*, vol. 1, no. 1, pp. 1239–1243, 2009.

[16] I. Arroyo, J. E. Beck, B. P. Woolf, C. R. Beal, and K. Schultz, "Macroadapting animalwatch to gender and cognitive differences with respect to hint interactivity and symbolism," in *Intelligent Tutoring Systems*. Springer, 2000, pp. 574–583.

[17] I. Arroyo, J. E. Beck, C. R. Beal, R. Wing, and B. P. Woolf, "Analyzing students' response to help provision in an elementary mathematics intelligent tutoring system," in *Proc. Papers AIED-Workshop Help Provision Help Seeking Interact. Learn. Environ.*, 2001, pp. 34–46.

[18] I. Arroyo, B. P. Woolf, D. G. Cooper, W. Burleson, and K. Muldner, "The impact of animated pedagogical agents on girls' and boys' emotions, attitudes, behaviors and learning," in *Proc. 11th IEEE Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2011, pp. 506–510.

[19] B. P. Woolf *et al.*, "The effect of motivational learning companions on low achieving students and students with disabilities," in *Intelligent Tutoring Systems*. Springer, 2010, pp. 327–337.

[20] A. Mitrovic and B. Martin, "Evaluating the effectiveness of feedback in SQL-tutor," in *Proc. Int. Workshop Adv. Learn. Technol.*, Dec. 2000, pp. 143–144.

[21] H. Kazi, P. Haddawy, and S. Suebnukarn, "Employing UMLS for generating hints in a tutoring system for medical problem-based learning," *J. Biomed. Inform.*, vol. 45, no. 3, pp. 557–565, 2012.

[22] A. Mitrović, "Experiences in implementing constraint-based modeling in SQL-Tutor," in *Intelligent Tutoring Systems*. Springer, 1998, pp. 414–423.

[23] H. Passier and J. Jeuring, "Ontology based feedback generation in design-orientated e-learning systems," in *Proc. IADIS Int. Conf. e-Soc.*, Ávila, Spain, 2004, pp. 992–996.

[24] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[25] H. Kazi, P. Haddawy, and S. Suebnukarn, "Leveraging a domain ontology to increase the quality of feedback in an intelligent tutoring system," in *Intelligent Tutoring Systems*. Springer, 2010, pp. 75–84.

[26] H. Kazi, P. Haddawy, and S. Suebnukarn, "Clinical reasoning gains in medical PBL: An UMLS based tutoring system," *J. Intell. Inf. Syst.*, vol. 41, no. 2, pp. 269–284, 2013.

[27] J.-R. Duboc, "Dynamic feedback generation in virtual patients using semantic Web technologies," M.S. thesis, Univ. Southampton, Southampton, U.K., 2013.

[28] F. Frutos-Morales *et al.*, "An extension of the OeLE platform for generating semantic feedback for students and teachers," *Procedia-Social Behav. Sci.*, vol. 2, no. 2, pp. 527–531, 2010.

[29] M. del Mar Sánchez-Vera, J. T. Fernández-Breis, D. Castellanos-Nieves, F. Frutos-Morales, and M. P. Prendes-Espinosa, "Semantic Web technologies for generating feedback in online assessment environments," *Know.-Based Syst.*, vol. 33, pp. 152–165, Sep. 2012.

[30] D. Castellanos-Nieves, J. T. Fernández-Breis, R. Valencia-García, R. Martínez-Béjar, and M. Iniesta-Moreno, "Semantic Web technologies for supporting learning assessment," *Inf. Sci.*, vol. 181, no. 9, pp. 1517–1537, 2011.

[31] E. Gouli, A. Gogoulou, K. A. Papanikolaou, and M. Grigoriadou, "An adaptive feedback framework to support reflection, guiding and tutoring," in *Advances in Web-Based Education: Personalized Learning Environments*. Hershey, PA, USA: IDEA Group, 2006, pp. 178–202.

[32] J.-R. Duboc, M. Weal, H. Davis, and S. Choi, "Automatic feedback generation in virtual patients using semantic Web technologies," in *Proc. Int. Comput. Assist. Assessment*, 2011, pp. 1–11.

[33] A. Papasalouros, K. Kanaris, and K. Kotis, "Automatic generation of multiple choice questions from domain ontologies," in *Proc. e-Learn.*, 2008, pp. 427–434.

[34] A. Papasalouros, K. Kotis, and K. Kanaris, "Automatic generation of tests from domain and multimedia ontologies," *Interact. Learn. Environ.*, vol. 19, no. 1, pp. 5–23, 2011.

[35] M. Cubric and M. Tosic, "Towards automatic generation of e-assessment using semantic Web technologies," *Int. J. e-Assessment*, vol. 1, no. 1, pp. 1–9, 2011.

[36] A. Grubišić, "Adaptive students knowledge acquisition model in elearningsystems Adaprive student's knowledge acquisition model in e-learning systems," M.S. thesis, Univ. Zagreb, Zagreb, Croatia, 2012.

[37] A. Grubišić, S. Stankov, and B. Žitko, "Stereotype student model for an adaptive e-learning system," in *Proc. Int. Conf. Inf. Intell. Syst.*, 2013, pp. 1–8.

[38] M. Al-Yahya, "OntoQue: A question generation engine for educational assesment based on domain ontologies," in *Proc. 11th IEEE Int. Conf. Adv. Learn. Technol. (ICALT)*, Jul. 2011, pp. 393–395.

[39] M. Al-Yahya, "Ontology-based multiple choice question generation," *Sci. World J.*, vol. 2014, Mar. 2014, Art. no. 274949.

[40] B. S. Bloom, *Taxonomy of Educational Objectives: The Classification of Educational Goals*, vol. 1. New York, NY, USA: Longmans, Green, 1956.

[41] D. R. Krathwohl, "A revision of Bloom's taxonomy: An overview," *Theory Pract.*, vol. 41, no. 4, pp. 212–218, 2002.

[42] (2002). *Assessment of Higher-Order Skills*. [Online]. Available: http://www.caacentre.ac.uk/resources/faqs/higher.shtml

[43] R. M. Felder and R. Brent, "Objectively speaking," *Chem. Eng. Educ.*, vol. 31, no. 3, pp. 178–179, 1997.

[44] S. R. Murugan, P. Bala, and G. Aghila, "An ontology for exploring knowledge in computer networks," *Int. J. Comput. Sci. Appl.*, vol. 3, no. 4, pp. 13–21, 2013.

[45] D. Allemang and J. Hendler, *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. Amsterdam, The Netherlands: Elsevier, 2011.

[46] (2012). *W3C XML Schema Definition Language*. [Online]. Available: http://www.w3.org/TR/xmlschema11-2/

[47] M. N. Demaidi, M. M. Gaber, and N. Filer, "Evaluating the quality of the ontology-based auto-generated questions," *Smart Learn. Environ.*, vol. 4, Dec. 2017, Art. no. 7.

[48] R. L. Bangert-Drowns, C.-L. C. Kulik, J. A. Kulik, and M. Morgan, "The instructional effect of feedback in test-like events," *Rev. Educ. Res.*, vol. 61, no. 2, pp. 213–238, 1991.

[49] C. Glover and E. Brown, "Written feedback for students: Too much, too detailed or too incomprehensible to be effective?" *Biosci. Educ.*, vol. 7, no. 1, pp. 1–16, 2006.

[50] G. Rubner, "mbclick—An electronic voting system that returns individual feedback," in *Proc. IEEE 7th Int. Conf. Wireless, Mobile Ubiquitous Technol. Educ. (WMUTE)*, Mar. 2012, pp. 221–222.

[51] *Classroom Response System*. Accessed: Jan. 2012. [Online]. Available: https://mbclick.manchester.ac.uk/

[52] *Java Programming*. Accessed: Jan. 2012. [Online]. Available: http://www.eee.manchester.ac.uk/study/undergraduate/choosingcourse/courses/mte/mechatronic-engineering-4-years-meng/course-unit-spec/?unitcode=EEEN10035

[53] C. Matuszek, J. Cabral, M. J. Witbrock, and J. DeOliveira, "An introduction to the syntax and content of cyc," in *Proc. AAAI Spring Symp., Formalizing Compiling Background Knowl. Appl. Knowl. Represent. Question Answering*, 2006, pp. 44–49.

[54] *Data Networking*. Accessed: Jan. 2013. [Online]. Available: http://www.eee.manchester.ac.uk/study/undergraduate/choosingcourse/courses/eee/electrical-electronicbeng3years/course-unit-spec/?unitcode=EEEN30024

[55] *ComputerNetworks Course*. [Online]. Available: http://studentnet.cs.manchester.ac.uk/ugt/COMP28411/syllabus

[56] (2013). *Computer Networks Course*. [Online]. Available: http://trove.cs.man.ac.uk/moodle/

[57] L. Cohen, L. Manion, and K. Morrison, *Research Methods in Education*. Abingdon, U.K.: Routledge, 2013.

[58] P. Black and D.Wiliam, *Inside the Black Box: Raising Standards Through Classroom Assessment*. Granada, Spain: Learning Sciences, 1998.

[59] S. Bedford and G. Price, "A study into the use of computer aided assessment to enhance formative assessment during the early stages of undergraduate chemistry courses," in *Proc. 11th CAA Int. Comput. Assist. Conf.*, 2007, pp. 57–69.

MONA NABIL DEMAIDI received the B.Sc. degree (Hons.) from An-Najah National University, Palestine, in 2010. Her graduation project there, which received two awards, focused on building a 3-D e-Learning platform. She received the M.Sc. degree from The University of Manchester, Manchester, U.K., in 2011, where her award winning thesis aimed to teach students about ray tracing in a 3-D interactive world. She is currently pursuing the Ph.D. degree with Birmingham City University, U.K. She is also an Assistant Professor with An-Najah National University. Her research interests include developing e-Learning platforms which help students grasp the educational concepts in an interactive manner. She is currently working with Prof. Mohamed who is very experienced in the field of personalized e-Learning systems.

MOHAMED MEDHAT GABER received the Ph.D. degree from Monash University, Australia, in 2006. He then held appointments with the University of Sydney, CSIRO, and Monash University, all in Australia. Prior to joining Birmingham City University, U.K., he was a Reader in computer science with Robert Gordon University, U.K., and with the University of Portsmouth as a Senior Lecturer in computer science, U.K. He is currently a Professor in data analytics with the School of Computing and Digital Technology, Birmingham City University. He has published over 150 papers, co-authored two monograph-style books, and edited/co-edited six books on data mining and knowledge discovery. He is a member of the International Panel of Expert Advisers for the Australasian Data Mining Conferences. In 2007, he received the CSIRO Teamwork Award. His work has attracted well over 3000 citations, with an h-index of 32. He has served in the program committees of major conferences related to data mining, including ICDM, PAKDD, ECML/PKDD, and ICML. He is a fellow of the British Higher Education Academy. He has also co-chaired numerous scientific events on various data mining topics.

NICK FILER received the Ph.D degree in computer science from the University of Manchester, U.K., in 1988. He describes himself as a Generalist rather than a one topic expert. Aside from lots of computer science, he took modules from a few other departments but, mainly electrical engineering. He moved to The University of Manchester in 1981 doing research initially in signal routing for chips and printed circuit boards and then looking at how techniques from artificial intelligence could help electronic systems designers. Until his retirement in 2017, he was a Staff Member with the School of Computer Science at The University of Manchester. He produced several so called expert systems which could solve design problems and attempt to explain decisions to the designer on request. This leads to attempts to improve the quality of the generated explanations, which needed deeper knowledge (semantics) so that the underlying reasons could be elucidated.