Adaptive feedback in computer-based learning environments: a review

Andrew Thomas Bimba¹, Norisma Idris¹, Ahmed Al-Hunaiyyan², Rohana Binti Mahmud¹ and Nor Liyana Bt Mohd Shuib³

Abstract
Adaptive support within a learning environment is useful because most learners have different personal characteristics such as prior knowledge, learning progress, and learning preferences. This study reviews various implementation of adaptive feedback, based on the four adaptation characteristics: means, target, goal, and strategy. This review focuses on 20 different implementations of feedback in a computer-based learning environment, ranging from multimedia web-based intelligent tutoring systems, dialog-based intelligent tutoring systems, web-based intelligent e-learning systems, adaptive hypermedia systems, and adaptive learning environment. The main objective of the review is to compare computer-based learning environments according to their implementation of feedback and to identify open research questions in adaptive feedback implementations. The review resulted in categorizing these feedback implementations based on the students’ information used for providing feedback, the aspect of the domain or pedagogical knowledge that is adapted to provide feedback based on the students’ characteristics, the pedagogical reason for providing feedback, and the steps taken to provide feedback with or without students’ participation. Other information such as the common adaptive feedback means, goals, and implementation techniques are identified. This review reveals a distinct relationship between the characteristics of feedback, features of adaptive feedback, and computer-based learning models. Other information such as the common adaptive feedback means, goals, implementation techniques, and open research questions are identified.

Keywords
Adaptation, learning environment, problem-solving, student modeling, learner model

I. Introduction
The process of learning involves mistakes and errors. In these situations, students often review course materials and search the Internet or other sources to assist them in solving their problems (Ghauth & Abdullah, 2010). Seeking solution is usually time consuming and does not always insinuate a better learning experience. Having a system which generates effective feedback that guides students to the solution can improve the learning process (Muñoz-Merino et al., 2011). Feedback is frequently provided in a typical classroom setting; however, most of the information is poorly received because feedback is presented to groups and so often students do not believe such feedback is relevant to them (Hattie & Gan, 2011). Currently, the gap between students who excel the most and those who excel less is a challenge that teachers, school administrators, and government officials face frequently (Luckin & Holmes, 2016).

Adaptive learning environments provide personalization of the instruction process based on different parameters such as sequence and difficulty of task, type and time of feedback, learning pace, and others (Brusilovsky et al., 1999; Stoyanov & Kirchner, 2004). One of the key features in learning support is the personalization of feedback (Advisors, 2013). Adaptive feedback support within a learning environment is useful because most learners have different personal characteristics such as
prior knowledge, learning progress, and learning preferences. Tailoring feedback according to learner’s characteristics and other external parameters is a promising way to implement adaptation in computer-based learning environment (Narciss et al., 2014). Adaptive feedback unlike generic feedback is dynamic, as learners work through instructions where different learners will receive different information (Le, 2016). Addressing this need, many researchers have proposed various approaches to help students in learning (Farid, Ahmad, & Alam, 2015). As a result, they have identified gaps and have been developing various frameworks and educational systems that are able to analyze student learning and provide adaptive feedback.

The main objective of this review is to compare computer-based learning environments according to their implementation of feedback and to identify major open research questions in adaptive feedback implementations. Not all the implementations selected have adaptive feedback as their main design aim. The reason for our selection is to provide readers an insight to how adaptive feedback is implemented by comparing a wider range of applications.

Previous researchers have conducted reviews of adaptive feedback systems. Le (2016) analyzed the approaches used in developing educational systems for programming and introduced a classification for adaptive feedback supported by these systems. Hepplestone, Holden, Irwin, Parkin, and Thorpe (2011) explored various literature supporting the appropriate use of technology for providing feedback to students. Our current review follows similar methodologies. However, this study reviews various implementation of feedback, based on the four adaptation characteristics: means, target, goal, and strategy (M. E. Specht, 1998). Based on our knowledge, there has not been any review of adaptive feedback implementations according to the four adaptation characteristics. This classification scheme provides an overview of the field. It emphasizes the aspects of the technology, demonstrates open research questions, possible research opportunities, and offers opportunity for researches to identify key characteristics, while implementing adaptive feedback systems. The main objective is to compare feedback implementations according to these adaptation characteristics and identify major open research questions in adaptive feedback implementations.

The structure of the article is as follows: First, the background study on adaptive feedback, explaining the characteristics of adaptation and feedback, is discussed in Section 2. In Section 3, we provide the outline of the review process. The results of the review of adaptive feedback implementations according to the characteristics of adaptation and feedback are discussed in Section 4. We discuss our findings in Section 5. Future directions and conclusion are presented in Sections 6 and 7, respectively.

2. Background

Brusilovsky (1998) defined systems that model student’s learning style, prior knowledge, goals, and preference as adaptive, while those systems which use artificial intelligence (AI) techniques to perform the role of an instructor in tutoring and correcting are referred to as intelligent systems. Learning environments can be either one or a combination of both adaptive and intelligent elements. According to Chieu (2005), there are five main adaptation techniques which are related to the key components of constructive learning environment as follows:

1. Adaptive presentation of learning contents. The course designer should define which learning contents are appropriate to a specific learner at any given time, for example, simpler situations and examples for a novice learner than for an expert.
2. Adaptive use of pedagogical devices. The course designer should define which learning activities are appropriate to a specific learner, for instance, simpler tasks to a novice learner than to an expert.
3. Adaptive communication support. The course designer should identify which peers are appropriate to help a specific learner, for example, learners with more advanced mental models help learners with less advanced ones.
4. Adaptive assessment. The course designer should identify which assessment problems and methods are appropriate to determine the actual performance of a specific learner, for instance, simpler tests for a novice learner than for an expert.
5. Adaptive problem-solving support. The tutor should give appropriate feedback during the problem-solving process of a specific learner, for example, to show the learner his or her own difficulties and provide him or her with the way to overcome those difficulties.

These adaptation techniques rely on a learner model; an essential component which, among other student relevant data, keeps data about the student’s knowledge of the subject domain under study.

Adaptive learning involves multiple disciplines such as Educational Psychology, Cognitive Science, and Artificial Intelligence. This complexity prompted the structuring of research on adaptivity along the methodological questions distinguishing means, target, goal, and strategy (M. E. Specht, 1998):

1. Adaptation means. What information about the learner such as knowledge level, cognitive style, learning style, gender, student’s current activity, previous achievements and difficulties, and misconception is known and used for adaptation?
Table 1. Characteristics of Feedback.

<table>
<thead>
<tr>
<th>Characteristics of feedback</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>Feedback can be provided in relation to the instructional goals and objectives. For example, feedback is provided based on cognitive functions such as promoting information processing, motivational functions such as developing and sustaining persistence or provide correct response.</td>
</tr>
<tr>
<td>Timing</td>
<td>Feedback can be given with respect to timing. It could be in advance, appearing before an action; it could be immediate, appearing immediately after an action or delayed, appearing at a longer time after the action has been made. The feedback is intended to advise, notify, recommend, alert, inform, or motivate the learner about some concerns.</td>
</tr>
<tr>
<td>Scheduling</td>
<td>Feedback can also be made available at scheduled instances. For example, when the learner exceeds a certain time threshold, expertise level, after solving certain questions or after every subtopic.</td>
</tr>
<tr>
<td>Type</td>
<td>There are various feedback types resulting from function, timing, and scheduling. For example, verification feedback, avoidance feedback, correction feedback, informative feedback, cognitive feedback, emotional feedback, scheduled feedback, dynamic feedback, immediate feedback, advanced feedback, delayed feedback, comparative feedback, and isolation feedback.</td>
</tr>
</tbody>
</table>

2. Adaptation target. What aspect of the instructional system (pedagogy and domain model) is adapted based on the learner model?

3. Adaptation goal. What are the pedagogical reasons for the system to adapt to the learner model? Is the system aiding inductive or deductive learning; is the system adapting to a specific instructional method based on the learner model?

4. Adaptation strategy. What are the steps and techniques used to adapt the system to the learner model, and how active or reactive are the learners and system to the adaptation process?

In a learning environment, feedback is seen as the teacher’s (artificial or real) response to the student’s action. There are four main characteristics of feedback: function, timing, schedule, and type (Carter, 1984). Although other researchers (Economides, 2006) suggested other characteristics of feedback, we adhere to the characterization by Carter (1984) because it encompasses all other characteristics. These characteristics are briefly explained in Table 1.

For developing an effective adaptive feedback system, the characteristics of adaptation and feedback have to be taken into consideration. The next section discusses our approach to reviewing implementations of adaptive feedback based on these characteristics.

3. Materials and method

Scientific journals related to learning, computer technology for education, and artificial intelligence in education from five main digital libraries were searched, with the aim of reviewing adaptive feedback implementations based on the characteristics of adaptation and feedback. These libraries include Scopus, Web of Science, IEEE Xplore, Google Scholar, and ACM. The libraries were selected based on their impact evaluation and wide coverage of peer-reviewed journals in multiple academic disciplines. In addition to searching these databases, snowballing technique was used to identify similar implementations of adaptive feedback systems. Only publications from years 2000 to 2016 were collected since most implementations of adaptive feedback in learning environments were realized during this period. To search for potential articles, keywords such as feedback, adaptive feedback, intelligent tutoring system, adaptive learning system, computer-based tutor, pedagogical agents, and computer-assisted learning were used.

For the searched articles, two criteria were considered: (1) publications from year 2000 to year 2016, which indicated evidence of implementation and scientific evaluation of an adaptive feedback system and (2) recent articles with implementations or a clear proposed approach. These criteria allow us to consider publications that have demonstrated practical relevance and also take into account recently developed adaptive feedback systems. We also narrowed down our selection based on three views of adaptive feedback systems. First, adaptive learning systems that provide different information to different learners as they work through instructions and second, adaptive learning systems which generate feedback based on a learner model which distinguishes different learners. Third, we focused on the proposed adaptive feedback frameworks, with practical implementation strategies. Publications that do not fall within this focus area or meet the target criteria were excluded.

A total of 1709 articles were found after searching through the five major digital libraries based on keywords as shown in Figure 1. Using EndNote desktop application (a software tool for managing articles and citation), we eliminated the duplicates and selected the articles that met part of our criteria through relevance sorting. This process resulted in 185 articles excluding the subject descriptive articles which are mentioned in
the introductory parts. Furthermore, we selected 24 eligible articles and added 4 more from snowballing according to 20 different implementations of adaptive feedback. These implementations range from multimedia web-based ITS, dialog-based ITS, web-based intelligent e-learning system, adaptive hypermedia system, theoretical feedback frameworks, and intelligent and adaptive learning environment. The analyzed articles consisted of journal articles, conference proceedings, books, and serials. They were examined based on the publication years, availability, and relevance to the research domain.

4. Results

4.1. Classification of adaptive feedback implementations

A computer-based learning environment represents knowledge in the form of models. The three key models in a computer-based learning environment are the pedagogical model, domain model, and learner model. Research regarding the design and development of adaptive learning environments is highly multi-disciplinary, uniting research from computer science and engineering, psychology and psychotherapy, cybernetics and system dynamics, instructional design, and empirical research on technology enhanced learning (Specht, Kravcik, Klemke, Pesin, & Hüttenhain, 2002). While the educational scientists give attention to development, evaluation, and approval of adaptive instruction algorithms, computer scientists are concerned more with the development of better algorithms, models (pedagogy, domain, and user), and intelligent adaptation. The complexity which arises by the union of these disciplines initiated the need for structuring implementations of adaptivity according to the methodological questions distinguishing means, target, goal, and strategy (Specht et al., 2002).

Similarly, we adopt this methodology as a classification scheme to review different implementations of adaptive feedback in learning environments. Adaptive feedback implementations can be grouped and analyzed based on adaptation methodology and feedback characteristics. Several adaptive learning systems have utilized learner’s characteristics to provide adaptive feedback. In this review, we discuss the following implementations of adaptive feedback.
Wayang Outpost is a multimedia web-based intelligent tutoring system, designed to help students solve mathematics problems. It promotes meaningful and effective ways of learning (Arroyo et al., 2003; Arroyo et al., 2014). Gerdes’ tutor is an interactive functional programming tutor, which supports stepwise development in Haskell programming language (Gerdes, Jeuring, & Heeren, 2012). The E-Tutor, which was developed at Simon Fraser University in Canada, is a web-based intelligent computer-assisted language learning (iCALL) system for beginner to advanced level German grammar exercises. It consists of German grammar concepts and vocabulary tasks, which is used by students in North American Universities (Heifl & Schulze, 2007). AutoTutor is an intelligent tutoring system which uses natural language and adaptive dialog to help students in understanding concepts in Newtonian physics, critical thinking, and computer literacy (D’Mello & Graesser, 2012). The intelligent Teaching Assistant for programming (ITAP) is a data-driven tutoring system that provides personalized help to students while working on code-writing problems (Rivers & Koedinger, 2015).

DeepTutor is another dialog-based intelligent tutoring system that uses scaffolding to improve student’s knowledge during problem-solving (Rus, Niraula, & Banjade, 2015). ACTIVEMATH is a web-based intelligent e-learning system that offers access to various mathematical learning objects, which supports the constructivist learning approach (Melis, Moormann, Ullrich, Gogudze, & Libbrecht, 2007). Guru, on the other hand, is an intelligent tutoring system which consists of exercises in high school biology, supporting conversation with students and virtual instructional materials (Olney et al., 2012). INSPIRE is an adaptive educational hypermedia system which provides meaningful tasks to students, based on their preferred way of learning (Papanikolaou, Grigoriadou, Kornilakis, & Magoulas, 2003). FIT Java Tutor is an intelligent and adaptive learning environment which integrates several pedagogical approaches to assist students in learning Java programming (Gross & Pinkwart, 2015).

ANDES is an intelligent tutoring system which encourages students to construct new knowledge in introductory physics (Gertner & VanLehn, 2000; VanLehn et al., 2005). SQL-Tutor is an intelligent tutoring system which teaches database query language by helping students learn from their mistakes (Mitrovic, 2003; Mitrovic & Ohlsson, 1999; Mitrovic, Ohlsson, & Barrow, 2013). COMPASS uses concept maps as a learning tool which allows students to undertake assessment activities (Gouli, Gogoulou, Papanikolaou, & Grigoriadou, 2006). Excel Tutor is an intelligent novice tutor which provides feedback through error detection and correction skills (Mathan & Koedinger, 2005). The pedagogical motivation for feedback in Excel Tutor is to guide students in error detection and provide an opportunity to reason about the causes and consequences of the errors.

Adaptive feedback frameworks have also been proposed by other researchers. Mason and Bruning’s (2001) theoretical framework enables the creation of feedback based on a variety of conditions such as the complexity of task, student’s prior knowledge, student’s achievement, timing of feedback, and learner control. A conceptual framework for designing informative tutoring feedback forms was put forward by Narciss and Huth (2002). The framework is aimed at deriving general principles for designing informative feedback based on cognitive task and error analysis.

Other mathematics-based intelligent tutoring systems which provide feedback have been proposed. Animalwatch is an ITS which integrates mathematics and biological sciences for teaching arithmetics to elementary school students (Arroyo, Beck, Woolf, Beal, & Schultz, 2000). It builds empirical models of the student’s behavior through an analyses of their interaction with the mathematics tutor. Tsouvalti and Fiedler (2003) used a natural language dialog module to implement a mathematical tutoring system for teaching naive set theory. An integrated learning environment for secondary school mathematics was realized by Bokhove, Koolstra, Boon, and Heck (2007). The aim of the learning environment is to provide students easily accessible practice mathematics problems and intelligent feedback when interacting with the content materials. A tool called web-based authoring tool for Algebra Related domains (WEAR) assist teachers while authoring exercises, monitors students during problem-solving, and provides appropriate feedback (Virvou & Moundridou, 2000). WEAR combines knowledge of authoring algebraic equations which is applicable in other non-mathematical domains and student error diagnosis (Virvou & Moundridou, 2001). It adapts the interaction with students to provide individualized feedback (Moundridou & Virvou, 2002). In the next sections, we concentrate on the key factors that distinguish these systems in their implementation of adaptive feedback.

4.1.1. Adaptive feedback means. An adaptive learning system alters its behavior based on how a learner interacts with it. These alterations are decided based on the learner’s characteristics which are represented in the learner model (Lo, Chan, & Yeh, 2012). It involves the accurate tracking of learner’s activity, monitoring their individual characteristics, and providing timely adaptive feedback according to effective pedagogical principles (Narciss et al., 2014). Tailoring feedback according to learner’s characteristics and other external parameters is a promising way to implement adaptation in computer-based learning environment (Narciss et al., 2014). Adaptive feedback means, poses the question,
What information about the learner is known and used for providing adaptive feedback? These information consist of students’ characteristics such as knowledge level, cognitive style, learning style, and gender.

Wayang Outpost provides adaptive feedback in the form of hints. Two types of hints are provided: (1) a computational and numeric approach to solve a problem and (2) spatial transformations and visual estimations to make the problem easier to solve (Arroyo et al., 2003). The choice of the hint provided by Wayang Outpost is based on the learner’s cognitive profile. The learner profile is built based on an online assessment to determine the learner’s math proficiency which includes accuracy and speed of arithmetic computation and spatial ability. Wayang Outpost provides hints that capitalize on the learner’s cognitive strength when one ability is clearly better than the other. Otherwise, if both skills are low, computational help is provided, whereas if both are high spatial help is provided (Arroyo et al., 2003). Similar to Wayang Outpost, Gerdes’ tutor provides feedback in the form of hints. However, the hints and feedback provided by Gerdes’ tutor does not utilize the characteristics of the learner (such as knowledge level and learning style), instead it is generated automatically from an organized hierarchy according to the syntax tree of the model solution (Gerdes et al., 2012).

The student model in E-Tutor captures the path a student has taken and the underlying source of the error. It then provides instructional feedback based on the students’ prior performance (Heift & Schulze, 2007). The novice, intermediate, and expert are the three types of learners assumed in the student model. These student levels are used to determine the specificity of the feedback provided. In AutoTutor, feedback is provided in the form of a dialog. The dialog moves, pumps, hints, prompts, and assertions are selected based on the students’ knowledge (D’Mello & Graesser, 2012). It constructs a cognitive model of the students’ knowledge level based on the analysis of their typed or spoken responses. ITAP automatically generates feedback in the form of hints. Using the path construction algorithm, it generates hints based on the students’ solution strategy as determined in the solution space (Rivers & Koedinger, 2015). DeepTutor utilizes the students’ knowledge level in order to determine the type and frequency of feedback (Rus et al., 2015). As the knowledge level of the student increases, less amount of feedback is provided.

In ACTIVEMATH, generic computer algebra systems (CASs) are used to diagnose student’s actions in order to provide hints, flag feedback (correct/incorrect), and correct solution (Melis et al., 2007). It does not use any learner characteristics in deciding the type of feedback to be provided. Similarly, Guru provides feedback incrementally based on student’s knowledge level (Olney et al., 2012). INSPIRE supports adaptive navigation support and adaptive presentation of learning content only (Papanikolaou et al., 2003). It does not use any student characteristics in providing feedback. Whenever a learner requires help, examples and hints are provided based on the theory presented (Papanikolaou et al., 2003). The FIT Java Tutor provides feedback based on the students’ structured solution space, which comprises student solution attempts and sample solutions (Gross & Pinkwart, 2015). Andes provides immediate feedback at each stage of problem-solving. The system provides immediate feedback based on the students’ current knowledge or mental state (Gertner & VanLehn, 2000). Feedback in SQL-Tutor is provided based on the number of student’s unsuccessful solution attempts (Mitrovic & Ohlsson, 1999).

Feedback in COMPASS is generally personalized based on identified error in a student’s concept maps, knowledge level, preferences, and interactive behavior (Gouli et al., 2006). Based on the theoretical framework proposed by Mason and Bruning (2001), an effective feedback design should take into consideration the student’s achievement level and prior knowledge. While Narciss and Huth’s (2002) conceptual framework suggests that the necessary information required for providing informative feedback are the student’s learning objectives, prior knowledge, learning strategies, and procedural and meta-cognitive skills. In Excel Tutor, the system just considers the error made by students, it does not take into consideration any other personalized characteristics of the student (Mathan & Koedinger, 2005).

Adaptive feedback in Animalwatch is provided according to the student’s cognitive development and gender (Arroyo et al., 2000). In the process of assisting students in learning naive set theory, Tsouvaltzi and Fiedler (2003) developed a taxonomy of hints which is provided to the students based on their current and previous answers. Unlike the other approaches, Bokhove et al. (2007) do not take into consideration any characteristics of the student while providing feedback. Instead, it provides intelligent feedback based on expert knowledge, common mistakes, and knowledge about the learning domain. However, in WEAR, feedback is provided based on the student’s knowledge level (Virvou & Moundridou, 2000).

4.1.2. Adaptive feedback target. In a computer-based learning environment, the pedagogical model represents the knowledge and expertise of teaching. Specific knowledge represented in the pedagogical model includes effective teaching techniques (deductive and inductive); the various instructional methods (lectures, problem-based learning, inquiry learning, etc.); instructional plan that define phases, roles, and sequence of activities (Scheuer, Loll, Pinkwart, & McLaren, 2010); feedback types, depending on a learner’s action; and assessment to inform and measure learning (Luckin &
Holmes, 2016). The domain model represents knowledge of the subject been learned. It mainly consists of concepts such as how to add, subtract, multiply numbers; Newton’s law of motion; how to structure an argument; and different approaches to reading (Luckin & Holmes, 2016). Adaptive feedback target is involved with the aspect of the instructional system (pedagogy and domain knowledge) that is adapted to provide feedback based on the learner characteristics. Within various instructional methods, there are certain conditions that affect the type of feedback provided.

The hint provided by Wayang Outpost could be in various forms based on three multimedia learning theories which include modality principle, contiguity principle, and animation principle (Arroyo et al., 2014). The modality principle represents words in form of speech, the contiguity principle aligns text to corresponding graphics while animation principles produce an illusion of characters adhering to the basic laws of physics. These principles guide the videos which show how instructors solve maths problems; synchronized sound, animation, and contiguous explanations of maths problem and worked examples. Adaptive feedback in Wayang Outpost is involved with both the pedagogical (multimedia learning theories) and domain models (maths solution in form of worked examples, speech, graphics, and animation), selecting various components of these models based on the learner characteristics (Arroyo et al., 2014). Gerdes’ tutor provides interactive feedback to students (Gerdes et al., 2012). These interactions give hints to students on the next step to take, list of possible ways to proceed, point-out errors, and provide complete worked-out examples. These information are all part of the instructional material, but they are not provided based on the student’s characteristics. Instead, students are provided with an option to choose what type of hint they desire.

Unlike Gerdes’ tutor, E-Tutor alters its instructional feedback within the domain model based on the student’s proficiency as determined in the student model by the percentage of previously correct answers (Heift & Holmes, 2007). AutoTutor provides feedback in the form of dialog. The decision to provide a specific form of feedback, either prompts, hints, an answer or a prompt, depends on the information received by the tutor from the student (Nye, Graesser, & Hu, 2014). ITAP extends the Hint Factory (domain knowledge) by automatically generating hints that are tailored to an individual’s solution to a problem (Rivers & Koedinger, 2015). DeepTutor provides scaffolding and a sequence of progressive hints, based on the student’s knowledge level as articulated in the student model (Rus, Conley, & Graesser, 2014). In ACTIVEMATH, the domain reasoner generates and provides hints, flag feedbacks, and correct solution based on the diagnosis of student’s solution steps (Melis, Goguadze, Libbrecht, & Ullrich, 2009). Depending on the type of error made by a student, the Guru tutor assesses the student’s knowledge and response with a positive feedback, negative feedback, neutral feedback, or elaborative feedback. Even though INSPIRE provides adaptation for navigation support and learning content based on student’s learning style, it does not provide various forms of help according to the student’s learning style.

When there is no information about the quality of a student’s solution (without representative solution), FIT Java Tutor provides feedback in form of self-reflection prompts. At instances where the quality of a student’s solution is partially known, feedback F1, F2, or a combination of F1 and F2 are provided based on previous successes of theses strategies on similar solutions qualities (Gross, Mokbel, Paassen, Hammer, & Pinkwart, 2014). F1 feedback strategy is when the student’s solution differ partially from the correct solution but implements the same problem-solving strategy, the difference is highlighted without showing the actual solution. F2 feedback strategy is when the student’s solution is contrast with the actual solution but the correct problem-solving strategy is used, the solutions are contrasted to allow the student to compare and find the possible mistakes. Andes provides flag feedback accompanied by hints and error messages, which the student can decide to consult when they stuck. These hints are generated using the solution graph in the domain model, based on the state of the student model (Gertner & VanLehn, 2000). In SQL-Tutor, feedback messages are provided as right/wrong, error flag, hints, partial solution, and complete solution (Mitrovic & Ohlsson, 1999). These forms of feedback, which are part of the pedagogical module, are provided to the student based on the number of successful solution attempts.

The process of generating effective feedback in COMPASS depends on the student’s answer during problem-solving (Gouli et al., 2006). In COMPASS, feedback is provided based on an answer categorization scheme. According to the scheme, a student’s answer can be characterized based on completeness, accuracy, and missing out. However, based on (Mason & Bruning, 2001) theoretical framework, less specific feedback is provided as the learning tasks and student’s knowledge level increases. Students with higher achievement levels can benefit more from feedback which provides general information. Thus, allowing them to identify their errors and accurately seek the correct solutions (Mason & Bruning, 2001). In Narciss and Huth’s (2002) conceptual framework, the aspects of the instructional content that affects the type of feedback provided are the instructional goals, learning tasks, issues, and learning problems. This framework recommends a careful alignment of the type of feedback provided and the characteristics of the instructional content.
In *Excel Tutor*, feedback is generated based on an *intelligent novice model*. The diagnostic capabilities of the model supports the provision of context-specific feedback to students (Mathan & Koedinger, 2005). Similarly, in *Animalwatch*, hints are classified based on the *degree of hint symbolism and interactivity* (Arroyo et al., 2000). However, these hints are provided randomly regardless of the student’s interactions. However, a hinting algorithm is used by Tsolaltzis and Fiedler (2003) to evaluate the student’s performance and to provide relevant hint. Different forms of local feedback are provided by Bokhove et al. (2007). The feedback could be a comment regarding the accuracy of a response or an explanation based on the student’s answer. But, in WEAR, there is no clear aspect of the instructional system that is adapted in providing feedback. However, the instructor and student model in WEAR interact with each other to mimic a one-to-one tutoring setting (Moundridou & Virvou, 2002).

### 4.1.3. Adaptive feedback goal

Feedback can serve different purposes based on pedagogical principles or a particular learning theory that is been applied. From an *objectivist* perspective, feedback is regarded as a reinforcement, which is aimed at guiding the learner to progress from a simpler task to a more complex one. The information processing theory suggests that the goal of feedback is not only to reinforce correct answers but also to serve as corrective information to allow learners correct their errors (Hattie & Gan, 2011). *Socioculturalism* considers feedback as a reciprocal dialog, where meaning is reconstructed by peers. The goal of feedback from this view is the consolidation, reorganization, and making knowledge explicit through exchange of ideas between peers (Pryor & Crossouard, 2008). Visible learning theory views feedback in the context of student’s learning (alone, with peers, or adults), at different levels of expertise (novice, proficient, or expert) and level of understanding (surface, deep, and conceptual) (Hattie & Gan, 2011). In an inductive teaching method such as discovery learning, feedback is provided only based on a student’s effort and not as a direct guide for those efforts (Prince & Felder, 2006). In developing adaptive feedback systems, the designer needs to consider the goal of providing feedback. The adaptive feedback goal identifies pedagogical reasons for providing feedback based on the learner model, thus differentiating various implementations. These characteristics revile the function of feedback.

In *Wayang Outpost*, adaptive feedback is provided based on the theory of cognitive apprenticeship, where a master teaches skills to an apprentice. The main aim of this theory is to encourage learners to accomplish more difficult problems than they can accomplish without a guide. Thus, adaptive feedback in *Wayang Outpost* is aimed at providing motivational support and encouraging engagement in the learning process. Help is provided in the form of similar work examples, which enable students to solve similar or harder problems than the current problem (Arroyo et al., 2014). The feedback and hints in *Gerdes*’ tutor are provided based on teacher-specified annotations of solutions (Gerdes et al., 2012). There is no pedagogical principle or learning theory involved. The purpose of providing feedback in *E-Tutor* is based on the language teaching pedagogy (Heift & Schulze, 2007). This learning theory ensures that the amount of feedback provided does not confuse the student (Heift & McFetridge, 1999). *AutoTutor* provides feedback with the goal of stimulating active construction of knowledge based on the *constructivist* principle (D’Mello & Graesser, 2012).

In ITAP, hints are provided as either *point* hint or *bottom-out* hint (Rivers & Koedinger, 2015). The purpose of providing such feedback does not depend on any pedagogical principle or learning theory. *DeepTutor* provides different types of hints which are dynamically sequenced based on a constructivist scaffolding strategy (Rus et al., 2014). Feedback in *ActiveMath* is based on the moderate constructivist approach. It is aimed at providing a reasonable amount of guidance which allows learners to choose and reflect on their work (Melis et al., 2007). Even though, *Guru* provides incremental feedback which is aimed at tailoring conversations based on individual student’s knowledge level, it does not base this feedback on any pedagogical principle or learning theory (Olney et al., 2012).

*INSPIRE* uses the instructional design theory and learning style theory to provide individualized instructions. However, the hints provided by *INSPIRE* only indicate right or wrong, but do not depend on any pedagogical principle or learning theory (Papanikolaou et al., 2003). *FIT Java Tutor* provides feedback which is aimed at guiding students toward self-reflection based on the example-based learning theory (Gross et al., 2014). Feedback in *Andes* is aimed at encouraging constructive learning (Gertner & VanLehn, 2000). Thus, providing little feedback to students unless they request for it. The aim of feedback in *SQL-Tutor* is to promote the acquisition of new cognitive skills, through the state constraint theory (Mitrovic & Ohlsson, 1999). The constraint theory suggests that acquiring new cognitive skills is based on the transfer of knowledge from the evaluative to the generative component. Thus, feedback in *SQL-Tutor* is provided when students submit solutions for evaluation.

The goal of feedback in *COMPASS* is to support reflection by guiding the students to reconstruct their knowledge (Gouli et al., 2006). Mason and Bruning’s (2001) theoretical framework proposes the incorporation of immediate feedback when the goal of instruction is teaching new concepts or the facilitation of concept
acquisition. While delayed feedback should be provided when the instructional aim is to develop higher order skills like abstract reasoning or comprehension (Mason & Bruning, 2001). According to Narciss et al.’s (2014) conceptual framework, feedback can be viewed in the context of self-regulated learning, behavioral, and cognitive learning theories. Based on these theories, feedback can either have a goal of tutoring or guiding the learner, re-enforcement of a concept, or a source of information (Narciss et al., 2014).

The intelligent novice model used in Excel Tutor is aimed at supporting a student in the generative and evaluative aspects of learning a new skill. It explicitly models these skills and guides students through the process of error detection and correction (Mathan & Koedinger, 2005). Tsouvalti and Fiedler’s (2003) natural language dialog module provides hints based on the Socratic tutoring strategy in order to achieve self-explanation (Tsouvalti & Fiedler, 2003). While feedback is provided in Animalwatch (Arroyo et al., 2000), Bokhove et al. (2007), and WEAR (Moundridou & Virvou, 2002), there is no clear pedagogical reason for its provision.

4.1.4. Adaptive feedback strategy. Adaptive feedback strategies combine several feedback components to assist learners in identifying gaps that exist between their current and desired knowledge state (Narciss, 2013). These feedback strategies could be in several forms which include adaptive bottom-up feedback (where detailed feedback is provided and as proficiency increases the feedback changes to general), adaptive top-down feedback (general feedback is provided first, if there is no improvement then a detailed feedback is provided), outcome feedback (indicate right or wrong), hints on how to proceed, and location of mistakes (Billings, 2012). Most of the time, a combination of these strategies are used to ascertain appropriate feedback conditions (Narciss, 2013). In some situations, the learner is given an option to interact with the system and determine the need for feedback. In implementing these strategies, several modeling and artificial intelligence techniques are used. The adaptive feedback strategy looks at the steps taken in providing feedback based on changes in student proficiency, how active or reactive are the learners in the feedback process, and the modeling and artificial intelligence techniques used in implementing adaptive feedback. The implementation of an adaptive feedback strategy determines the timing and scheduling of feedback.

The strategy used by Wayang Outpost is through a step-by-step instruction and guidance to a solution. Adaptive feedback is provided only when the learner requests for help. There is no explicit process for providing feedback as learner’s proficiency increases (Arroyo et al., 2014). Help is provided when a learner has difficulty in one problem, and then a similar problem is provided, encouraging a transfer of knowledge to subsequent problems. The approach used by Wayang Outpost to implement this strategy is a data-centric Bayesian Network which produces a probability model based on student’s previous interaction with the system. Help seeking is modeled to see how hint is related to skills (Arroyo et al., 2014). The Bayesian network has nodes corresponding various hints and skills. Hints in the Gerdes’ tutor is provided in steps (Gerdes et al., 2012). When a student is stuck, they can request for help from the tutor. The tutor provides options based on the annotated teacher-specified feedback. If a choice is made, the student can ask for further details if the first explanation is not clear. Afterward, the tutor responds with more details and a bottom-out hint. To provide a semantically rich feedback, Gerdes’ tutor used techniques such as parsing, rewriting, and program transformation (Gerdes et al., 2012). In E-Tutor, the feedback process is iterative. Student’s errors are identified and communicated one at a time. These iterative processes continue until the student gets the correct answer or decides to submit a solution (Heift, 2010).

Unlike Gerdes’ tutor, E-Tutor does not provide students with a feedback choice. Instead, feedback is generated based on the correlation between the result of the linguistic analysis of a student solution and an error-specific feedback message. A parser and head-driven phase structure grammar (HPSG) are used to determine grammatically incorrect sentences and associate the errors detected with feedback messages (Heift, 2016; Heift & Nicholson, 2001). AutoTutor provides feedback to a student’s initial answer by first providing a short feedback, an elaborative feedback, and then an encouragement (D’Mello & Graesser, 2012). During this process, the student is actively involved in a conversation with the tutor. The latent semantic analysis (LSA) algorithm in AutoTutor is used to determine the information within a student’s response that matches an expectation in the ideal answer, while a subthreshold expectation selection algorithm determines the prescribed sequential order to present expectations (D’Mello & Graesser, 2012). When a student makes an error, ITAP provides hints in two levels. The first level (point hint) informs the student about the type of change required and where the change should be. While the second level (bottom-out hint) provides all the information needed to correct the error (Rivers & Koedinger, 2015). The path construction algorithm is used to generate a chain of hints that leads to a correct solution state.

DeepTutor provides feedback using the tell-or-elicit tactic (Rus et al., 2014). This strategy is based on the scaffolding-modeling-fading theory. DeepTutor implements the theory by eliciting a step, if not comprehended by the student then it tells. This ensures an active participation by the student through a two-way conversation with the tutor. The management of this dialog is implemented using production rules. In
ActiveMath, the domain reasoner generates feedback such as flag feedbacks, correct solution, next step hint, correct input, and number of steps to final solution (Melis et al., 2009). These feedback forms are not presented in any sequence. However, students can request for hints when needed (Melis et al., 2007). The domain reasoner in ActiveMath is implemented using rule-based techniques. The feedback provided by Guru is provided in form of dialog. The response of the tutor on the type of feedback to be provided is based on an assessment of the student’s knowledge. In a case where the student makes an error, the incorrect relationship is highlighted and an explanation is provided for the meaning of the relationship; however, if the student has little background knowledge, an extended direct instruction is provided (Olney, Person, & Graesser, 2011). Guru uses LSA and concept map to align the students’ utterances with the domain and students models to determine if an input is correct or wrong.

In INSPIRE hints and examples are provided to students to indicate right or wrong or on request. However, these feedbacks are provided without any considerations of a specific type or sequence of presentation. The FIT Java Tutor uses a consecutive combination of the F1 and F2 strategy in providing feedback, depending on the learners' needs and progress. With the aim of providing support which is relevant to the student’s needs, FIT Java Tutor provides feedback with varying levels of detail according to the student’s learning progress (Gross et al., 2014). The automated provision of feedback is developed based on clustered solution space. ANDES provides help in a sequence based on three levels. It provides flag feedback in form of a pop up message when the error is likely a slip and not lack of knowledge, and if it is not recognized as a slip, it is highlighted red. The second level is the what’s wrong help, where students can click on a red entry and find out the reason behind the error. Finally, students can request for help when they are not sure of what to do next (VanLehn et al., 2005). During this process, the student is actively involved in selecting the sequence of hints provided. In order to provide immediate feedback, ANDES uses a context-free parser to detect errors in student’s input and a solution graph which contains relevant solution entries. SQL-Tutor postpones feedback until the end of problem-solving steps. At the end of problem-solving, the student is presented with all the errors, but feedback is given for only one error. The feedback is based on the amount of information they provide. They are in five levels: right/wrong, error flag, hint, partial solution, and complete solution. The levels of feedback are provided based on the student’s unsuccessful solution attempts (Mitrovic & Ohlsson, 1999). However, the student can request for a partial or complete solution. Violations in a student’s solution are determined with the aid of constraint-based modeling, relevance, and satisfaction networks.

The steps taken to provide feedback in COMPASS are a gradual provision of various types of feedback based on a four-layered structure and the category of a student's answer (Gouli et al., 2006). Feedback in COMPASS is implemented with the help of concept maps used for identifying student’s errors. Mason and Bruning’s (2001) theoretical framework suggests a different strategy, where the student is provided with a knowledge-of-response feedback and then allowed to decide if they require additional feedback. Mason and Bruning (2001) suggest that this strategy will help to develop the student’s understanding in situations where the correct answer was a guess. A proposed guideline for selecting and specifying different forms of feedback is presented by Nareiss et al.’s (2014) conceptual framework. These guidelines aim at ensuring the student receives the appropriate feedback based on the learning task.

In Excel Tutor, an immediate corrective feedback is provided to the student at the formulas correction step. However, if the student requests for help in correcting the error, the system provides a gradual two-step feedback process. The first step focuses on error detection and the second step involves error correction (Mathan & Koedinger, 2005). Corrective feedback in Excel Tutor is implemented using production rules associated with error free and efficient task performance (Mathan & Koedinger, 2005). Whenever a student enters a wrong answer during a tutoring session in Animalwatch, a hint is provided. The first hint provides little amount of information and if the student keeps providing the wrong answer, the system guides them through the whole process of problem-solving (Arroyo et al., 2000). For implementing feedback in Animalwatch, machine learning techniques, linear regression, and analysis of variance (ANOVA) are used to predict hint effectiveness. Similarly, Tsouvalti and Fiedler’s (2003) natural language dialog module supports a gradual provision of hints based on the number of hints given, the number of wrong answers, and the category of the student’s answer. A combination of ontology and production rules are used to generate feedback in Tsouvalti and Fiedler’s (2003) natural language dialog module.

5. Discussion

In this study, 20 different implementations of adaptive feedback were reviewed and analyzed. These implementations were selected based on their impact and
contribution in computer-based adaptive learning research. We present our analysis on these implementations based on the classification criteria for adaptive feedback, highlighting their levels of adaptive feedback provision. Subsequently, we compared the various implementations of adaptive feedback based on adaptive feedback means, target, goal, and strategy; domain of implementation; adaptive feedback implementation techniques; and evaluation method.

5.1. Adaptive feedback implementation categories

In Figure 2, the adaptive feedback implementations are presented based on the classification scheme discussed in Section 4. We categorized the feedback implementation based on their implementation of adaptive feedback means, target, goal, and strategy. Figure 2 also shows how the adaptive feedback characteristics are aligned to the pedagogical domain and student models of a computer-based learning environment. The adaptive feedback target, strategy, and goal are determined by concepts in the pedagogical model. However, adaptive feedback target and strategy can be implemented using concepts represented in the domain model. Finally, the adaptive feedback means is determined by factors in the student model.

Similarly, there is a relationship between the characteristics of feedback and the features of adaptive feedback. As shown in Figure 2, the implementation of an adaptive feedback strategy determines the timing and scheduling of feedback. The adaptive feedback goal identifies pedagogical reasons for providing feedback based on the learner model. These characteristics reveal the function of feedback. Within various instructional methods, there are certain conditions that affect the type of feedback provided. These reveal a distinct relationship between the characteristics of feedback, features of adaptive feedback, and computer-based learning models (pedagogy, domain, and student models).

5.2. Comparison of adaptive feedback implementations

A detailed comparison of the various implementations of adaptive feedback is presented in Table 2. The main objective is to identify the common ways for implementing adaptive feedback means, target, goal, and strategies; common domains for adaptive feedback implementations; adaptive feedback implementation techniques; and common evaluation techniques. Based on the results of the comparison, the following conclusions were obtained:
<table>
<thead>
<tr>
<th>Feedback implementation</th>
<th>Domain</th>
<th>Adaptive feedback target</th>
<th>Adaptive feedback goal (pedagogical principle)</th>
<th>Adaptive feedback strategy</th>
<th>Implementation technique</th>
<th>Evaluation technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wayang Outpost (Arroyo et al., 2014)</td>
<td>Mathematics</td>
<td>Student’s cognitive profile</td>
<td>Multimedia learning theories (pedagogical knowledge). Worked-examples, speech, graphics, and animation (domain knowledge)</td>
<td>Based on theory of cognitive apprenticeship</td>
<td>Step-by-step instruction. Active learner</td>
<td>Data-centric Bayesian Network</td>
</tr>
<tr>
<td>Gerdes’ Tutor (Gerdes, Jeuring, &amp; Heeren, 2012)</td>
<td>Programming</td>
<td>N/A</td>
<td>N/A (hints and worked examples are provided equally, without considering students’ characteristics)</td>
<td>N/A</td>
<td>Levels of detail. Active learner</td>
<td>Parsing, rewriting, and program transformation</td>
</tr>
<tr>
<td>E-Tutor (Heift &amp; Schulze, 2007)</td>
<td>Language learning</td>
<td>Student’s knowledge level</td>
<td>Instructional feedback (domain knowledge)</td>
<td>Based on language teaching pedagogy</td>
<td>Iterative error detection. Reactive learner</td>
<td>Parser and head-driven phase structure grammar (HPSG) Latent semantic analysis (LSA) and subthreshold expectation selection algorithm Path construction algorithm Production rules</td>
</tr>
<tr>
<td>AutoTutor (D’Mello &amp; Graesser, 2012).</td>
<td>Physics, computer literacy, and critical thinking</td>
<td>Student’s knowledge level</td>
<td>Pump, hints, answers, and prompts (domain knowledge)</td>
<td>Based on constructivist principle</td>
<td>Sequence of feedbacks. Active learner</td>
<td>Latent semantic analysis (LSA) and subthreshold expectation selection algorithm Path construction algorithm Production rules</td>
</tr>
<tr>
<td>ITAP (Rivers &amp; Koedinger, 2015)</td>
<td>Programming</td>
<td>Student’s solution strategy</td>
<td>Hint factory (domain knowledge)</td>
<td>N/A</td>
<td>Levels of details. Active learner</td>
<td>Analysis of log data in the report manager Bystander Turing test. Expert comparison</td>
</tr>
<tr>
<td>DeepTutor (Rus, Niraula, &amp; Banjade, 2015)</td>
<td>Physics</td>
<td>Student’s knowledge level</td>
<td>Scaffolding and sequential hints (domain knowledge)</td>
<td>Based on a constructivist scaffolding strategy</td>
<td>Levels of details. Active learner</td>
<td>Analysis of log data in the report manager Bystander Turing test. Expert comparison</td>
</tr>
<tr>
<td>ActiveMath (Melis, Moormann, Ulrich, Goguadze, &amp; Libbrecht, 2007)</td>
<td>Mathematics</td>
<td>Student’s action</td>
<td>Hints, flag feedbacks, and correct solution (domain knowledge)</td>
<td>Based on moderate constructivist approach</td>
<td>No explicit strategy for presenting feedback. Active learner Level of details. Active learner</td>
<td>Analysis of log data in the report manager Bystander Turing test. Expert comparison</td>
</tr>
<tr>
<td>Guru (Olney et al., 2012)</td>
<td>Biology</td>
<td>Student’s knowledge level</td>
<td>Positive feedback, negative feedback, neutral feedback, or elaborative feedback (domain knowledge)</td>
<td>N/A</td>
<td>Levels of details. Active learner</td>
<td>LSA and concept maps</td>
</tr>
</tbody>
</table>

(continued)
<table>
<thead>
<tr>
<th>Feedback implementation</th>
<th>Domain</th>
<th>Adaptive feedback means</th>
<th>Adaptive feedback target</th>
<th>Adaptive feedback goal (pedagogical principle)</th>
<th>Adaptive feedback strategy</th>
<th>Implementation technique</th>
<th>Evaluation technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>INSPIRE (Papanikolaou, Grigoriadou, Kornilakis, &amp; Magoulas, 2003)</td>
<td>Computer architecture</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Outcome feedback. Active learner</td>
<td>N/A</td>
<td>Questionnaire and data tracking in activity logs</td>
</tr>
<tr>
<td>FIT Java Tutor (Gross &amp; Pinkwart, 2015)</td>
<td>Programming</td>
<td>Student’s solution strategy</td>
<td>Self-reflection prompts, F1 and F2 feedback strategies</td>
<td>Example-based learning theory</td>
<td>Level of detail based on consecutive combination of F1 and F2 strategies. Active learner</td>
<td>Clustered solution space. Relational neural gas (RNG) cluster</td>
<td></td>
</tr>
<tr>
<td>ANDES (VanLehn et al., 2005)</td>
<td>Physics</td>
<td>Student’s mental state</td>
<td>Provides flag feedback accompanied by hints and error messages according to the solution graph (domain model)</td>
<td>Based on constructivist learning theory</td>
<td>Sequence of hints with levels of detail. Active learner</td>
<td>Questionnaire</td>
<td></td>
</tr>
<tr>
<td>SQL-Tutor (Mitrovic, Ohlsson, &amp; Barrow, 2013)</td>
<td>Programming</td>
<td>Student’s unsuccessful solution attempts</td>
<td>Right/wrong, error flag, hints, partial solution, and complete solution (pedagogical module)</td>
<td>Based on state constraint theory</td>
<td>Levels of detail. Active learner</td>
<td>Constraint-based modeling. Relevance and satisfactory networks</td>
<td>Questionnaire</td>
</tr>
<tr>
<td>COMPASS (Gouli, Gogoulou, Papanikolaou, &amp; Grigoriadou, 2006)</td>
<td>Introductory informatics</td>
<td>Student’s knowledge level, preferences, and interactive behavior</td>
<td>Completeness of solution, accuracy, and missing out (domain module)</td>
<td>Support for reflection</td>
<td>Gradual provision of feedback based on four layers. Inactive learner</td>
<td>Concept maps</td>
<td>Empirical studies</td>
</tr>
<tr>
<td>Mason and Bruning (2001)</td>
<td>Domain independent</td>
<td>Student’s achievement level and prior knowledge</td>
<td>Increase difficulty of learning task (domain module)</td>
<td>Support for new concept acquisition and abstract reasoning</td>
<td>Knowledge of response feedback. Active learner</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Narciss and Huth (2002)</td>
<td>Domain independent</td>
<td>Student’s learning objectives, prior knowledge, learning strategies, procedural, and meta-cognitive skills</td>
<td>Instructional goals, learning tasks, issues, and problems (domain module)</td>
<td>Self-regulated learning, behavioral and cognitive learning theories</td>
<td>Knowledge of correct response and elaborative feedback based on specified guidelines. Inactive learner</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Excel Tutor (Mathan &amp; Koedinger, 2005)</td>
<td>Computer science</td>
<td>N/A</td>
<td>Intelligent Novice Model which represents the errors made by students (domain module)</td>
<td>Supports generative and evaluative learning</td>
<td>Immediate corrective feedback, gradual 2-step process. Active Learner</td>
<td>Production rules</td>
<td>ANOVA</td>
</tr>
</tbody>
</table>
Student’s knowledge level is mostly used for deciding what feedback to provide during problem-solving. The forms of feedback in the domain knowledge is the most common aspect of the instructional system that changes based on the student’s characteristics. The goal of providing adaptive feedback is mostly due to the constructivist learning theory. Students are usually involved actively in the feedback process. Adaptive feedback is usually presented with a certain level of detail, based on the student’s initial response. LSA and parsers are the most common techniques used in implementing adaptive feedback. The most common techniques used for evaluating adaptive feedback implementations are through questionnaires, pre-test and post-test, and analysis of log data.

Adaptive feedback seems to be more easily implemented in the programming domain as seen from the full implementation of adaptive feedback features in FIT Java Tutor and SQL-Tutor. This could be as a result of the logical and procedural nature of the programming domain.

6. Future direction

Adaptive feedback support is necessary in a computer-based learning environment because of the difference in students’ characteristics. Based on our review, providing full adaptive feedback is yet to be implemented in non-procedural domains. Further research is required to tackle this issue. Subsequently, there is a need for an adaptive feedback framework which can accommodate the various adaptive feedback criteria presented and support multiple adaptive feedback means, target, goal, and strategy. This will allow for a better evaluation of the following:

- Is there a right combination of adaptive feedback means, target, goal, and strategy which caters for a particular student?
- Can multiple student characteristics be used efficiently for providing efficient feedback?

An area which requires further investigation is the complexity of aligning multiple adaptive feedback characteristics to a specific student’s needs.

7. Conclusion

Feedback is an effective tool used in typical classroom settings during teaching. However, the feedback provided is usually to a group of students with different
characteristics. This results in a gap between students who excel the most and those who excel less. Providing adaptive feedback that caters for students based on their individual characteristics have been implemented in computer-based learning environments with effective results. This review focuses on 20 different implementations of adaptive feedback in computer-based learning environment, ranging from intelligent tutoring system (ITS), multimedia web-based ITS, dialog-based ITS, web-based intelligent e-learning system, adaptive hypermedia system, and intelligent and adaptive learning environment. These implementations were carefully selected based on their impact in providing feedback to students. The main objective of the review is to compare adaptive feedback systems according to feedback adaptation characteristics and identify major open research questions in adaptive feedback implementations.

The review resulted in categorizing these feedback implementations based on the students’ information used for providing feedback (adaptive feedback means), the aspect of the domain or pedagogical knowledge that is adapted to provide feedback based on the students’ characteristics (adaptive feedback target), the pedagogical reason for providing feedback (adaptive feedback goal), and the steps taken to provide feedback with or without students’ participation (adaptive feedback strategy). Other information such as the common adaptive feedback means, goals, and implementation techniques are identified. This review reviles a distinct relationship between the characteristics of feedback, features of adaptive feedback, and computer-based learning models (pedagogy, domain, and student models).

Declaration of Conflicting Interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the University of Malaya Research Grant (RP040B-15AET, 2015).

References


**About the Authors**

**Andrew Thomas Bimba** received a BEng in Electrical and Electronics Engineering in 2006 and a Master’s degree in Computer Science (Artificial Intelligence) in 2014. He is currently a PhD student in Computer Science at University of Malaya. His research interest includes cognitive knowledge base, natural language processing, artificial intelligence in education, machine learning, and computer–human interaction.

**Norisma Idris** received a BSc degree, Master’s degree, and PhD in Computer Science at the University of Malaya. Her area of interest includes artificial intelligence in education (summariization, summary sentence decomposition, heuristic rules, understanding and categorization, essay grading system) and natural language (Malay text processing, text normalization, stemming algorithm). She is currently the head of Artificial Intelligence Department at the University of Malaya.

**Ahmed Al-Hunaiyyan** received a BS in BA in 1983 at Kuwait University, an MS degree in MIS from Aurora University, Illinois, USA, in 1988, and a PhD in Computer Science at Hertfordshire University, UK, in 2000. He has lecturing and training experience in multimedia applications and authoring, database systems and application, management information systems, programming languages, just to mention a few. His research interest includes web-based tutors, multimedia applications, e-learning, human–computer interaction, and knowledge base. He is currently an assistant professor in Computer and Information Systems Department, Public Authority for Applied Education and Training, Kuwait.

**Rohana Binti Mahmud** is a senior lecturer at the Department of Artificial Intelligence, University of Malaya. She received a BSc degree at Waikato University, New Zealand, an MSc degree at Universiti Sains Malaysia, and a PhD at University of Manchester, UK. Her area of expertise includes natural language (Discourse Structure, Lexical Relation, Malay Language Text Processing), expert system (Knowledge Base System, Multi-agent Expert System, Expert Tutoring System), and education (AI in Education, Soft skills, Higher Order Thinking Skills).
Nor Liyana Bt Mohd Shuib is a lecturer at Department of Information System in the Faculty of Computer Science & Information Technology, University of Malaya. She graduated with BCS (Information System) in 2005 from Universiti Teknologi Malaysia and MIT (Information Technology) from Universiti Kebangsaan Malaysia, in 2007. She obtained her PhD in 2013 from Universiti of Malaya, Malaysia. Her research interests include data mining, education technology, and information retrieval tools.