# Feedback Design Patterns for Math Online Learning Systems

PAUL SALVADOR INVENTADO, Department of Computer Science, California State University Fullerton
PETER SCUPELLI, School of Design, Carnegie Mellon University
CRISTINA HEFFERNAN, Mathematical Sciences Department, Worcester Polytechnic Institute
NEIL HEFFERNAN, Computer Science Department, Worcester Polytechnic Institute

Increasingly, computer-based learning systems are used by educators to facilitate learning. Evaluations of several math learning systems show that they result in significant student learning improvements. Feedback provision is one of the key features in math learning systems that contribute to its success. We have recently been uncovering feedback design patterns as part of a larger pattern language for math problems and learning support in online learning systems. In this paper, we present three feedback design patterns developed from the application of the data-driven design pattern methodology on a large educational dataset collected from actual student data in a math online learning system. These design patterns can help teachers, learning designers, and other stakeholders construct effective feedback for interactive learning activities that facilitate student learning.

Categories and Subject Descriptors: **K.3.1 [Computers and Education]**: Computer Users in Education—*Distance learning*; **H.5.2 [Information Interfaces and Presentation]**: User Interfaces—*User-centered design*

General Terms: Design, Human Factors

Additional Key Words and Phrases: design patterns, online learning systems, problem solving, feedback, data driven design pattern production, ASSISTments

**ACM Reference Format:**

Paul Salvador Inventado, Peter Scupelli, Cristina Heffernan, and Neil Heffernan. 2017. Feedback Design Patterns for Math Online Learning Systems. EuroPLoP'17. (July 12-16 2017), 15 pages.

## 1. INTRODUCTION

We define online learning systems as computer-based systems accessible over the internet that help instructors manage teaching resources, deliver content to their students, and facilitate student learning. Examples of such systems include learning management systems, intelligent tutoring systems, and massive open online courses (MOOCs). Such systems can be used to augment classwork and homework in traditional classroom settings, to deliver completely online courses, to manage flipped classrooms, and so forth. Research in online learning systems suggests incorporating interactive learning activities with associated feedback to further improve student learning (Clark & Mayer, 2016). In fact, Koedinger and colleagues (2015) reported that students enrolled in a Psychology MOOC learned about six times more when they additionally engaged in interactive learning activities with associated feedback.

Pedagogical feedback often refers to providing students information about their performance. Educators agree that feedback is important, but it is difficult to design effective feedback. In the same way, designing effective feedback for online learning systems is not trivial because several factors need to be considered such as the learning environment, the subject taught, students' learning history, individual differences, and others. Design patterns have recently been considered to facilitate the selection and application of solutions that address educational challenges

This material is based upon work supported by the National Science Foundation under DRL-1252297.
Author's address: Paul Salvador Inventado, Department of Computer Science, California State University, Fullerton 800 N. State College Blvd. Fullerton, CA 92834; email: pinventado@fullerton.edu; Peter Scupelli, School of Design, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213; email: pgs@andrew.cmu.edu; Cristina Heffernan, Mathematical Sciences Department, Worcester Polytechnic Institute, 100 Institute Road Worcester, MA 01609-2280; email: ch@wpi.edu; Neil Heffernan, Computer Science Department, Worcester Polytechnic Institute, Worcester Polytechnic Institute, 100 Institute Road Worcester, MA 01609-2280; email: nth@wpi.edu

*EuroPLoP'17*, July 12—16, 2017, Irsee, Germany

ACM ISBN 978-1-4503-4848-5/17/07…$15.00
https://doi.org/10.1145/3147704.3147738

in online-learning contexts. One such challenge that these patterns address is the creation of feedback. For example, design patterns have been written to manage MOOCs, to design e-learning content, and to construct feedback for interactive learning activities (Warburton & Mor 2015, Rusman, Lutgens & Ronteltap 2005, Zimmerman, Herding & Bescherer 2014).

This paper contributes three design patterns for constructing feedback to student responses in math online learning systems namely: **Incorrect Example Explanation**, **Common-wrong-answer Feedback**, and **Increasing Hint Specificity**. Instructors, teachers, learning designers, and other stakeholders can use these patterns in developing such systems. The following sections describe prior work in design patterns for online learning systems, the data-driven design pattern production methodology (3D2P) used to develop the patterns discussed in this paper, the pattern language containing these patterns, the design patterns, and future work.

## 2. RELATED WORK

Several design pattern languages and collections have been developed for online learning systems. For example, the e-Len project developed 42 e-learning design patterns, which were categorized into four special interest groups namely: SIG 1: Learning resources and learning management systems (LMS); SIG 2: Lifelong learning, SIG 3: Collaborative learning; and SIG 4: Adaptive learning (Rusman, Lutgens & Ronteltap 2005). Another example is Mor and Warburton's 32 MOOC design patterns that addressed various aspects of MOOC design including participation, community, structure, learning, and orientation (Warburton & Mor 2015). Finally, Mor, Mellar, Warburton, and Winters (2014) compiled 29 design patterns for teaching and learning with technology, which covered learner-centered designs, learning communities, social media and learner interaction in social spaces, and assessment and feedback.

Among these pattern languages and collections, only a few patterns described designs for feedback strategies. A notable example is the **Hint on Demand** design pattern, which suggests giving students the option to request hints so that knowledgeable students have the flexibility to answer problems on their own, while students struggling to answer a problem can get help (Zimmerman, Herding & Bescherer, 2014). Although feedback design patterns outside the online-learning-system domain can resolve online learning system issues, they may need adaptation to address contextual differences. Some examples include **Feedback Sandwich**, **Differentiated Feedback**, and **Peer Feedback**, which are part of Bergin et al.'s (2012) pedagogical design patterns.

## 3. DESIGN PATTERN MINING METHODOLOGY

Figure 1 illustrates a pattern language we have been developing for online learning systems, which include feedback-specific design patterns. A unique feature of the design patterns in this pattern language is that they were developed using the 3D2P methodology (Inventado & Scupelli 2016b). Details about the 3D2P methodology and the design patterns produced using the methodology can be found in Inventado and Scupelli (2015b, 2016a,b,c). Inventado & Scupelli defines 3D2P as:

> ... a four-step iterative process used to uncover design patterns from data collected in a particular domain. 3D2P starts by prospecting data to find interesting relationships in the data. These relationships are investigated further in the pattern-mining step to develop hypotheses based on recurring problems and high-quality solutions uncovered. Literature and experts in the field are consulted to test the validity of the hypotheses. Resulting hypotheses are used to write proposed patterns, which are further refined with the help of the design pattern community through mentoring and pattern writing workshops. Accepted design patterns are evaluated by implementing them in existing systems and evaluating their performance. Randomized controlled trials are conducted to compare the resulting outcome measures (e.g., learning gain, time on task) between applying the design pattern and not applying the design pattern. Results of the evaluation are used to further refine the design pattern as needed.

Currently, our pattern language contains 19 complete design patterns indicated by the solid-lined boxes in Figure 1. Tables 1 and 2 summarize six design patterns that are currently in development and are indicated by the broken-lined boxes in the figure. There are five general design pattern themes namely *Problems*, *Mastery Learning*, *Motivation*, *Personalized Learning*, and *Learning Feedback*. Design patterns under the *Problems* theme address challenges related to the creation of problems in online learning systems. Design patterns under the *Mastery*

*Learning* theme address challenges related to ensuring students' mastery of a concept or skill. The *Motivation* theme contains design patterns that help maintain student motivation while learning. The *Personalized Learning* theme contains design patterns that enable systems to adapt to students' skill levels. The *Learning Feedback* theme contains design patterns that address challenges in generating feedback for students learning through an online learning system. *Hints*, *Examples*, and *Scaffolding* are different strategies to generate feedback, which further split the *Learning Feedback* theme. *Feedback Content* is another subtheme that focuses on the content of the feedback that is utilized by different feedback strategies. The three design patterns described in the next section fall under the *Learning Feedback* theme and are highlighted in blue in Figure 1. These patterns are **Incorrect Example Explanation**, **Common-wrong-answer Feedback**, and **Increasing Hint Specificity**.

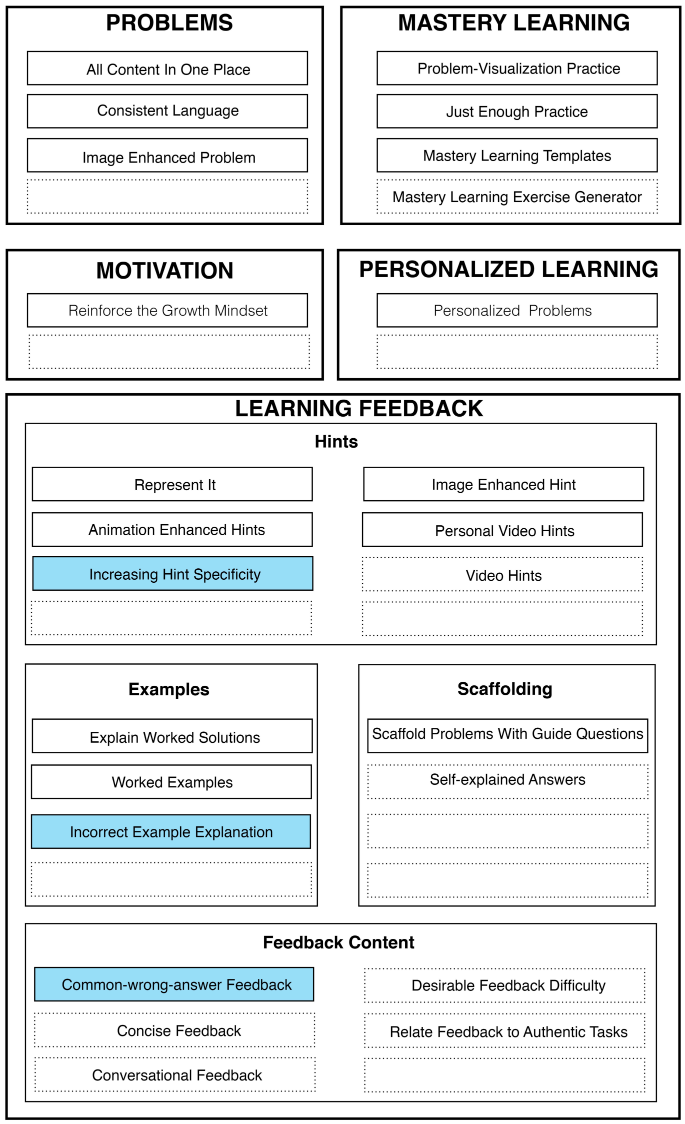



Fig. 1. Pattern Language for Math Problems and Learning Support in Online Learning Systems (Image courtesy of the Learning Environments Lab).

## 4. LIMITATIONS

The design patterns presented in this paper were developed primarily for math because data used for the pattern prospecting and mining steps of the 3D2P methodology were from math problem sets in the ASSISTments online learning system (Heffernan & Heffernan 2014). The data used in the analysis were collected between 2012 and 2013 which contained about 5.8 million student interactions with 179,908 problems. The selected problems covered a range of topics mostly designed for grades six through eight as defined by the Common Core State Standards for

Mathematics (CCSSI 2010). We suspect that the design patterns presented in this paper may apply to other domains, but we opt to limit the scope of these patterns to math until we gather sufficient evidence of their effectiveness in other domains.

## 5. DESIGN PATTERNS

The pattern format used in this paper separates each section with a heading much like other pattern formats (c.f., Carlsson 2004, Dearden & Finlay 2006). It contains the commonly used *context*, *forces*, *problem*, and *solution* sections. The *benefits* section describes how the solution addresses the forces in the problem and the *liabilities* section presents issues that may arise from implementing the solution. The *evidence* section provides theoretical foundations that explain why the problem recurs and why a solution might effectively resolve it. Forces, benefits, liabilities, and evidence are ordered and aligned to facilitate readability. For example, force 1 is addressed by benefit 1 but could result in liability 1, which is supported by evidence 1. The *known uses* section presents successful applications of the design pattern that validate its effectiveness. Finally, the *related patterns* section lists other patterns that the design pattern references or references it.

Table 1 provides summaries of the three design patterns that are presented in the following subsection as well as patterns that are currently under development. We anticipate that the patterns under development will reference the three patterns discussed in this paper. Table 2 provides summaries of design patterns that are referenced by the patterns in this paper.

Table I. Feedback Design Patterns for Math Online Learning Systems

| **Design Pattern** | **Status** | **Summary** |
|---|---|---|
| **Incorrect Example Explanation** | PIP | Ask students to explain incorrect examples to help them understand and avoid common mistakes and misconceptions. |
| **Common-wrong-answer Feedback** | PIP | Identify common wrong answers for a given problem and construct feedback to address the underlying misconception. |
| **Increasing Hint Specificity** | PIP | Allow students to request progressively elaborate hints in which the last hint contains the correct answer. |
| **Mastery Learning Exercise Generator** | UD | Generate and assign problem variations that test a particular skill to help students master that skill. |
| **Video Hints** | UD | Use a video to present feedback that helps students visualize the problem, capture their attention, and minimize their tendency to skip feedback. |
| **Relate Feedback to Authentic Tasks** | UD | Use examples that are based on real-world settings to help students understand the value of the skill taught. |
| **Desirable Feedback Difficulty** | UD | Consider what students already know to provide feedback that will challenge them. |

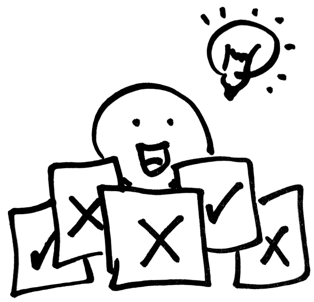
**Note*: PIP - presented in this paper; UD - under development.

Table II. Referenced Design Patterns

| **Design Pattern** | **Status** | **Summary** |
|---|---|---|
| **Pitfall Diagnosis and Prevention** (Anthony 1996) | P | Pay special attention to vital concepts and emphasize them when it has shown that last time you taught the concept students had trouble with it. |
| **Worked Examples** (Inventado & Scupelli 2015b) | P | Provide students with an example similar to the problem they are asked to solve so they understand how to solve the problem without revealing the answer. |
| **Explain Worked Solutions** (Inventado & Scupelli 2016a) | P | Provide students with clearly explained worked solutions when they are unable to answer problems correctly despite receiving support. |
| **Self-explained Answers** | UD | Ask students to explain their answer to ensure they understand it, to help reinforce their understanding, and to encourage them to make generalizations from their solutions. |
| **Concise Feedback** | UD | Avoid extraneous and unnecessarily long feedback explanations so students can focus on the information they need to solve the problem. |
| **Conversational Feedback** | UD | Use a conversational style in communicating ideas with students to make material more engaging. |

**Note: P - published; UD - under development.*

## Incorrect Example Explanation



**Context**: An instructor wants to deepen students' understanding of tricky math concepts or skills. Such topics are tricky because they are complex or they involve special cases that change how a concept is understood or how a skill is applied. The instructor uses an online learning system to assign exercises that can help students expand their knowledge and practice their skills.

**Problem**: Students can understand and apply basic concepts and skills, but they lack knowledge and experience to answer unfamiliar or advanced questions.

**Forces**:
1. Students struggle to identify what makes their answers incorrect.
2. Students are unable to uncover underlying misconceptions that lead to incorrect answers.
3. Many students often commit the same mistakes and share the same misconceptions on target skills.

**Solution**: *Provide students with several related incorrect examples and ask them to explain what and why they are wrong*. Before creating incorrect examples, it helps first to identify common wrong answers to problems that will be included in the math exercise because students are likely to commit the same mistakes. A good source of incorrect examples is previous students' answers. Use the common wrong answers to construct incorrect examples and interleave them with usual problems you would include in an exercise. Ask students to identify what is wrong with the incorrect example and to explain why it is wrong so they can identify features that make it incorrect and uncover the underlying misconceptions that cause the error. Consider asking students to select from a list of possible explanations instead of writing a free-form explanation to simplify the task especially for novice learners. Present several incorrect examples, so students see different error variations and gain more experience. Feedback on students' self-explanation may also facilitate learning.

**Benefits**:
1. By evaluating incorrect examples, students learn to identify features of a solution that make it incorrect, which will help them identify errors in their answers.
2. Students can better identify underlying misconceptions from observing several incorrect examples.
3. Different students can learn from the same incorrect examples, which may address their shared misconceptions and help them avoid making the same mistakes in the future.

**Liabilities**:
1. High-performing students may find the self-explanation of incorrect examples too elaborate, time-consuming, or distracting, which may hinder learning.
2. Content creators will need to design several incorrect examples for each problem.
3. Incorrect examples alone may not describe how to solve the problem. Correct examples are also needed. **Worked Examples** (Inventado & Scupelli 2015b), for example, would complement this pattern well.

**Evidence**:
Ohlsson's (1996) theory of learning from errors suggests that individuals need to learn to detect misconceptions, to identify the features that caused it, and to explain what additional conditions or features will make it correct.

Using incorrect examples, correct examples, and self-explanation during practice can improve student learning (Booth et al., 2013, Durkin & Rittle-Johnson 2012, Hang, Liu & Shiu 2008). Specifically, self-explanation of correct examples may facilitate learning because it forces students to make their knowledge explicit (Chi 2000, Roy & Chi 2005) and self-explanation of incorrect answers can help students identify features that make the solution incorrect and recognize their misconceptions (Siegler 2002). Separate correct and incorrect examples or both correct and incorrect examples together may be interspersed with practice problems. Both strategies are shown to be effective (Durkin & Rittle-Johnson 2012, Booth, Lange, Koedinger, and Newton 2013).

Students often make the same types of mistakes when they answer math problems, which are called error patterns, or common wrong answers that dates back to the work of Radatz (1979). Several research studies have been conducted to address students' common errors (Peng & Luo 2009, Shulman 1986).

**Known Uses**:
**Incorrect Example Explanations** have been applied in different learning contexts. In ASSISTments for example, it is easy for content creators to construct and intersperse incorrect examples in a problem set. Figure 2 shows an example of an incorrect example explanation problem. Students are asked to choose the statement that explains why the answer was incorrect. The top left side of the image shows that the student has already answered a practice problem before answering the incorrect explanation problem. The design of this problem set is being finalized before it will be deployed to ASSISTments users. Student performance in answering this problem set will be evaluated when enough data is collected.

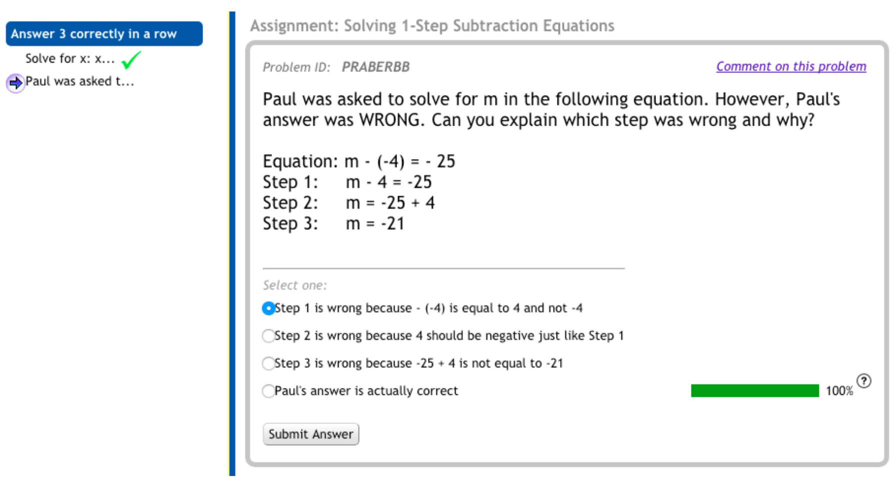



Fig. 2. Screenshot of an incorrect example explanation problem used in an ASSISTments problem set (Image courtesy of ASSISTments https://assistments.org).

The work of Booth, Lange, Koedinger, and Newton (2013) shows an example of incorrect examples and self-explanation that was deployed in the Algebra 1 Cognitive Tutor system. Students using the system answered guided practice problems for two-step equations in Algebra with interspersed correct and incorrect examples. Their methodology involved asking students to self-explain correct and incorrect examples. Unlike usual self-explanation questions, students used menu options to select what they thought was done in the example (e.g., add, subtract, multiply, divide), and why the step was correct or incorrect (e.g., "It was illegal because it combined terms that were not like terms"; "It was legal but not helpful because it did not reduce the number of terms"). Practice problems that were interleaved within the same exercise involved students solving problems that were automatically checked by the system. Students received correctness and explanatory feedback when they submitted their answers and could request hints to get help. Booth et al. also conducted experiments to compare student performance when exposed to (1) practice only, (2) correct examples with self-explanation and practice, (3) incorrect examples with self-explanation and practice, or (4) correct examples and incorrect examples with self-explanation, and practice.

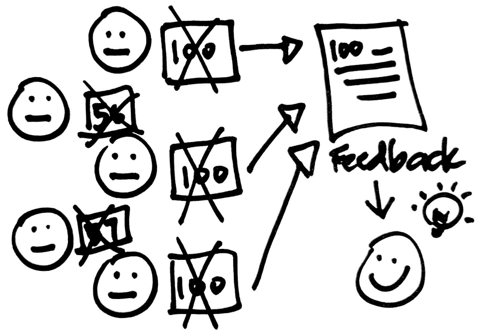
The results of the study showed that students exposed to practice, correct and incorrect examples, and self-explanation (i.e., condition 4) outperformed the other three conditions.

Huang, Liu, and Shiu (2008) implemented a computer-aided system for learning decimal concepts. Students using the system answered decimal problem exercises that asked them to re-evaluate their incorrect answer to help them figure out their mistakes and understand concepts more effectively. For example, after giving an incorrect answer, the system may ask "Does the "4" of "5.4" pancakes mean there are "four" pancakes?" so that students can focus on the meaning of the decimal value. Huang et al. ran a study to compare students' performance in pre-, post-, and delayed posttests when they either learned from incorrect examples through the computer-aided system (experimental condition) or from answering test-sheet questions for practice without access to incorrect examples (control condition). The results of the experiment showed that students exposed to incorrect examples performed significantly better in immediate and delayed posttests compared to students who only engaged in practice.

Durkin and Rittle-Johnson (2012) introduced both correct and incorrect examples and self-explanation questions in decimal-magnitude practice problems that students answered. In their case however, students used pen and paper and not a learning system. Nevertheless, results from their experiment aligned with other research showing the value of incorrect examples. Specifically, students in an experimental condition were given pairs of correct and incorrect examples for the same problem. For each pair, they were asked to explain why an example was correct or incorrect, how the two examples were similar or different, and how they would teach another student to solve the problem. After studying three example pairs, students were asked to answer a practice problem. This process was repeated four times so that students worked on 12 example pairs and four interleaved practice problems. In the control condition, students were only shown correct examples instead of an example pair but were exposed to the same methodology of studying 12 examples and answering four interleaved practice problems. Students from both conditions were given pre-, post-, and delayed posttests, which showed that students exposed to both correct and incorrect examples performed significantly better than students exposed to only correct examples.

**Related Patterns**: Common wrong answers to a particular problem can be used to construct incorrect examples, which may help prevent students from performing similar mistakes in the future. It is a good idea to **Use Student Solutions** (Köppe et al. 2015) to find incorrect examples because students are more likely to commit similar mistakes. Such a case implements **Pitfall Diagnosis and Prevention** (Anthony 1996). **Incorrect Example Explanation** can also be used in conjunction with **Worked Examples** (Inventado & Scupelli 2015b) and **Self-explained Answers** to support further student learning.

**Common-wrong-answer Feedback**




**Context:** A content creator designs a math exercise in an online learning system. It is a good idea to provide students with feedback, so they understand concepts better and learn skills properly.

**Problem**: It is difficult to create feedback that addresses each student's incorrect answer to a math problem.

**Forces**:

1. Students who struggle to answer a problem may be unable to solve it unless they receive help.
2. Students often share the same mistakes and underlying misconceptions that result in common wrong answers to given problems.
3. It usually takes time and effort to encode feedback for a math problem in an online learning system.

**Solution**: *Construct feedback that addresses students' common wrong answers to a given problem*. Common wrong answers need to be identified first before feedback can be created. Common wrong answers may be specific incorrect values associated with a particular problem, or an incorrect procedure for solving a problem sometimes referred to as buggy rules (Brown & Burton 1978, Sleeman 1982). Experts can be consulted to identify common wrong answers and buggy rules for a given problem based on their experience. Alternatively, content creators may create problems sets in an online learning system to collect initial student data that will identify common wrong answers. Once common wrong answers are identified, static explanatory feedback can be designed for corresponding incorrect answers to a particular math problem. Such feedback can then be presented to students when they submit a specific incorrect answer. In the case of buggy rules, explanatory-feedback templates can be designed so that specific incorrect answers can be merged with the template and presented to the student when they submit an answer that violates a buggy rule. Consider, for example, a math problem that asks "Imagine that 2 out of three 3 balls are green. What is the percentage of green balls from the set?" When the student submits a particular answer like 1.5, that answer can be checked against a mapping of incorrect answers and buggy rules. In this case, the associated buggy rule may be: *dividing the second value by the first value*; a feedback template may be: "*Are you sure you should divide <val2> by <val1>?"*; and the merged feedback presented may be: *"Are you sure you should divide 3 by 2?"*

**Benefits**:

1. Students receive feedback based on their incorrect answer that may help them solve the problem.
2. Common wrong answers can capture students' common mistakes and misconceptions, which are addressed by the feedback specifically designed to help resolve them.
3. Content creators only need to create feedback for common wrong answers and buggy rules instead of each possible wrong answer for each question.

**Liabilities**:

1. The system may be unable to provide feedback for uncommon wrong answers unless default feedback is provided, which can be less effective.
2. Content creators may need to consult experts constantly to identify and address common wrong answers.

3. The learning system will need to support functionalities that allow the provision of associated feedback to incorrect answers. Also, submitting the same incorrect answer will provide the same explanatory feedback, which might not be helpful for the student.

**Evidence**:
Students who struggle to understand a concept or acquire a skill may learn to do so with appropriate feedback and guidance (Vygotsky 1962). Human tutors often provide feedback based on students' incorrect answers to address their underlying misconceptions (Graesser et al. 1999, Person et al. 2003, Hume et al. 1996). Students often share similar misconceptions that lead them to commit similar mistakes (Brown & Burton 1978, Sleeman 1984).

The development of learning environments is often expensive and time-consuming (Murray et al. 2003). Students' errors need to be analyzed to understand their underlying misconceptions and provide appropriate feedback (Peng & Luo 2009, Shulman 1986). Learning systems have used buggy rules to assign appropriate feedback to common misconceptions (Brown & Burton 1978, Sleeman 1984).

**Known Uses**:
Tutoring systems are designed to provide automated support for students learning in varied learning settings. It is difficult to manually create feedback for every learning setting, so it makes sense to utilize mechanisms that generalize over commonly occurring student errors.

ASSISTments is an example of such online learning systems that allow teachers to easily construct and assign problem sets to their students, to automatically evaluate students' performance, and to generate reports that identify common misconceptions, which may need to be further discussed in class (Heffernan & Heffernan 2014). ASSISTments collects data from students' interactions with problems, which allows it to identify common wrong answers associated with the problem. Teachers and content creators can use this data to design bug messages that are shown to students whenever they submit such common wrong answers for a given problem. Figure 3 shows a screenshot of a student answering a math problem in ASSISTments and Figure 4 shows a screenshot of ASSISTments' authoring tool that displays the common wrong answers associated with the problem. Figure 4 shows three bug messages that will be shown to students who answer 2, 3, or 4, which are common wrong answers associated with the problem. The light blue box below the problem in Figure 4 shows the associated bug message shown to a student who entered the common wrong answer, 3. Although no specific experiments have been conducted to evaluate the effectiveness of common-wrong-answer feedback in ASSISTments, several experiments show that its design (including the use of common wrong answers) lead to significant learning gains in real classroom settings (Roschelle et al. 2016).

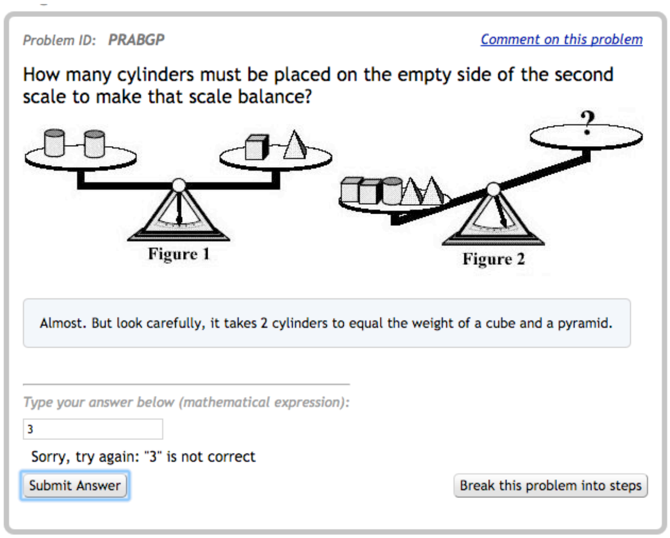



Fig. 3. Screenshot showing a student's screen when a common wrong answer was submitted. The light blue box is the bug message associated with the common wrong answer, 3 (Image courtesy of ASSISTments https://assistments.org).

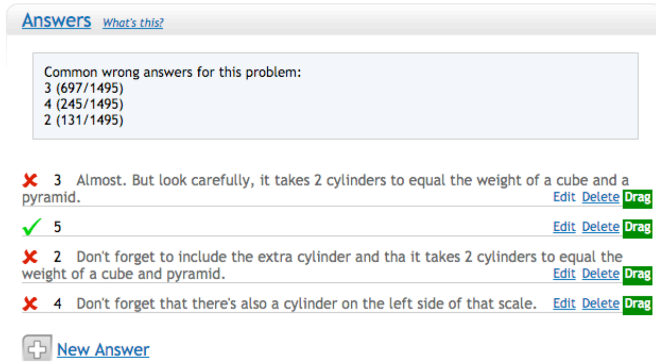


Fig. 4. Screenshot of the ASSISTments authoring tool showing the common wrong answers associated with a particular problem and the bug messages that will be shown when a particular answer is submitted by the student (Image courtesy of ASSISTments https://assistments.org).

Geometry Tutor (Anderson, Boyle, & Yost 1985) is a tutoring system that also utilizes common wrong answers to provide automated feedback. Specifically, a set of ideal and buggy rules (IBR) was developed from theoretical analysis and empirical observations of student behavior to provide the system with domain knowledge. These rules were used by the system to identify students' misconceptions on a particular geometry problem and to provide the corresponding message associated with the rule. When the student submitted an answer that matched a buggy rule such as $\angle AMF \cong \angle BFE$, for example, the system would provide the following feedback "No, it is not useful to make that vertical inference here. It is useful to make the vertical angle inference when the angles are corresponding parts of triangles you want to prove congruent. In this problem, why don't you try to make involving the fact that M is the midpoint of AB". Experiments conducted on Geometry Tutor showed a large positive impact on student learning for using the tutor in class (Anderson et al. 1995).

The Practical Algebra Tutor (PAT) is another tutoring system that utilized buggy rules to detect students' incorrect answers and to provide associated feedback that addressed their misconceptions (Koedinger et al. 1997). PAT focused on helping students apply basic algebra and reasoning skills in day-to-day situations such as checking the amount of a paycheck, estimating the cost of a rental car for a trip, and choosing between long-distance telephone services. Koedinger et al.'s experiments showed significant learning gains in real classroom settings for students who used the tutor compared to those who did not.

**Related Patterns**: **Image-enhanced Hints** (Inventado & Scupelli 2016b), **Concise Feedback**, and **Conversational Feedback** can be used to help construct effective **Common-wrong-answer Feedback**.

**Increasing Hint Specificity**


**Context:** A content creator designs hints for a problem in a math exercise published on an online learning system.

**Problem**: It is difficult to construct appropriate hints for students with different levels of background knowledge.

**Forces**:
1. Students need hints to recall or clarify key concepts that hinder them from answering problems correctly.
2. Students differ in the amount of information they need to figure out the answer.
3. Providing too little information may not be enough to help the student identify the answer.
4. Providing too much information may reveal the answer too quickly.

**Solution**: *Provide students with a sequence of hints that they can request progressively and that begins with general hints and increases in specificity*. A good strategy for designing a hint sequence is starting with the solution to the problem and breaking it into the individual steps of the process. **Explain Worked Solutions** (Inventado & Scupelli 2016a), for example, can guide the construction of the solution because it contains information needed to explain each step. Construct feedback for each step so that it provides enough information to guide students toward the next step, but does not give away answers for the current or succeeding step. **Concise Feedback** may help clarify the content that will be included in the feedback for each step. It is common for the final step to reveal the answer to the problem, often called the bottom-out hint, so that students do not get stuck answering the same problem and are allowed to move on to the next problem in the exercise.

**Benefits**:
1. Reading through hints in the sequence allows students to recall and clarify key concepts they need to solve the problem.
2. The system provides appropriate help for each step in the sequence, which the student can request progressively as necessary.
3. Students could request more hints if the hints they received were unable to help them solve the problem.
4. Hints presented earlier in the sequence reveal less information that could give away the answer; students can only access more specific information when they request it.

**Liabilities**:
1. Students need to go through each hint in the sequence to find relevant information. However, this process can be tedious and can potentially bore or frustrate students.
2. Students do not always seek help even if they need assistance and when they do seek help, they may not use it effectively.
3. Content creators need to construct appropriate hints for each step in the sequence, which is difficult and time-consuming.
4. Students can “game” the system by revealing the bottom-out hint even if they did not use prior hints productively to help them solve the problem.

**Evidence**:

1. According to the Zone of Proximal Development, expert guidance can help students achieve difficult tasks that they are not capable of completing on their own (Vygotsky 1962).
2. Human tutors often tailor their feedback according to students' prior actions and their perception of what the student knows or does not know (Person et al. 2003). The challenge of help-seeking facilities in online learning systems is that students who need assistance often fail to seek help and those who do seek help do not use such functionalities effectively (Puustinen, 1998, Ryan et al., 1998).
3. Students perform better when they request hints possibly because they receive timely help compared to proactive help, which could be distracting or annoying (Razzaq & Heffernan 2010).
4. Tutors often help students who struggle to learn a concept, procedure, or skill by starting with hints that are as far away from the sought-after answer and provide more specific help until they can understand it (Hume et al. 1996, Wood & Wood 1999). Unfortunately, students have also been shown to use help unproductively due to several reasons including dislike of the subject matter, the learning environment, or computers in general, lack of educational self-drive, low self-efficacy, and poorly designed help (Baker et al. 2008).

**Known Uses**:
Most tutoring systems employ **Increasing Hint Specificity** because it mimics a strategy commonly used by human tutors. For example, system developers, learning designers, and teachers design and implement hint sequences for each problem in the ASSISTments online learning system (Heffernan & Heffernan 2014). Most problems in ASSISTments are for math, but it also contains problems for other domains like English, chemistry, and physics. The system does not require content creators to construct hint sequences in increasing specificity, but it is a common practice they follow in ASSISTments. Students who answer ASSISTments problems with assigned hint sequences can access hints using a *"Show Hint"* button. Every time the button is clicked associated hints in the sequence are progressively revealed. All hints are left on the screen so students can easily review them. Figure 5 shows a screenshot of a problem in ASSISTments wherein the student has already requested two out of three hints in the sequence. Clicking on the button again will reveal the correct answer to the problem, which is 60.9.

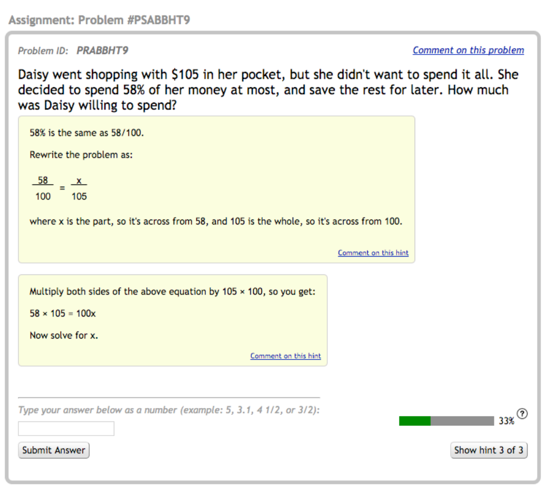



Fig. 5. Screenshot of an ASSISTments problem where a student has requested two out of the three hints available for the problem (Image courtesy of ASSISTments https://assistments.org).

Algebra Cognitive Tutor is another example of a tutoring system that allows students to request multiple levels of hints to help them solve algebra problems (Koedinger & Aleven 2007). Students can progressively request hints, which provide more specific advice based on the solution strategy they used. This feature means that the hint content is specific to the current state of students' answers, which make hints more appropriate. Unfortunately, there is limited information on how content creators conceptualized hints in this research, but we know they encoded them.

Geometry Cognitive Tutor is another system developed by largely the same group of researchers who developed Algebra Cognitive Tutor. It uses a similar hinting strategy to support students who are learning Geometry (Roll et al.

2011). There is active research in the Cognitive Tutor family of tutoring systems to further improve student learning such as helping students develop better help-seeking skills (Roll et al. 2011).

As of writing this paper, we were unable to find experiments that evaluated the effectiveness of using progressive help in any of these three systems. However, experiments that compared student performance with and without using the system have consistently reported improved learning gains (Koedinger & Aleven 2007, Roll et al. 2011, Roschelle et al. 2016).

**Related Patterns**: **Hint on Demand** (Zimmerman, Herding & Bescherer, 2014) is often used in tandem with **Increasing Hint Specificity** because it results in better student learning compared to proactively providing hints. **Explain Worked Solutions** (Inventado & Scupelli 2016a) involves the identification of solution steps that may also be used to craft individual hints in the hint sequence to provide **Increasing Hint Specificity**. **Image-enhanced Hints** (Inventado & Scupelli 2016b), **Concise Feedback**, and **Conversational Feedback** can be used to ensure the quality of each hint.

## 6. SUMMARY AND NEXT STEPS

The paper discussed three design patterns for constructing feedback design patterns for math online learning systems namely **Incorrect Example Explanation**, **Common-wrong-answer Feedback**, and **Increasing Hint Specificity**. Online learning system developers, content creators, and teachers can use these patterns to guide the creation of feedback while ensuring its effectiveness in facilitating student learning.

We plan to apply and test the effectiveness of our design patterns in other domains such as physics, chemistry, computer programming, and so forth. Similarly, it would be interesting to evaluate how well these patterns can translate to other learning environments like traditional classrooms. Design patterns that are found to be effective in other domains or learning environments may be generalized, and those that are not might lead to the development of new patterns that adapt to the specific constraints of the domain or environment.

The 3D2P methodology is currently being used on data collected from the ASSISTments online learning system to uncover more patterns that will be part of the Pattern Language for Math problems and Learning Support in Online Learning Systems. The design patterns are being compiled in an online design pattern repository (http://learningenvironmentslab.org/openpatternrepository) and work is being done to foster collaboration between design pattern authors, domain experts, and design pattern users to continue writing, evaluating, and refining design patterns.

## 7. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under DRL-1252297. We would like to thank our shepherd Elissaveta Gourova for her invaluable advice and feedback on this work and the participants of the writing workshop at EuroPLoP 2017 who helped us refine and improve our paper. We also thank Sharris Francisco-Inventado for her help with the visualization of the Pattern Language for Math problems and Learning Support in Online Learning Systems.

### REFERENCES

Anderson, J. R., Boyle, C. F., & Yost, G. 1985. The geometry tutor. In IJCAI (pp. 1-7).

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. 1995. Cognitive tutors: Lessons learned. The journal of the Learning Sciences, 4(2), 167-207.

Anthony, D.L. 1996. Patterns for classroom education. In John Vlissides, James Coplien, and Norman Kerth (Eds.) Pattern Languages of Program Design 2. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 391–406.

Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. 2008. Why students engage in" gaming the system" behavior in interactive learning environments. Journal of Interactive Learning Research, 19(2), 185.

Bergin, J., Sharp, H., Chandler, J., Sipos, M., Eckstein, J., Völter, M., Manns, M. L., Wallingford, E., Marquardt, K. 2012. Pedagogical patterns: advice for educators. Joseph Bergin Software Tools; New Paltz, NY.

Booth, J. L., Lange, K. E., Koedinger, K. R., & Newton, K. J.2013. Using example problems to improve student learning in algebra: Differentiating between correct and incorrect examples. Learning and Instruction, 25, 24-34.

Brown, J. S., & Burton, R. R. 1978. Diagnostic models for procedural bugs in basic mathematical skills. Cognitive science, 2(2), 155-192.

Carlsson, D. 2004. A categorization of HCI patterns. Department of Computing Science, Umea University, Sweden.

Common Core State Standards Initiative. 2010. Common Core State Standards for Mathematics. Washington, DC: National Governors Association Center for Best Practices and the Council of Chief State School Officers.

Chi, M.T.H. 2000. Self-explaining expository texts: The dual processes of generating inferences and repairing mental models. In Glaser, R. (Ed.) Advances in Instructional Psychology, Mahwah, NJ: Lawrence Erlbaum Associates, pp. 161-238

Clark, R. C., & Mayer, R. E. 2016. E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning. John Wiley & Sons.

Durkin, K., & Rittle-Johnson, B. 2012. The effectiveness of using incorrect examples to support learning about decimal magnitude. Learning and Instruction, 22(3), 206-214.

Graesser, A. C., Wiemer-Hastings, K., Wiemer-Hastings, P., Kreuz, R., & Tutoring Research Group. 1999. AutoTutor: A simulation of a human tutor. Cognitive Systems Research, 1(1), 35-51.

Heffernan, N. T., & Heffernan, C. L. 2014. The ASSISTments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. International Journal of Artificial Intelligence in Education, 24(4), 470-497.

Huang, T. H., Liu, Y. C., & Shiu, C. Y. 2008. Construction of an online learning system for decimal numbers through the use of cognitive conflict strategy. Computers & Education, 50, 61-76.

Hume, G., Michael, J., Rovick, A., & Evens, M. 1996. Hinting as a tactic in one-on-one tutoring. The Journal of the Learning Sciences, 5(1), 23-47.

Inventado, P.S. & Scupelli, P. 2016a. Design Patterns for Math Problems and Learning Support in Online Learning Systems. In Proceedings of the 11th Viking Conference on Pattern Languages of Programs (VikingPLoP 2016), ACM, New York, NY, USA.

Inventado, P.S. & Scupelli, P. 2016b. Design Patterns for Helping Students to Learn to Visualize Math Problems in Online Learning Systems. In Proceedings of the 21st European Conference on Pattern Languages of Programs (EuroPLoP '16).

Inventado, P.S. & Scupelli, P. 2016c. Learning-Support Design Patterns for Student Interactions with Math Problems. In Proceedings of the 23rd Conference on Pattern Languages of Programs (PLoP'16).

Inventado, P.S. & Scupelli, P. 2015a. Towards an open, collaborative repository for online learning system design patterns. eLearning Papers, 42(Design Patterns for Open Online Teaching):14-27.

Inventado, P.S. & Scupelli, P. 2015b. A Data-driven Methodology for Producing Online Learning System Design Patterns. In Proceedings of the 22nd Conference on Pattern Languages of Programs (PLoP 2015).

Koedinger, K. R., & Aleven, V. 2007. Exploring the assistance dilemma in experiments with cognitive tutors. Educational Psychology Review, 19(3), 239-264.

Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. 1997. Intelligent Tutoring Goes To School in the Big City. International Journal of Artificial Intelligence in Education, 8, 30-43.

Koedinger, K. R., Kim, J., Jia, J. Z., McLaughlin, E. A., & Bier, N. L. 2015. Learning is not a spectator sport: Doing is better than watching for learning from a MOOC. In Proceedings of the Second (2015) ACM Conference on Learning@ Scale (pp. 111-120). ACM.

Köppe, C., Niels, R., Holwerda, R., Tijsma, L., Van Diepen, N., Van Turnhout, K., & Bakker, R. 2015. Flipped classroom patterns: designing valuable in-class meetings. In Proceedings of the 20th European Conference on Pattern Languages of Programs (EuroPLoP 2015). New York:ACM.

Mor, Y., Mellar, H., Warburton, S., and Winters, N. (Eds.). 2014. Practical design patterns for teaching and learning with technology. Rotterdam, Netherlands: Sense Publishers.

Murray, T., Blessing, S., & Ainsworth, S. 2003. Authoring tools for advanced technology learning environments: Toward cost-effective adaptive, interactive and intelligent educational software. Springer Science & Business Media.

Ohlsson, S. 1996. Learning from error and the design of task environments. International Journal of Educational Research, 25(5), 419-448.

Peng, A., & Luo, Z. 2009. A framework for examining mathematics teacher knowledge as used in error analysis. For the learning of mathematics, 29(3), 22-25.

Person, N. K., Graesser, A. C., Kreuz, R. J., & Pomeroy, V. 2003. Simulating human tutor dialog moves in AutoTutor. International Journal of Artificial Intelligence in Education (IJAIED), 12, 23-39.

Puustinen, M. 1998. Help-seeking behavior in a problem-solving situation: Development of self-regulation. European Journal of Psychology of education, 13(2), 271-282.

Radatz, H. 1979. Error analysis in mathematics education. Journal for Research in Mathematics Education, 10(3), 163-172.

Razzaq, L., & Heffernan, N. 2010. Hints: is it better to give or wait to be asked? In Intelligent Tutoring Systems (pp. 349-358). Springer Berlin/Heidelberg.

Roll, I., Aleven, V., McLaren, B. M., & Koedinger, K. R. 2011. Improving students' help-seeking skills using metacognitive feedback in an intelligent tutoring system. Learning and Instruction, 21(2), 267-280.

Roschelle, J., Feng, M., Murphy, R.F., & Mason, C.A. 2016. Online mathematics homework increases student achievement. AERA Open, 2(4), 1-12.

Roy, M. & Chi, M.T.H. 2005. Self-explanation in a multimedia context. In R. Mayer (Ed.), Cambridge Handbook of Multimedia Learning (pp. 271-286). Cambridge Press.

Rusman, E., Lutgens, G., and Ronteltap, F. 2005. The production of e-learning design patterns, and a research road map for e-learning. (Ref num: 101421-CY-2002-1-CY-MINERVA-MMP). Retrieved from E-LEN project website: http://www2.tisip.no/E-LEN/documents/ELEN-Deliverables/Report_WP3_ELEN-Roadmap.pdf

Ryan, A. M., Gheen, M. H., & Midgley, C. 1998. Why do some students avoid asking for help? An examination of the interplay among students' academic efficacy, teachers' social–emotional role, and the classroom goal structure. Journal of Educational Psychology, 90(3), 528.

Shulman, L. S. 1986. Those who understand: Knowledge growth in teaching. Educational researcher, 15(2), 4-14.

Siegler, R.S. 2002. Microgenetic studies of self-explanations. In N. Granott & J. Parziale

(Eds.), Microdevelopment: Transition processes in development and learning (pp. 31- 58). New York: Cambridge University

Sleeman, D. 1984. An attempt to understand students' understanding of basic algebra. Cognitive Science, 8(4), 387-412.

Sleeman, D. 1982. Assessing aspects of competence in basic algebra. In Intelligent tutoring systems (pp. 185-199). Academic Press, New York.

Vygotsky, L. S. 1962. Language and thought. Massachusetts Institute of Technology Press, Ontario, Canada.

Warburton, S., and Mor, Y. 2015. A set of patterns for the structured design of MOOCs. Open Learning: The Journal of Open, Distance and e-Learning, 1-15.

Wood, H., & Wood, D. 1999. Help seeking, learning and contingent tutoring. Computers & Education, 33(2), 153-169.

Zimmerman, M., Herding, D., & Bescherer, C. 2014. Pattern: Hint on Demand. In Mor, Y., Mellar, H., Warburton, S., and Winters, N (Eds.) Practical design patterns for teaching and learning with technology. Rotterdam, Netherlands: Sense Publishers.