Developing a Comic-Creation Assignment and Rubric for Teaching and Assessing Algorithmic Concepts

Briana Bettin
Computer Science Department
Michigan Technological University
Houghton, USA
bcbettin@mtu.edu

Michelle Jarvie-Eggart
Engineering Fundamentals Department
Michigan Technological University
Houghton, USA
mejarvie@mtu.edu

Kelly S. Steelman
Cognitive and Learning Sciences Department
Michigan Technological University
Houghton, USA
steelman@mtu.edu

Charles Wallace
Computer Science Department
Michigan Technological University
Houghton, USA
wallace@mtu.edu

***Abstract*—Comics as a pedagogical tool have shown promise in conveying ideas and increasing retention in STEM disciplines. Within the context of introductory programming courses, the visual, real-world analogies to programming problems that comics offer may allow engineering students to better grasp abstract algorithmic concepts that can be obscured by low-level programming language details. Creating comics also provides an opportunity for students to exercise metacognition, as they are asked to reflect on what they know and how to express it in a new way. Further work is needed, however, to develop comic assignments with rigorous rubric-based assessment.**

Our work in progress focuses on an intervention where students develop their own algorithmic comics and share them with peers. The design of the assignment and assessment rubric leverages prior work on using context-bounded analogy to teach programming concepts. Students will create a comic illustrating an algorithmic concept and then use it as a medium for further communication, responding to peer feedback in order to further refine their understanding. These communications, we hypothesize, will increase correctness of representations and scenarios in the comics through refinement and likewise, in their understanding of programming concepts. The rubric associated with the assignment will serve as a model for assessment of relevant criteria within comic-based student activities.

***Index Terms*—Comics, Analogy, Metaphor, STEM Education, Computer Science Education, Introductory Programming, Assessment**

I. Introduction

Comics, cartoons, and graphic novels are gaining popularity as tools in STEM education, both as a means of elucidating concepts through instructor-generated comics, and as a means for students to express their understanding by generating their own comics. Studies of comic-oriented interventions within STEM classes indicate benefits in increasing student engagement [1] and confidence about subject matter [2], as well as increasing understanding and retention of difficult topics [3], [4], [5]. In addition, graphic novels may ease anxieties about learning [6] and make classes more enjoyable [7].

The use of comics specifically in introductory programming contexts has also been explored. Research-based workshop designs that encourage storyboarding and planning for development of coding have received favorable responses from participants [8]. Further exploration into the use of comics alongside code snippets revealed student success in identifying the code that the comic characterized [9].

Farinella argues "one of the main benefits of comics in science communication could be the mapping of abstract scientific concepts onto everyday objects and experiences" [10], but he notes that assessments tend to lack objectivity and tend to focus on the appeal and humor of the comics. He argues that an essential research question to be addressed is "What constitutes a 'useful' visual metaphor in science communication?" [10]. Educators in computing, engineering, and other STEM disciplines need more robust tools to help them assess the comparisons established within student-generated comics. The degree to which a student can create a useful comparison when assigned a comic can be considered a metric for assessing how well students grasp concepts.

We are currently developing an assignment and associated rubric for assessing comics as analogies, as a foundation for future work in this area. In §II we discuss the theory and prior work on analogy as a pedagogical tool in computing education that forms the foundation for our intervention. In §III we discuss the design of the assessment for the intervention, particularly the rubric for assessing student comic contributions. In §IV we discuss the methods for our intervention, including details on the student assignment and assessment of the rubric. We conclude in §V with a discussion of how this intervention fits within our broader research goals.

II. Positioning Comic Design and Assessment Using Analogy Theory

Interpreting and constructing educational comics both require analogical reasoning to form links between the comic illustration and the educational topic at hand. In exploring assessment and design methods for comic-based assignments, we use prior work on analogy as a theoretical framework. Gentner's structure mapping theory [11] suggests that analogies are understood through the relationships entities share with each other, not simply the entities themselves. Gentner's theory has already been applied in the creation of a design and critique framework for programming analogies [12], which will be extended in this work as a theoretical framework in the space of programming comic assessment.

Context often governs the viable relationships we discern between entities, and context is also posited as key in programming analogy design. The commonly used analogy likening a program variable to a box is often used to illustrate shortcomings in analogy for programming education, as a box and a variable behave quite differently in many ways. Clearly the analogy is not effective in isolation; it requires contextualization to be valuable in understanding the relationship between a physical box and a non-physical concept like a variable. Tightening the scope through context promotes the choice of targeted relationships for representation, lessening the likelihood that relationships become mismatched among several different entities, or that two entities become conflated in their narrative role.

In an action-reaction frame, an entity's relationship with the environment and other entities is affected by actions and their results. This style of framing can be useful in creating comics and analogies that help explain "how programming works". Changes to the environment and the entities representing concepts like loops and variables will parallel changes one might observe in the code from similar interactions. The action-reaction frame that is promoted through this lens also draws upon the three-part storytelling arc of:

1) scene setting (introduction of the entities and any key prior relations)
2) conflict (actions upon or between entities)
3) resolution (portrayal of the results of those actions)

Personalized activities are valuable to instruction for increasing engagement and sense of belonging in learners [13], [14], but they complicate assessment, particularly when the student contribution is so open-ended. The student's choice of domain to explore and their artistic skills are two sources of variation that are not relevant to our goals. Using analogical theory as a framework to drive assignment design and assessment design shifts the focus appropriately to assessing parallels in the relationships represented in the comic and the relationships within the educational context being portrayed.

III. Assessment and Rubric Design

In developing a rubric for comic-based assignments, our first objective is to assess its utility and usability, by assessing inter-rater reliability and eliciting feedback on the rubric from course TAs. Our second objective is to assess the relationship between performance on the comic assignment and performance on a separate, non-comic-based programming assessment.

We propose a rubric design that centers the components of structure mapping in analogy. Comics must (1) be targeted to some problem space, (2) utilize elements that parallel those found in that problem space, (3) showcase relationships between these elements that map to interactions within such a scenario, and (4) show these with correctness and consistency across the comic.

While this framework leverages creativity, it does not center it in the traditional way. Creativity in this framework is defined as students personalizing the assignment by finding domains that are relevant to them to draw analogies (and subsequently their comics) from. Artistic ability is not imperative; creativity is shown in choosing a viable domain to reflect the STEM concept that is "outside the box" of the discipline.

Finally, in order to facilitate critical analogical design, our study proposes a reflection portion in which students describe what their comic means. Our inspiration is the *Explain xkcd* wiki [15], based on the popular *xkcd* comic series, that provides the scientific and technical background necessary to fully appreciate each comic. Our rubric assesses if students have clearly articulated what their comic actually portrays, and reflected on how that portrayal relates to the concept they are representing. This reflection promotes comics as a communication medium, and engages the student in summarizing what they attempted to communicate, and how. We intend to design a rubric specific to the reflection, but the comic rubric's reflection competency is a check for "what the comic says" and "what the student intends the comic to say".

Category	**Description**
Problem Space	Comic clearly represents a specific application of or situation using the coding concept
Entities	Each character or "set piece" in the comic consistently portrays the same concept or entity in coding (ie, a loop, the programmer, etc)
Relationships	Characters and set pieces are shown to interact in ways that parallel the interactions expected of the concepts or entities from programming
Competency	The comic clearly indicate an understanding of the programming concepts showcased
Creativity	The comic shows application of source domains outside of programming to represent coding concepts in their comic
Reflection	Summarizing of the comic in writing accurately represents the coding concept portrayed, and indicates how knowledge of these concepts was intentionally applied in the comic's design

TABLE I
Rubric outline for comic assessment

The xkcd comic in Figure 1, illustrating the dynamics of an infinite loop, serves as an excellent model for what we have in mind. This comic clearly shows **entities** that represent

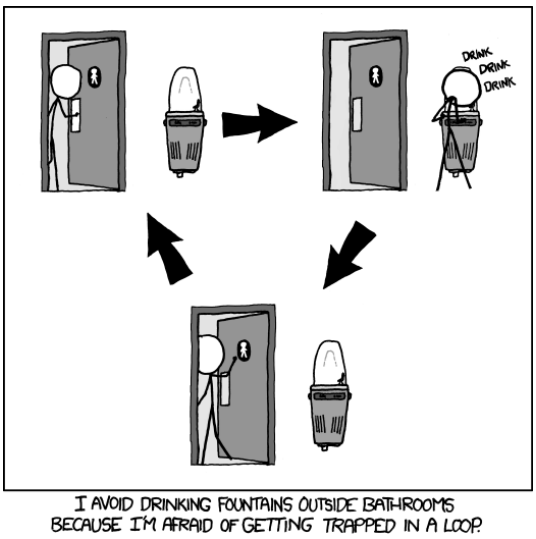

Fig. 1. xkcd's "Drinking Fountains" Comic (https://xkcd.com/986/)

something about program execution in how they interact and exhibit **relationships** within the comic space. For instance, the stick figure in the comic may represent the program counter, the restroom and fountain locations may represent particular instructions, and the arrows between locations may represent control relationships between instructions, with changes in the figure's location corresponding to program execution following the control flow. These relationships and entities show **competency** in expressing the essential dynamics of an infinite loop, which could be recognized as the **problem space** assigned. The comic shows **creativity**, not in artistic sophistication, but in the choice of representing a programming concept within a context from everyday life. A **reflection** on the comic, explaining its ties to computer programming, can be found on the Explain xkcd [15] website.

IV. Methods

A. Instructional Context

At our home institution, all first year engineering students complete a common course ENG1 *Engineering Analysis and Problem Solving*, an introduction to the engineering profession and to its various disciplines. Coursework and activities focus on developing problem-solving skills, computational skills, and communication skills. During the first semester of the course, students are introduced to MATLAB as a problem-solving tool that will be used throughout the semester. Within ENG1, students are introduced to beginning programming concepts through MATLAB applications.

The approximately 800 incoming first year engineering students are divided into classes of 100 each. Each class meets twice per week for two-hour studio sessions with their instructor and near-peer teaching assistants (TAs). Each TA is assigned to a cohort of approximately 20 students, with whom they work throughout the semester. Each cohort also meets once per week for a one hour session with only their TA. All sessions are taught in a flipped-class format, where students complete readings and watch lecture videos prior to attending class sessions. During these sessions the students collaboratively work in four-person semester-long teams.

B. Participants

Participants will include 100 first-year engineering students in one section of ENG1. Although some may have previous programming experience, historically the majority of students in ENG1 are novice programmers. The university is 88.1% white, 72.6% male, and approximately 11% transfer students. We expect the sample of students in ENG1 to approximately mirror this university distribution. Data will be analyzed only for those students who provide their informed consent and complete the intervention assignments. The course TAs will also serve as participants. TAs will be responsible for grading the comic assignment using the rubric outlined above. TAs will be asked to provide feedback on the usability of the rubric.

C. Comic Assignment

The comic assignment will be presented in week 10 of the Fall 2021 semester. At this point, students will already have completed an introduction to Matlab and should be familiar with basic programming concepts including user-defined functions, plotting, relational and logical operators, and conditional statements.

Introducing the assignment: The comic assignment will be initially introduced in the large-format studio session. The lesson will begin with an introduction to the use of comics in STEM fields, emphasizing how comics can be used to explain STEM concepts in ways that are accessible. Examples of several comics will be provided, especially the well-known *xkcd* comics [16]. The xkcd comics were selected because they focus on a wide variety of STEM concepts and feature simple drawings with stick figure characters, to emphasize comics as a means of communication rather than as an artistic endeavor. We hope that introducing comics that focus on subject matter rather than the aesthetic quality of the artwork will help alleviate students' concerns about being asked to do a project that uses skill sets outside of their comfort zone.

The xkcd comics are also ideal examples because of the associated wiki that explains the science behind the comics [15]. We will also share data [17] with the students that shows that the ability to explain a programming concept to others is associated with better programming abilities. Moreover, we will emphasize that the ability to explain STEM concepts to lay audiences is a critical skill for engineers in and of itself. We hope to reinforce that this assignment allows students to practice explaining programming concepts in a way that encourages them to both check their own understanding and to demonstrate their knowledge outside of a typical programming assignment or unit assessment.

Creating the comic: Students will be provided with the following prompt:

> *Consider a problem where a loop would be used in programming. For example, you might use a loop to*

calculate how interest compounds in your retirement account as you make monthly contributions (you can use this idea if you are stuck, but try to think of your own). Create a comic to show how a loop is used and executes in this scenario.

Students can draw their comics on one of several printable comic strip templates or produce it digitally with the software of their choice. Many options exist for generating e-comics, which do not require background in comic design [18]. Several free and easy-to-use options will be presented to students as resources, such as StoryboardThat [19]. Students will also be asked to write a narrative reflection that explains their concept, similar to the examples that were provided. We will provide students with guided questions that ask them to consider the underlying entities and relations that are depicted in the strip.

As we work to refine this study based on our preliminary design, we will also incorporate an alternative text requirement for each comic panel, in which students describe the literal design elements of the panel isolated from their reflection on the meaning of those elements. This will serve a role in teaching students about necessary accessibility considerations and also distinguishing between communicating the literal representation they created and communicating their intention with it. Any students with visual impairments will be able to complete the assignment in an alternative narrative story form. We will also introduce students to the rubric, again emphasizing that we are assessing their ability to communicate about key programming concepts and that creativity is based more on their ability to represent and apply programming concepts in novel contexts rather than their artistic skills.

Sharing the comic: Students will bring the completed comics to their small-group sessions with the TA. Students will then trade comics with a partner without providing any additional context or description, such as their reflections. Using the same set of guided questions, students will be asked to write an explanation for their partner's comic. Afterward, students will share their responses and discuss opportunities for improving their comics.

Revising the comic and description: Finally, students will be asked to revise their comic and written explanation based on their peer feedback.

D. Grading the assignment

Both initial and final comics and reflections will be assessed using the rubric. All TAs will be provided with an introduction to the rubric and will go through a norming activity to help ensure consistency in grading. Students will also be graded on participation in the revision activity through sharing a summary of feedback communicated during the session.

E. Assessing the utility and usability of the rubric

To assess inter-rater reliability, all TAs will be asked to assess final comics from two of the cohorts, their own and one other. After completing the grading, TAs will be provided with a survey to reflect on the usability of the rubric. In particular, we will inquire about whether they found the performance levels and performance criteria descriptions to be distinct, clear, and meaningful [20]. To determine whether scores on this programming assignment can discriminate between students with strong and weak understanding of loops, we will also assess the correlations among scores on the final comics and on student scores on the loops unit assessment and other unit programming assignments.

V. Conclusion

This work-in-progress introduces a comic-creation assignment and rubric for teaching and assessing algorithmic concepts presented within them. In contrast to other comic rubrics, creativity is assessed based on student's ability to represent coding concepts through a source domain outside of programming rather than on artistic skill. This rubric leverages analogical theory and a structure-mapping based framework to emphasize how well students have represented the problem space, algorithmic entities, and their relationships. This relationship-based design promotes a storytelling action-reaction structure, which can be helpful in describing "how programming works" based on the action of writing some idea in code and the reaction of the process result shown. In previous work, we have seen evidence that engineering students often struggle with the idea of "how programs are constructed" [12]. Communicating how the concepts comprising the programs work may be beneficial to students understanding of programming construction. Although our current focus is on first-year engineering students, the approach is clearly applicable to students in other programs; later we intend to compare the relative effectiveness of comic-creation activities across different curricula.

With the comic approach come certain risks. We acknowledge that some students may feel uncomfortable and perhaps even resistant to participating in a visual "artistic" activity as part of their computing education. More seriously, a comic-based assignment without accommodation for visual or physical disabilities carries a risk of marginalizing students who are already in a challenging position. These concerns make it all the more important to develop rubrics that place emphasis appropriately on the establishment of strong analogies, and assignments that offer accessible alternatives to the standard comic strip format. We actually see an opportunity to "teach accessibility" [21] in these assignments, by asking students to provide accessible text alternatives to their visual creations, thereby forming good practices in accessible design.

The forthcoming work will assess and refine the rubric to ensure its reliability and usefulness in evaluation of student work. This rubric will support future work to assess the effectiveness of comic-creation assignments for improving engineering students' confidence and competence with programming.

References

[1] L. Landherr, "By students for students: Using course projects to create learning materials for future classes," in *2020 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 2020. [Online]. Available: https://www.asee.org/public/conferences/172/papers/29375/view

[2] ——, "Integrating comics into engineering education to promote student interest, confidence, and understanding," in *2019 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 06 2019. [Online]. Available: https://www.asee.org/public/conferences/140/papers/25700/view
[3] E. Diehl, "Using cartoons to enhance engineering course concepts," in *2018 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 06 2018. [Online]. Available: https://www.asee.org/public/conferences/106/papers/21772/view
[4] G. O. Ilhan, G. Kaba, and M. Sin, "Usage of digital comics in distance learning during covid-19," in *International Journal on Social and Education Sciences (IJonSES)*, 2021, pp. 161–179. [Online]. Available: https://www.ijonses.net/index.php/ijonses/article/view/106
[5] C. D. Wylie and K. A. Neeley, "Learning out loud (lol): How comics can develop the communication and critical thinking abilities of engineering students," in *2016 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 2016. [Online]. Available: https://www.asee.org/public/conferences/64/papers/16545/view
[6] P. Johanes, "The graphic novel: A promising medium for learning research," in *2018 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 06 2018. [Online]. Available: https://www.asee.org/public/conferences/106/papers/23152/view
[7] J. Pascal and T. L. Pascal, "Using student developed comics to promote learning of transport phenomena concepts," in *2017 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 2017. [Online]. Available: https://www.asee.org/public/conferences/78/papers/19848/view
[8] S. Suh, M. Lee, G. Xia, and E. Law, "Coding strip: A pedagogical tool for teaching and learning programming concepts through comics," in *In IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE, 2020. [Online]. Available: https://sanghosuh.github.io/assets/pdf/codingstrip_vlhcc.pdf
[9] S. Suh, C. Latulipe, K. J. Lee, B. Cheng, and E. Law, "Using comics to introduce and reinforce programming concepts in cs1," in *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 369–375. [Online]. Available: https://doi.org/10.1145/3408877.3432465
[10] M. Farinella, "The potential of comics in science communication," *Journal of Science Communication*, vol. 17, 01 2018. [Online]. Available: https://jcom.sissa.it/archive/17/01/JCOM_1701_2018_Y01
[11] D. Gentner, "Structure-mapping: A theoretical framework for analogy," *Cognitive Science*, vol. 7, no. 2, pp. 155 – 170, 1983. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0364021383800093
[12] A. Authors, "Anonymized for review," in *Anonymized for Review*, XXXX.
[13] J. M. Harackiewicz, J. L. Smith, and S. J. Priniski, "Interest matters: The importance of promoting interest in education," in *Policy insights from the behavioral and brain sciences*, vol. 3. National Institutes of Health, 2016, pp. 220—227. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5839644/
[14] T. Shepard, "Implementing first-year design projects with the power of choice," in *2013 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 06 2013. [Online]. Available: https://peer.asee.org/19722
[15] C. Wiki, "Explain xkcd," 2021. [Online]. Available: https://www.explainxkcd.com
[16] R. Munroe, "xkcd comics," 2021. [Online]. Available: https://xkcd.com/
[17] P. von Lockette, "Algorithmic thinking and matlab in computational materials science," in *2006 ASEE Annual Conference & Exposition*. American Society for Engineering Education, 06 2006. [Online]. Available: https://peer.asee.org/725
[18] F. Lazarinis, A. Mazaraki, V. S. Verykios, and C. Panagiotakopoulos, "E-comics in teaching: Evaluating and using comic strip creator tools for educational purposes," in *2015 10th International Conference on Computer Science Education (ICCSE)*, 2015, pp. 305–309.
[19] S. That, "Storyboard that: Storyboard creator," 2021. [Online]. Available: https://www.storyboardthat.com/storyboard-creator
[20] P. Arcuria and M. Chaaban, "Best practices for designing effective rubrics," 2019. [Online]. Available: https://teachonline.asu.edu/2019/02/best-practices-for-designing-effective-rubrics/
[21] M. Lawrence and M. Bellard, "Teach access: Preparing computing students for industry (abstract only)," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, ser. SIGCSE '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 700. [Online]. Available: https://doi.org/10.1145/3017680.3022392