“I didn’t order that!”: Demystifying Distributed Systems

Hongyi Li, Hongyi Liu, Olivier Marin
hl2752@nyu.edu,hl2638@nyu.edu,ogm2@nyu.edu
New York University Shanghai, China

ABSTRACT

On a daily basis, the general public uses online services without any knowledge of the concepts these services rely on. Even though this is fine most of the time, there are situations where such obliviousness puts the users at risk [4, 6]. We aim to popularize key distributed systems concepts, and make them accessible to K-16 students, so as to give future generations a better idea of what to expect of online services. This should help make better decisions regarding when to use them, and when to trust them. We adopt the method of story telling, and embed distributed systems concepts into an interactive story available on the Internet. Our preliminary assessment on CS1 students shows that our interactive game may have a positive effect on their overall grasp of the embedded notions. The effect is marginal, but our study provides interesting insights on how to make the technique effective.

KEYWORDS

Popularization, Storytelling, Distributed Systems

ACM Reference Format:
Hongyi Li, Hongyi Liu, Olivier Marin. 2020. “I didn’t order that!”: Demystifying Distributed Systems. In *Proceedings of 52nd ACM Technical Symposium on Computer Science Education (SIGCSE ’21).* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Even though distributed systems are ubiquitous, they are rather poorly understood by the general public. People mostly expect their online services to handle their requests instantly and flawlessly. In reality, a request typically generates multiple rounds of communication over the network, and failures may occur anywhere, anytime. In 2018, a food delivery service app crashed massively in the UK, and failed to confirm orders [6]. Many customers ended up submitting duplicate orders and got charged for them. Placing and confirming online orders is a classic problem in distributed systems, associated with the Two Generals’ Problem [13]. The Iowa Caucus in 2020 [4] is another infamous incident caused by incorrect distributed software. The above examples illustrate that, although users tend to trust online services, such services must abide by the natural limitations of the distribution at their core. In particular, it is theoretically impossible to guarantee consensus in an asynchronous network where delay is unbounded [5].

SIGCSE ’21, March 17–20, 2021, Toronto, ON, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

We aim to help K-16 students form a reasonable understanding and a correct *notional machine* [2, 15] of distributed systems. Armed with a correct mental representation, these future users will be empowered to make better decisions regarding whether to trust online services and how to use them.

Our approach is to use storytelling to introduce common distributed computing problems and protocols, while hiding their complexity. We embed the Two-Phase Commit Protocol (**2PC**) in a play script and implement an internet-based video game to deliver the play in the form of interactive story. We present a preliminary empirical study on college freshmen. It indicates that an interactive game may have a positive effect on their overall grasp of the embedded notions. Our results are not statistically significant to pass null hypothesis testing; yet students’ accounts of their experience with our game show that storytelling is more engaging, and does a better job of hooking their interest than traditional materials like textbooks. Our results also rule against an overabundance of storytelling elements which can impede the learning experience.

2 THE STORYTELLING APPROACH

Storytelling consists in embedding computer science concepts in a story, with settings and personae that are familiar to the target audience. This allows to convey the gist of the embedded ideas without any specialized terminology. Besides, and a good story can hook the interest of the audience.

Ipsum Lauren [1] introduces algorithmic concepts on a very high level. The book targets middle school children, and the story mimics *Alice in Wonderland* with its playful and clever use of names. Parham-Mocello et al. offer a Computer Science Orientation course based on story programming [12], where students read stories [3] to learn the concepts. Several works use the constructionist learning approach [7, 10, 11], where learners and/or actors impersonate the “characters”. One of these [7] is of particular interest to us because it models the behavior of processes in a distributed system in order to explain the concept of self-stabilization; however, it targets graduate students. The evaluations of all these works produce mixed results. A quantitative study [12] shows that story programming is a viable approach for teaching a CS orientation class. Other studies [8, 9] show that children could develop various incorrect mental models through constructionist learning. The authors of [7] base their claim of effectiveness and efficiency on a single survey.

Our work stands out over two aspects. Firstly, we use storytelling to teach K-16 students distributed systems, a more advanced topic than traditional beginner-level ones such as computer programming. Secondly, our assessment focuses on the efficacy of storytelling in teaching distributed computing concepts: we use null hypothesis testing for our quantitative evaluation, and students’ feedback for our qualitative evaluation.

3 DRAMATIZATION OF THE 2PC PROTOCOL

Our simple idea is to tell stories that embed distributed systems concepts. We choose to begin with the Two-Phase Commit protocol [14]. The goal of the Two-Phase Commit protocol (2PC) is to reach a consensus among the participants of a network: either all the participants commit to the transaction, or they all abort. The 2PC is designed to guarantee consistency: participants must eventually share the same view of the system. The protocol requires one of the participants to act as coordinator. In most variants of the 2PC algorithm, all the other participants only communicate with the coordinator. Possible failures during the procedure include message losses, network partitions,and site crashes. Assuming all sites recover independently (without communicating with other sites) there is no way to guarantee consistency in the event of a failure [14].

We draw the background setting for the story directly from *Gulliver's Travels* [16]. To briefly introduce the background, Gulliver the Englishman travels to an archipelago in the South Indian Ocean, where there are two island empires, Lilliput and Blefuscu. The two empires have conflicting laws over which end of an egg to break, namely the Small-Endian and the Big-Endian convention. Based on this setting, we come up with a scenario where the emperor of Lilliput wishes to restore peace with Blefuscu by changing the law on breaking eggs (so as to eliminate the conflict). Reflecting the goal of consistency in the 2PC, there will be a severe consequence if one empire carries out the proposal but the other does not. And similarly to the 2PC model, they rely on a coordinator, namely Gulliver, to make the decision.

We implemented the story as an interactive game[1] on top of a Python-based engine called Ren'Py. The user can interact with the story by selecting decisions from a choice menu. The player goes through the main plot and gets to decide what happens at key moments of the story, which leads to different branches. At the beginning, the player can only go through the perfect scenario where no site failure or network problem occurs. If both emperors agree to the proposal, there will finally be peace between the empires. The player is also free to control the emperors' decisions, causing the agreement to possibly abort. When the player reaches the perfect ending, a new story branch is unlocked. Now the player can explore more complicated scenarios such as a message loss and its consequences after a decision is made.

When the player finishes all the story branches, a recap section will appear. The purpose of this section is to teach the audience/player how the story connects to the concepts of the 2PC, so as to strengthen their understanding. The section includes a revisit of the plot, an explanation of the protocol and how it is represented in the story, followed by a pop-up quiz as an exercise.

4 ASSESSMENT

To evaluate our work, we identify our point of interest as one's understanding of the distributed concepts embedded in the story, and we want to know how it changes before and after one interacts with our interactive story. We formulate our research question as:

RQ: Does the audience understand the distributed systems concepts embedded in our interactive story better after playing it?

To answer the research question, we conduct a within-subject experiment in an intensive five-week Introduction to Computer Programming summer course aimed at students who haven't declared their majors. We measure a subject's understanding of the 2PC protocol with a quiz consisting of five multiple choice questions[2] (select-all-that-apply style). In keeping with the spirit of vulgarization and engagement, our quiz uses popular personas in the Marvel Universe instead of scientific terminology to introduce the problem in a dramatic setting. The quiz questions cover the normal operation of 2PC, 2PC under network partition and site failures [14], and the FLP impossibility [5]. The quiz doesn't mention these problems by their names to prevent the bias of subjects actively looking these up after the first quiz session.

The null hypothesis for RQ is that there won't be any difference in quiz performance before and after interacting with our story. The mean for the quiz score is 3.40 (out of 5) pre-game, compared to 3.72 post-game. The paired difference is +0.318. However, paired-t test shows that the difference is not significant ($p = 0.1815$) so we cannot reject the null hypothesis for RQ. A problem with our experiment design is that the within-subject experiment design cannot mitigate a possible bias: students who have taken the quiz are more familiar with questions when they take it the second time.

We lacked the resources to conduct a between-subject experiment to compare our storytelling approach with traditional methods such as textbooks. As a substitute, we introduced the formal 2PC protocol in the survey questionnaire and provided a link to its Wikipedia page[3]. We asked the students who took the time to read the Wikipedia page to rate their agreement to the two following statements on a 5-point Likert scale: (1) playing this interactive story is more engaging than reading the Wikipedia page; (2) the interactive story makes the Two Phase Commit protocol easier to understand. As a result, 12 students out of 20 (60%) strongly agree that our game is more engaging, 7 (35%) agree, and one (5%) strongly disagrees. Out of 19 students, 15 (78.9%) agree that our story makes the 2PC easier to understand, 3 (15.9%) strongly agree, while only 1 (5.3%) strongly disagrees.

Student feedback also offers insights into flaws of our work. Most students agree that the story elements made the learning process more interesting, but some felt that this was not an efficient way of learning. Others felt that the plots and characters/universe settings overshadowed the distributed concepts embedded in the story.

5 CONCLUSION

Our storytelling approach aims to help K-16 students (future decision makers) make sense of how online services operate. Our online interactive story dramatizes the 2PC protocol, personifying network nodes to explore complex notions such as consistency and the FLP impossibility. It requires no prior knowledge of computer science. The results of our experimental assessment hint at some improvement of the subjects' understanding of these notions, but fail to reject the null hypothesis. The feedback we gathered highlights the importance of an adequate balance between the storyline and the concepts: overly rich plot lines hinder the learning experience, while explicit rehash of the concepts helps improve it.

[1] Available at https://bit.ly/2D1UwSU

[2] Available at https://bit.ly/3hrPqOV

[3] https://en.wikipedia.org/wiki/Two-phase_commit_protocol

REFERENCES

[1] Carlos Bueno. 2014. *Lauren Ipsum: A Story About Computer Science and Other Improbable Things*. No Starch Press, 245 8th St. San Francisco, California 94103 USA. Google-Books-ID: 62TDBQAAQBAJ.

[2] Benedict Du boulay, TIM O'shea, and JOHN Monk. 1999. The black box inside the glass box: presenting computing concepts to novices. *International Journal of Human-Computer Studies* 51, 2 (Aug. 1999), 265–277. https://doi.org/10.1006/ijhc.1981.0309

[3] Martin Erwig. 2017. *Once Upon an Algorithm: How Stories Explain Computing*. The MIT Press, One Rogers Street, Cambridge, MA.

[4] Pam Fessler. 2020. After Iowa Debacle, Nevada Democrats Will Not Use An App For Their Caucuses. https://www.npr.org/2020/02/07/803941738/after-iowa-debacle-nevada-democrats-will-not-use-an-app-for-their-caucuses

[5] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of distributed consensus with one faulty process. *J. ACM* 32, 2 (April 1985), 374–382. https://doi.org/10.1145/3149.214121

[6] Helen Knapman. 2018. Deliveroo app crash leaves customers furious as some are charged SEVEN times. https://www.thesun.co.uk/money/7284287/deliveroo-app-crash-customers-furious-charged-seven-times/

[7] Boris Koldehofe and Philippas Tsigas. 2001. Using actors in an interactive animation in a graduate course on distributed system. In *Proceedings of the 6th annual conference on Innovation and technology in computer science education (ITiCSE '01)*. Association for Computing Machinery, New York, NY, USA, 149–152. https://doi.org/10.1145/377435.377670

[8] D. Midian Kurland and Roy D. Pea. 1995. Children's Mental Models of Recursive Logo Programs:. *Journal of Educational Computing Research* 1, 2 (Jan. 1995), 235–243. https://doi.org/10.2190/JV9Y-5PD0-MX22-9J4Y Publisher: SAGE PublicationsSage CA: Los Angeles, CA.

[9] D. Midian Kurland, Roy D. Pea, Catherine Clement, and Ronald Mawby. 1995. A Study of the Development of Programming Ability and Thinking Skills in High School Students:. *Journal of Educational Computing Research* 2, 4 (Jan. 1995), 429–458. https://doi.org/10.2190/BKML-B1QV-KDN4-8ULH Publisher: SAGE PublicationsSage CA: Los Angeles, CA.

[10] Seymour Papert. 1980. *Mindstorms: children, computers, and powerful ideas*. Basic Books, Inc., USA.

[11] Seymour Papert. 1986. On Logo. https://el.media.mit.edu/logo-foundation/resources/onlogo/index.html

[12] Jennifer Parham-Mocello, Shannon Ernst, Martin Erwig, Lily Shellhammer, and Emily Dominguez. 2019. Story Programming: Explaining Computer Science Before Coding. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. Association for Computing Machinery, Minneapolis, MN, USA, 379–385. https://doi.org/10.1145/3287324.3287397

[13] Tom Scott. 2019. The Two Generals' Problem. https://www.youtube.com/watch?v=IP-rGJKSZ3s

[14] D. Skeen and M. Stonebraker. 1983. A Formal Model of Crash Recovery in a Distributed System. *IEEE Transactions on Software Engineering* SE-9, 3 (May 1983), 219–228. https://doi.org/10.1109/TSE.1983.236608 Conference Name: IEEE Transactions on Software Engineering.

[15] Juha Sorva. 2013. Notional machines and introductory programming education. *ACM Transactions on Computing Education* 13, 2 (July 2013), 8:1–8:31. https://doi.org/10.1145/2483710.2483713

[16] Jonathan Swift. 1826. *Gulliver's Travels*. Jones & Company, 3, Action Place, Kingsland Road, London. Google-Books-ID: ta1uaL7RF5gC.