Making the Liskov Substitution Principle Happy and Sad

Elisa Baniassad
University of British Columbia
Vancouver, BC
ebani@cs.ubc.ca

ABSTRACT

The Liskov Substitution Principle states, among other constraints, that a subtype is not substitutable for its super type if it strengthens its operations' preconditions, or weakens its operations' postconditions. We found that students in two subsequent courses had trouble remembering these rules. Their major stumbling block appeared to be recalling which condition (pre- or post-) could be strengthened and which could be weakened. We developed a simple visual reminder to help: A method is happy if it is substitutable—A smile is wider at the top than at the bottom, suggesting weaker/looser/wider pre-conditions, and stronger/tighter/narrower post conditions.; A method is sad if it isn't substitutable—a frown is narrower at the top, suggesting stronger/tighter/narrower preconditions, and wider at the bottom, suggesting weaker/looser/wider postconditions. Though the technique is far from perfect, we found that it allowed students to move on to the more interesting design questions around the LSP.

CCS CONCEPTS

• **Social and Professional Topics** → *Software Engineering Education*

KEYWORDS

Software engineering education

ACM Reference Format:
Elisa Baniassad 2018, Making the Liskov Substitution Principle Happy and Sad, In Proceedings of 40th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET'18). ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3183377.3183380

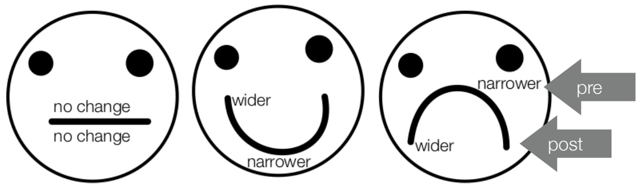


Figure 1: Happy and sad operations

1 THE LISKOV SUBSTITUTION PRINCIPLE (LSP)

The Liskov Substitution Principle (LSP) [1] states that a subclass should not break the expectations set by its superclass: A Penguin is not substitutable for a Bird if users of Bird expect it to be able to fly; A Square is not substitutable for a Rectangle if users expect to be able to change width and height independently. Encoding this rule involves two constraints about pre- and postconditions of operations: a subclass can only be a substitute for its superclass if its operations' preconditions are not strengthened, and its operations' postconditions are not weakened.

2 EDUCATIONAL CONTEXT

We teach the LSP in two courses: a second year introduction to software construction (SC) and its follow-on third year software engineering course (SE), in classes with multiple sections of 160 students each. A mix of students take these courses, but most are computer science majors. In the SC course we introduce the LSP generally, and in the SE course we delve into it more deeply in the context of a broader range of design principles.

We took a very standard approach to teaching the LSP, using typical examples (Square versus Rectangle, Circle versus Ellipse) in both courses. Because of the lapse in time between taking second year Software Construction and third year Software Engineering, students reported little confidence when asked to recall the LSP. Thus, we found it necessary to repeat most of the LSP content in the SE course almost identically to its introduction in the SC course.

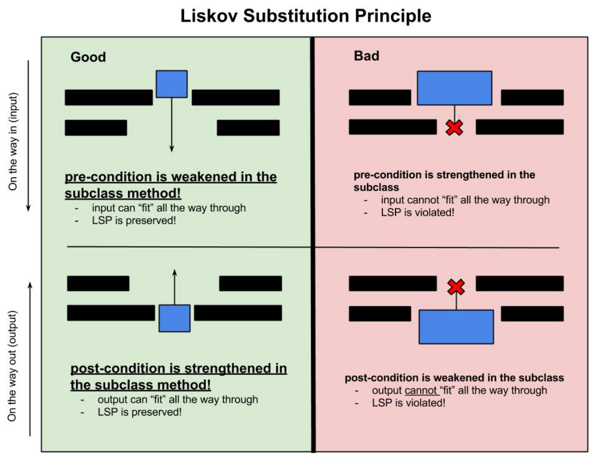

Liskov Substitution Principle
Good
Bad
On the way in (input)
pre-condition is weakened in the subclass method!
- input can "fit" all the way through
- LSP is preserved!
pre-condition is strengthened in the subclass
- input cannot "fit" all the way through
- LSP is violated!
On the way out (output)
post-condition is strengthened in the subclass method!
- output can "fit" all the way through
- LSP is preserved!
post-condition is weakened in the subclass
- output cannot "fit" all the way through
- LSP is violated!

Figure 2: Student's diagram prior to us introducing the happy and sad operations notation

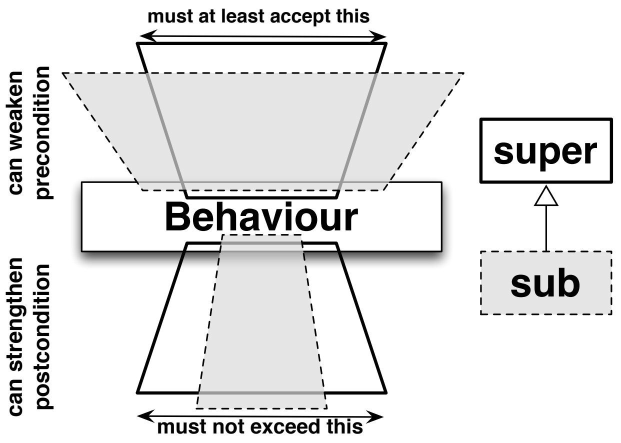


Figure 3: Our original (confusing) visualisation

To adjust for the level of experience of the second year SC students (still learning about Object-Orientation, and somewhat new to Java), we chose to associate weakening and strengthening conditions, with the simpler and more visualisable concepts of widening and narrowing ranges of inputs/outputs. We then leveraged that association in those standard examples: A Penguin accepts a narrower range of inputs than a Bird, because it would not implement the *fly* method; a Square's *setWidth* operation produces a wider range of outputs than a Rectangle's because it changes both width and height; if a Doctor's *bookAppointment* operation accepted hours between 9am and 5pm, then a Specialist subtype of Doctor would be violating the LSP by implementing its *bookAppointment* operation as only permitting hours of 10am to 2pm, hence narrowing the range of acceptable inputs.

3 OUR ORIGINAL EXPERIENCE TEACHING THE LSP: STUCK ON WHICH IS WHICH

We noticed that students in both courses were stuck on recalling which of the conditions should be strengthened and which should be weakened for substitutability to be maintained. We called this the *which is which* question. Students asked for clarification regularly on the forum, and even confident students in both courses mixed it up when answering others. All the students' responses in both the SC and SE forums were focused on indicating *which is which*, rather than getting at the "expectations of the type" spirit of the LSP. Examples of narrowing and broadening ranges were discussed in each forum, but all of those threads began with establishing *which is which*.

One student in the second year SC course created her own visualisation (Figure 2) to help her remember *which is which*, and shared it with her classmates on the forum (they indicated appreciation!) Reassuringly, the student had correctly mapped narrower to stronger, and wider to weaker. However, the diagram's purpose perfectly illustrates that the student was entirely focused on *which is which* — there was no domain insight about what strengthening and weakening actually implied.

We provided our own complicated visualisation (Figure 3) for both courses and this did help with leveraging dual-mode learning, but it was not easy to remember or redraw from memory, and did not seem to help students with their recall of *which is which.* The forum questions for both courses remained focused on that question, and one student posted a note complaining of confusion with the diagram, and asking domain-knowledge questions about how the visualisation worked.

Upon talking to students and monitoring the forums we saw that the question of *which is which* had, for some, become the implicit learning *outcome*, as opposed to a learning *mechanism.* Advanced students actually used the implications of the LSP to derive the rule for *which is which*: "If you were expecting to make a Doctor's appointment at 9am, you wouldn't want to be told you couldn't, right? So the preconditions MUST be wider or the same!" These more proficient students were able to follow the circular approach, but because reasoning backwards required nuanced intuition, prior knowledge, and design sense, it failed to give novices or struggling students solid ground upon which to base their recall of *which is which.* This line of reasoning was the opposite of what we wanted when teaching the LSP. The goal for learning the LSP is to impress on students that you would not want a subclass to break the expectations of users of

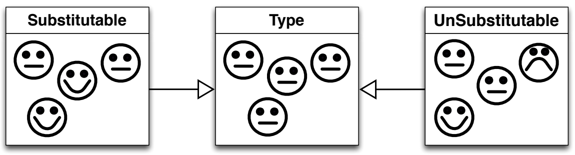


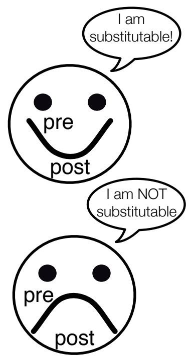
Figure 4: Happy and Sad classes

the superclass. *Which is which* should just be a vehicle for expressing and underscoring that subtle insight.

4 HAPPY AND SAD OPERATIONS

To get students past the question of *which is which* we transformed the rule into a simple notation that was facilitated by this line of reasoning:

1) An operation is happy if it can be substituted for its super type's method and sad if it cannot.
2) A smile is wider at the top than at the bottom; a frown is the opposite.
3) Preconditions are at the top of the smile/frown, postconditions are at the bottom of the smile/frown.
4) Ergo: Preconditions can be wider (looser/weaker), postconditions can be narrower (tighter/stronger).



We also needed to point out that it was fine if there was no change to either the precondition, postcondition, or both. The three main combinations were summarised with the drawing in Figure 1. We could then expand the notion of happy and sad operations to discuss the substitutability implications for entire classes, as shown in Figure 4. A single sad operation makes the entire class sad, and (not depicted) makes users of the class sad.

The happy/sad memory device was introduced simultaneously in both courses.

5 WHAT WE NOTICED AFTER INTRODUCING HAPPY AND SAD

The notation took under 5 minutes to introduce to the classes, and it did not need to be explained beyond stating the four points. Students did not ask clarifying questions about the happy and sad faces in class or on the forum. While introducing the happy and sad faces in the second year SC class, one student at the back of the room (of 160 seats) actually said "Aaah!" so loudly that it was audible up at the front.

After introducing the happy and sad notation, there were no questions on the second year forum related to LSP (other than asking if it would be on the exam).

In the third year SE forum we no longer saw confusion or fixation about the *which is which* question. The discussions moved on in style and substance to clarifying the technical manifestations of strengthening and weakening (a reduction in a range? Fewer methods implemented? More or fewer outputs or exceptions thrown?). These questions were raised without the *which is which* clarification as their preamble.

For the first time we saw instances of students synthesising and relating the LSP to other topics and contexts. For instance, one student asked whether a violation of one case of the Interface Segregation Principle [2] (no client should depend on methods it does not use) would imply a violation of the LSP (because portions of an interface might not be sufficiently implemented). Another student specifically asked if the faces' meanings changed if they were placed in the context of classes. Originally Figure 4 was only illustrated in class while Figure 1 was also in the PDF of the handouts. We subsequently added Figure 4 to the PDF of the notes so students who missed class would be sure to see it. But it was encouraging that the student was able to ask about how operations' substitutability would influence a whole class's substitutability, suggesting reasoning about the broader implications of the LSP rather than on *which is which.*

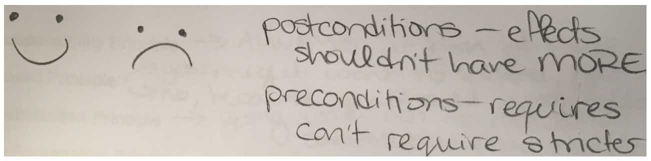

postconditions – effects shouldn't have MORE
preconditions – requires can't require stricter

Figure 5. Example of a Happy/Sad annotated exam

We asked an LSP question on the SE final exam without including the memory device. The question was answered mostly correctly, and the happy and sad faces were drawn by some students in the margins next to the LSP questions (always accompanying a correct answer). This suggests that the happy and sad mnemonic was remembered and applied correctly though it was roughly a month later.

6 WHY SIMPLE VISUALS HELP

Simple visual memory devices are common. Our happy and sad faces draw inspiration from iconic approaches such as the *Right Hand Rule* for remembering the direction of magnetic force on a moving charge, and the "alligator mouth eats the bigger number" approach for helping children recall *which is which* in the less than and greater than symbols.

As educators, we often seek to provide opportunities for dual-mode representations of subjects: supplying visuals that underscore the topics we are covering. However, according to Cook [3] visual representations can be problematic if prior knowledge is key to their understandability: understanding the diagram becomes its own learning task. The need for prior knowledge probably accounts for why the more complex visuals we initially introduced weren't as successful, and didn't produce the "Ahh" moment that the happy and sad faces did. The students were forced to then take a sidebar and interpret the visualisation, rather than using it to facilitate a more advanced question, as is evidenced by the forum question related to the image. Simple visuals, like happy and sad faces, the direction of fingers on a hand, or an open alligator mouth, require no prior domain knowledge to interpret, and so impose little additional cognitive load.

7 HUMOUR HELPS PEOPLE REMEMBER

The happy and sad faces are a little sillier than concepts we usually see in computing classes, but maybe that is why they work. The field of humour and learning can tell us a great deal about how to intersperse distinctiveness, or absurdities, to great effect in our lectures.

In 1994, [4] Schmidt ran a study looking at whether humorous sentences were easier to recall than serious ones conveying the same meaning. He found that they were! But he also found that people's concept of humour mattered: so if something wasn't funny to them, then they wouldn't remember it as well as something that was. The happy and sad faces are, if not actively "funny", at least humorous and somewhat absurd. They are a light touch, in an otherwise bland or dry subject. This levity might be a proxy for humour to some extent, and so might explain why students were quickly able to initially grasp and later recall the stronger/weaker pairing and then move on to the implications of that rule.

Puns have been found to be the best form of humour for recall[5]. The authors hypothesise that its the constrained nature of the pun that affords better recall, by limiting the breadth of information that can fit into the punchline. While the happy and sad faces are not traditional puns, they do have a very simple punchline that relates back to the meaning of what they represent: a happy face makes substitutability happy, a sad face makes substitutability sad. This is a degree of semantic constraint that may be similar to that of the pun.

All that said, we don't want to go overboard by making everything funny. Study subjects showed worse recall for sentences in a list where every sentence was absurd, than for sentences in a list where every sentence made sense. But in a mixed-list, with some absurd, and some sensible sentences, the absurd ones were remembered better [6]. The fact that the happy and sad faces were used as a single absurd/silly message may have helped them stand out as memorable learning tools.

9 FUTURE DIRECTIONS

This evaluation of the happy and sad face approach was based on an analysis of forum comments, recollections of in-class responses, and observation of exam performance and students' exam annotations. Evaluation such as this is limited if we want to be sure we can isolate the effects of, and derive reproducible positive outcomes from the happy and sad face memory cue. A controlled study may serve to solidify evidence for the approach. However, there is anecdotal evidence that happy and sad faces work: We have employed the happy/sad technique since the initial semester captured in this paper, and obtained the same results—no questions about *which is which*, and lots of content-driven discussion about the design implications of the LSP.

ACKNOWLEDGMENTS

Thanks to Alice Campbell for pedagogical context for cognitive load related to visualisations, and to Laura Barton for the use of her Substitutability diagram. Thanks also to the reviewers for ICSE-SEET, and to the reviewers and participants from SPLASH-E 2017, who helped workshop an earlier version of this paper.

REFERENCES

[1] B. H. Liskov and J. M. Wing, "A Behavioral Notion of Subtyping," ACM Transactions on Programming Languages and Systems, vol. 16 (6), pp. 1811-1841, 1994

[2] Martin, R.C., "The Interface Segregation Principle", C++ Report, Aug. 1996.

[3] Cook, M. P. (2006), Visual representations in science education: The influence of prior knowledge and cognitive load theory on instructional design principles. Sci. Ed., 90: 1073–1091. doi:10.1002/sce.20164

[4] Schmidt, Stephen R. Effects of humor on sentence memory. Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol 20(4), Jul 1994, 953-967. http://dx.doi.org/10.1037/0278-7393.20.4.953

[5] Hannah Summerfelt, Louis Lippman, and Ira E. Hyman Jr. The Effect of Humor on Memory: Constrained by the Pun. The Journal of General Psychology Vol. 137 , Iss. 4,2010

[6] McDaniel, Mark A.; DeLosh, Edward L.; Merritt, Paul S. Order information and retrieval distinctiveness: Recall of common versus bizarre material. Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol 26(4), Jul 2000, 1045-1056. http://dx.doi.org/10.1037/0278-7393.26.4.1045