Metaphors We Teach By

Joseph P. Sanford
Dept. of Computer Science
Tufts University
Medford, MA, USA
joe@cs.tufts.edu

Aaron Tietz
Dept. of Computer Science
Tufts University
Medford, MA, USA
aaron.tietz@tufts.edu

Saad Farooq
Dept. of Computer Science
Tufts University
Medford, MA, USA
sfarooq@cs.tufts.edu

Samuel Guyer
Dept. of Computer Science
Tufts University
Medford, MA, USA
sguyer@cs.tufts.edu

R. Benjamin Shapiro
Dept. of Computer Science
Dept. of Education
Tufts University
Medford, MA, USA
ben@cs.tufts.edu

ABSTRACT

In this paper we present an initial study of how metaphors are used by university-level Computer Science instructors. The goal of this research is to gain a better understanding of the role that metaphors play in Computer Science education, to catalog the kinds of metaphors that are used, and to assess their effectiveness in supporting learning. We interviewed 10 educators in Computer Science about the metaphors they have used in the classroom, with a focus on introductory "CS1" programming courses. We analyze these interviews with an existing theory of metaphors, which provides a framework for describing their structure and features. The theory predicts that most metaphors have limitations, and eventually fall apart. Therefore, we also asked educators to assess how far they could push their metaphors with and to describe what happens at the breaking point. Our preliminary findings provide a foundation to inform and guide more in-depth analyses in the future.

Categories and Subject Descriptors

K.3.2 [**Computer and Information Science Education**]: Computer Science Education

General Terms

Human Factors, Theory

Keywords

metaphors, CS1, pedagogical content knowledge, teaching

SIGCSE '14 March 5–8, 2014, Atlanta, Georgia USA

> "[...] unless you are at home in the metaphor, unless you have had your proper poetical education in the metaphor, you are not safe anywhere. Because you are not at ease with figurative values: you don't know the metaphor in its strength and its weakness. You don't know how far you may expect to ride it and when it may break down with you. You are not safe with science; you are not safe in history."
> – Robert Frost[2]

1. INTRODUCTION

Research on teaching increasingly points to the importance of teachers' knowledge about how to teach in ways that are not only responsive to the general needs of learners, but to particularities of learning in different disciplinary areas [17, 3, 16, 5]. Knowledge that is particular to the intersection of content knowledge and teaching has been called Pedagogical Content Knowledge (PCK)[18]. A common way in which teachers adapt pedagogical knowledge for the particular content of Computer Science (i.e., generate PCK) is by utilizing metaphors that explain computing concepts with the aid of ideas that are already familiar to students.

Many fields of teaching use metaphor to explain disciplinary ideas. They do so because metaphors offer a mapping from a concept familiar to students (the source domain) to a new concept (the target domain). Investigations of metaphors in teaching have been conducted for other STEM fields of study such as Biology[14], Chemistry[19], and Mathematics[1]. While the use of metaphors is commonplace in Computer Science classrooms, there is a paucity of research on this practice. While metaphorical language in teaching Computer Science has been previously touched on[15], including discussions of particular metaphors[4, 12], metaphorical language has not been examined in detail, such as through investigation of the various metaphors used by teachers to explain different CS topics, or through examination of the cognitive consequences for students of different metaphorical explanations of the same topics.

One facet of metaphors that should be of interest to researchers and teachers is the limitations of metaphors, both in general and in particular instances. A feature of many metaphors is that when their mappings are pushed beyond their user's intended limits, they eventually break down[11]. Those intended limits may be unknown to learners, and thus opportunities for spurious inferences arise. It may be helpful to instructors to be able to anticipate common misunderstandings caused by learners' overstretching of metaphors into mismatches between metaphorical source domains and their explanatory targets, leading to explanatory breakdown. As yet, Computer Science educators have no published literature about commonly used metaphors, which metaphors are most helpful to students' learning, what the common breakdown points of those metaphors are, and how those breakdowns are manifested in student activity. As Computer Science education continues to grow as a field of study, investigating metaphor use, particularly with an eye toward discovering how to use metaphors more productively in the classroom, may enable the development of pedagogies to support the retention and success of more students in computing.

This paper reports on a study of the use of metaphors for teaching Computer Science concepts through interviews of CS1 instructors. The goal of this preliminary study is to identify CS1 topics where metaphors are used, what metaphors are used to explain those topics, to search for any common themes that are present across these metaphors, and to gather data about how instructors think about breakdowns of the metaphors that they use.

2. BACKGROUND

Metaphorical language is employed in everyday life, both in thought and action, and is now considered to be a crucial component of thinking[7]. In particular, conceptual metaphors (i.e., cognitive mechanisms which project from a source domain to a target domain in order to facilitate understanding of a concept[13] in the target domain) are of interest in educational environments.

To fully comprehend the cognitive effects of combining metaphor and pedagogy, a general framework of the correlation between metaphor and understanding must be defined. The theories behind metaphors and their use has been recently redefined, thus leading to a "contemporary theory of metaphor" [6]. This theory, proposed by Lakoff, introduces the nature, structure, and aspects of metaphors, and provides a framework in which metaphors can be defined, explained, and evaluated.

Lakoff's theory explains that metaphors are the primary means in which we comprehend abstract concepts, while also expanding on the Aristotelian definition of metaphors. The differences between these two theories lies primarily in several assumptions, particularly that everyday language is rich with metaphor, and therefore widely used (including pedagogically), and that information can be understood without the use of metaphor.[7]. In both Lakoff and Aristotle's definitions, metaphors are defined as mappings across different conceptual domains that allow for understanding of relatively abstract or unstructured subject matter in the terms of a more defined subject matter. These mappings are from a source domain to a target domain, with the source domain defined as the familiar, real-world concept that maps to the target domain, defined as the concept that is being taught. For instance, the metaphor "pointers are like file folders" has a source domain of file folders and a target domain of pointers. At their core, these mappings are:

Partial in that no source domains can map to every aspect of a target domain.

Fixed in mappings, with a static set of correspondences between entities in a source domain and those in a target domain.

Based upon everyday knowledge in order to create mappings from source domains that are already understood.

These aspects of metaphor are important to consider when analyzing metaphors, as they not only demonstrate the importance of source domains used in metaphors, but also explain that no metaphor contains a full, one-to-one mapping across domains, which may lead to limitations. Using this structure and theory of metaphor provides a foundation upon which metaphors can be broken down and examined. Lakoff himself used this theory when discussing understanding mathematical concepts[8].

Thornbury used Lakoff's theory to investigate and evaluate metaphors used for teaching and training[20]. This provided a framework of the nature, structure, and aspects of metaphors for reviewing awareness of metaphor use, evaluation of metaphors, and arguments for using metaphors in training within the scope of teaching English as a second language. In a similar vein, this paper will analyze and compare metaphors using the same definitions laid out by Lakoff.

3. THE STUDY

One-on-one, semi-structured interviews of approximately 15 minutes were conducted with each participant via telephone, VOIP, or face-to-face. Each audio-recorded interview consisted of 7 questions, along with any follow-up questions for clarification.

3.1 Participants

Participants were 10 instructors that teach or have taught CS1 at a university level in the United States. The participants were found through recommendations by the researchers' colleagues and a posting on the SIGCSE mailing list, and volunteered to participate. Participants' mean number of years of Computer Science teaching experience was 12.9 years, with a minimum of 2 years and a maximum of 40 years.

3.2 Study Protocol

Interviewers asked all participants the following questions:

1. At what levels have you taught Computer Science?
2. How long have you been teaching Computer Science?
3. Please tell me about a metaphor you have used in your teaching to help explain some idea or technique from Computer Science.
4. Can you describe how the metaphor explains the CS topic?
5. What are common questions that students have?
6. Where does the metaphor start to break down?
7. How do you handle that when you're teaching?

Follow-up questions were asked as appropriate; in some cases, multiple metaphors were discussed. Upon request by participants, the questions and motivation for the interview were provided in advance, allowing them time to contemplate their responses.

4. FINDINGS

All but two participants offered at least one metaphor. Each interview was analyzed utilizing the previously discussed theory of conceptual metaphors. Given the defining factors and features of metaphors under Lakoff's definition, we were able to investigate the metaphorical mappings, how these mappings hold, target domains of the metaphors, and therefore make conjectures on how effective the metaphors used can be. Table 1 details the mappings of target and source domains of the metaphors discussed by participants.

The components of metaphors used by instructors can be categorized by what they ask students to focus on. Categories include a process (e.g., a recipe mapping to a program), a whole action (e.g., a verb mapping to a function), a collection of spatial relationship between two locations (e.g., consecutive street addresses mapping to an array), a physical relationship between two objects (e.g., a box and its contents mapping to a variable and its contents), and a relationship between a value and a location (e.g., a card in a card catalog and the book it describes mapping to a pointer and its target). Many metaphors contain components in multiple categories. For example, a portkey was offered as a metaphor for a return statement in response to a large proportion of students incorrectly responding to exam questions on the topic. Many students believed that multiple return statements in one function enumerated a set of values that are returned, rather than a single value that is returned once the return statement is executed. This shows that some metaphors are selected due to particular student misunderstandings. This metaphor also contains one component that describes a relationship between two physical locations, one that describes an action (immediately leaving one context for another), and one that describes a value standing for something else (some knowledge in an item representing another location).

4.1 Atomic vs. Complex Metaphors

Another feature found was the difference in scope of the metaphors. Some metaphors mapped a single feature between the source and target domains rather than a complex, algorithmic process. These algorithmic processes have multiple mappings between the source and target domains, some of which are explicitly elaborated in the intended usage. The most striking example was one participant's use of the portkey from *Harry Potter*. In *Harry Potter*, a portkey is a magical object that when touched, returns a person to a specific location. This was employed as a metaphor for return statements. The participant stated that the usage of the metaphor arose from confusion among the students regarding the number of values returned by a function that had multiple return statements, as in the following pseudocode:

```
def foo():
    if bar == 1:
        return 1
    else if bar == 2:
        return 2
    return False
```

Source Domain	**Target Domain**
Harry Potter portkeys	Return statements
Cookbooks and recipes	Functions, subroutines, looping, etc.
Cooking	Functions, encapsulation, storing data, etc.
Driving a car	Sequence of execution in code
Verbs	Functions
The Grim Reaper	Garbage collector
Learning a human language	Learning to program
Inception	Recursion
Street addresses	Recursive array indexing
Boxes	Variables and arrays
Containers	
Bookmarks	Return addresses
Telling a story	Algorithms
Card catalogs	Pointers
File folders	
Zombies	
Splitting students into two groups	Relationship of memory and CPU
Prioritizing customers for car repairs	Heaps
Campus dining hall dishes	Stacks and heaps
Cookie cutters and cookies	Classes and objects

Table 1: Source and target domains for metaphors reported by participants.

The portkey metaphor itself only deals with one specific action and is rooted in the fact that the source domain has additional, minor components, that are related to one predominant property, that it transfers context. It does not explain sequential control flow in programming in general but only one atomic action, as opposed to the aforementioned metaphors that deal with an entire process and are therefore more complex in their mappings with the source domain.

Boxes and containers were also atomic metaphor source domains that were noted on three separate occasions by different participants. The domains were all mapped to the same target domain, variables, in the same manner. Containers were presented as being able to hold an object (or value) in a simple one-to-one mapping. The only extension of this metaphor was the target domain of an array. This is arguably still the same metaphor repeated, and still maintains its atomicity.

Another atomic metaphor very similar to the portkey metaphor was the use of the source target of a physical bookmark mapped to a target domain of a return statement. The participant who discussed this metaphor stated that it only had one simple use, and the source and domain mappings could no longer be expanded. It was noted that adding another feature, such as a value to a return statement, invalidated the metaphor. Despite this, the participant still noted that this atomic metaphor was useful as a subset of the cookbook source domain.

While many atomic metaphors may be seen as simple, they have been reported to still be of value – whether considered individually or as a subset of a larger metaphor. It was reported that these metaphors never elicited any questions from students. This may indicate that atomic metaphors are much easier for students to grasp.

4.2 Usage of Same Source Domain Mappings

We also found that the same source domains were mapped to more than one target domain. One participant with over 10 years of experience explained that a cookbook provided an excellent way to teach new students about the techniques of programming during their first two weeks of class. These mappings include a recipe's steps being mapped to imperative commands, acting differently based on a food property being mapped to selection, stirring until smooth being mapped to looping, sub-recipes (e.g., a sauce for a dish) being mapped to subroutines, a book mark being mapped to a return address, and referencing a recipe written in full only once mapping to abstraction, easily updated code, and code reuse. This participant admitted that after the first two weeks the metaphor wasn't of much use, but expressed that students had few problems with it and that it helped them get used to a new set of ideas.

Another participant, also with greater than 10 years of experience, felt that the recipe metaphor fell so short as to not be very useful. The participant explained that while students have a familiarity with cooking, they do not have one with programming language primitives. While this line of thinking led the first participant to use the cooking metaphor, the second expressed the need to identify with the discomfort students were feeling about learning a new language. This participant compared learning to program to learning a human language.

The use of the same source domains with mappings to different target domains is notable as elements can represent various aspects of different target domains. This may indicate that some source domains may be more useful than others, and may even be used repeatedly throughout an instructor's pedagogy. They may also be richer and more complex, allowing for more mappings between source and target domains.

4.3 Usage of Same Target Domain Mappings

During the study, it was also found that the same target domain was mapped to more than one source domain. Several participants described how they used metaphors to explain pointers, though the source domains that they drew from – and therefore the mappings to the target domain as well – were different. One participant used the source domain of a card catalog, mapping the cards to the pointers themselves and shelves to memory, with the data residing on the shelves. Another participant used a somewhat similar source domain of the organization of file folders, with indices of both which drawer the paper (i.e., data) was stored, and which folder within that drawer. Both of these metaphors mapped to a physical source domain with similar features, such as objects containing other objects and a referent to where the "data" could be found. However, the file folders source domain is slightly more extensible, as it can refer to double indirection. While the target domain was the same as the previous two metaphors, a third participant described pointers with the source domain of zombies. This was described as zombies containing nothing in their head, and can just point to "real" data, or humans. Unlike the first two source domains for pointers, this domain does not use inanimate objects that students could physically interact with, but instead actual agents that the students themselves could represent.

The first source domain of a card catalog was cited as a poor source domain, due to being a concept that students may be less familiar with. However, the second source domain of file folders was perceived to be easily understood by the students, and it was cited that the physical aspect and extensible nature of this domain led to this understanding. The third source domain followed in the same pattern, as it was described as extensible as well. It was reported that as students had certain questions, they were easily answered within the domain (e.g., "What does a zombie initially point to?" which was answered as zombies are useless until they are assigned a victim). However, the zombie and file folder source domains may promote different levels of understanding, as one contains actual agents instead of inanimate objects. Though the source domains varied, there seemed to be similar features present across all domains.

The use of the same target domains with mappings to different source domains is notable as it shows that some target domains can be mapped to more than one source domain. While this may be simply a difference in pedagogical strategies between respondents, it may also show insight into these metaphors. The difficulty in understanding target domains might lead to a variety of source domains being used for a single target, as instructors attempt to find a useful tool to mitigate student confusion, but this is yet to be studied. It may also serve as a means of comparison between source domains.

4.4 Limitations of Metaphors

While limitations of any pedagogical strategy may be seen as a negative aspect, understanding the limitations of metaphors in Computer Science can aid in making them more effective. When asked questions 6 and 7 of the protocol, many instructors responded by stating that once a metaphor has reached its limit, they no longer use it in the classroom. Others, however, reported that sometimes these limitations in what a metaphor can explain can sometimes lead to more confusion. For example, one respondent noted that they believed students attempted to extend a metaphor to a point where they believed false, additional mappings between domains. Limitations were found across many different facets of the metaphors discussed. Limitations were found regarding the source domain, target domain, and/or mappings of these domains. For example, source domains were reported to sometimes be obscure (e.g., certain films that students may have not seen), and mappings between domains were reported to sometimes be limited (e.g., the variables are containers metaphor is limited in extensibility). Other limitations were aspects of metaphor defined under Lakoff's definition of metaphor. Table 2 lists some of the main limitations both reported by participants, and others of note.

Out of date or obscure source domains can be a large limitation of using metaphors in the classroom. For example, the source domain of a card catalog may be foreign to many students as many students may have not used a card catalog previously. If students are not familiar enough with the source domain of a metaphor, their understanding of the concept may not be increased. Instructors reported that

Out of date/obscure source domains	Cultural differences
Mapping to low-level details often fails	Increases in target domain complexity
Limited mappings	None reported

Table 2: Common limitations found across responses.

the most effective metaphors were those that had source domains that students were very familiar with. One respondent in particular noted that their source domains for their metaphors often changed once the domains were no longer part of popular culture. The participant stated that certain metaphors, such as the source domain *Inception* mapping to the target domain of recursion may not be used much longer by the participant. This is due to the participant recognizing the limitations of the metaphor, as the source domain will eventually be less familiar to students, and a new source domain would be used in its place. It was reported that this has occurred in the past, and changing to more modern source domains has always been the participant's response to out-of-date source domains. Cultural differences are also a limitation of metaphors as their source domains, as students from different cultures may not have any familiarity with the source domain.

Another reported limitation of metaphors is that, at a certain point, they begin to break down and can no longer be expanded to further explain a concept (or related concepts). Three participant instructors stated that while metaphors can offer a high-level framework for a concept, they cannot be extended to low-level features of the target domain (such as the implementation of a data structure). This is an inherent feature of metaphors, as source and target domains are not identical. Because metaphors are only partial mappings between source and target domains, they may break down as learners explore a concept in more detail, or examine a concept as part of a larger system.

4.5 Students' Understanding of Metaphors

Three participants stated that there were no limitations of their metaphors. No participants were able to give examples of students' misunderstandings caused by confusion about metaphorical mappings, aside from some of them mentioning the places they tend to break down. It has been previously reported that all metaphors do in fact break down[11], leading us to hypothesize that either *a*) there is something special about metaphors in CS that enable them to withstand breakdown or *b*) instructors have poor understandings of their students' thinking about the metaphors that are used in their courses, including how metaphorical breakdown leads to misunderstandings of CS1 concepts. Given that *b* seems far more likely, there is a need for research into what these misunderstandings are and how teachers can better reason about student thinking around them in the classroom. If limitations are known, certain metaphors may be abandoned or limitations disclaimed to students in order to reduce confusion. Further research involving teaching assistants and students will help to illuminate where gaps in teachers' knowledge exist, and how they might be closed.

5. CONCLUSIONS

This preliminary study provides initial insights into the use of conceptual metaphors in Computer Science education. While metaphor may be a very useful pedagogical tool, it is also one that must be used with knowledge of its limitations. What those limitations are for different Computer Science topics, and for mappings between particular source and target domains, would, thus, seem to be vital Pedagogical Content Knowledge. Given our findings that CS1 teachers use a wide variety of metaphors but that those same instructors were generally unable to explain common questions and misunderstandings students had when using those metaphors, further research is needed into student thinking about CS1 topics when scaffolded by metaphors. This may aid in broader understanding of the range of metaphors that are currently used, and deeper understandings of which are more effective in presenting new concepts, and why.

6. FUTURE WORK

One direction for future research is to analyze how metaphors break down. Further insight into why, when, and how some metaphors break down is needed. Three participants noted that metaphors they used never ceased to explain the target domain, however attempting to extend these metaphors past their limits will likely show that, in fact, all metaphors do break down. Additionally, determining exactly when certain metaphors fall apart may show common features of the metaphors themselves, and may help identify when other metaphors lose their effectiveness, ultimately supporting a theory of breakdown which might predict student misunderstandings. Such a theory might encompass different dimensions which affect metaphor breakdown. The mapping between source and target domains may not be strong initially, or the similarities between domains may be shallow.

Despite the limited size of this preliminary study, we found shared target and source domains in instructor metaphor use. Examining the use of the same source domains may offer insight into how effective these domains may be for students, as they map to the same target domain. Also of note is the use of the same target domain, as the overlap in mappings may also have an effect on students' understanding. Another similar observation was that the complexity of mappings can affect the implementation of metaphors in instruction. Atomic metaphors may have less potential for future confusion due to their simplicity of mappings. However, these mappings may be limited and less extensible. Future studies may show if these features of source and target domains continues to be a dominate feature of metaphor use in CS1.

Another area of interest would be examining the views of students with regards to certain metaphors used in the classroom. As students' understanding is a facet of Pedagogical Content Knowledge, it may offer further insight regarding how useful metaphors can be in Computer Science. While three participants responded that no students had any follow up questions, that does not necessarily rule out any confusion or misunderstanding that the students may have. Instructors may also hold false assumptions of some metaphors used, and may overestimate students' understanding. Additionally, metaphors may break down to students much earlier than some participants believed. Students may also attempt to extend metaphors on their own, not knowing the limitations of these metaphors. Gauging student understanding may be accomplished through the use of clickers for instant feedback for instructors, or student interviews. Studying the classroom as a whole may be most beneficial, as both sides of understanding and pedagogy of the use of metaphors could be examined simultaneously through the viewpoints of instructors, teaching assistants, and students.

Not only would this further research be of academic interest (supporting our understandings of how learning happens in Computer Science courses), but it is also of crucial practical importance. Research on metaphor and other pedagogical techniques in CS1 could produce a useful, empirically grounded knowledge base[3] for teachers illuminating the most productive ways to support students' learning of different computational ideas. At minimum, it should offer a guide to what the most effective metaphors are to explain particular ideas, and ideally should go well beyond this to include guidance on how to construct in-class activities for students that push them to explicate, refine, and build upon their emerging computational ideas. Such a knowledge base could be a nexus of strong partnership between researchers of Computer Science education and teachers that could not only advance research on learning, but improve outcomes for a large number of students.

7. ACKNOWLEDGMENTS

We would like to thank all of the participants for offering their time to this study.

8. REFERENCES

[1] P. Boero, L. Bazzini, and R. Garuti. Metaphors in teaching and learning mathematics: a case study concerning inequalities. In *PME Conference*, volume 2, pages 2–185, 2001.

[2] R. Frost. Education by poetry. *Selected Prose of Robert Frost*, pages 33–46, 1966.

[3] J. Hiebert, R. Gallimore, and J. W. Stigler. A knowledge base for the teaching profession: What would it look like and how can we get one? *Educational researcher*, 31(5):3–15, 2002.

[4] R. Jiménez-Peris, C. Pareja-Flores, M. Patiño-Martínez, and J. Á. Velázquez-Iturbide. The locker metaphor to teach dynamic memory. In *ACM SIGCSE Bulletin*, volume 29, pages 169–173. ACM, 1997.

[5] V. Kind. Pedagogical content knowledge in science education: perspectives and potential for progress. *Studies in science education*, 45(2):169–204, 2009.

[6] G. Lakoff et al. The contemporary theory of metaphor. *Metaphor and thought*, 2:202–251, 1993.

[7] G. Lakoff and M. Johnson. *Metaphors we live by.* University of Chicago press, 2008.

[8] G. Lakoff and R. E. Núñez. *Where mathematics comes from: How the embodied mind brings mathematics into being*. Basic books, 2000.

[9] A. M. Leavy, F. A. McSorley, and L. A. Boté. An examination of what metaphor construction reveals about the evolution of preservice teachers' beliefs about teaching and learning. *Teaching and Teacher Education*, 23(7):1217–1233, 2007.

[10] M. A. Martinez, N. Sauleda, and G. L. Huber. Metaphors as blueprints of thinking about teaching and learning. *Teaching and Teacher education*, 17(8):965–977, 2001.

[11] S. J. McCarthey and E. B. Moje. Identity matters. *Reading Research Quarterly*, 37(2):228–238, 2002.

[12] W. W. Milner. A broken metaphor in java. *ACM SIGCSE Bulletin*, 41(4):76–77, 2010.

[13] R. E. Núñez. Mathematical idea analysis: What embodied cognitive science can say about the human nature of mathematics. 2000.

[14] N. A. Paris and S. M. Glynn. Elaborate analogies in science text: Tools for enhancing preservice teachers' knowledge and attitudes. *Contemporary Educational Psychology*, 29(3):230–247, 2004.

[15] R. T. Putnam, D. Sleeman, J. A. Baxter, and L. K. Kuspa. A summary of misconceptions of high school basic programmers. *Journal of Educational Computing Research*, 2(4):459–472, 1986.

[16] E. Seung, L. A. Bryan, and M. P. Haugan. Examining physics graduate teaching assistants' pedagogical content knowledge for teaching a new physics curriculum. *Journal of Science Teacher Education*, 23(5):451–479, 2012.

[17] L. S. Shulman. Those who understand: Knowledge growth in teaching. *Educational researcher*, 15(2):4–14, 1986.

[18] L. S. Shulman. Knowledge and teaching: Foundations of the new reform. *Harvard educational review*, 57(1):1–23, 1987.

[19] G. P. Thomas and C. J. McRobbie. Using a metaphor for learning to improve students' metacognition in the chemistry classroom. *Journal of Research in Science Teaching*, 38(2):222–259, 2001.

[20] S. Thornbury. Metaphors we work by: Efl and its metaphors. *ELT Journal*, 45(3):193–200, 1991.