Financial support. The funds to conduct this research were provided by the research program Ethics of Socially Disruptive Technologies (ESDiT), which is funded through the Gravitation program of the Dutch Ministry of Education, Culture, and Science and the Netherlands Organization for Scientific Research (NWO grant number 024.004.031).

Competing interest. None.

References

Abernethy, K., White, L., & Wickings, E. (2002). Hordes of mandrills (*Mandrillus sphinx*): Extreme group size and seasonal male presence. *Journal of Zoology*, *258*(1), 131–137.

Chapman, C. A., & Chapman, L. J. (2000). Determinants of group size in primates: The importance of travel costs. In S. Boinski & P. A. Garber (Eds.), *On the move: How and why animals travel in groups* (pp. 24–41). University of Chicago Press.

Chapman, C. A., & Teichroeb, J. A. (2012). What influences the size of groups in which primates choose to live? *Nature Education Knowledge*, *3*(10), 9.

Dunbar, R. (1992). Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, *22*(6), 469–493.

Ganas, J., & Robbins, M. M. (2005). Ranging behavior of the mountain gorillas (*Gorilla beringei beringei*) in Bwindi Impenetrable National Park, Uganda: A test of the ecological constraints model. *Behavioral Ecology and Sociobiology*, *58*, 277–288.

Janson, C. H., & van Schaik, C. P. (1988). Recognizing the many faces of primate food competition: Methods. *Behaviour*, *105*, 165–186.

Krause J., & Ruxton, G. D. (2002). *Living in groups*. Oxford University Press.

Lehmann, J., & Boesch, C. (2003). Social influences on ranging patterns among chimpanzees (*Pan troglodytes verus*) in the Taï National Park, Côte d'Ivoire. *Behavioral Ecology*, *14*(5), 642–649.

Markham, A. C., Gesquiere, L. R., Alberts, S. C., & Altmann, J. (2015). Optimal group size in a highly social mammal. *Proceedings of the National Academy of Sciences*, *112*(48), 14882–14887.

Meder, A. (2013). Great ape social systems. In W. Henke & I. Tattersall (Eds.), *Handbook of paleoanthropology* (pp. 1–34). Springer.

Snaith, T. V., & Chapman, C. A. (2005). Towards an ecological solution to the folivore paradox: Patch depletion as an indicator of within-group scramble competition in red colobus. *Behavioral Ecology and Sociobiology*, *59*, 185–190.

Snaith, T. V., & Chapman, C. A. (2007). Primate group size and socioecological models: Do folivores really play by different rules? *Evolutionary Anthropology*, *16*, 94–106.

Teichroeb, J. A., & Sicotte, P. (2009). Test of the ecological-constraints model on ursine colobus monkeys (*Colobus vellerosus*) in Ghana. *American Journal of Primatology*, *71*, 49–59.

Tomasello, M. (2020). The adaptive origins of uniquely human sociality. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *375*, 20190493.

Computational theories should be made with natural language instead of meaningless code

Peter DeScioli

Department of Political Science, Stony Brook University, Stony Brook, NY, USA
pdescioli@gmail.com; pdescioli.com

doi:10.1017/S0140525X23001231, e332

Abstract

The target article claims that we should speak in code to understand property, because natural language is too ambiguous. Yet the best computer programmers tell us the opposite: Arbitrary code is too ambiguous, so we should use natural language for variables, functions, and classes. I discuss how meaningless code makes Boyer's theory too enigmatic to properly debate.

Boyer has made some of the deepest insights into the psychology of property (Boyer, 2015, 2018). The target article probably adds to these insights if someone can decipher the scrolls of cryptic notation, such as P() tags, Min(A), and Brep[Xrep[Min(A)]].

The article goes wrong with the notation introduced in Table 2. On its face, we should be skeptical that speaking in code will help us understand property. Throughout academia, scholars are mesmerized by secret codes, encrypted texts, and appearances of technicality. So we need to ask whether an artificial code is actually useful or yet another indulgence in scholarly solipsism.

On this point, the author provides little justification: a single sentence which claims that the notation helps by "avoiding the ambiguities of natural language glosses" (target article, sect. 5.2). The claim echoes a common refrain among modelers and jargon peddlers of many persuasions. Natural language is too ambiguous, they say, and arbitrary terms like Min(A) and Brep[] make ideas clearer. But this claim is dubious.

Cognitive scientists know that natural language is an evolutionary marvel of expressive power. Yes, language can be ambiguous; so can reality itself. But arbitrary code is far more ambiguous than language. Its meaning is hidden in the author's mind, it lacks the advantages of words and grammar, and it is untested by the demands of daily use. Besides, the author's notation does not escape natural language. Instead, readers must memorize a table of definitions in language, and then mentally substitute words for notation while reading, adding ambiguity in translation.

Consider the advice of professional programmers, whose computational programs need to actually run, and who therefore put the greatest premium on precision and clarity. Programmers constantly complain that they cannot understand each other's code, due to the bad habit of arbitrary notation. The best programmers tell us the remedy is natural language. For instance, the prominent guide *Clean Code* advises programmers to use informative names for variables and functions (Martin, 2009). In chapter 2, "Meaningful names," Martin advises programmers to replace ambiguous variables like "d" with names like "daysSinceCreation," and to replace general labels like "theList" with specifics like "gameBoard" (p. 18). Google and Microsoft agree (Fig. 1). More particularly, the book recommends noun phrases for classes and objects, such as "Customer" and "addressParser," and verb phrases for functions and methods, such as "deletePage" and "postPayment" (p. 25).

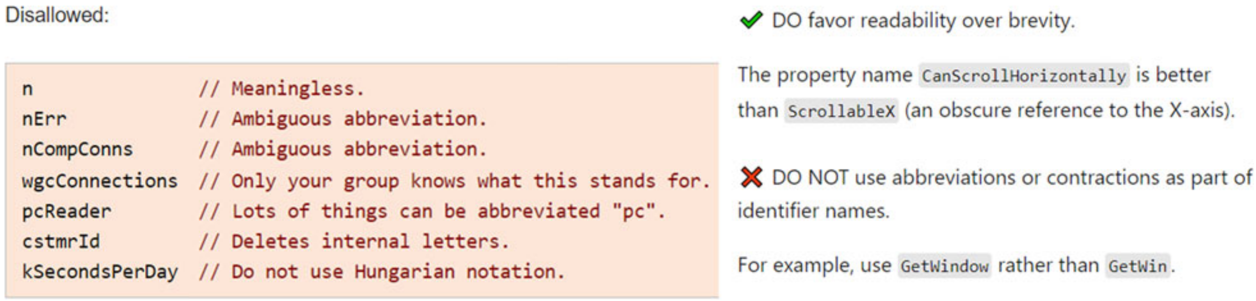
In fact, Martin (2009) applies evolutionary psychology to make the case. The section, "Use pronounceable names," explains:

> Humans are good at words. A significant part of our brains is dedicated to the concept of words. And words are, by definition, pronounceable. It would be a shame not to take advantage of that huge portion of our brains that has evolved to deal with spoken language. So make your names pronounceable. (pp. 21–22)

Continuing, Martin explains that arbitrary strings are difficult to discuss.

> If you can't pronounce it, you can't discuss it without sounding like an idiot. "Well, over here on the bee cee arr three cee enn tee we have a pee ess zee kyew int, see?" This matters because programming is a social activity. (p. 22)

Indeed, cognitive science is also a social activity, requiring a fluent vocabulary. We cannot fully debate a theory while tongue-



Disallowed:

```
n               // Meaningless.
nErr            // Ambiguous abbreviation.
nCompConns      // Ambiguous abbreviation.
wgcConnections  // Only your group knows what this stands for.
pcReader        // Lots of things can be abbreviated "pc".
cstmrId         // Deletes internal letters.
kSecondsPerDay  // Do not use Hungarian notation.
```

✓ DO favor readability over brevity.

The property name `CanScrollHorizontally` is better than `ScrollableX` (an obscure reference to the X-axis).

✗ DO NOT use abbreviations or contractions as part of identifier names.

For example, use `GetWindow` rather than `GetWin`.

Figure 1 (DeScioli). Rules for naming from Google's Style Guide, left, and Microsoft's Naming Guidelines, right (https://google.github.io/styleguide/jsguide.html#naming, https://learn.microsoft.com/en-us/dotnet/standard/design-guidelines/general-naming-conventions).

tied in twisters like, "how do pee parentheses tags relate to bee rep open bracket min parentheses ay close bracket?"

Just as the programmers warned, Boyer's arbitrary code spawns confusion. First, what is a P() tag? Does the "P" allude to "property," "possession," or something else? Is a P() tag the same as a P() representation, P() relation, or P-intuition, and are P-cues the same as P-indices? If so, synonyms multiply the ambiguous jargon. The author seems to prefix "P-" to any word, blurring technical terms with ordinary words bearing an arbitrary prefix. I cannot tell whether the model of P() in Figure 1 makes specific claims because the inputs and outputs are open-ended.

Second, what is Min(A)? If it means that A would cooperate with someone, then we have adjectives like *cooperative*, *friendly*, and *nice*, which has served as a technical term for cooperative strategies (Axelrod, 1984). Nearby, "Brep[]" seems to mean that B believes something, which can be expressed with verbs like *think* or *believe*. "Brep[Min(A)]" in natural language means "B thinks A is friendly." Overall, I cannot tell whether the code introduces new ideas about cooperation or expresses familiar ideas in Min(A) and Brep[].

Third, what is an L() tag? What does the "L" allude to? L() combines the previous elements, compounding the confusion. The discussion enumerates lists of familiar ideas expressed in code and tortuous language. For instance, "A is in a P() relation with thing t" (target article, sect. 7.2.2) is not more precise but merely a circumlocution for "someone owns something" or "someone has something," convoluted by the extraneous concepts of *be*, *in*, *relation*, and *with*.

The main point of L() seems to be that people weigh the value of a cooperative partner when deciding whether to respect or defend that person's property. What is notable about this claim? Later, the author argues that thieves do not infer their victims' ownership (target article, sect. 8.2.1). But then the thief would not hide their thievery. A skilled thief needs the ability to judge ownership impartially to anticipate the moves of owners and observers.

Boyer acknowledges these terms are meaningless and argues that saying what they mean "would be misguided and misleading" because they refer to "mappings of inputs to outputs" (11.1). This amounts to saying, "I have a theory but it would be misleading to tell you what it is." That sort of theory is incomprehensible and thus unfalsifiable. Again, the author's justification assumes that meaningless code is less ambiguous than language and does not require translation to language to be understood. This assumption does not hold for real programs that map inputs to outputs, nor for theoretical programs in the mind.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

Competing interest. None.

References

Axelrod, R. (1984). *The evolution of cooperation*. Basic Books.

Boyer, P. (2015). How natural selection shapes conceptual structure: Human intuitions and concepts of ownership. In S. Laurence & E. Margolis (Eds.), *The conceptual mind* (pp. 185–200). New Directions in the Study of Concepts, The MIT Press.

Boyer, P. (2018). *Minds make societies*. Yale University Press.

Martin, R. C. (2009). *Clean code*. Pearson Education.

A cooperative–competitive perspective of ownership necessitates an understanding of ownership disagreements

Margaret Echelbarger[a] and Stephanie M. Tully[b]

[a]College of Business, Stony Brook University, Stony Brook, NY, USA and [b]Marshall School of Business, University of Southern California, Los Angeles, CA, USA
margaret.echelbarger@stonybrook.edu
tullyste@marshall.usc.edu

doi:10.1017/S0140525X23001486, e333

Abstract

Boyer's cognitive model of ownership, based on cooperation and competition, underscores the importance of studying disagreements in ownership. We argue that exploring the factors that can lead to different perceptions and experiences of ownership will uniquely inform our understanding of legal, psychological, and perceived ownership beliefs.

Boyer proposes a cooperative/competitive framework to understand ownership, arguing that ownership intuitions are derived from competing cognitive systems grappling with the need to allocate resources across people in a society. The idea that cooperation and competition serve as drivers of ownership (legal, psychological, and perceived) brings to light an underexplored area of research on ownership: disagreements in ownership. Indeed,