Temporally Learning Dependent Types

The magazine article wants readers to be hooked on the magazine, while the paper wants people to be hooked on the concept of dependent types. For example, in “Dependent Types: A New Paradigm?” by Paul Callaghan, he references previous months many times in the introduction, followed by setting up future months in the conclusion. However, Edwin C. Brady's paper “Programming and Reasoning with Algebraic Effects and Dependent Types” mentions other sources, previous work, and future work so the reader has more opportunities to understand the material. Both of these works center around the programming language Idris and its main feature, dependent types. The rhetorical differences between these works are where the basic knowledge comes from (past), how the author increases the reader's knowledge (present), and what the reader is expected to do after (future).

Callaghan's article leans on the previous issues of the magazine PragPub for support, whereas Brady's paper looks to outside resources for clarity on concepts. The article's first sentence references the previous issue. It uses this not only to hook new readers, but also to provide a transition for continuing readers. It helps those familiar with the previous issue by refreshing their memories. In the case of the paper, more than 80% of the references are from other authors. The paper utilizes these different resources because many of the base concepts have already been explored in detail.

Once the previous connections have been made, Callaghan goes on to forge new connections, while Brady seeks to further the understanding of those connections. The magazine article seeks to take the reader on a journey through learning dependent types by first giving a “Hello World” program[2], then a simple proof with holes[2], followed by a comparison with tests[2], and ending with the author offering help in their online forum[2]. Each example is progressively more difficult. And Callaghan offers to make a connection with the reader by mentioning PragPub's online forum. The magazine needs to make new connections because it needs readers to be interested in order to continue publishing. At the end of the day, PragPub is a business, so it must have interested readers in order to remain profitable. The paper, in contrast, exists to spread information. Brady strengthens the connections by exploring where monad transformers become clunky, offering up algebraic effects and dependent types as an alternative, implementing a simple interpreter with algebraic effects and dependent types, and finally showing that algebraic effects are a less powerful abstraction than monads in general[1]. One of Brady's goals is to ensure that readers understand the connections with previous work and the motivation for algebraic effects.

With the reader fully vested in the connections, Callaghan entices the reader to find new connections next month, meanwhile Brady encourages the reader to explore new connections at their leisure. The magazine article uses the cliché, “...tune in next month”[2] followed by a list of interesting forays into dependent types to hook readers into coming back. This is good, because this article barely touches the surface of dependent types and in some sense makes it seem like a novelty rather than a mind altering shift in computer science. The future articles provide more interesting examples that explore more of what motivates this branch of type theory. Brady's paper ends by explaining that algebraic effects and dependent types are still in their infancy[1]. It also lists some of the shortcomings of the current implementation of algebraic effects and offers suggestions for improvements[1]. Finally, it mentions that the entire implementation is written in Idris as a library rather than as a language extension[1]. This is a point which should be more celebrated than it is in the paper, and is almost entirely because of dependent types, meaning that Idris is more expressive than vast majority of languages that currently exist.

References

1. E. Brady. Programming and Reasoning with Algebraic Effects and Dependent Types. In *Proceedings of the 18th ACM SIGPLAN International Conference on Functional Programming*, pages 133-144. ACM, 2013.
2. P. Callaghan. Dependent Types: A New Paradigm? *PragPub*, #46, pages 33-42 2013.