How are Explanations Explained? Towards an Explanation-Oriented Programming DSL

Eric Walkingshaw
School of EECS
Oregon State University
Corvallis, Oregon, USA

Jeffrey Young
School of EECS
Oregon State University
Corvallis, Oregon, USA

***Abstract*—ABSTRACT TRY 1: Slow to build, and then, Crescendo!**

Programs often produce incorrect results. In such a scenario how should a user respond? Explanation-Oriented Programming's [XOP] raison d'etre is to provide a user with the "how" and "why" in such a scenario. It is computationally trivial to provide the "how", but the "why" remains an area of active research in XOP literature. We make two observations: 1) Theories of Explanation are not the lingua franca of computer scientists and 2) Pedagogic Theory is rife with examples of typologies that quantize Teacher-Pupil explanations and interactions in the classroom. In this paper we appropriate one such typology to academic explanations of common algorithms in CS curricula. Utilizing this typology, we generate a data set from real world, academic, didactic documents. From this data, we observe patterns of teaching; known as *teaching cycles* in pedagogic literature. Finally, we show how such typologies form the basis for an Explanation-Oriented Domain-Specific Language of Algorithms.

ABSTRACT TRY 2: This one is fast and hard. It's a punk song from the late 80s a la Operation Ivy.

We propose a Typology of algorithmic explanation which allows one to quantize pedagogic explanations of common algorithms in Computer Science curricula. Utilizing this Typology we show that many real world, didactic, academic documents exhibit patterns of explanation; known as *teaching cycles* in pedagogic literature. Finally, we propose that such typologies are imperative to the design and construction of Explanation-Oriented Domain Specific Languages.

ABSTRACT TRY 3: This one has the flow right. It's a Sleater-Kinney Song from 95. Could probably use some word-smithing.

The goal of Explanation-Oriented Programming is to provide a user with an explanation for incorrect or confusing output. But where is such an explanation to come from? In this paper we review Pedagogic Theory to compose a Typology that is able to quantize academic explanations of common algorithms in CS curricula. We show that by using such a Typology to encode academic, didactic documents, a language designer is able to determine common pedagogical patterns of explanation; known as *teaching cycles* in educational literature. Finally, we propose that such a Typology forms the foundation for the design and construction of an Explanation-Oriented Domain Specific Language of Algorithms.

I. Introduction

This demo file is intended to serve as a "starter file" for IEEE conference papers produced under LaTeX using IEEEtran.cls version 1.8b and later. I wish you the best of success.

mds

August 26, 2015

A. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

II. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.