

Linfeng Li

3/21/2021

CS 540

### Critique Paper: Adding Interactive Visual Syntax to Textual Code

The article, “Adding Interactive Visual Syntax to Textual Code,” by the authors, Leif Andersen, Michael Ballantyne, and Matthias Felleisen, presents an extensive and clearly articulated argument analyzing the application of interactive visual syntax as an extension to the commonly used textual programming code. In its context, the text scrutinizes the topic from various realms, including providing an overview of the theatrics involved in the use of textual code; how this could be improved through the implementation of interactive syntax, and proceeds on to provide a substantial discussion of what involves visual syntax code. Consequently, this text aims at providing critical analysis to the article, “Adding Interactive Visual Syntax to Textual Code,” by highlighting its nature and analyzing the significance it plays in contributing to the topic regarding visual syntax.

In the article, the authors take a unique approach in vouching for the necessity of implementing Interactive Visual Syntax to the already existing programming technique that involves the use of textual code in the development of working automated solutions for existing problems. Notably, the onset of the article begins by explaining the theatrics of performance for textual code, which includes the transformation of geometrical ideas into lines of code based on a particular programming language. Subsequently, the article highlights how this poses a disadvantage to thoughts transformed into lines of code since this often hampers the geometric sense of the development process, including the use of tables, hierarchical structures, objects,

and graphs among others. As a result, this leads to the well-stipulated thesis of the article, which includes the proposition for Visual Syntax as a novel mechanism aimed at providing a convenient means of extension for the programming languages.

To actualize the article's main goal, which includes a highlight on the significance of Visual Syntax, the paper presents a working proposition that would solve the existing problem with textual programming techniques. The authors acknowledge that programming techniques involving textual code often pose a challenge to future developers to the extent of reducing productivity since this requires that a developer develop an in-depth comprehension of code in order to reconstruct thoughts involving geometrical relationships to tables, objects, trees, and graphs from linear text. In the same section, the paper then proceeds to highlight how Visual syntax provides an interactive solution to the construction of a design. As a result, this proves the importance of using the graphical syntax as an extension to the linear text as a means for developing working solutions.

Subsequently, the text offers working demonstrations to prove the significance of Visual Syntax to extend textual code. Markedly, *tsuro tile*, a game that involves the use of graphs to provide output for unit tests is highlighted as an example where features provided by Visual Syntax enhance the use of linear text in constructing instances through user interface actions such as keystrokes and menu selections to develop graphical representations of a particular niche. In addition, the text proves how Visual Syntax occurs in multiple natures including objects expanded sentences, sequential outputs, and multiple choices among others. For instance, the paper suggests the use of *racket*, a general-purpose programming technique, as a design methodology that involves the use of schemes to provide a working solution. The paper further proves the advantages of using Visual syntax such as the bridging of edit time and compile time

as both seem to run concurrently when using more complex Visual syntaxes such as form builders unlike in textual code where one has to write lines of code before implementing or running a developed solution. As a result, the section proves the powerfulness of Visual Syntax in decoding complex solutions, as it tends to generate linear texts of code from a developed graphical construct.

Finally, the authors prove the authenticity of their work by providing an evaluation of related works to Visual Syntax including the presentation of Video language documented at ICFP in Oxford. This includes working propositions that set out to provide interactive syntax for actualizing Graphical syntax, that relied on scripts and programs that had Visual elements embedded for Video production. This section also highlights shortcomings for prior graphical elements such as the re-using of existing libraries for developing Graphical User Interfaces and provides a workable solution for how this could be improved in the future such as the development of more interactive Visual Syntax such as canvas editors that have the capability to embed controls in multiple windows. To conclude, the authors provide references to their work proving to the readers of the paper's extensive coverage.

Overall, the authors present clearly articulated research that is both comprehensive and easy to understand. The brief overview, an extensive introduction explaining the history of textual programming and the detrimental effect it has on future developers' sets the foundation for the rest of the paper. Subsequently, the use of examples in explaining the theatrics of Visual Syntax and its significance as an extension of linear code together with references appeals to the readers and makes it easier to understand the topic. Ultimately, the paper proves the authors' prowess in the line of Visual syntax and graphical elements.

## Works Cited

Andersen, Leif, et al. "Adding Interactive Visual Syntax to Textual Code." *OOPSLA*, Article 222  
2020, pp.1-28.