Andrew Blinn / Statement of Purpose / Michigan ID 55418716

I intend to pursue a PhD at the intersection of PL and HCI. I see this as a natural extension of the work I've been doing over the last two years, most concretely expressed through the creation of a structured editing engine, Fructure, which I presented[1] at RacketCon 2019. I want to investigate applying formal semantics to underpin rich editing processes designed to capture, communicate, and augment user intent.

One recurring theme for me has been the expression of edit-time language semantics via augmenting target-language syntax with meta-syntactical affordances. These serve to reify UI primitives like cursors and selections at the language-level, acting as scaffolding to support semantically-aware transformation. My confidence in this approach has been bolstered through a literature search revealing past successes in exposing semantics via syntactic augmentation, including Felleisen et al's 'A syntactic theory of sequential control' and, more directly, Cyrus Omar et al's 'Toward Semantic Foundations for Program Editors'.

In connection, I want to explore the boundary between language and editor semantics. Learning Racket, I was fascinated by how, in the REPL, language behaviors around scope and control flow are subtly modified. Increasingly we interact with languages across many modalities; REPLs, notebooks, spreadsheets, steppers, debuggers, and embedded editors for rich multimedia literals. These modes often require ad-hoc changes to language behavior. Can we systematize these modifications in a language-oriented way, reshaping our interactions with complex systems as nested workflows of semantically-aware embedded editors?

While I am currently enjoying working on my own projects, attending conferences, and making my way through the existing literature, I have become aware of my own shortcomings, both in putting my ideas in the context of existing work, and in circumscribing their scope to produce incremental, publication-oriented contributions. Thus I am seeking a supervisor to help further my academic development.

Topical Background

Pursuing the above directions, I have been reading papers in structured & projectional editing, live programming, term-rewriting, incremental parsing and evaluation, type-directed editing, language-oriented programming, bidirectional transformation, and direct manipulation. I also have a supporting side interest in FRP, reactive UI, and stylesheet languages.

In particular, I have been reading (and running) works on live programming including Omar's Hazel, Jonathan Edward's Subtext experiments, and Sean McDirmid's APX, and in the direct manipulation space, Ravi Chugh's (and Brian Hemple's) work on Sketch-n-Sketch. I've also been following work in this space outside of academia including Lamdu, Unison, DarkLang, Chris Granger's EVE, and JetBrain's MPS projectional editing system. I've been trying to engage with the deeper history of these topics, including GRAIL and the early GUI work at Xerox Park, as well as Alan Kay's VPRI, and Bret

[1] Fructure: https://www.youtube.com/watch?v=CnbVCNIh1NA, https://github.com/disconcision/fructure

Victor's more recent demos in the same vein. As a Racket fan, Language-Oriented Programming angles are of particular interest, such as Leif Anderson's vision of DSL-specific editor extensions in Dr. Racket.

Over the last two years I’ve been engaging with some of the above researchers in this space, at conferences and through social media. Through these sources I’ve become viscerally aware of the amount of work done in this space, some of which dates back to the 1960s. Most of these attempts have gone nowhere, at least insofar as integration into the exoteric culture of programming is concerned. Often it's not clear to me whether this is because these ideas were flawed, perhaps in non-obvious ways, or if they were too early, or perhaps simply unlucky. Increasingly, academia feels like the best bet to avoid what appears to be a vicious cycle of reinvention and redundancy in this space. To this effect, I want to learn to better justify and situate my thoughts through publication.

My Background & Experience

Until recently a math student, in 2017 I co-organized a category theory reading group where I first encountered FP (via Haskell) and learned about UofT’s PL class, which I took the following semester. Learning about language semantics via macro systems and rewriting strongly evoked previously idle thoughts I’d had about making a general-purpose visual interactive environment for building intuition about mathematical abstractions. I was immediately hooked, pivoting on-the-spot towards CS. The following summer, I convinced (Toronto PL lecturer) Gary Baumgartner to supervise an independent research course on structured editing, a topic synergizing my interests in PL and math-oriented UI.

During the following four semesters, I TAed for the class, and was hired in the summer to design future assignments. That summer I also attended OPLSS; the three weeks I spent there left me with a sense of connection and cultural affinity with academic PL and PL-adjacent students and researchers. The following school year, I applied and was accepted to attend ICFP2018 on a PLMW scholarship, where I had the opportunity to meet Cyrus Omar and Ravi Chugh, as well as attend Leif Anderson's RacketCon workshop on her VideoLang DSL and its accompanying embedded editor.

I left eager for direct PL research experience, unfortunately hard to come by in Toronto. I did manage to find professor Marsha Chechik; though her focus is on Software Engineering, she had one fairly speculative PL-adjacent project on Haskell-based automated deep-lifting of algorithms to operate on variational data structures. This was based on previous work by Eric Walkingshaw, passingly familiar to me as his paper on projective editing in a variational context had come up in my literature search. Although our line of exploration has not yet resulted in publishable work, it gave me the opportunity to leverage some PL tech, including using PLT Redex to model a system of GHC rewrite-rules I created to accomplish the desired lifting.

During this time, I was contacted by Jay McCarthy, who had seen some of my Racket-based structured editing demos I had posted online. He invited me to come speak about Fructure at RacketCon 2019. I was already intending to attend the attached *How To Design Languages* summer school, and Jay's invitation galvanized me to firm up my various demos into a presentable product. My presentation on Fructure was well-received, and I've since been talking to a few researchers, including professor Jesse Tov at Northwestern, about incorporating it into a computing education setting.