The Plaid Programming Language

Peter Mikitsh, Ethan Peters, Steven Brunwasser

History and Motivation

- 2010: John Aldrich
- Permission-Based Programming Paradigm: objects with multiple references have each individualized read/write permissions. This is applied to achieve more effective implementation of concurrency.
- State-Based Programming: Like classes, typestates define a unique interface and unique instance variables.

Special Design Considerations

- Typestate orientation: an extension to OOP that better models the real world for object's state. States can be implemented clean and easily in Plaid.
- Concurrency by default: programmers must apply access permissions to objects.
- Heavily Java-based: Runs in a JVM; allows for standard calls to Java libraries.

Syntax, Semantics, & Pragmatics

- Syntax is borrowed heavily (nearly exclusively) from Java. However, Plaid does allow the use of duck-typing, which isn't a part of Java (unless reflection is applied).
- Pragmatics: This language should be applied in a large, state-heavy project.
- Semantics: The language employs typestates for compile-time checking.

Strengths

- Offers a reasonable to state-based programming.Typestates are easy to understand and would appear as an attractive option to experienced programmers.
- Plaid offers solutions and handling concurrency. Access permissions control how references have access to object’s data through read/write permissions.

Limitations

- The Access Permissions are not easy to summarize. They are complex to work with when it comes to understanding the proper ways one can split and merge permissions from references.
- Software Engineering Design Patterns exist for state machines in OOP languages. Creating a new PL for an issue that has been more or less resolved in the OOP community may restrict the language in its growth.

Sample Program

Using the Plaid Language

- Good Approach

```
state idle case of my_base_class {
  method deposit_coin(amount) [idle >> coin_deposit]  {
        this <- coin_deposit{ var amount_deposited = amount;
  }
}
```

- Bad Approach

```
state my_base_class {
 var flag;
  method do_some_things()  {
      if { flag == 0; } {
         java.lang.System.out.println("I'm in state 1");
      };
  }
}
```

Current Status & Future

- Plaid is currently under development at Carnegie Mellon University. Its developer, Jonathan Aldrich, is advising PhD students researching the language.
- There is currently no standalone compiler available to the public, the only option in that department is one built into a web page on the CMU site.