Glib: the Typesafe Event Publishing House

Daniel Libicki

Celequest Corporation

The Problem

- in an object-oriented event driven architecture, an event consists of one or more objects
- objects are sent from one system to another
- the receiving system may not have a class definition for the object
- if the language uses the object’s original class definition from the sending system, how can it guarantee type safety?

The Arrival Example (in Glib)

```
for sub in $subscriber-machines:
    outside SouthwestArrival arrival : me.get-last-arrival;
    $sub.insert<FlightArrival>("arrival", $arrival);
    $sub.execute(
          "$subscriber-object.flight-arrived($arrival);"
    );
in order;
```

Where is the Class Defined?

- the event subscriber system has a definition for the base class Arrival, but not for the implementation class SouthwestArrival
- Glib will send the definition of SouthwestArrival over the wire so the objects will behave correctly
- SouthwestArrival is dynamically loaded
- by the way, any new expressions invoked by the SouthwestArrival class will trigger a dynamic class loading from the remote system

Type Safety

- Cardelli: type systems catch classes of errors at compile time
- a type safe language guarantees that no program that passes the type checker will execute an illegal operation on a object
- informally, type systems guarantee that "NoSuchMethod" exceptions are never thrown at runtime

```
//this call should throw a ClassNotFound exception
Class.forName(languageName + ".lang.NoSuchMethodException");
```

Type Checking

- In order to type check a program, all expressions in the program must be assigned a type
- program fragments are composable if checked against a single set of type definitions
- loading code dynamically composes two code fragments, each of which was checked against its own set of type definitions
- in general, this is not type safe
- can it be done in a type safe fashion?

The Solution

- a runtime type equivalence check
- both systems must have a definition for the base class – in this case, Arrival
- are the two definitions of the class Arrival type equivalent?
- if so, the receiving system can safely use the sending system's definition of SouthwestArrival

String Equivalence is NOT Type Equivalence

- false negatives (intuitive) : implementation of methods
- false positives (counterintuitive) : dependency graph

It's Kind of Like .h Files

to compare two classes for type equivalence:

1 strip out method definitions

2 define dependency relation

3 find transitive closure of dependencies

4 do steps 1-3 for both classes and compare the resulting sets

The Typestamp: an Implementation Note

- Glib has class declarations like C++
- traverse graph of class declarations
- arrive at a set of strings
- order cannonically and concatenate
- this is the typestamp of a class

Conclusion

- The Story So Far
  – Full language definition
  – Masters thesis
  – Prototype implementation
- Where to Find Additional Information
  – http://www.typestamp.com/glib/thesis.html
  – dlibicki@celequest.com
  – OOPSLA 2006 Poster: Semantics of Persistence in the Glib Programming Language
  – OOPSLA 2006 Lightning Talk: “Simplicity is not Enough”