Static Assurance of Runtime Architectures

Conformance Checking Strategy

We extend the extract-abstract-check strategy

- Document as-designed architecture
- Abstract as-built architecture from code
  - Annotate code to clarify architectural intent
  - Extract sound approximation of runtime object graphs
- Abstract into as-built runtime architecture
- Check and measure structural conformance
  - Structurally compare as-built and as-designed views
  - Trace to code unexpected conformance finding

Conformance Checking Analysis

- Consider as-designed view more authoritative
- Allow as-built view to contain low-level details
- Account for all communication in as-built view that is not in as-designed view
- Include transitive communication through elided objects
- Conformance check highlights key differences:
  - Convergence: node or edge in both as-built and as-designed view
  - Divergence: node or edge in as-built but not in as-designed view
  - Absence: node or edge in as-designed but not in as-built view

Illustration of End-To-End Approach on Aphyds (8-KLOC)

Aphyds as-designed architecture, drawn by original developer:

```java
class Circuit {
    public domain DR; // Declare public domain DR
    domain OWNED; // Declare private domain OWNED
    DR Node nodes; // Declare Node reference in DR
    // Other annotation is for container; instance for its elements
    OWNED NodeTable<String, DR Node> nodes;
}
class Viewer OP { // Declare domain parameter M
    // Circuit circuit; // Declare Circuit reference in M

    class Main {
        domain MODEL, UI; // Declare top-level domains
        MODEL Circuit circuit;
        // Bind domain parameter M to actual domain MODEL
        UI Viewer<MODEL> viewer;
    }
}
```

Ownership Domain annotations express:
- object encapsulation
- logical containment
- architectural tiers
- communication permissions

In Eclipse AcmeStudio perspective, document as-designed architecture in architecture description language.

ArchCog abstraction tool maps hierarchical runtime object graph to as-built runtime architecture.

ArchRecJ architectural extraction tool extracts representation of as-built hierarchical runtime object graph.

Relate architectural elements to code. Fix serious architectural violations. Or refine as-designed architecture.

ArchConJ conformance checking tool displays results Study conformance view. Investigate differences.

Conformance check highlights key differences:

- Convergence: node or edge in both as-built and as-designed view
- Divergence: node or edge in as-built but not in as-designed view
- Absence: node or edge in as-designed but not in as-built view

Add ownership domains as Java 1.5 annotations. ArchCheckJ typechecking tool shows warnings in Eclipse problem window.