



Graph Tracking Using Bytecode Instrumentation

Master thesis project for: ...

Student ID: ...

The Ideal Graph Visualizer is a tool for displaying evolving graph data structures. Currently this is done by serializing a snapshot of the graph at various times. The goal of this project is to build a tool that instruments Java programs such that changes to the graph data structures are automatically detected. It should use an existing byte code instrumentation library (e.g. BCEL) and the `java.lang.instrument` package.

The user specifies the Java classes that are part of the graph. Objects of these classes are nodes and references between the objects are edges of the graph. The tool should be able to deal with arrays as well as indirect references (e.g. through a Java collection class). Changes to fields of the specified classes should be tracked and recorded. The output of the tool can then be viewed in the Ideal Graph Visualizer.

The implementation should be tested and evaluated on selected open source Java programs.

Challenges

- *Execution speed:* The performance cost of instrumentation should be low.
- *Data size:* The size of the recorded data is significant for large applications. The tool can consider treating several small changes as one atomic change.
- *Usability:* The specification of the instrumented classes should be as easy as possible.

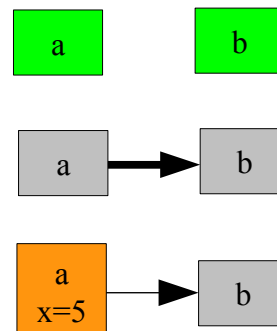
Example program

```
class Node {
    int x;
    Node next;
}
```

```
Node a = new Node();
Node b = new Node();
```

```
// Intercepted field access =>
// a new link between a and b is recorded
a.next = b;
```

```
// Intercepted field access =>
// the property x of node a changed
a.x = 5;
```



Links

- Ideal Graph Visualizer: <http://ssw.jku.at/General/Staff/TW/igv.html>
- BCEL: <http://jakarta.apache.org/bcel/>
- `java.lang.instrument` package: <http://www.j2ee.me/j2se/1.5.0/docs/api/java/lang/instrument/package-summary.html>

Contact: Dipl.-Ing. Thomas Wuerthinger (wuerthinger@ssw.jku.at)