

<RA>

RA: ResearchAssistant  
for the  
Computational Sciences

*Daniel Ramage*

*Adam J. Oliner*

*Department of Computer Science*

*Stanford University*



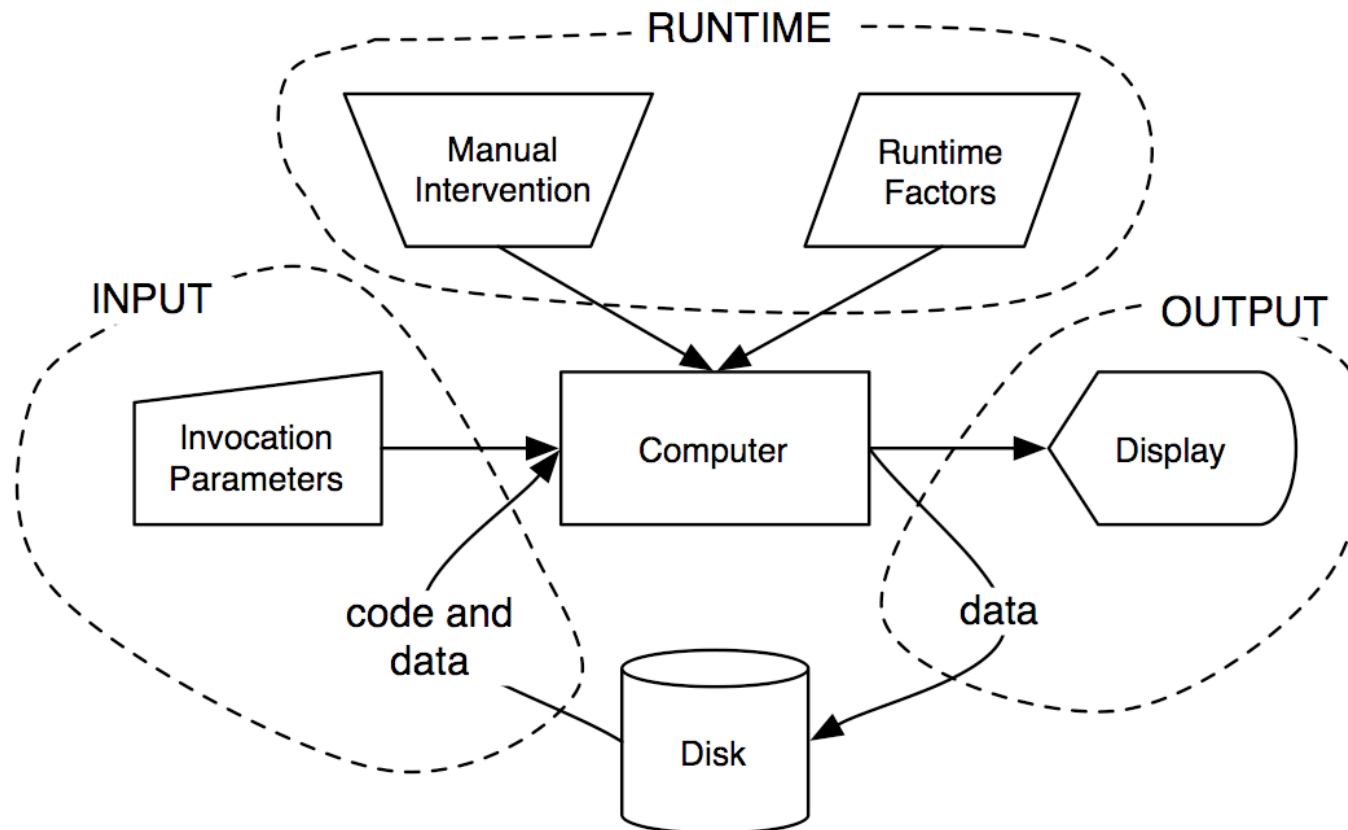
# Goal

---

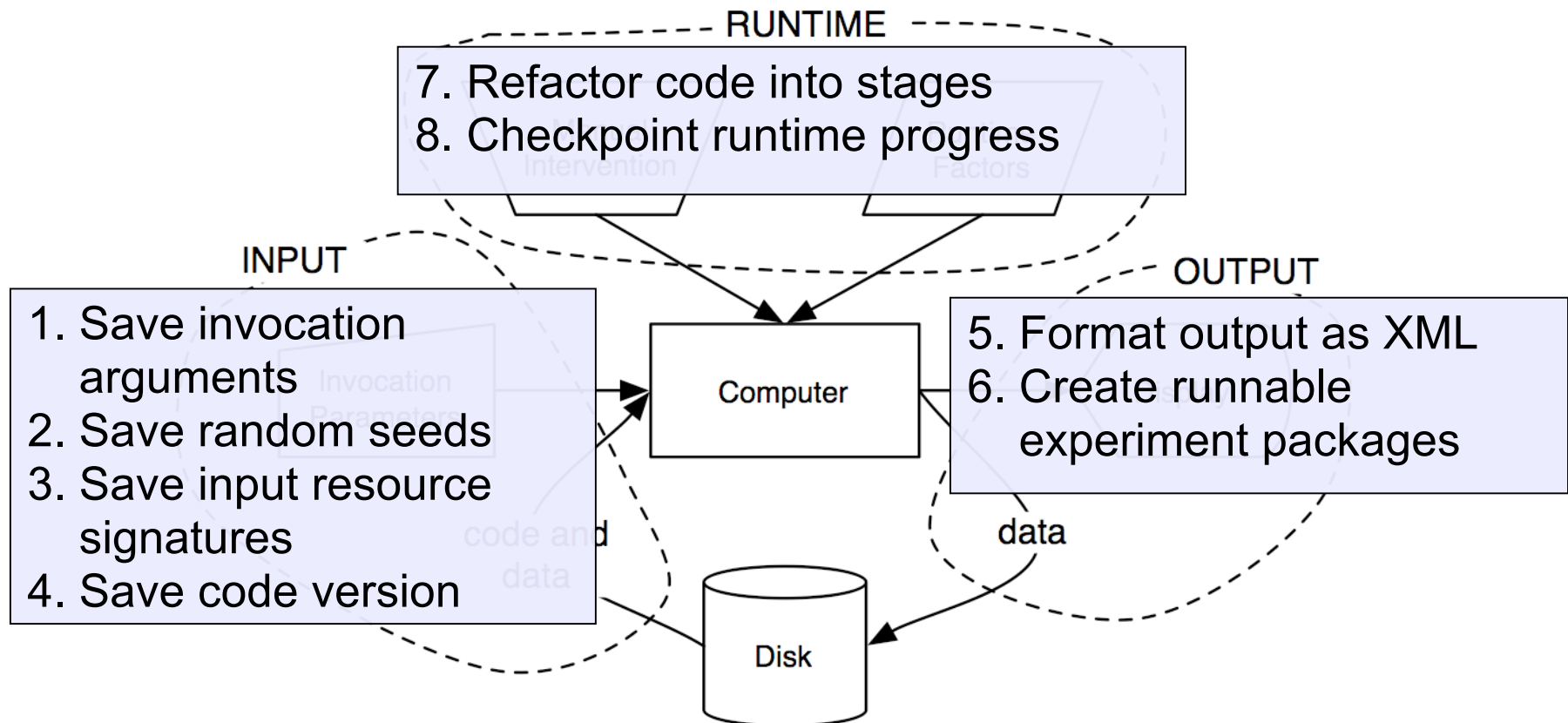
- Software as a scientific tool
- Valuable experiment information
  - Identify
  - Manage
  - Capture
  - Organize



# Experiment Information



# 8 Recommendations



# RA: ResearchAssistant

---

- Implements 8 recommendations
- Java Programming toolkit
  - Automatic bookkeeping
  - Experiment management
  - Runnable packages
- ~10k lines of Java 1.5



# Key Example: WebCounts

---

```
public class WebCounts {
    public static void main(String argv[]) {
        String text = textFromURL(argv[0]);
        Map<String,Integer> counts = tokenCounts(text);

        for (String token : counts.keySet()) {
            System.out.println(counts.get(token)
                               + " " + token);
        }
    }
}
```



# WebCounts Output

---

Grabbing text from <http://www.nytimes.com/>

45 a

6 about

1 absente

...

6 york

4 you

1 young

6 your

1 zimbabw



# One-Line Migration

---

- Add new first line
  - `RA.begin(argv) ;`
- 6 recommendations implemented:
  1. Save invocation arguments
  2. Save random seeds
  3. Save input resource signatures
  4. Save code version
  5. Format output as XML
  6. Output runnable experiment packages



# WebCounts XML

---

```
<invoke class="example.WebCountsOneLineMigration"
  starttime="Wed Feb 14 16:26:20 PST 2007"
  seed="8745931257572013">
  <environment>
    <arguments><arg>http://www.nytimes.com/</arg></arguments>
  </environment>
  <stderr>Grabbing text from http://www.nytimes.com/</stderr>
  <ReadResource name="http://www.nytimes.com/" type="URL"
    hash="453d1e99248cc5fec2b9cdf2a8a5daed" />
  <stdout>39 a</stdout>
  <stdout>7 about</stdout>
  ...
  <stdout>1 zone</stdout>
  <exit code="0" endtime="Wed Feb 14 16:26:23 PST 2007"
    runtime="P0Y0M0DT0H0M3.421S" />
</invoke>
```

---



# Why XML?

---

- Text-based
- Structured heterogeneous data
  - RA wraps stdout and stderr
  - API for custom tags
- Community, tools
- Scriptable
  - `xml sel -t -m "//stdout" -v "text()" -n`



# Repeatable Randomness

---

- RA maintains seed
  - `Math.random` → `RA.random`
  - `new java.util.Random()` → `RA.newRandom()`
- Repeatable
- Caveats
  - Scheduling
  - Internal structures



# Workbook API

---

- Record input resource accesses
  - `new InputStream(file) → Workbook.getInputStream(file)`
  - Hash, file modification written to XML



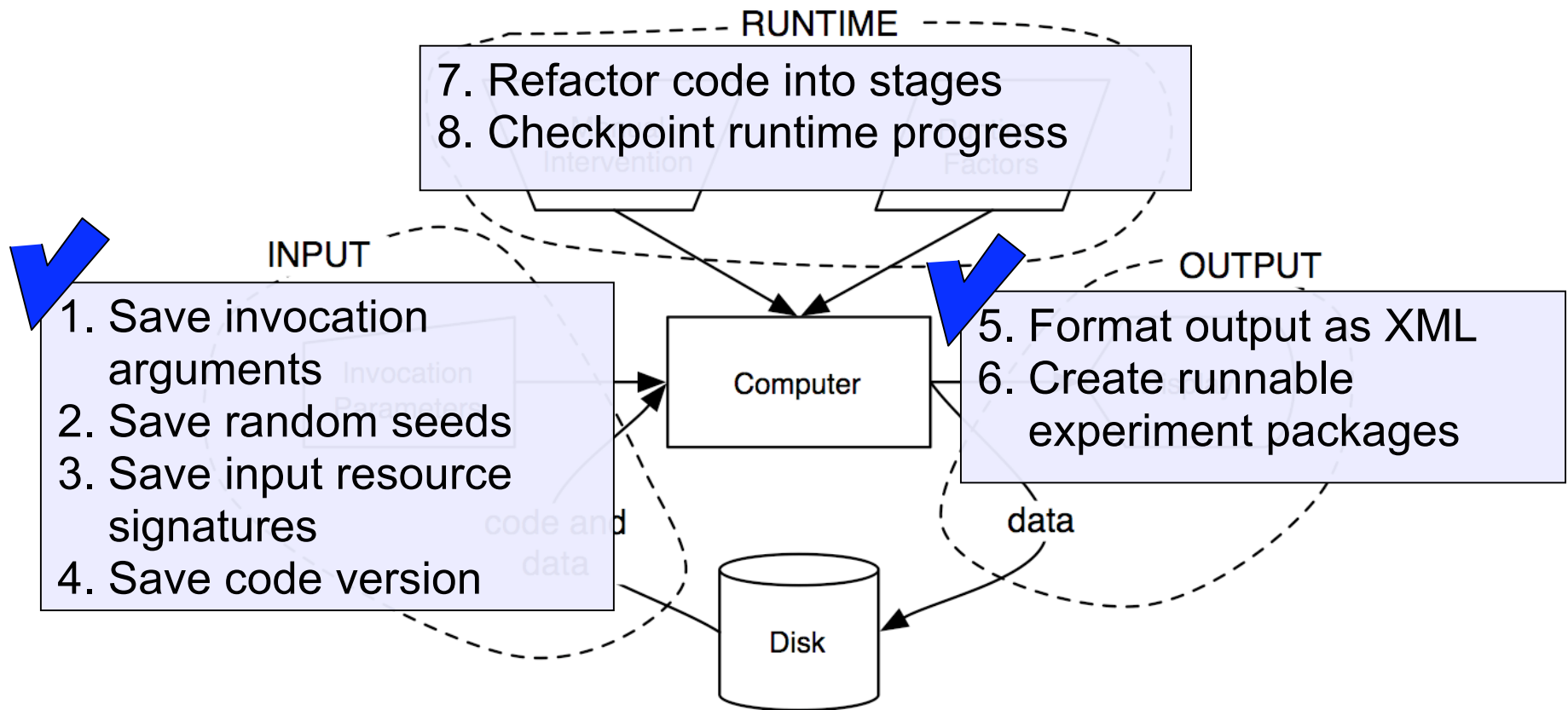
# Experiment Packages

---

- Source directory snapshot
  - Subversion version control
- Generates Java Archive (.jar)
  - Source version
  - Command line
  - Output XML
  - All classes



# 8 Recommendations



# StageWise

---

- Recommendations
  7. Refactor code into stages
  8. Checkpoint runtime progress
- Annotate dependencies
- Experiment Fields
- Arguments



# WebCounts in Stages

---

```
@Stage.Requires(Load.class)
class Count implements Stage {
    @Stage.ImportField("WebCounts:text")
    String text;

    public void run() {
        Map<String,Integer> counts = tokenCounts(text);
        for (String token : counts.keySet()) {
            RA.stream.line("token", token,
                          "count", counts.get(token));
        }
    }
}
```



# Checkpointing

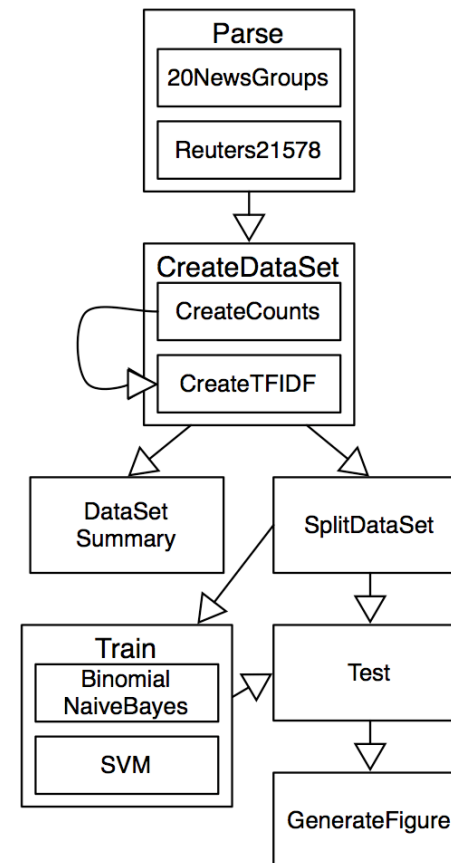
---

- Stages may be serialized
- Re-run from checkpoints
  - Saves time
  - Repeatability



# Scaling

- Dependency graphs
- Modularity
- NLP [Hughes 07]



# Contributions

---

- Characterize lost information
  - Eight recommendations
    - Save input: random seeds, invocation parameters, input signatures, code version
    - Output: XML, experiment packages
    - Runtime: refactor code into stages, checkpoint runtime progress
  - Embodiment in ResearchAssistant  
<http://www.stanford.edu/~dramage/ra/>
- 

