Formal Methods for Preserving Privacy while Loading Big Data

M. Brian Blake, PhD

Professor of Computer Science, Electrical and Computer Engineering, and Human Genetics

Dr. Iman Saleh, Research Scientist

NSF Big Data 2014, September 16, 2014, Dallas, Texas
Web-Scale Workflow - not as envisioned...

Human Actor(s) in the Loop

Data and capabilities from human-user and organizational data

Capability availability from functional, domain-specific services

Data/capabilities from legacy applications in understood formats

Proprietary and domain-specific capabilities providers

Open, Network-Accessible Services (i.e. information search, calculation tools)

Previously Established Data

Legacy data

Relational Data

XML

Business Intelligence

Web Service Application Servers

Process Management Tools

Adaptive Components/Intelligent Agents

Proactivity

Reactivity

Learning

Autonomy

Knowledge Management

Process Management

Data Management

1. Open, Network-Accessible Services (i.e. information search, calculation tools)

2. Capability availability from functional, domain-specific services

3. Data/capabilities from legacy applications in understood formats

1a. Proprietary and domain-specific capabilities providers

1b. Data/capabilities from legacy applications in understood formats

1c. Capability availability from functional, domain-specific services

1d. Data and capabilities from human-user and organizational data
Data-Oriented Prevalent On-line Web Services

1

Instant search
jQuery + Ajax

Social Networking & Mobile

Electronic Commerce

Utility Services

Content Management

Big Data: Volume, Variability, and Velocity

A Rising Growth of ETL Software
(extract-transform-load)
Must preserve privacy across services

• Sensitive information can be released when overlapping specific data points.
• For example, blood-type, eye-color, familial relationships, can suggest relation to diseases such as cancer and heart disease or even genetic defects.

“Health InfoScape”: a visualization of the relationships between medical conditions on the basis of the frequency of co-occurrences.

Systematic validation with confidence

Formal Definition of Privacy in Loading Applications
Open Challenges

• Customized, contemporary software lifecycles and frameworks for big data testing
  – Loading data from a high variety of sources with confidence and repeatability

• New approaches for automated test generation including approaches that leverage the efforts of crowds/social networks
  – Testing cannot be limited a small testing team considering variability

• Algorithms for continuously ensuring privacy and data quality as ETL applications and data evolve
  – Need to address nonfunctional concerns at loading time.
Current and Future Experimentation

RQ1: Formal specification of privacy? Machine-readable privacy rules
RQ2: How to extract privacy rules from natural language descriptions
RQ3: Formal definition of privacy to reason about privacy conformance
RQ4: Automatically generating tests from formal specification of privacy

Privacy Terms

Our Privacy Policy explains:
What information we collect and why we collect it.
How we use that information.
The choices we offer, including how to access and update information.
We’ve terms like cookies, IP addresses, pixel tags and browsers, then read about

Big Data Application

Privacy Violations

>WARNING
>ERROR

Tests

RQ1
Privacy Formalization

RQ2
Machine-Readable Privacy Rules

RQ3
Static Code Analysis

RQ4
Automatic Test Case Generation

Syntax Tree

>WARNING
>ERROR

Privacy Violations

RQ1: Formal specification of privacy? Machine-readable privacy rules
RQ2: How to extract privacy rules from natural language descriptions
RQ3: Formal definition of privacy to reason about privacy conformance
RQ4: Automatically generating tests from formal specification of privacy
Early Prototype: JPrivacy

PrivacyInspector classifier represents the suggested model of privacy, its implications on the patterns of potential violations in the code under inspection, and how the proposed framework manipulates the AST for detecting them.

Every generalized classifier (parent class) has a static factory operation (method) that are called by the Initiator (Inspector) to create an instance of the appropriate subclass without explicitly mentioning it.

Setters, getters, and constructors are not mentioned in the class diagram.

The initiator (ASTInspector) is responsible for the initialization, management, control, and finalization of establishing the shown map of association.
### Meet the Team....

<table>
<thead>
<tr>
<th>Research Scientist</th>
<th>Graduate Students</th>
<th>Undergraduates (More Recent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dr. Iman Saleh Moustafa</td>
<td>Mr. Damian Clarke CS, PhD Student</td>
<td>Alex Yale-Loehr Senior, CS</td>
</tr>
<tr>
<td></td>
<td>Mr. Julian Jarrett CS, PhD Student</td>
<td>(SMART Scholar)</td>
</tr>
<tr>
<td></td>
<td>Mr. Luis Nunez Degwitz CS, MS Student</td>
<td>Ian Schlesinger Google, CS,</td>
</tr>
<tr>
<td></td>
<td>Iman Moustafa CS, PhD Feb 2012</td>
<td></td>
</tr>
<tr>
<td></td>
<td>David Wei CS, PhD June 2013 (Microsoft Corporation)</td>
<td></td>
</tr>
</tbody>
</table>

**Undergraduate Alums:**

- **Michael Nowlan**, PhD (2014), Yale University
  - SMART and NDSEG Fellowship Awardee
  - ACM National Research Competition Finalist
  - Georgetown Computer Science Award
- **David Cumming**, MS (2012), Stanford University
- **Amy L. Sliva**, PhD (2010), University of Maryland-College Park (Assistant Professor Northeastern University)
  - ACM CRA Research Award Runner-Up
  - ACM National Research Competition Finalist
- **Wendell Norman**, Software Engineer, The MITRE Corporation
- **Georgina Saez**, Software Engineering Consultant, Accenture
- **Todd Cornett**, MS Student, Stanford University
- **Tepring Piquado**, PhD (2010), Brandeis University (Post Doc, UC-Irvine)
Questions Now?

Later?

M. Brian Blake, PhD
Vice Provost and Dean of the Graduate School
Professor of Computer Science
Concurrent Professor of Electrical and Computer Engineering
Concurrent Professor of Human Genetics

M.Brian.Blake@miami.edu
http://www.cs.miami.edu/~blake
Background

Native of Savannah, Georgia

Education

Bachelor of Electrical Engineering, Georgia Tech
Master of Electrical Engineering, Mercer University
PhD, Software Engineering, George Mason University

Professional Experience:

Engineer and Defense Contractor (6 yrs)
Professor & Chair, Computer Science, Georgetown University
Associate Dean & Professor, University of Notre Dame
Vice Provost & Dean of the Graduate School, University of Miami

Family:

Wife, Bridget, BME GA Tech / MBA Johns Hopkins
Brendan (8yrs old), Bryce (2yrs old)