

Research Abstract for Semantic Anomaly Detection in Dynamic Data Feeds with Incomplete Specifications

Orna Raz
School of Computer Science
Carnegie Mellon University
Pittsburgh PA 15213 USA
orna.raz@cs.cmu.edu

1. The thesis

We can infer useful characteristics of the normal behavior of a dynamic data feed and use these characteristics as proxies for missing specifications, to augment any existing specifications.

These augmented specifications suffice for practical semantic anomaly detection. Their inference can be done automatically, to a large extent.

2. Motivation

Everyday software must be dependable enough for its intended use. Because this software is not usually mission-critical, it may be cost-effective to detect improper behavior and notify the user or take remedial action. Detecting improper behavior requires a model of proper behavior. Unfortunately, specifications of everyday software are often incomplete and imprecise.

The situation is exacerbated when the software incorporates third-party elements such as commercial-off-the-shelf software components, databases, or dynamic data feeds from online data sources. The latter case is especially difficult, because the proprietor of the data feed may change its semantics, format, or availability while it is being used; further, specifications for data feeds are often even sketchier than those for software components. Examples of data feeds include stock quotes for a specific company, airfare for specific origin and destination, and news for a specific topic.

We want to make the use of dynamic data feeds more dependable. We are specifically interested in *semantic* problems with these feeds—cases in which the data feed is responsive, it delivers well-formed results, but the results are inconsistent, out of range, incorrect, or otherwise unreasonable. We focus on a particular facet of dependability: availability or readiness for usage [9], and change the fault model from the traditional “fail-silent” (crash failures) to “semantic”. We investigate anomaly detection as a step towards increasing the semantic availability of dynamic data feeds.

3. Approach

The **research challenge** is to enable cost-effective semantic anomaly detection when specifications are incomplete. This is a first step towards enabling assessment and control of semantic failures.

The **solution** we propose is a method and tools for *inferring* characteristics of a data feed from its behavior, using and adapting existing statistical and machine learning techniques for inference. The inferred characteristics, in the form of boolean and statistical invariants, serve as proxies for missing specifications. We demonstrate the usefulness of such invariants for on-going *semantic anomaly detection* in the data feed: identifying occasions when a dynamic data feed is delivering unreasonable values, even though its behavior may be superficially acceptable (i.e., it is delivering parsable results in a timely fashion).

Initial results [12] demonstrate the feasibility of our approach (in the context of stock market tickers).

This approach presents the **challenges** of: inferring invariants that are strong enough to separate normal behavior from abnormal behavior, selecting a model (recommending amount of training data, tuning parameters of a technique, and preprocessing the data), and choosing a good-enough combination of techniques.

Our invariant inference framework has two **major stages**: setup and usage. For each, our goal is to supply tools (semi to fully automated) to aid in performing the related tasks. Domain knowledge and explicit specifications are not required for our approach, but they may be incorporated in each step of both the setup and the usage stages, and should improve results.

The setup stage can be viewed as a gray-box into which the user enters a data feed, and from which the user gets a suite of invariant inference techniques, tuned for the given data feed. Producing the suite includes the following steps: (1) match candidate techniques with the data, (2) if necessary: augment the techniques (e.g., to handle noise) and/or pre-process the data (e.g., bin continuous values), (3) find an effective way to use each technique over the data (do model selection), and (4) determine a good-enough subset of techniques.

In the usage stage: first, each of the techniques in the suite is used for invariant inference over a moving window of the data feed. Second, each invariant is evaluated over fresh data. An anomaly is detected when an invariant is evaluated to false.

4. Expected contribution

We expect to: (1) provide a method and tools for inferring characteristics of normal behavior of a dynamic data feed that can be used as proxies for missing specifications, (2) demonstrate the usefulness of these characteristics for semantic anomaly detection, and (3) qualitatively demonstrate cost-effectiveness.

We use the inferred invariants for semantic anomaly detection because it is a first step towards increasing the dependability of dynamic data feeds. However, the inferred invariants may have **additional usages**, such as: assessing independence of data feeds, helping to deduce likely specifications, and detecting mismatches between specifications and actual behavior. Moreover, our approach for inferring invariants may be useful not only for consumers but also for producers. For example, historical behavior within a context may be used to automatically establish sanity checks for entering data into a database.

5. Validation

We map the thesis statement to validation criteria: we need to address breadth, cost, and strength/benefit.

Breadth. Our experimental plan aims to ensure this research explores an interesting subset of dynamic data feeds. This subset includes examples of the major classes of data feeds that are especially vulnerable to semantic failures: data feeds that report sensor data and data feeds that rely on humans to enter the data.

Cost. We expect the main cost factor to be human attention. This includes the amount and nature of human intervention and the amount of false positives. The computation cost appears to be reasonable (on the order of magnitude of seconds) so we do not take it into account.

For each step in our approach, we plan to ensure the frequency of using this step matches the level of human attention required in this step. Steps that are infrequent may require intensive human intervention, whereas more frequent steps should be less demanding of human attention. For example, adapting and adding an existing technique to the invariant inference tool-kit is done once. It will be satisfactory if we provide a procedure (that may be human attention intensive) for doing so. Detecting anomalies over newly observed data is done very frequently (as frequently as every observation). Therefore, it should be automated.

Strength/benefit. Anomaly detection is not only a first step towards increasing dependability but also a means we use to validate the strength of the inferred invariants: we measure how effective these invariants are for semantic anomaly detection. We use classification accuracy to quantify this effectiveness both for each technique in separation and for the suite of invariant inference techniques.

Because perfect anomaly detection is not practical in our setting, we settle for detection that is comparable to the kind of anomalies an “ideal” (able to pay attention) human would find.

6. Related work

Efforts to increase dependability include prevention and detection/mitigation of problems. Our approach concentrates on detecting semantic problems by the client of a data feed. In general, prevention and detection/mitigation are complementary as complete prevention is rare.

Preventing problems. Our approach deals with the situation as it is today. The Semantic Web [1] suggests a grand vision for a comprehensive solution to syntax/form and semantic failures, which requires many additions to the current Web. Even if the infrastructure is in place, semantics will need to be supplied and may not match behavior.

We concentrate on detecting semantic anomalies. Web Services promote a new development paradigm: building applications using as elements services available on the Web. The emphasis is on service discovery and automatic information exchange/integration. This is mainly related to connectivity and syntax/form failures.

Our work can be viewed as concerned with data quality. Most data quality research is concerned with the producer of the data. Our emphasis is on measuring and increasing data quality by the consumer.

Detecting/Mitigating problems. For improved dependability, solutions to all types of problems are necessary. There are many existing solutions to specific connectivity and syntax/form problems. However, solutions to semantic problems are scarce and either require domain knowledge [11, 10, 3, 7] or provide a specific technique [6].

Our approach of inferring the characteristics of a data feed from its behavior is similar to work in the areas of program analysis [5, 4, 2] and intrusion detection [8]. However, program analysis work naturally has a different domain, and often concentrates on a specific technique. The major difference between our work and intrusion detection is the fault model. Our model is semantic, unintentional faults, whereas intrusion detection assumes malicious faults. In addition, intrusion detection often concentrates on specific techniques.

7. Acknowledgements

I thank my thesis committee members: Mary Shaw, Michael Ernst, Christos Faloutsos, and Philip Koopman.

This research is supported by the National Science Foundation under Grant CCR-0086003 and by the Software Industry Center at Carnegie Mellon University.

References

- [1] W3C. The Semantic Web, Activity.
- [2] G. Ammons et al. Mining specifications. In *POPL*, 2002.
- [3] M. Bauer et al. Trias: Trainable information assistants for cooperative problem solving. In *Agents*, 1999.
- [4] D. Engler et al. Bugs as deviant behavior: A general approach to inferring errors in systems code. In *SOSP*, 2001.
- [5] M. Ernst et al. Dynamically discovering likely program invariants to support program evolution. In *IEEE TSE*, 2000.
- [6] Rulequest. GritBot. www.rulequest.com/gritbot-info.html. Accessed January 2002.
- [7] N. Kushmerick. Regression testing for wrapper maintenance. In *AAAI-99*, 1999.
- [8] T. Lane et al. Approaches to online learning and concept drift for user identification in computer security. In *KDD*, 1998.
- [9] J. Laprie. *Dependability: Basic Concepts and Terminology*. Springer-Verlag, Vienna, 1991.
- [10] K. Lerman et al. Learning the common structure of data. In *AAAI*, 2000.
- [11] V. Raman et al. Potters wheel: An interactive data cleaning system. In *VLDB*, 2001.
- [12] O. Raz et al. Semantic anomaly detection in online data sources. In *ICSE*, 2002.