Fast Abstract: Data Dynamics for Testing Systems

Conference or Workshop Item

Fast Abstract: Data Dynamics for Testing Systems

Julian Harty
Open University
Milton Keynes, UK
julianharty@gmail.com

***Abstract*—Data is the lifeblood of business systems, and accordingly having useful data for testing is both an interesting research challenge and a headache for many involved in projects in industry. There are few good answers of where and how to source such data cost- and time- effectively. In the author's experience across various industries globally the state of practice is poor. The poor practices potentially compromises both the projects and potentially the reputation and revenues of businesses, especially given fines based on a percentage of turnover.**

This paper aims to stimulate discussion and research into ways to first understand and then devise safe yet potent data sets for testing systems where the data are appropriate to the needs and context of the software project.

***Index Terms*—database, performance, privacy, quality in use, security, software testing, systems, test data, utility**

I. Introduction

Systems without data are lifeless. For businesses, their data is often one of their vital elements and a major intangible asset in the value of that business. Misuse of production data, *e.g.* for development testing, may lead to an erosion of trust, and sometimes large fines [1] and regulatory oversight. So data is important and has a value, therefore an organisation needs to process their data efficaciously and safely (protecting it from harm or abuse). There is a dichotomy between protecting the data (*e.g.* assets, confidentiality, embarrassment, revenues, and secrets) and testing the underlying systems that house and use the data for business-as-usual.

Data can play several roles in a system. It may be both acted upon (*e.g.* queries, updates, deletes) and used in a payload. Insert operations are a hybrid where existing data in a system may be indirectly affected by additions, whereas the data being inserted comes from the payload.

The behaviours of a system often depend on the data being used, this includes data being provided, for instance in a database query or API request and the target data. The characteristics of the data may also be relevant in terms of its fidelity for the purposes and context it is being used for, for instance when searching a map region for somewhere to eat then post codes, meta data about business locations such as hotels, restaurants, fast food outlets, *etc.* are germane to the query and results.

In practice data, or data fields and storage, are sometimes used in unexpected ways that the IT team does not expect. This is important in two ways: the User Experience *(UX)* view of how people use a system, and also in data migration – what you have in the source field may not be what the target field expects. Some uses will be opaque to the IT team unless they actually read the data. When the data is migrated the meanings of the data may have to be migrated adding time, complexity and risk. Data is not static, it changes by location, over time, because the person whose data it is changes, *etc.* What we are allowed to store about people changes as custom/norms change and legislation changes – so IT policies tend to be quite rigid, but information is fluid. People will combine data in new ways, or want to do that. And data is migrated between systems, sometimes on an ongoing basis which establishes dependencies and may require data to be identified and matched.

1) Life of data: Data is around for longer than many people think from testing or usage perspectives, for example, a pension fund could have 40 years of paying in and 40 years of paying out. Therefore it is pertinent to address ways data is stored and processed that allow for change, migration, longevity, historical characteristics, and for forward planning.

2) Performance characteristics: Realistic characteristics of data from a performance perspective is worth considering. As an example when testing the performance characteristics of searching names, real-world distributions are not-uniform [2] so test data may need to have a similar distribution for realistic performance testing.

3) Observation Points: In network security analysis various monitoring software tools are used to discover and observe network traffic [3]. We may be able to use a similar approach, generalised as *observation posts* to learn about data is it is transferred from one system to another, or from the boundary of our ecosystem in or out of our system in order to establish suitable criteria for data that would be appropriate to use in testing the system. As we learn more about the data we may also be able to generate interesting variants, or mutants, to help test how both current and future systems would perform.

4) Test Data and Test Environments: For the purposes of this work, test data is the data that is used in systems being tested and in the interactions with those systems. It may include user interactions - performance metrics based on interactions with the system. As an old example, the inter-character delays between keystrokes can be used to guess a password and presumably other sensitive data [4]. It excludes data about the practice or the results of testing (unless the system being tested is a test management system, for example).

Using a layered approach to testing where results from less sensitive test environments can inform more sensitive ones may also be useful [5].

 DOI:10.1109/ICSTW50294.2020.9374728

5) Ethics under pressure?: One of the conundrums for individuals and organisations is their approach to using production data for testing. Many years ago, employees simply used a copy of production data, for instance by copying and using the production database. Such practices are no longer condoned and may contravene what the company commits to do and what legislation forbids them to do. Furthermore, some end users may be upset if they discover such practices and may also be adversely affected by the results. And yet, there are likely to be team members who continue to use such data anyway. Perhaps they perceive doing so is necessary and/or expedient? And could those organisations be applying a *"Don't ask, don't tell"* [6] policy (almost certainly unofficially and undocumented).

6) Valid Uses of Production Data?: There are various reasons why use of production data may be valid, and perhaps necessary. These include access by researchers into social media data for 'harm' research [7], for auditing, for testing ethical claims, and for legal investigations, amongst other reasons. Note, this does not necessarily mean *direct access to the data*, instead access might be running blind queries (similar in concept to blind SQL injection [8]) in a sandboxed, secure environment where query results are used to assess validity of results.

A. Research Questions to Consider

1) How much data is necessary and how much data is optimal?: Data costs and has risks associated with its preservation and use. There are numerous practical challenges involved in managing and using such data, for instance data going stale. Software-as-a-Service (SaaS) providers and Cloud Providers charge based on data volumes, these costs can adversely impact testing as well as production. Conversely, are a couple of examples sufficient to establish behaviours and characteristics of a system?

2) What is the useful life of data? When should it be retired?: With great power comes great responsibility, and these days also financial and reputational risk. Conversely project teams have an imperative to satisfice their testing.

3) What are some ethical considerations of test data?: When is it appropriate to use production data as a source of data in the system under test? When datasets need to be constructed that include elements related to people how much variety of the data is desirable and practical (without compromising real people)?

4) What characteristics would data need to correct for biases?: Particularly for *machine learning*, there have been many examples where the system learns and applies biases based on the data it was trained on [9]. Does the same apply with data we use for testing?

5) What is the Role of Data in Bugs Investigation?: As many developers and testers have learned, sometimes the data forms a vital element in the discovery of bugs. Users may report bugs that affect them and their use of a system.

II. Method

Our current research is ongoing and based on industry practices and research into related fields and topics. The aims of the research include *i*) establishing a possible taxonomy of approaches to obtain suitable data that meets the needs and context of the testing, *ii*) devising an approach to assess characteristics of data including *data quality* and fitness-for-purpose. Ultimately we wish to help establish efficacious, reliable and healthy engineering discipline around the data that is used in systems, particularly for testing and evaluating those systems. We are devising a layered model for understanding the *data dynamics* of data in systems and a concept of *atomic implementation units* (systems or subsystems that are treated as a single indivisible unit for our purposes); these may include a production database server, a payment processing service, *etc*. We also aim to establish a taxonomy of approaches to obtain data for testing, such as: generating data using a statistical model, or extracting and transforming production data [10].

III. Discussion

This work is based on several decades of experience working across a wide range of industries and domains internationally. It is further informed by ongoing discussions with people who self-identify as working in software testing. This paper aims to help accelerate the research aspects through discussion, critique and collaboration across academia and industry.

IV. A Call to Action

Data used for testing systems is endemic and the value of such data and the consequences of using the data are both immense. Practitioners in industry often seem to muddle through and make do with what they have or can envisage, and even those who pay for software tools to assist them would benefit from well-researched methods and approaches to work more effectively with data used when testing systems. We challenge researchers to get actively involved in the various practical challenges and devise some of these approaches.

References

[1] Powered by Trunomi. (2019) EUGDPR - information portal. [Online]. Available: https://eugdpr.org/

[2] R. Wicklin. (2011) Two-letter initials: Which are the most common? [Online]. Available: https://blogs.sas.com/content/iml/2011/01/14/two-letter-initials-which-are-the-most-common.html

[3] R. Bejtlich, *The Tao of network security monitoring: beyond intrusion detection*. Pearson Education, 2004.

[4] M. Zalewski, *Silence on the wire: a field guide to passive reconnaissance and indirect attacks*. No Starch Press, 2005.

[5] J. Harty. (2018) Testing the beast. Kafka Summit 2018. [Online]. Available: https://kafka-summit.org/sessions/testing-the-beast/

[6] (2020) Don't ask, don't tell. Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Don%27t_ask,_don%27t_tell

[7] BBC. (2020) Social media data needed for 'harm' research, say doctors. [Online]. Available: https://www.bbc.co.uk/news/health-51134545

[8] K. Spett, "Blind SQL injection," Technical report, SPI Dynamics, Tech. Rep., 2003.

[9] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *arXiv preprint arXiv:1908.09635*, 2019.

[10] A. Kottas. (2019) Secure your sandbox data with Salesforce data mask. Salesforce. [Online]. Available: https://www.salesforce.com/blog/2019/11/data-mask-secure-sandbox.html