

Digital Permanence

(or: Apps Considered Harmful – Again)

Dr. Ora Lassila

Technology Architect

Pegasystems, Inc. (Cambridge, MA, USA)

2015-11-23

My motivation

Revisiting an old theme:
“Longevity” of data.

There is something fundamentally wrong in how we develop information systems, and it affects permanence as well.

Note: I still have more questions than answers...

Love Thy Data (or: Apps Considered Harmful)

Dr. Ora Lassila

2012-06-11

Principal Technologist
Cloud Analytics Team
Nokia Location & Commerce

&

Elected Member
Advisory Board
World Wide Web Consortium (W3C)



CIDOC2012
HELSINKI • FINLAND



Use cases where permanence matters

Museums, digital artwork,

Libraries, archives,

Legal & financial documents,

Research data, logs, etc.

Use cases where permanence matters

Museums, digital artwork,

Libraries, archives,

Legal & financial documents,

Research data, logs, etc.

} ***automation***

Some Threats to Permanence

Hardware

- no support for older hardware,
- not possible to replace older hardware.

Storage media

- media degrades over time,
- no compatible hardware for older media.

Software

- “bit rot” + no support for older software,
- no hardware to run older software,
- older documents not supported by new software.

Most of those are not really problems

Hardware

- older hardware can be emulated (there are good examples from computer games).

Storage media

- content of older media can be migrated to new storage solutions (typically, cloud storage).

Software

- older software can run on emulated hardware,
- **non-proprietary, open document formats** are considered a **Good Thing** (i.e., multiple vendors for compatible software).

Open document formats...?

Lately, we have regressed in this regard. Huh...?

Rather than using the ultimate open platform (= the WWW), there is a tendency to build narrow-scope, proprietary applications.

“There is an app for that!” (this is a **Very Bad Idea**).

Apps are **silos** which **imprison data** and harmfully fragment our information spaces.

Let's define what an “app” is



data

logic

presentation

data + logic + presentation

A way to package/deliver/deploy the three

- this is an antiquated notion that serves developers/publishers (users don't care)

We see different kinds of apps, including

1. perform a specific function (e.g., “camera”)
2. present users with some specific data (e.g., the “NY Times” app)

Specifically with #2, one is left wondering, why not just use the Web...

Issues with data



data

logic

presentation

Data in a “silo” w/ opaque semantics:

- proprietary data models (semantics),
- proprietary data formats (syntax).

This makes the data hard to

- access (from outside the app),
- reuse (by other systems),
- integrate (with data from other sources).

An app typically “owns” its data, locking users to this particular app.

Issues with logic

data

logic

presentation

Logic is “embedded” in the app and has (at best) opaque semantics.

This makes it hard to

- access the logic – associate data with this logic except through (and in the context of) the app,
- reuse the logic in some other system.

Issues with presentation

data

logic

presentation

Typically, presentation is “fixed”

- (i.e., decided by developers of the app).

This makes it hard to

- flexibly change the presentation per desires and preferences of the user,
- reuse the presentation in some other context.

“Packaging” content in a (native) app excludes the good the Web would give us:

- no linking, no bookmarking,
- no accessibility features (unless the platform provides those; cf. reuse of data/content).

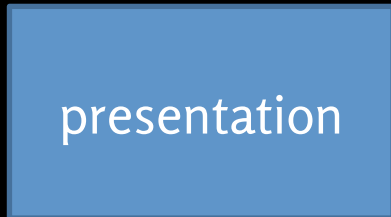
What does all this mean...?



Each data model is a new **“vocabulary”** (particularly proprietary models)



Each service is a **new interface** with own semantics (assuming logic is accessible)



Each presentation, separate from others, speaks a **new “language”** (its own)

Whether we are talking about data, logic or presentation, locking these in an un-reusable “silo” only further fragments our information space

This is our future



"Tower of Babel", Pieter Bruegel the Elder, 1563; Kunsthistorisches Museum, Wien

Always, always, always focus on data

Apps and systems come and go, but data has (or should have) **longevity**.

Models are important.

Specifically, focus on things that make **sharing** possible:

- open formats and models,
- “accessible” semantics,
- also: don't forget data provenance.

Permanence, sharing and interoperability

Permanence is aided by our ability to share things (data, documents).

Sharing is enabled by interoperability.

Why? Interoperability typically ensures

- multiple software solutions for any particular kind of data,
- that data semantics are captured in a non-proprietary way.

What can we do to improve interoperability?

Interoperability and standardization

Traditional approach to interoperability is to create **standards** (“shared specifications”).

Unfortunately, traditional standards processes and approaches are slow and do not actually guarantee a happy future.

Rather than being prescriptive and “fixing the future”, we should standardize how we communicate and exchange semantics (delayed semantic commitment).

Prescriptive approaches will not work

Dictating how the world should (always) look like is destined to fail.

Leaving some room for future interpretation is better than limiting what's possible in the future.



Do we have technologies we could adopt?

Ontologies, knowledge representation.

Semantic Web and related approaches (e.g., “Linked Open Data”).

Note: If you thought “*I’ll just use JSON for all my data*” you just flunked the class. Sorry.

Some conclusions

Data should have longevity; apps **do not matter** in the long run.

Seek approaches that make semantics explicit; imagine how someone could “understand” your data in the future.

Prescriptive approaches to interoperability will work rather poorly in the long run.

Some conclusions

These are possibly not (big) problems:

- hardware fails, becomes obsolete
- storage media degrades, becomes obsolete
- software no longer supported

This is **definitely** a problem:

- semantics is “lost”, definition of semantics separate from data

Questions?

Email: ora.lassila@pega.com

WWW: <http://www.lassila.org/>

Twitter: [@gotsemantics](https://twitter.com/gotsemantics)