The Evolution of Data Products

The data that drives products is shifting from overt to covert

Mike Loukides

O'REILLY®

O'REILLY®
Strata
Making Data Work

The Evolution of Data Products

Mike Loukides

O'REILLY®
Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

The Evolution of Data Products
by Mike Loukides

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Editor: Mac Slocum

Printing History:
September 2011: First Printing.

ISBN: 978-1-449-31651-8

[LSI]

1316095778

Table of Contents

The Evolution of Data Products

In "What is Data Science," I started to talk about the nature of data products. Since then, we've seen a lot of exciting new products, most of which involve data analysis to an extent that we couldn't have imagined a few years ago. But that begs some important questions: What happens when data becomes a product, specifically, a consumer product? Where are data products headed? As computer engineers and data scientists, we tend to revel in the cool new ways we can work with data. But to the consumer, as long as the products are about the data, our job isn't finished. Proud as we may be about what we've accomplished, the products aren't about the data; they're about enabling their users to do whatever *they* want, which most often has little to do with data.

It's an old problem: the geeky engineer wants something cool with lots of knobs, dials, and fancy displays. The consumer wants an iPod, with one tiny screen, one jack for headphones, and one jack for charging. The engineer wants to customize and script it. The consumer wants a cool matte aluminum finish on a device that just works. If the consumer has to script it, something is very wrong. We're currently caught between the two worlds. We're looking for the Steve Jobs of data — someone who can design something that does what we want without getting us involved in the details.

Disappearing Data

We've become accustomed to virtual products, but it's only appropriate to start by appreciating the extent to which data products have replaced physical products. Not that long ago, music was shipped as chunks of plastic that weighed roughly a pound. When the music was digitized and stored on CDs, it became a data product that weighed under an ounce, but was still a physical object. We've moved even further since: many of the readers of this article have bought their last CD, and now buy music exclusively in online form, through iTunes or Amazon. Video has followed the same path, as analog VHS video-

tapes became DVDs and are now streamed through Netflix, a pure data product.

But while we're accustomed to the displacement of physical products by virtual products, the question of how we take the next step — where data recedes into the background — is surprisingly tough. Do we want products that deliver data? Or do we want products that deliver results based on data? We're evolving toward the latter, though we're not there yet. The iPod may be the best example of a product that pushes the data into the background to deliver what the user wants, but its partner application, iTunes, may be the worst. The user interface to iTunes is essentially a spreadsheet that exposes all of your music collection's metadata. Similarly, the "People You May Know" feature on social sites such as LinkedIn and Facebook delivers recommendations: a list of people in the database who are close to you in one way or another. While that's much more friendly than iTunes' spreadsheet, it is still a list, a classic data structure. Products like these have a "data smell." I call them "overt" data products because the data is clearly visible as part of the deliverable.

A list may be an appropriate way to deliver potential contacts, and a spreadsheet may be an appropriate way to edit music metadata. But there are many other kinds of deliverables that help us to understand where data products are headed. At a recent event at IBM Research, IBM demonstrated an application that accurately predicts bus arrival times, based on real-time analysis of traffic data. (London is about to roll out something similar.) Another IBM project implemented a congestion management system for Stockholm that brought about significant decreases in traffic and air pollution. A newer initiative allows drivers to text their destinations to a service, and receive an optimized route, given current traffic and weather conditions. Is a bus arrival time data? Probably so. Is a route another list structure, like a list of potential Facebook friends? Yes, though the real deliverable here is reduced transit time and an improved environment. The data is still in the foreground, but we're starting to look beyond the data to the bigger picture: better quality of life.

These projects suggest the next step in the evolution toward data products that deliver results rather than data. Recently, Ford discussed some experimental work in which they used Google's prediction and mapping capabilities to optimize mileage in hybrid cars based on predictions about where the driver was going. It's clearly a data product: it's doing data analysis on historical driving data and knowledge about road conditions. But the deliverable isn't a route or anything the driver actually sees — it's optimized engine usage and lower fuel consumption. We might call such a product, in which the data is hidden, a "covert" data product.

We can push even further. The user really just wants to get from point A to point B. Google has demonstrated a self-driving car that solves this problem.

A self-driving car is clearly not delivering data as the result, but there are massive amounts of data behind the scenes, including maps, Street View images of the roads (which, among other things, help it to compute the locations of curbs, traffic lights, and stop signs), and data from sensors on the car. If we ever find out everything that goes into the data processing for a self-driving car, I believe we'll see a masterpiece of extracting every bit of value from many data sources. A self-driving car clearly takes the next step to solving a user's real problem while making the data hide behind the scenes.

Once you start looking for data products that deliver real-world results rather than data, you start seeing them everywhere. One IBM project involved finding leaks in Dubuque, Iowa's, public water supply. Water is being used all the time, but sudden changes in usage could represent a leak. Leaks have a unique signature: they can appear at any time, particularly at times when you would expect usage to be low. Unlike someone watering his lawn, flushing a toilet, or filling a pool, leaks don't stop. What's the deliverable? Lower water bills and a more robust water system during droughts — not data, but the result of data.

In medical care, doctors and nurses frequently have more data at their disposal than they know what to do with. The problem isn't the data, but seeing beyond the data to the medical issue. In a collaboration between IBM and the University of Ontario, researchers knew that most of the data streaming from the systems monitoring premature babies was discarded. While readings of a baby's vital signs might be taken every few milliseconds, they were being digested into a single reading that was checked once or twice an hour. By taking advantage of the entire data stream, it was possible to detect the onset of life-threatening infections as much as 24 hours before the symptoms were apparent to a human. Again, a covert data product; and the fact that it's covert is precisely what makes it valuable. A human can't deal with the raw data, and digesting the data into hourly summaries so that humans can use it makes it less useful, not more. What doctors and nurses need isn't data, they need to know that the sick baby is about to get sicker.

Eben Hewitt, author of "Cassandra: The Definitive Guide," works for a large hotel chain. He told me that the hotel chain considers itself a software company that delivers a data product. The company's real expertise lies in the reservation systems, the supply management systems, and the rest of the software that glues the whole enterprise together. It's not a small task. They're tracking huge numbers of customers making reservations for hundreds of thousands of rooms at tens of thousands of properties, along with various awards programs, special offers, rates that fluctuate with holidays and seasons, and so forth. The complexity of the system is certainly on par with LinkedIn, and the amount of data they manage isn't that much smaller. A hotel looks awfully concrete, but

in fact, your reservation at Westin or Marriott or Day's Inn is data. You don't experience it as data, however — you experience it as a comfortable bed at the end of a long day. The data is hidden, as it should be.

I see another theme developing. Overt products tend to depend on overt data collection: LinkedIn and Facebook don't have any data that wasn't given to them explicitly, though they may be able to combine it in unexpected ways. With covert data products, not only is data invisible in the result, but it tends to be collected invisibly. It has to be collected invisibly: we would not find a self-driving car satisfactory if we had to feed it with our driving history. These products are frequently built from data that's discarded because nobody knows how to use it; sometimes it's the "data exhaust" that we leave behind as our cell phones, cars, and other devices collect information on our activities. Many cities have all the data they need to do real-time traffic analysis; many municipal water supplies have extensive data about water usage, but can't yet use the data to detect leaks; many hospitals connect patients to sensors, but can't digest the data that flows from those sensors. We live in an ocean of ambient data, much of which we're unaware. The evolution of data products will center around discovering uses for these hidden sources of data.

The Power of Combining Data

The first generation of data products, such as CDDB, were essentially a single database. More recent products, such as LinkedIn's Skills database, are composites: Skills incorporates databases of users, employers, job listings, skill descriptions, employment histories, and more. Indeed, the most important operation in data science may be a "join" between different databases to answer questions that couldn't be answered by either database alone.

Facebook's facial recognition provides an excellent example of the power in linked databases. In the most general case, identifying faces (matching a face to a picture, given millions of possible matches) is an extremely difficult problem. But that's not the problem Facebook has solved. In a reply to Tim O'Reilly, Jeff Jonas said that while one-to-many picture identification remains an extremely difficult problem, one-to-few identification is relatively easy. Facebook knows about social networks, and when it sees a picture, Facebook knows who took it and who that person's friends are. It's a reasonable guess that any faces in the picture belong to the taker's Facebook friends. So Facebook doesn't need to solve the difficult problem of matching against millions of pictures; it only needs to match against pictures of friends. The power doesn't come from a database of millions of photos; it comes from joining the photos to the social graph.

The Goal of Discovery

Many current data products are recommendation engines, using collaborative filtering or other techniques to suggest what to buy, who to friend, etc. One of the holy grails of the "new media" is to build customized, personalized news services that automatically find what the user thinks is relevant and interesting. Tools like Apple's Genius look through your apps or your record collection to make recommendations about what else to buy. "People You May Know," a feature common to many social sites, is effectively a recommendation engine.

But mere recommendation is a shallow goal. Recommendation engines aren't, and can't, be the end of the road. I recently spent some time talking to Bradford Cross (@bradfordcross), founder of Woven, and eventually realized that his language was slightly different from the language I was used to. Bradford consistently talked about "discovery," not recommendation. That's a huge difference. Discovery is the key to building great data products, as opposed to products that are merely good.

The problem with recommendation is that it's all about recommending something that the user will like, whether that's a news article, a song, or an app. But simply "liking" something is the wrong criterion. A couple months ago, I turned on Genius on my iPad, and it said things like "You have Flipboard, maybe you should try Zite." D'oh. It looked through all my apps, and recommended more apps that were like the apps I had. That's frustrating because I don't need more apps like the ones I have. I'd probably like the apps it recommended (in fact, I do like Zite), but the apps I have are fine. I need apps that do something different. I need software to tell me about things that are entirely new, ideally something I didn't know I'd like or might have thought I wouldn't like. That's where discovery takes over. What kind of insight are we talking about here? I might be delighted if Genius said, "I see you have ForScore, you must be a musician, why don't you try Smule's Magic Fiddle" (well worth trying, even if you're not a musician). That's where recommendation starts making the transition to discovery.

Eli Pariser's "The Filter Bubble" is an excellent meditation on the danger of excessive personalization and a media diet consisting only of stuff selected because you will "like" it. If I only read news that has been preselected to be news I will "like," news that fits my personal convictions and biases, not only am I impoverished, but I can't take part in the kind of intelligent debate that is essential to a healthy democracy. If I only listen to music that has been chosen because I will "like" it, my music experience will be dull and boring. This is the world of E.M. Forster's story "The Machine Stops," where the machine provides a pleasing, innocuous cocoon in which to live. The machine offers music, art, and food — even water, air, and bedding; these provide a context

for all "ideas" in an intellectual space where direct observation is devalued, even discouraged (and eventually forbidden). And it's no surprise that when the machine breaks down, the consequences are devastating.

I do not believe it is possible to navigate the enormous digital library that's available to us without filtering, nor does Pariser. Some kind of programmatic selection is an inevitable part of the future. Try doing Google searches in Chrome's Incognito mode, which suppresses any information that could be used to personalize search results. I did that experiment, and it's really tough to get useful search results when Google is not filtering based on its prior knowledge of your interests.

But if we're going to break out of the cocoon in which our experience of the world is filtered according to our likes and dislikes, we need to get beyond naïve recommendations to break through to discovery. I installed the iPad Zite app shortly after it launched, and I find that it occasionally breaks through to discovery. It can find articles for me that I wouldn't have found for myself, that I wouldn't have known to look for. I don't use the "thumbs up" and "thumbs down" buttons because I don't want Zite to turn into a parody of my tastes. Unfortunately, that seems to be happening anyway. I find that Zite is becoming less interesting over time: even without the buttons, I suspect that my Twitter stream is telling Zite altogether too much about what I like and degrading the results. Making the transition from recommendation to true discovery may be the toughest problem we face as we design the next generation of data products.

Interfaces

In the dark ages of data products, we accessed data through computers: laptops and desktops, and even minicomputers and mainframes if you go back far enough. When music and video first made the transition from physical products to data products, we listened and watched on our computers. But that's no longer the case: we listen to music on iPods; read books on Kindles, Nooks, and iPads; and watch online videos on our Internet-enabled televisions (whether the Internet interface is part of the TV itself or in an external box, like the Apple TV). This transition is inevitable. Computers make us aware of data as data: one disk failure will make you painfully aware that your favorite songs, movies, and photos are nothing more than bits on a disk drive.

It's important that Apple was at the core of this shift. Apple is a master of product design and user interface development. And it understood something about data that those of use who preferred listening to music through WinAmp or FreeAmp (now Zinf) missed: data products would never become part of our lives until the computer was designed out of the system. The user ex-

perience was designed into the product from the start. DJ Patil (@dpatil), Data Scientist in Residence at Greylock Partners, says that when building a data product, it is critical to integrate designers into the engineering team from the beginning. Data products frequently have special challenges around inputting or displaying data. It's not sufficient for engineers to mock up something first and toss it over to design. Nor is it sufficient for designers to draw pretty wireframes without understanding what the product is or how it works. The earlier design is integrated into the product group and the deeper the understanding designers have of the product, the better the results will be. Patil suggested that FourSquare succeeded because it used GPS to make checking into a location trivially simple. That's a design decision as much as a technical decision. (Success isn't fair: as a Dodgeball review points out, position wasn't integrated into cell phones, so Dodgeball's user interface was fundamentally hobbled.) To listen to music, you don't want a laptop with a disk drive, a filesystem, and a user interface that looks like something from Microsoft Office; you want something as small and convenient as a 1960s transistor radio, but much more capable and flexible.

What else needs to go if we're going to get beyond a geeky obsession with the artifact of data to what the customer wants? Amazon has done an excellent job of packaging ebooks in a way that is unobtrusive: the Kindle reader is excellent, it supports note taking and sharing, and Amazon keeps your location in sync across all your devices. There's very little file management; it all happens in Amazon's cloud. And the quality is excellent. Nothing gives a product a data smell quite as much as typos and other errors. Remember Project Gutenberg?

Back to music: we've done away with ripping CDs and managing the music ourselves. We're also done with the low-quality metadata from CDDB (although I've praised CDDB's algorithm, the quality of its data is atrocious, as anyone with songs by John "Lennnon" knows). Moving music to the cloud in itself is a simplification: you don't need to worry about backups or keeping different devices in sync. It's almost as good as an old phonograph, where you could easily move a record from one room to another, or take it to a friend's house. But can the task of uploading and downloading music be eliminated completely? We're partway there, but not completely. Can the burden of file management be eliminated? I don't really care about the so-called "death of the filesystem," but I do care about shielding users from the underlying storage mechanism, whether local or in the cloud.

New interfaces for data products are all about hiding the data itself, and getting to what the user wants. The iPod revolutionized audio not by adding bells and whistles, but by eliminating knobs and controls. Music had become data. The iPod turned it back into music.

The Drive Toward Human Time

It's almost shocking that in the past, Google searches were based on indexes that were built as batch jobs, with possibly a few weeks before a given page made it into the index. But as human needs and requirements have driven the evolution of data products, batch processing has been replaced by "human time," a term coined by Justin Sheehy (@justinsheehy), CTO of Basho Technologies. We probably wouldn't complain about search results that are a few minutes late, or maybe even an hour, but having to wait until tomorrow to search today's Twitter stream would be out of the question. Many of my examples only make sense in human time. Bus arrival times don't make sense after the bus has left, and while making predictions based on the previous day's traffic might have some value, to do the job right you need live data. We'd laugh at a self-driving car that used yesterday's road conditions. Predicting the onset of infection in a premature infant is only helpful if you can make the prediction before the infection becomes apparent to human observers, and for that you need all the data streaming from the monitors.

To meet the demands of human time, we're entering a new era in data tooling. Last September, Google blogged about Caffeine and Percolator, its new framework for doing real-time analysis. Few details about Percolate are available, but we're starting to see new tools in the open source world: Apache Flume adds real-time data collection to Hadoop-based systems. A recently announced project, Storm, claims to be the Hadoop of real-time processing. It's a framework for assembling complex topologies of message processing pipelines and represents a major rethinking of how to build data products in a real-time, stream-processing context.

Conclusions

Data products are increasingly part of our lives. It's easy to look at the time spent in Facebook or Twitter, but the real changes in our lives will be driven by data that doesn't look like data: when it looks like a sign saying the next bus will arrive in 10 minutes, or that the price of a hotel reservation for next week is $97. That's certainly the tack that Apple is taking. If we're moving to a post-PC world, we're moving to a world where we interact with appliances that deliver the results of data, rather than the data itself. Music and video may be represented as a data stream, but we're interested in the music, not the bits, and we are already moving beyond interfaces that force us to deal with its "bitly-ness": laptops, files, backups, and all that. We've witnessed the transformation from vinyl to CD to digital media, but the process is ongoing. We rarely rip CDs anymore, and almost never have to haul out an MP3 encoder.

The music just lives in the cloud (whether it's Amazon's, Apple's, Google's, or Spotify's). Music has made the transition from overt to covert. So have books. Will you have to back up your self-driving route-optimized car? I doubt it. Though that car is clearly a data product, the data that drives it will have disappeared from view.

Earlier this year Eric Schmidt said:

> Google needs to move beyond the current search format of you entering a query and getting 10 results. The ideal would be us knowing what you want before you search for it...

This controversial and somewhat creepy statement actually captures the next stage in data evolution. We don't want lists or spreadsheets; we don't want data as data; we want results that are in tune with our human goals and that cause the data to recede into the background. We need data products that derive their power by mashing up many sources. We need products that deliver their results in human time, rather than as batch processes run at the convenience of a computing system. And most crucially, we need data products that go beyond mere recommendation to discovery. When we have these products, we will forget that we are dealing with data. We'll just see the results, which will be aligned with our needs.

We are seeing a transformation in data products similar to what we have seen in computer networking. In the '80s and '90s, you couldn't have a network without being intimately aware of the plumbing. You had to manage addresses, hosts files, shared filesystems, even wiring. The high end of technical geekery was wiring a house with Ethernet. But all that network plumbing hasn't just moved into the walls: it's moved into the ether and disappeared entirely. Someone with no technical background can now build a wireless network for a home or office by doing little more than calling the cable company. Data products are striving for the same goal: consumers don't want to, or need to, be aware that they are using data. When we achieve that, when data products have the richness of data without calling attention to themselves as data, we'll be ready for the next revolution.