Architectural Strategy and Design Evolution in Business Ecosystems

Carliss Y. Baldwin
Discussion
AOM, August 10, 2013
Quick Overview

- Brief survey
- Big questions
- Most exciting discovery (and biggest bottleneck)
- Best research idea (if time)
Three different flavors of research on business ecosystems

- **Kapoor** (and others)—a consistent agenda, unfolding
- **Tee** (and Woodard)—exploratory research at the bleeding edge of technology
- **Prencipe** (and others)—the perennial problem of knowledge integration
Kapoor

- Has maps w/ flows of products
- Mainly stays away from software, computers, telecomm, the intransigent ecosystems
- Tests performance hypotheses (consistently)
- Sensitive to overlapping approaches
  - Has been beaten up by Strategy colleagues
Tee

- Has layer maps
- System is in motion, but at least there is a manageable number of firms
- Technically sophisticated
- Unabashedly exploratory
  - No performance hypotheses
- Primary source of data is interviews
Nice research question
- Whether, when, why systems integration is effective

Good sampling approach

Tests performance hypotheses

Survey/Interview data
- Subjective measures of technical properties, i.e., customization, modularity

Often technical properties are hidden even from technologists! (Especially true of modularity.)
The big questions I have about business ecosystems

- How does ecosystem research differ from platform research?
- Where do systems integrators come in?
- How do systems integrators differ from marketplaces and resellers?
- Why are ecosystems, platforms and integrators supplanting industries and products?
- What does it mean to be "platform-based" vs. "product-based"?

What is the most promising research idea (that I can’t pursue myself)?
Ecosystems and platforms

- You can have an ecosystem without a platform, but every successful platform has an ecosystem
  - True also of systems integrators
- Traditional firms don’t know they have an ecosystem
- Platform firms and systems integrators know and nurture their ecosystems
  - Traditional firms often “ate” their ecosystems as they got rich and had more resources
  - DEC and Microsoft
My most exciting discovery this year … about architecture…
This is a network graph of a software codebase (Baldwin et al. 2013)

A system of files linked by function calls
This is an enterprise architecture (Lagerstrom et al. 2013)

A system of applications linked by “communicates with” and “uses”
This is an industrial sector (Luo et al., 2012)

A system of firms linked by transactions
Startling similarities in network structure as we move up the layers

- Plus lots of cross-sectional variation within this basic “core-periphery” architecture
- Bottleneck is documenting network linkages
  - Software links can be automatically extracted using commercial packages
  - Enterprise architecture links obtained by interview
  - Transaction links were hand-collected

*Ecosystem research basically stuck with hand-collected and interview data …*
Our maps are expensive and inaccurate!
We will struggle until we have good maps at several layers!
Best research idea (this week)

How can you tell the difference between a platform firm and a product firm?
I owe this idea to a pointer from Andrei Hagiu to a rant by Steven Yegge

Steven Yegge

– Software developer at Google formerly employed by Amazon
– Concerned (to say the least) that Google doesn’t “get” what it means to be a platform company vs. a product company
Yegge’s Rant

“That one last thing that Google doesn't do well is **Platforms**. We don't understand platforms. We don't "get" platforms. … This has become painfully clear to me over the past six years. I was kind of hoping that competitive pressure from Microsoft and Amazon and more recently Facebook would make us wake up collectively and start doing universal services.”
Yegge’s Ideas

- Platforms = Universal Services
- Platforms are Accessible to everyone
  - Very different from traditional exclusion through property rights
- In software, accessibility come through APIs (application programmer interfaces)
Perhaps ecosystems arise when there “easy hookups”

“Easy hookups” are the result of modular interfaces
That hide information behind an abstraction (Parnas, 1974)
Yegge’s evidence (and the research opportunity)

- And if you go to [msdn.com](http://msdn.com), and spend some time browsing, and you've never seen it before, prepare to be amazed. Because it's staggeringly huge. They have thousands, and *thousands*, and THOUSANDS of API calls. They have a HUGE platform. Too big in fact, because they can't design for squat, but at least they're doing it.

- Amazon gets it. Amazon's AWS ([aws.amazon.com](http://aws.amazon.com)) is incredible. Just go look at it. Click around. It's embarrassing. We don't have any of that stuff.

- Apple gets it, obviously. They've made some fundamentally non-open choices, particularly around their mobile platform. But they understand accessibility and they understand the power of third-party development and they eat their dogfood. And you know what? They make pretty good dogfood. Their APIs are a hell of a lot cleaner than Microsoft's, and have been since time immemorial.
Yegge’s evidence (and the research opportunity)

- Facebook gets it. That's what really worries me. That's what got me off my lazy butt to write this thing. <He doesn’t say where to look for Facebook APIs.>

- After you've marveled at the platform offerings of Microsoft and Amazon, and Facebook I guess (I didn't look because I didn't want to get too depressed), head over to developers.google.com and browse a little. Pretty big difference, eh? It's like what your fifth-grade nephew might mock up if he were doing an assignment to demonstrate what a big powerful platform company might be building if all they had, resource-wise, was one fifth grader.
So there you have it

- Visible differences in practice
- Tied to “platform” vs. “product”
- Tied to performance of ecosystem
- Differences seem obvious (if you know what to look for!)

This is an avenue worth exploring!
Thank you!