

---

# The Evolution of Digital Ecosystems: Discussion

Carliss Y. Baldwin

Platform Strategy Conference

Boston University

July 14, 2016

---

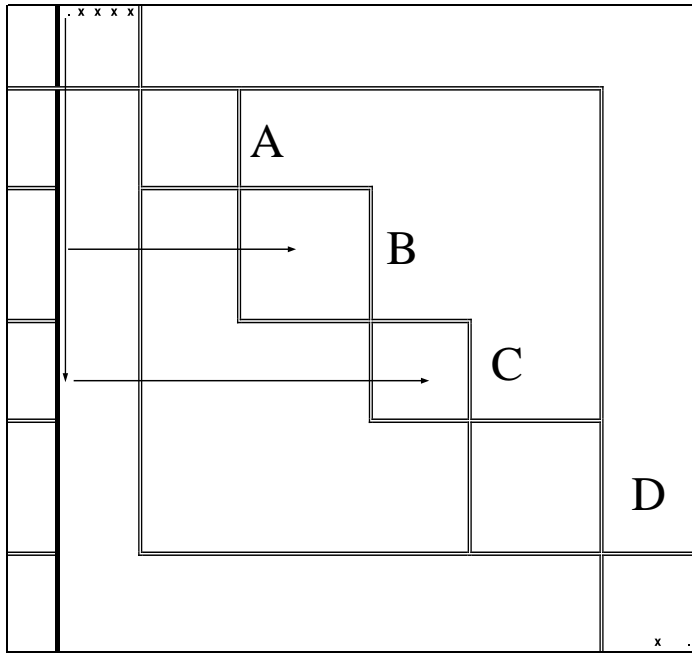
This is an important, exciting,  
provocative, frustrating,  
*exploratory work*

I will not be able to do it justice ...  
Apologies in advance

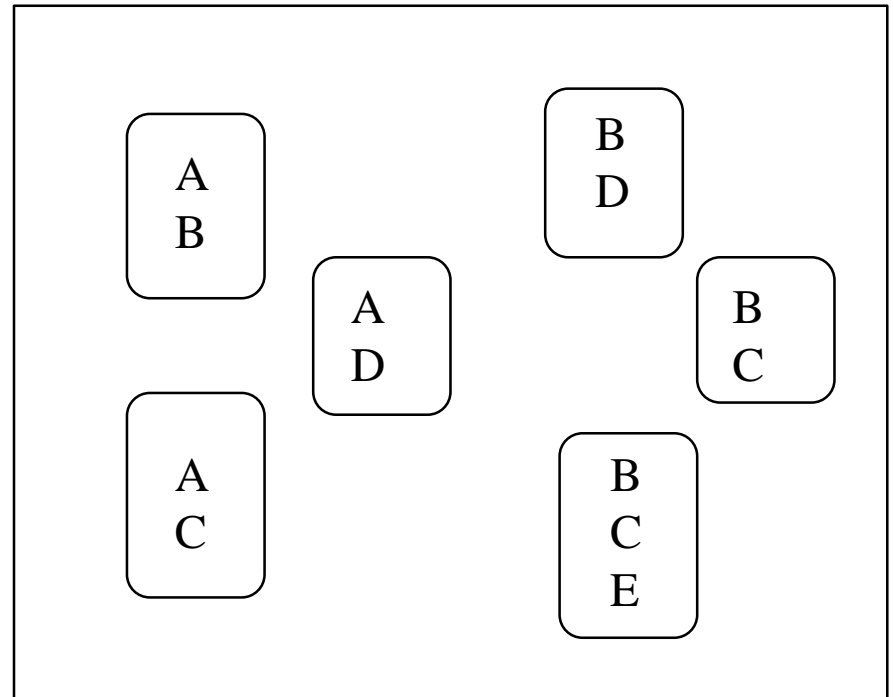
# The topic is recombination

- ◆ Modular evolution where “no explicit architecture provides a fixed boundary”

Not this—



But this—



---

# My first issue with the paper

# There IS a Framework Architecture

---

- ◆ There are design rules
  - *But they are very broad*
  - Byproduct of the WorldWideWeb
- ◆ Two technical versions of the architecture
  - Representational State Transfer (REST)
  - Simple Object Access Protocol (SOAP)
- ◆ *The nirvana of interoperability*

# What does it mean?

---

- ◆ Programs (called APIs) live on the Web
  - They are supplied by various people, often for free
  - To help users deal with commonly demanded tasks
- ◆ You can write a program of programs by setting up a bunch of APIs in a particular sequence
  - Like writing a paper by stringing together famous quotes...  
Washington, Jefferson, Lincoln, Shakespeare, Schumpeter, Dilbert, etc. ...
- ◆ APIs do generic things and then pass control back to your machine and the next API
- ◆ Platforms are not competitive without APIs

# “APIs do generic things and then pass control back to your machine and the next API”

---

- ◆ This is identical to “the creative recombination of existing modules”
- ◆ Thus today you can study the “evolution” of a software “ecosystem” by tracking usage of APIs
  - Particularly, which came first
  - And which are typically used together

---

That is the motivation of this  
paper

It is awesome



---

# That is the motivation of this paper

But remember... the APIs might know where they are going

But WE don't!

---

# My second issue with the paper

Authors are trying to do *normal science* when they are *really exploring a new domain*

# Implications

---

- ◆ There is no empirical base on which to build or test hypotheses
  - Throw out that section
- ◆ Unit of analysis is up for grabs
  - Component vs. program
  - Authors chose program, without discussing component
  - Reader wonders, “why this way?”

# Implications

---

- ◆ Must explain/justify methods
  - Not simply borrow from biology
  - Topological overlap measure is especially opaque (typo in formula?)
- ◆ Nature of evidence needs to be considered
  - Why are APIs important to the reader?
  - Is WordPress representative? Of what?
  - How has WordPress (NOT the network) evolved over time?
  - *Where would WordPress be without external APIs?*

---

*When you have a difficult problem  
and are using new, unproven  
methods, it helps to have a gripping  
question you can answer...*

# The gripping question...

---

- ◆ *How important are external APIs to the development of a specialized ecosystem?*
  - Paper ends here, it should *begin* here
  - This is a management question, consider your audience
  - You do not need clustering algorithms to answer this question (at the top level)

# And then, the next question ...

---

- ◆ *How do external APIs interact with platform-specific APIs?*
  - Clustering is useful here, but your clustering is too coarse
  - Need to show external/internal APIs in the Topological Overlay matrices

# The bottom line

---

- ◆ An interesting paper that opens up exciting new lines of inquiry
- ◆ A first attempt to look at the structure and evolution of a multi-platform ecosystem
- ◆ Uses state-of-the-art network theory from biology
  - but does not interpret it in the new context
- ◆ *Stay tuned! This methodology has legs.*