8-2006

Architectural Control and Value Migration in Platform Industries

C. Jason WOODARD
Singapore Management University, jason.woodard@olin.edu

School of
Information Systems

SMU
SINGAPORE MANAGEMENT UNIVERSITY

Architectural Control and Value Migration in Platform Industries

C. Jason Woodard
Assistant Professor
School of Information Systems
Singapore Management University

CMOST Lunch Time Seminar Series
National University of Singapore
19 October 2006

Outline

- Research agenda and tools
- Platform competition model
- Demo and simulation results

Research Agenda: Architectural Strategy

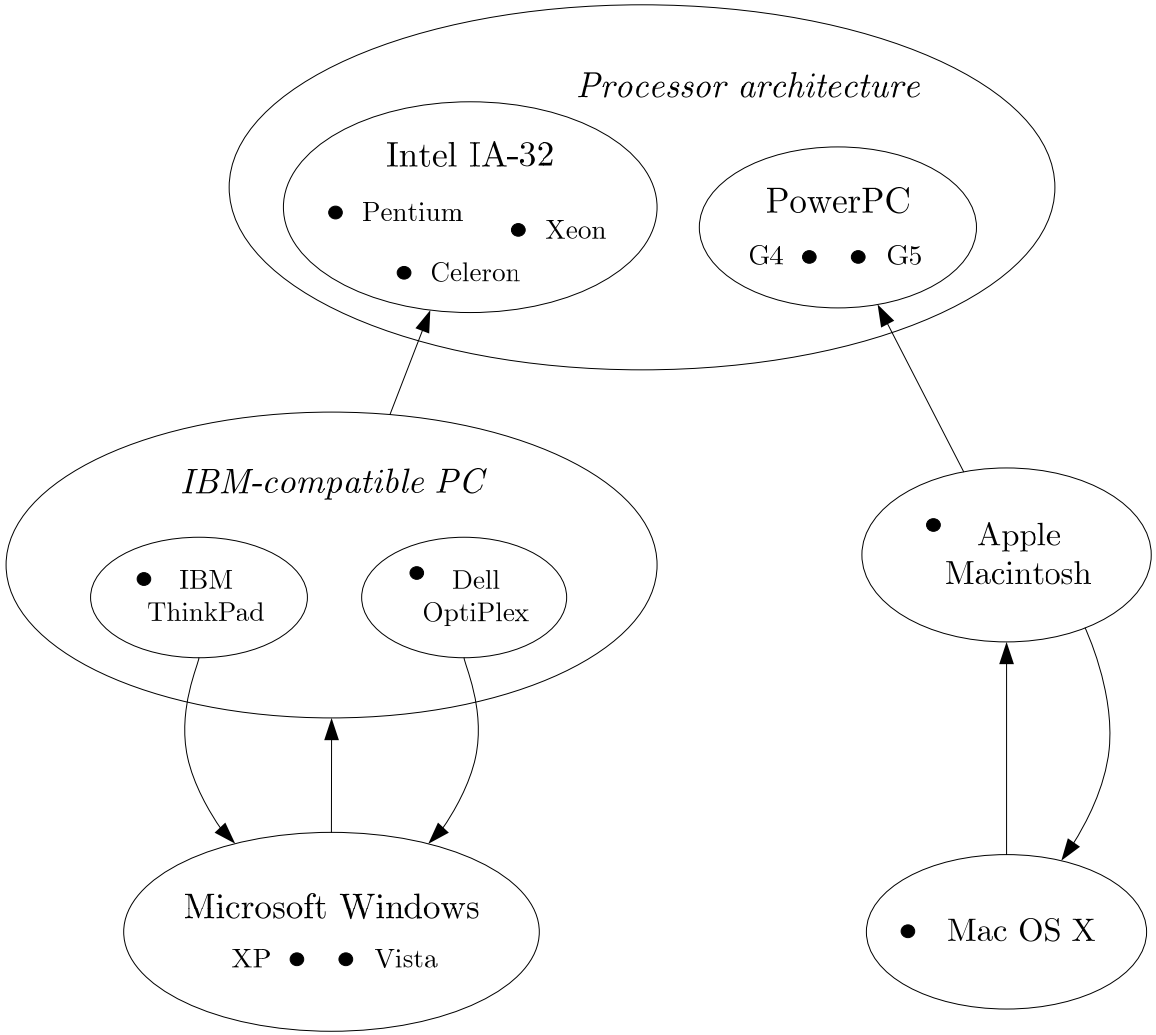
- How is value *created* and *captured* in complex engineered systems?

- How can (should) (do) value-seeking agents shape their structure and evolution?

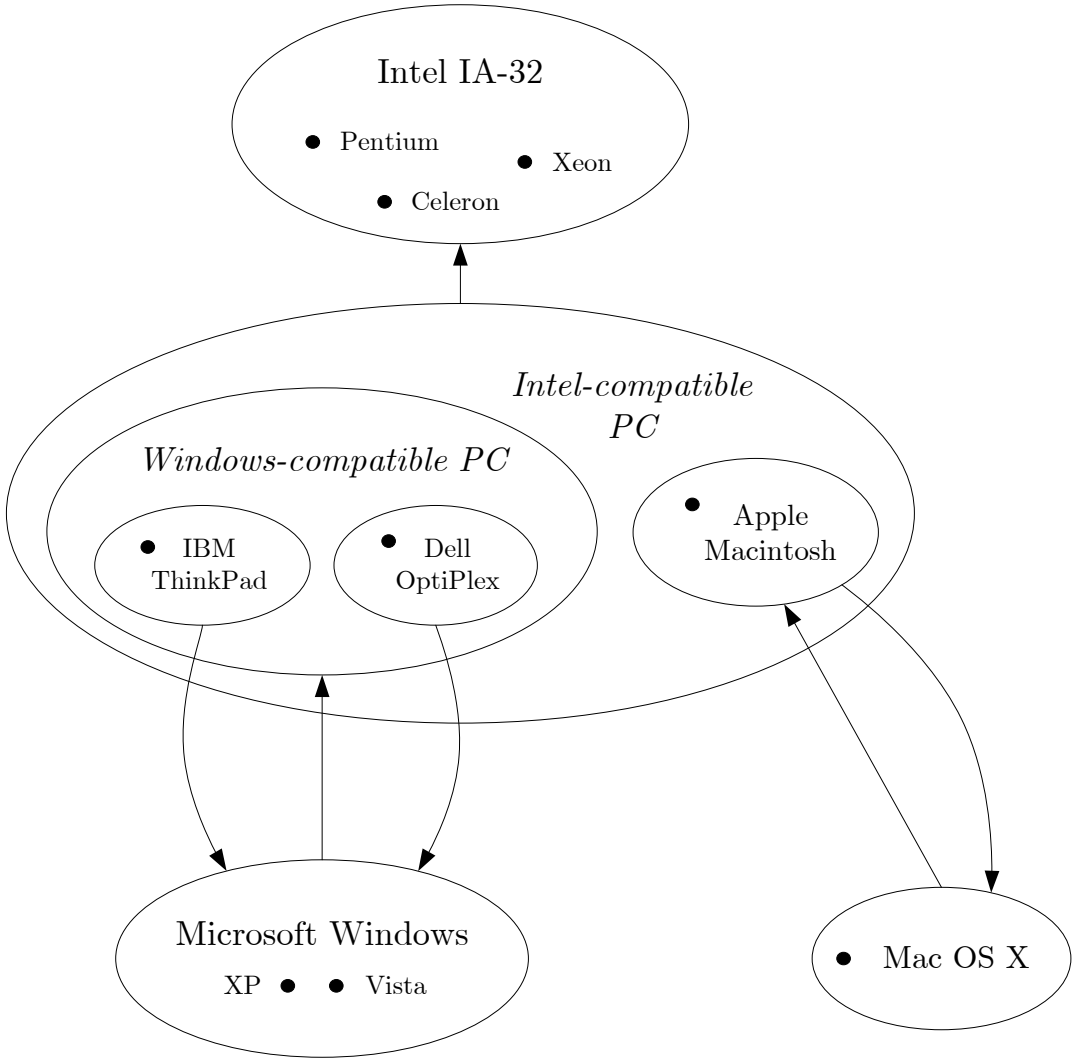
Motivation

- June 2005
  - Apple announces plans to transition Macintosh computers from PowerPC to Intel processors
- April 2006
  - Apple introduces Boot Camp software that enables Intel-based Macs to run Windows XP

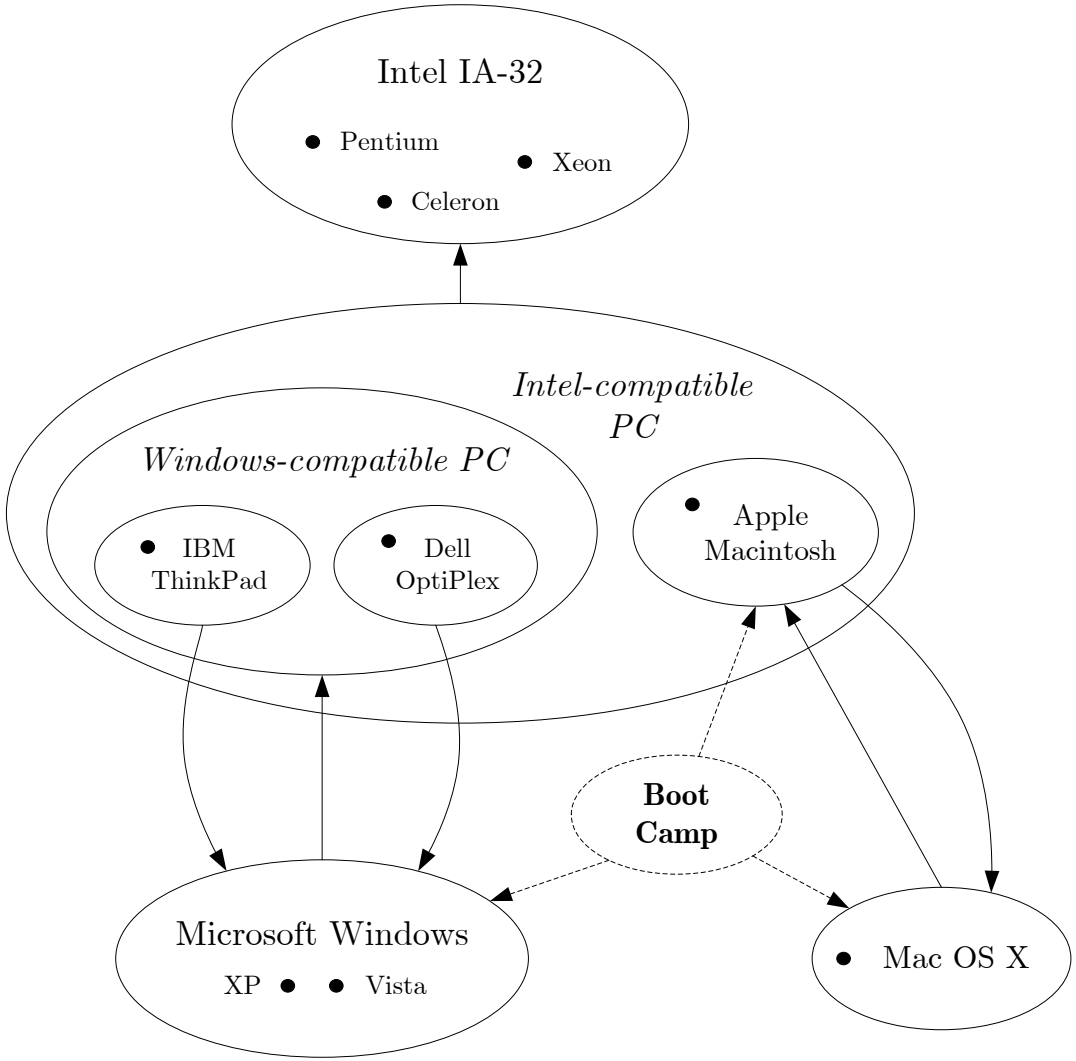
Old Game: Mac vs. PC



New Game: Mac *as* PC



Next: Mac as *Windows* PC?



Two Kinds of Research Challenges

- Explanation
  - Why did Apple make these design moves?

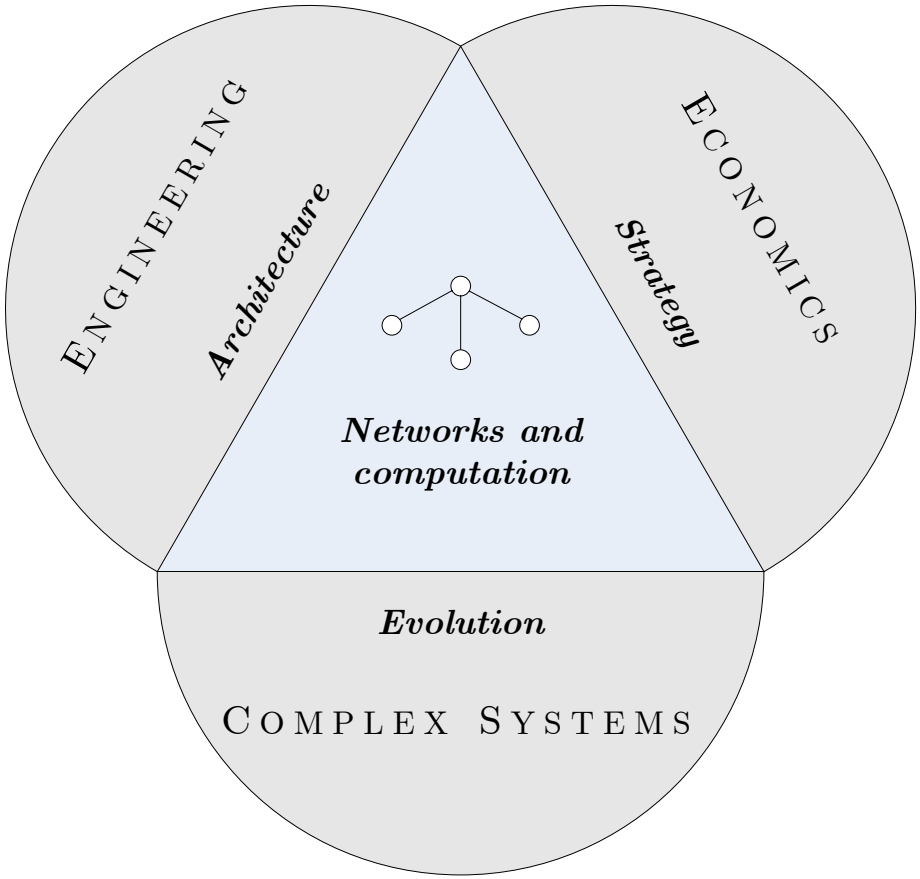
- Prediction
  - What will they do next?
  - How will their competitors respond?
  - What patterns will emerge at the industry level?

Why Is This Hard?

- Engineering tends to black-box “requirements”
- Economics tends to black-box “technologies”
  - Need to integrate across levels of abstraction

- But people do it in practice
  - Cases in point: Bill Gates, Scott McNealy

Related Fields and Concepts



Dissertation Work

- A formalism
  - Design structure networks (DSNs)
- A modeling approach
  - System design games (SDGs)
- Three models
  - Palm and Handspring in PDAs (analytical)
  - Value networks (computational / “closed”)
  - Platform competition (computational / “open”)

Dissertation Work

- A formalism
  - Design structure networks (DSNs)
- A modeling approach
  - System design games (SDGs)
- Three models
  - Palm and Handspring in PDAs (analytical)
  - Value networks (computational / “closed”)
  - **Platform competition (computational / “open”)**

Today’s focus!

A Model of Architectural Strategy in Platform Industries

- *Platform:* A system component (module) that is designed to be extended by *applications*
  - Software examples: Windows, Java, Google Earth
  - Examples from other engineering domains: NVIDIA nForce, Chrysler K-Car, Esplanade – Theatres on the Bay
  - What about chemical, biomedical, nanotech?

- Platforms may be built on top of other platforms ("*application-level platforms*")
    - Microsoft Outlook, Excel, Word
    - MSN Messenger? Skype?

Basic Setup

- Systems are tree-structured; new products and categories build on existing platforms

- Agents (firms) arrive in sequence, make a single product development decision
  - Where in the architecture to build?
  - How to attract applications and still make money?
    - What fraction of downstream profit to extract through *architectural control*?

Main Contributions

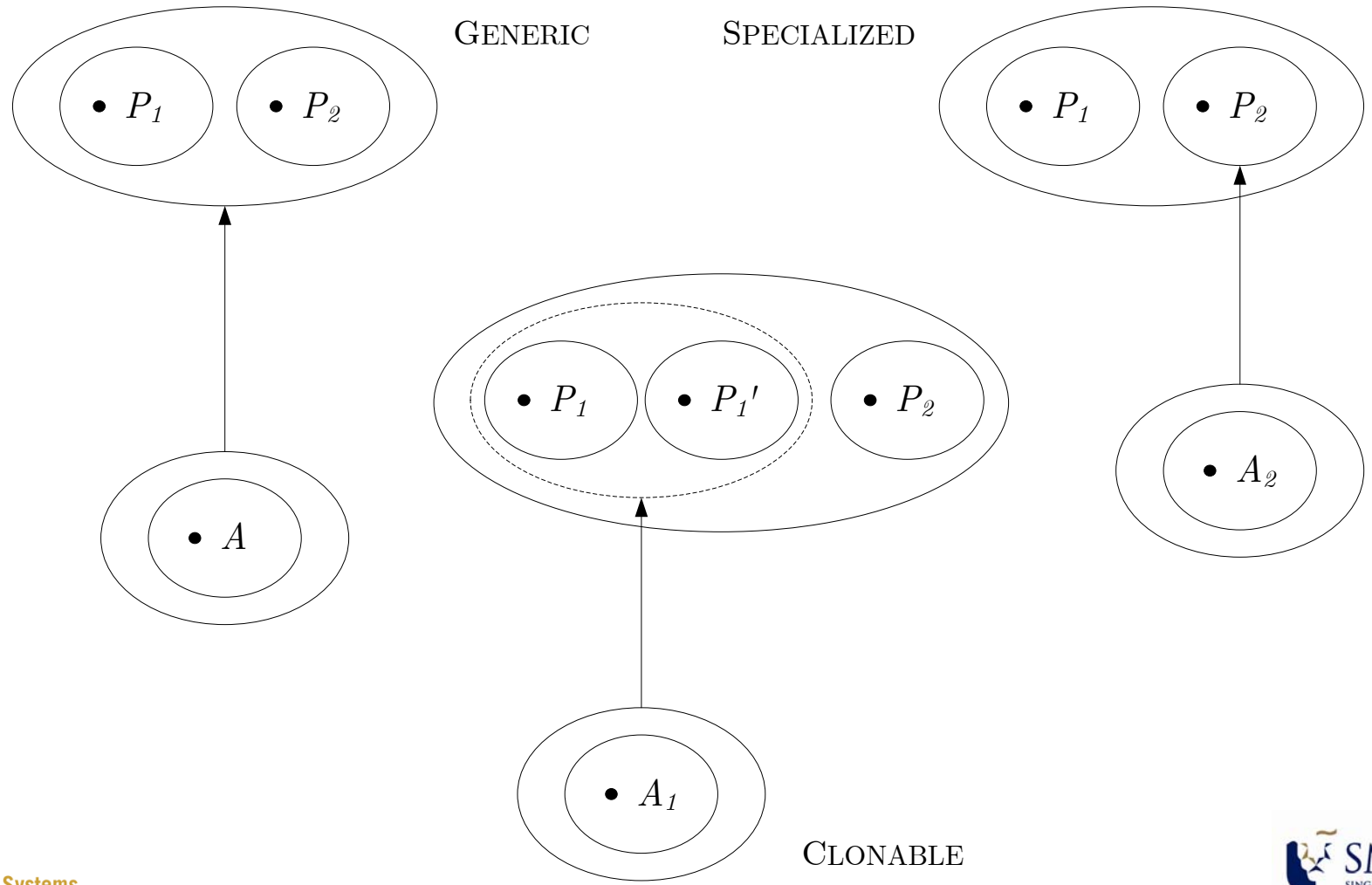
- Industrial economics: extends game-theoretic modeling in a “history-friendly” way
  - Network economics (Farrell & Saloner ’85, Katz & Shapiro ’94, …)
  - Industry evolution (Bresnahan & Greenstein ’99, Malerba et al. ’99, …)
- Technology management: connects industry dynamics to individual incentives
  - Platform competition (Baldwin & Clark ’00, Gawer & Cusumano ’02, …)
  - Industry evolution (Abernathy & Utterback ’78, Tushman & Murmann ’98, …)

Key Assumptions

- Firms are boundedly rational
  - Learn by observing payoffs of prior entrants
  - Choose action with highest expected payoff
    - Subject to cognitive limits; can only evaluate a fixed number of alternatives

- The future is uncertain
  - Innovation (product category arrival) is stochastic
  - Behavior of other firms also not fully predictable

Three Degrees of Dependence



From Use Value to Payoffs

- Each product category has a use value
  - Fixed at random upon arrival, then known to firms
  - Firms with products in category receive equal shares
    - Reduced form of a pricing game w/ differentiated goods

- Each platform owner chooses a “tax rate”
  - May be direct (e.g., licensing fees) or indirect (e.g., advantage in complementary product development)
    - Assumes platform owner can *exclude* application development through technical and/or legal means

From Use Value to Payoffs, Cont'd.

- Each firm's per-period payoff is its share of use value discounted by the tax rate of its parent platform
  - GENERIC case: no appropriability, so tax rate = 0
  - CLONABLE case: minimum of clones' tax rates

- Intuition check!
  - *A firm's payoff goes **down** over time as competitors enter its category, and **up** as complementors build on its product platform*

Learning Algorithm

- Firms run linear regressions of prior entrants' current NPV on characteristics of product location and tax choice
  - Category depth, width, and value
  - Parent tax, tax rate, tax^2, etc.

- Algorithm "seeded" by observations of prior runs (industry experience)

Questions for the Experiments

- Who makes the most money, and when?
- What is the optimal tax rate, and do firms learn to charge it?
- How does the level of taxation affect the "shape" of the system architecture?

Answers in a Nutshell

- Who makes the most money, and when?
  - *Generally top-level platforms (early in GENERIC case, later in others), but applications compensate through niche selection*
- What is the optimal tax rate, and do firms learn to charge it?
  - *Declines by level from ~0.7 → 0.2. Yes! (But high variance)*
- How does the level of taxation affect the "shape" of the system architecture?
  - *High taxes lead to shallow, "bushy" systems with few niches; low taxes to deep systems with "vine-like" dependencies*

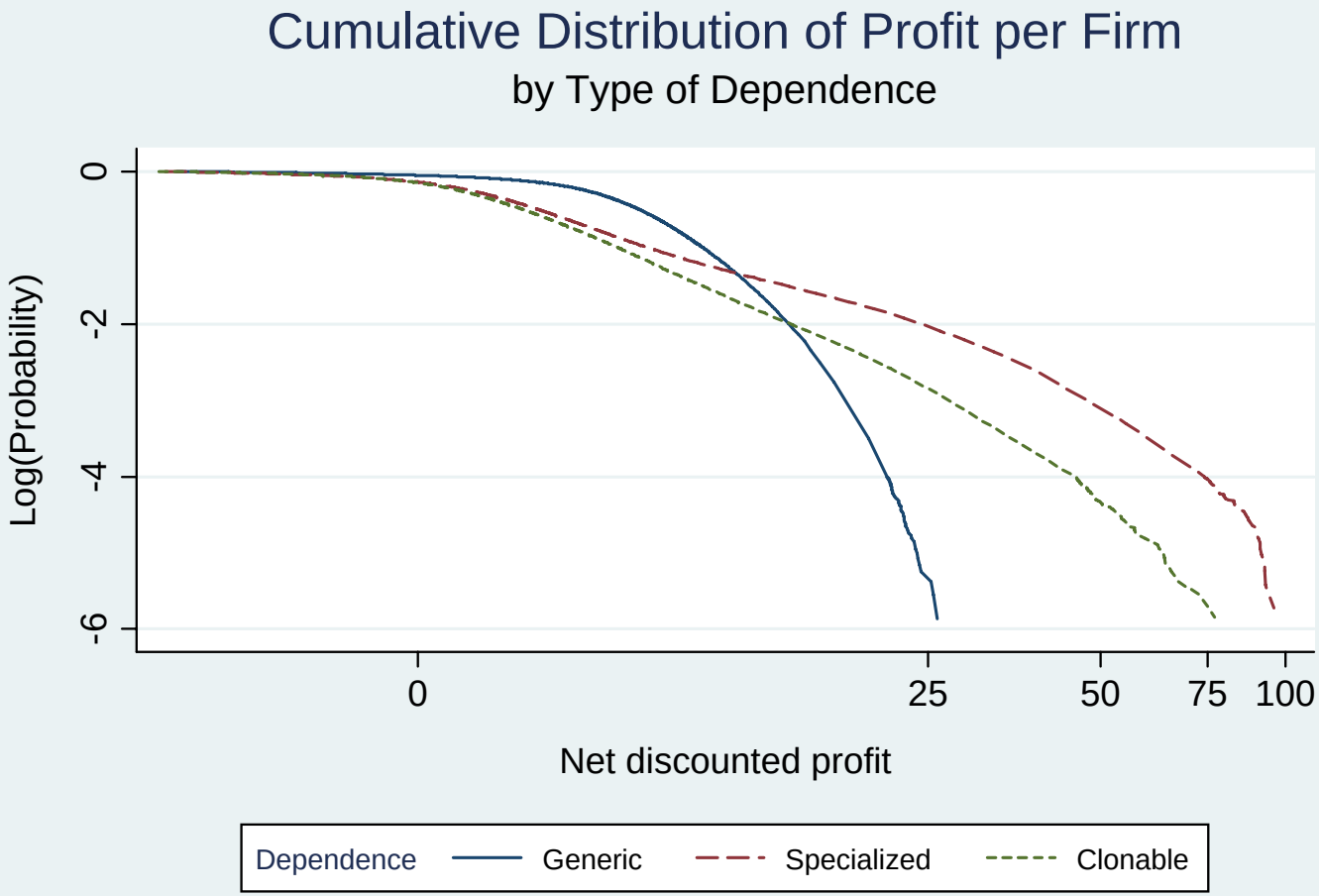
Key Takeaways

- Top-level architectural positions are scarce; it's generally "good to be the king"
- But platform owners voluntarily limit their exercise of architectural control to attract application developers
- Cause to be (cautiously) optimistic that market forces can sustain open multi-product systems even when appropriability is strong

Demo / Eye Candy

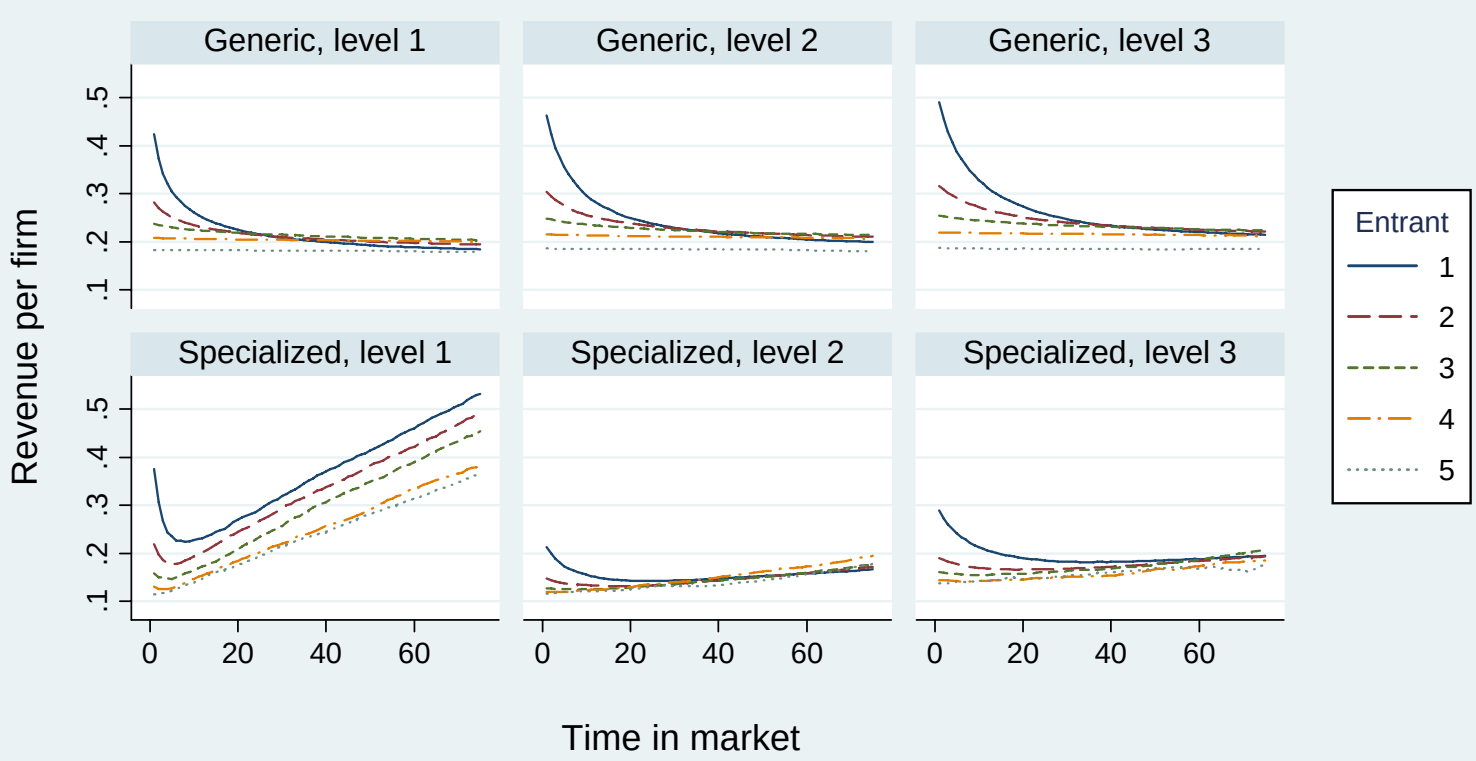
- How am I doing on time?

Cumulative Profits Are Power-Law Distributed



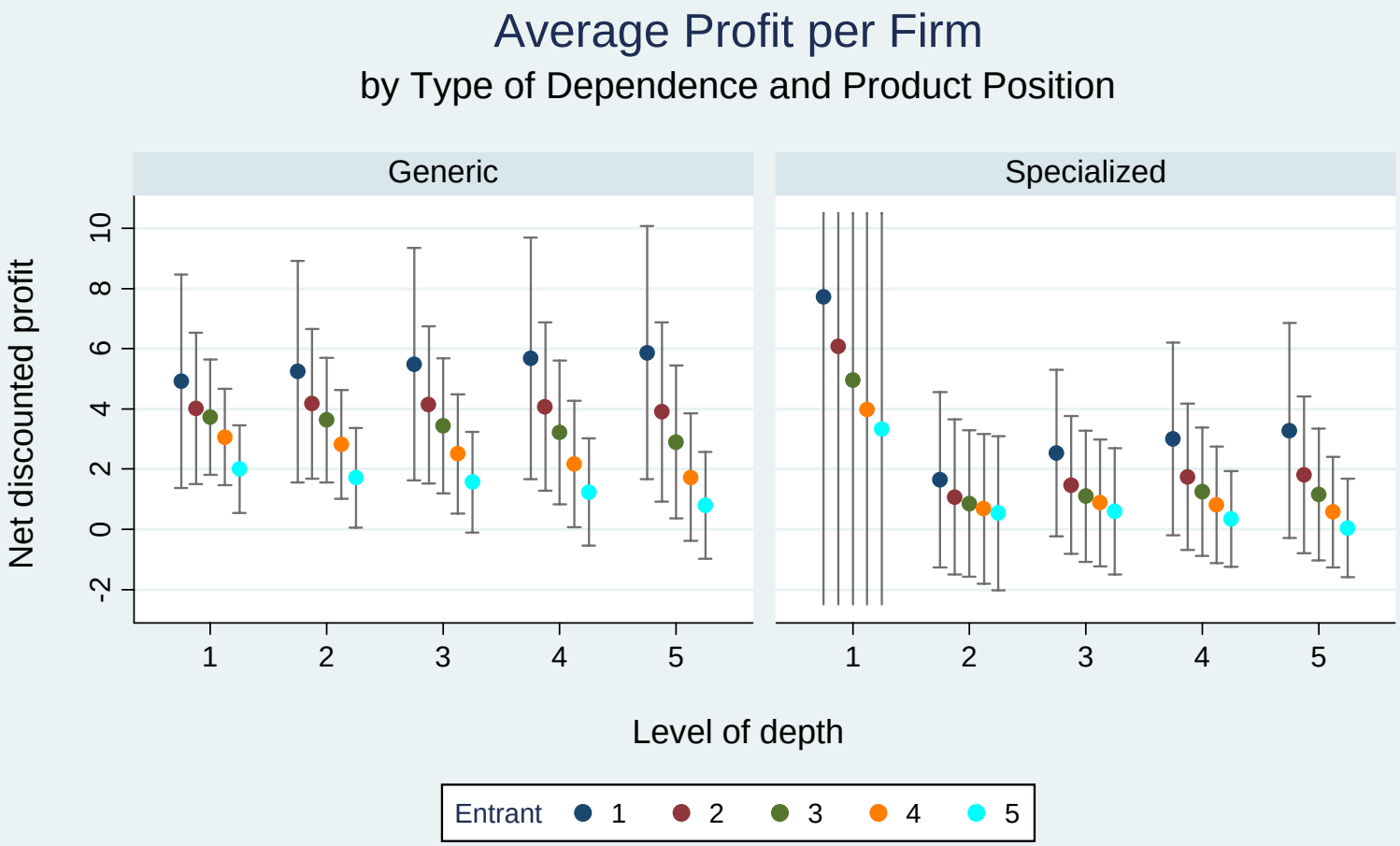
Experiments III-A, III-B, III-C

Top-Level Firms Enjoy Increasing Returns When Architectural Control is Strong



Experiments III-A, III-B

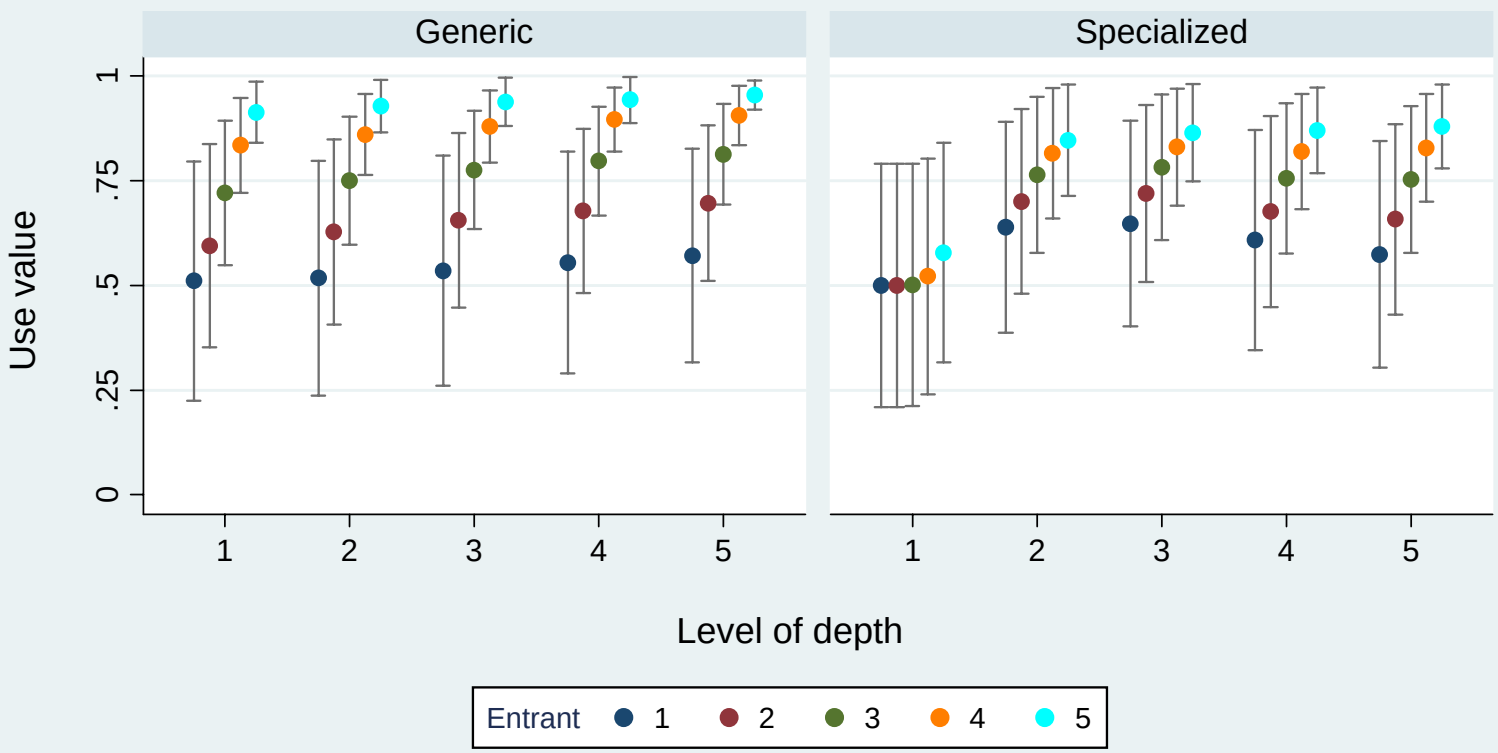
Leading to Higher Net Discounted Profit (But Note Upward Trend at Lower Levels)



Experiments III-A, III-B

Bars indicate +/- 1 s.d.

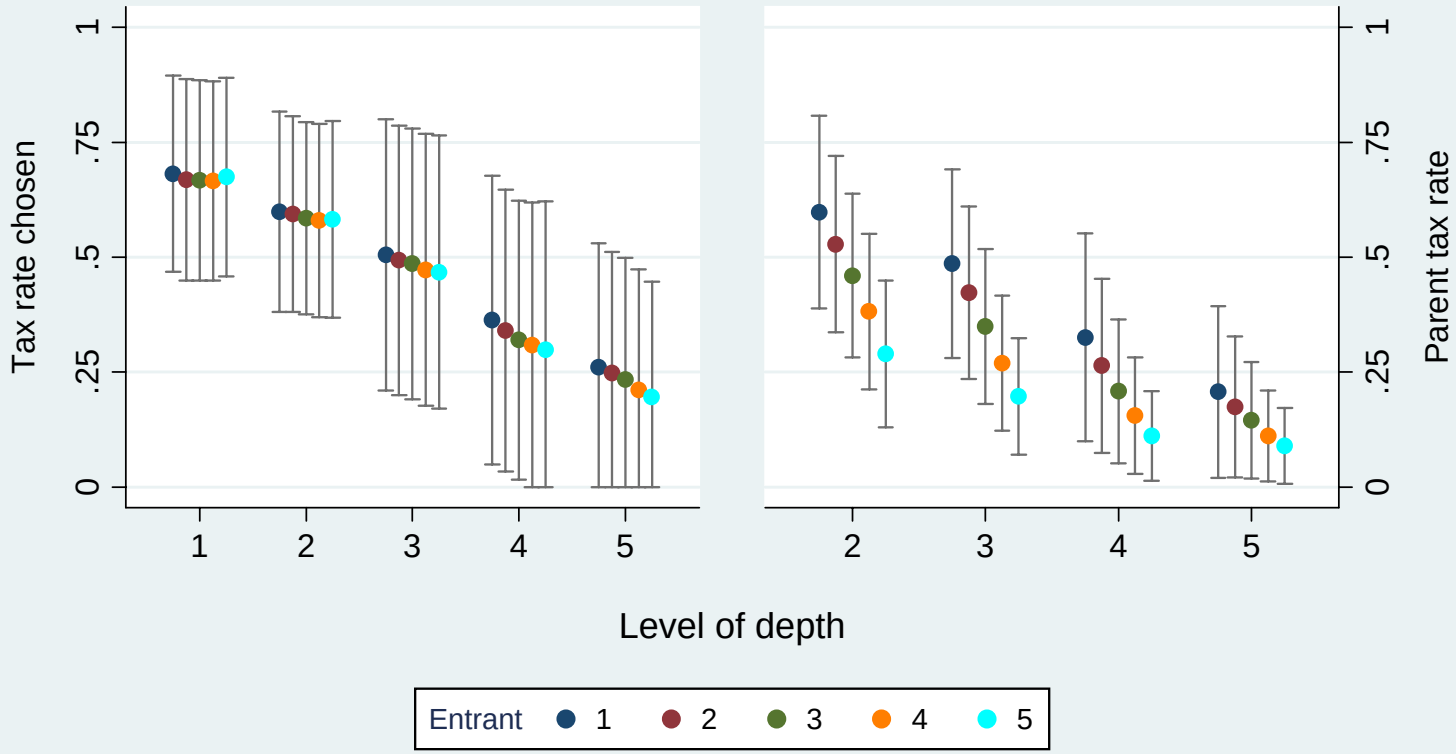
Revenge of the Applications (Part I)



Experiments III-A, III-B

Bars indicate +/- 1 s.d.

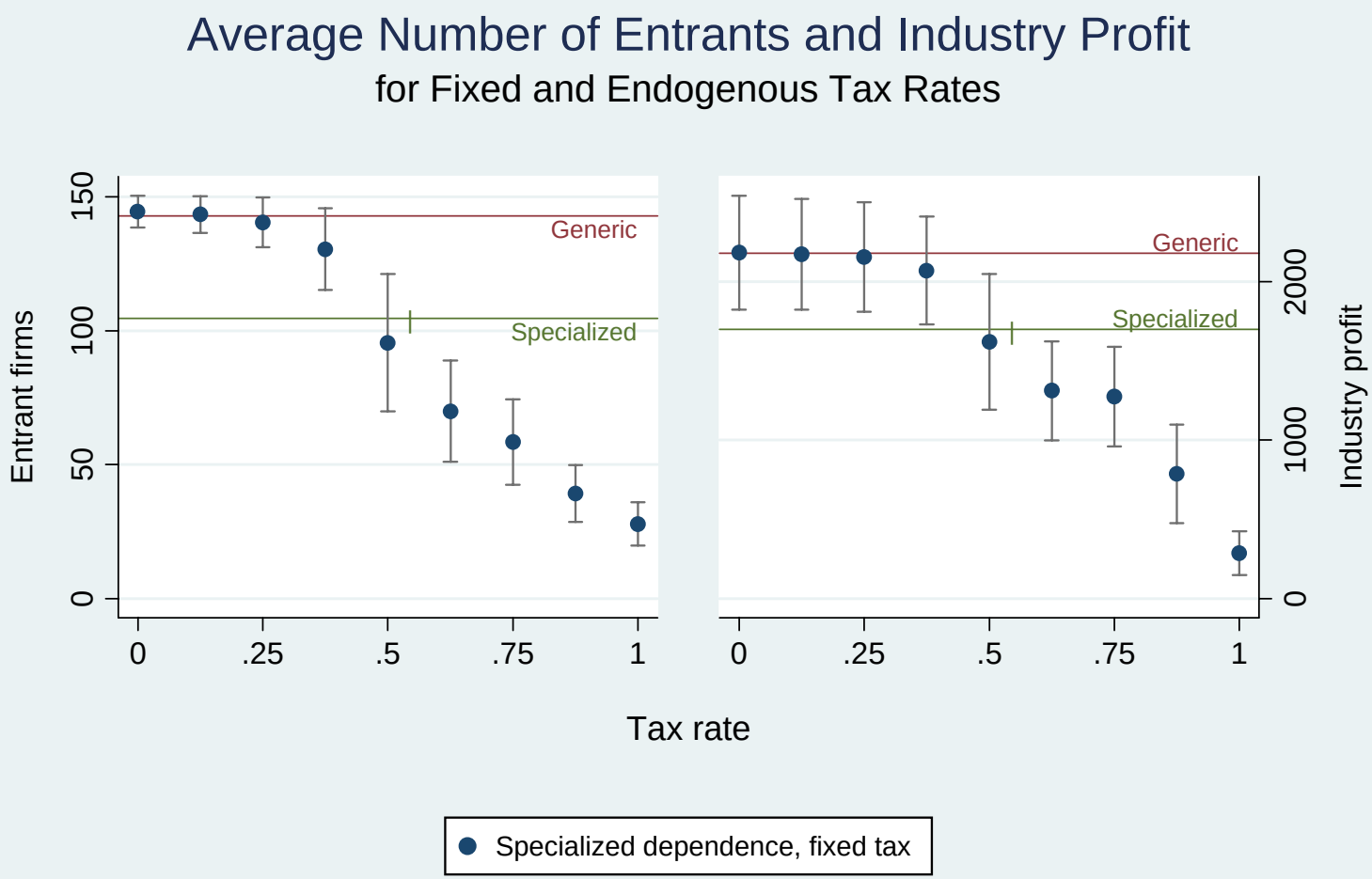
Revenge of the Applications (Part II)



Experiment III-B (specialized dependence)

Bars indicate +/- 1 s.d.

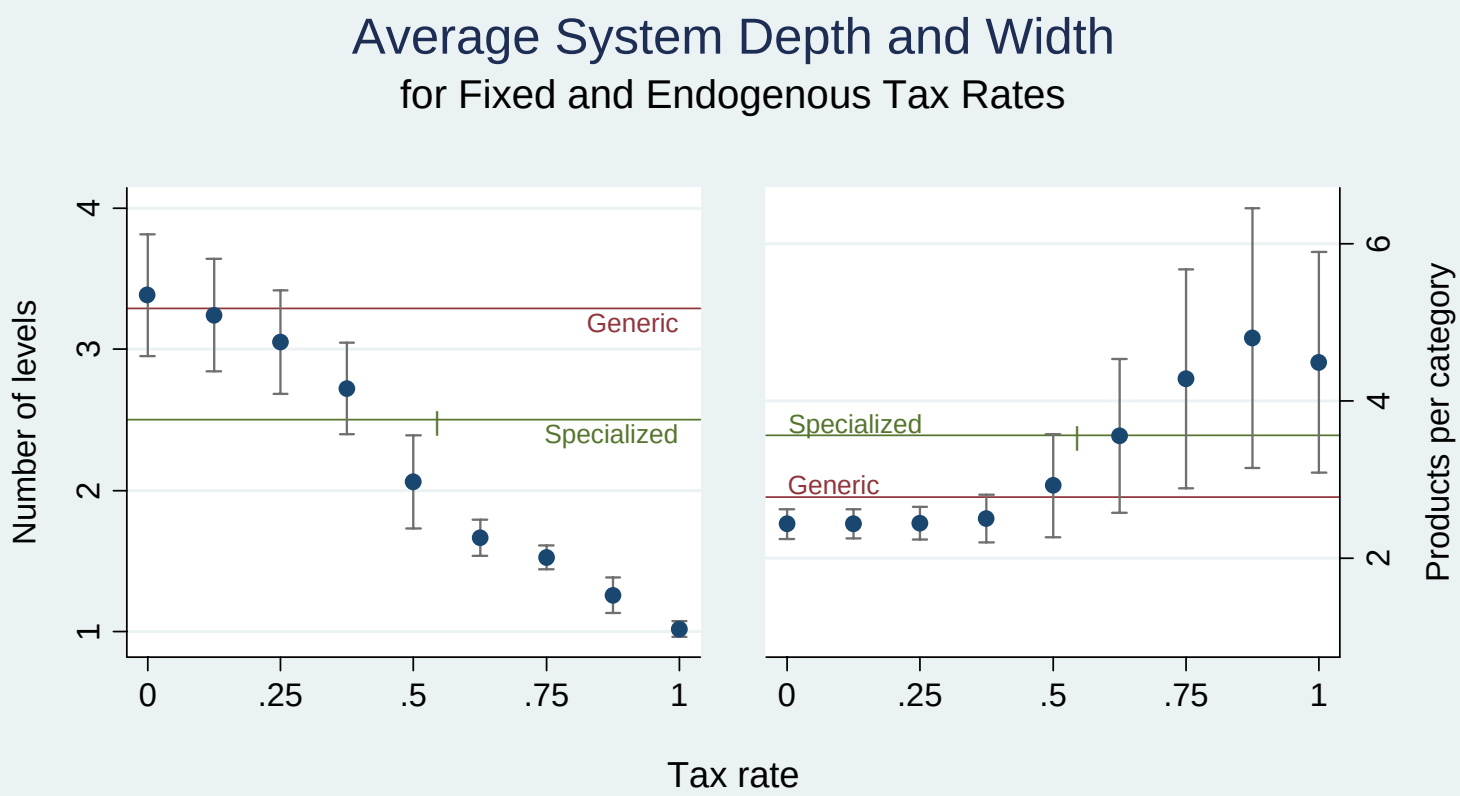
Fixed-Tax Result: Higher Taxes Lead to Less Entry, Lower Profit



Experiments II, III-A, III-B

Bars indicate +/- 1 s.d.

Under High Taxes, Firms Crowd Into Root Layers and Build Shallow Trees



Experiments II, III-A, III-B

Bars indicate +/- 1 s.d.

Summary

- Architectural strategy can and should be integrated with engineering design
- The resulting tools can help shed light on strategic moves like platform creation and the exercise of architectural control
- Computational models give dynamics for free – can study industry evolution *in silico*

Next Steps

- Platform model
  - More focus on value migration over time
  - Allow firms to develop multiple products
- Computational tools
  - An exercise in platform creation …
- Explanation and prediction
  - Engage directly with empirical evidence

Thank You!

- This paper and more at http://kuala.smu.edu.sg/~jason
- Or by email to jwoodard@smu.edu.sg