Making Money from Design Architecture

Carliss Y. Baldwin
Harvard Business School

Product Design and Development, MIT
February 27, 2007
Three Points to begin

◆ Large, complex, evolving designs
  – Are a fact of modern life
  – Need *design architectures* —
    » “Description of the entities in a system and their relationships”
    » Way of assigning work (Parnas)
    » Covers products and processes

◆ Designs create value
  – Value operates like a force in the economy
  – We fight to create it and to keep it—using *strategy*

◆ Design Architecture and Strategy
  – *How can you create and capture value in a complex evolving set of designs?*
  – Subject of this talk
In the economy, value acts like a force

Value = money or the promise of money

Consider the computer industry…
The changing structure of the computer industry

Andy Grove described a vertical-to-horizontal transition in the computer industry:

1980-“Vertical Silos”

1995-“Modular Cluster”
Andy’s Movie
Stack View in 1985

Top 12 Public Companies in US Computer Industry
Area reflects Market Value in Constant US $
Andy’s Movie
Stack View in 2005

Top 12 Public Companies in US Computer Industry
Area reflects Market Value in Constant US $
# Turbulence in the Stack

**Departures from Top 12:**
- Texas Instruments
- Intergraph Corp.
- Cray Research
- Automatic Data
- Advanced Micro
- Gould
- National Semiconductor
- Computervision
- Tandem

**Arrivals to Top 12:**
- Microsoft
- Google
- Cisco
- HP
- Dell
- Ebay
- SAP
- Yahoo
- TSMC

*Sic Transit Gloria Mundi … Sic Transit*
Contrast to the Auto Industry

Top 12 Public Companies in US Auto Industry

Area reflects Market Value in Constant US $
Two patterns

◆ “Manageable” designs = auto industry
◆ “Unmanageable” designs = computer industry

What makes computer designs so unmanageable?

This was the question Kim Clark and I set out to answer in 1987.
After studying the history of computer designs and correlating their changes with value changes

We concluded that *modularity* was part of the answer...
Modularity in computers—IBM System/360

◆ First modular computer design architecture (1962-1967)
  – Proof of concept in hardware and application software
  – Proof of option value in market response and product line evolution
  – System software was NOT modularizable
    » Fred Brooks, “The Mythical Man Month”
    » Limits of modularity
Strategically—IBM wanted to be the sole source of all of System/360’s Modules
By 1980, 100s of firms made S/360 “plug-compatible” components

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3570</td>
<td>Computer and Office Equipment</td>
<td>5</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>3571</td>
<td>Electronic Computers</td>
<td>1</td>
<td>8</td>
<td>29</td>
</tr>
<tr>
<td>3572</td>
<td>Computer Storage Devices</td>
<td>1</td>
<td>6</td>
<td>36 *</td>
</tr>
<tr>
<td>3575</td>
<td>Computer Terminals</td>
<td>2</td>
<td>5</td>
<td>23 *</td>
</tr>
<tr>
<td>3576</td>
<td>Computer Communication Equipment</td>
<td>1</td>
<td>1</td>
<td>10 *</td>
</tr>
<tr>
<td>3577</td>
<td>Computer Peripheral Devices, n.e.c.</td>
<td>3</td>
<td>5</td>
<td>12 *</td>
</tr>
<tr>
<td>3670</td>
<td>Electronic Components and Accessories</td>
<td>11</td>
<td>7</td>
<td>11 *</td>
</tr>
<tr>
<td>3672</td>
<td>Printed Circuit Boards</td>
<td>2</td>
<td>19</td>
<td>39 *</td>
</tr>
<tr>
<td>3674</td>
<td>Semiconductors and Related Devices</td>
<td>8</td>
<td>4</td>
<td>10 *</td>
</tr>
<tr>
<td>3678</td>
<td>Electronic Connectors</td>
<td>5</td>
<td>15</td>
<td>16 *</td>
</tr>
<tr>
<td>7370</td>
<td>Computer Programming, Data Processing, and Other Services</td>
<td>1</td>
<td>9</td>
<td>26 *</td>
</tr>
<tr>
<td>7371</td>
<td>Computer Programming Services</td>
<td>0</td>
<td>2</td>
<td>12 *</td>
</tr>
<tr>
<td>7372</td>
<td>Prepackaged Software</td>
<td>0</td>
<td>7</td>
<td>13 *</td>
</tr>
<tr>
<td>7373</td>
<td>Computer Integrated Systems Design</td>
<td>1</td>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td>7374</td>
<td>Computer Processing, Data Preparation and Processing</td>
<td>0</td>
<td>5</td>
<td>29 *</td>
</tr>
<tr>
<td>7377</td>
<td>Computer Leasing</td>
<td>0</td>
<td>10</td>
<td>7   *</td>
</tr>
</tbody>
</table>

* Firms in these subindustries make modules of larger computer systems.
  
* Firms making modules = 34 95 244
  
* Percent of total = 83% 88% 82%
Modularity and Option Value interact

- IBM did not understand the option value it had created
- Did not increase its inhouse product R&D
- Result: Many engineers left
  - to join “plug-compatible peripheral” companies
- San Jose labs \( \rightarrow \) Silicon Valley
Modularity in computers, cont.

- Bell and Newell, Computer Structures (1971)
  - General principles of modular design for hardware
  - Basis of PDP-11 design—another ORMDA
- Thompson and Ritchie, Unix and C (1971-1973)
  - Modular design of operating system software (contra Brooks Law)
- Parnas (1972) abstract data structures, info hiding
  - Object-oriented programming, C++, Java
- Mead and Conway, Intro to VLSI Systems (1980)
  - Principles of modular design for large-scale chips
Modularity in computers (cont.)

- IBM PC (1983)
  - DEC PDP-11 minimalist strategy (exclude and invite)
  - + Intel 8088 chip
  - + DOS system software
  - + IBM manufacturing
  - + Lotus 1-2-3

A modular design architecture with a mass market
Measuring modularity

- As scientists, we can visualize and measure modularity
  — after the fact
- DSMs, Design Hierarchies
  — Methods are tedious, non-automated
Comparison of different software systems with DSM tools

Mozilla just after becoming open source

Linux of similar size

Coord. Cost = 30,537,703
Change Cost = 17.35%

Coord. Cost = 15,814,993
Change Cost = 6.65%

© Alan MacCormack, Johh Rusnak and Carliss Baldwin, 2006
Conway’s Law: Different organizations deliver different architectures

Mozilla just after becoming open source

Coord. Cost = 30,537,703  
Change Cost = 17.35%

One Firm, Tight-knit Team, RAD methods

Distributed Open Source Development

Linux of similar size

Coord. Cost = 15,814,993  
Change Cost = 6.65%

© Alan MacCormack, Johh Rusnak and Carliss Baldwin, 2006
Refactoring for modularity

Mozilla Before Redesign

Mozilla After Redesign

© Alan MacCormack, Johh Rusnak and Carliss Baldwin, 2006
But modular ≠ unmanageable

- There are modular design architectures that are very manageable
  - Toyota Production System (TPS)

- The other property of a design architecture => option potential
- Denoted $\sigma$—measures variation of outcomes in a design search space
- $\sigma$/Option potential varies by system and by module
σ /Option potential

- Unmanageable systems => High σ
- Manageable systems => Low σ

- If System/360 had had lower σ /option potential, IBM would be like Toyota

*It would be a different world…*
An Option is

- The *right but not the obligation* to take an action
  - Action = Use a new design
  - If new is better than old, use new;
  - Otherwise, keep the old.

*Designs are Options!*
Evidence of $\sigma$ /Option value

- Successive, improving versions are evidence of option potential being realized over time—after the fact.

$\sigma = \text{Low} \quad \text{Medium} \quad \text{Zero} \quad \text{High}$
Option potential at work—Matlab programming contest
σ /Option potential is like dark matter in the universe

- Scientists can measure its effects but we can’t measure “it”
- “Wizards” can perceive σ /option potential
  - But wizards don’t talk to scientists!
- Thus we lack ways to measure σ /option value scientifically
  - It is a “research frontier”
Sources of $\sigma$ /option value

- **Physics** —
  - Moore’s Law (dynamics of miniaturization) applies to MOSFET circuits and systems (Mead and Conway)
  - Power and heat systems vs. logic systems (Dan Whitney)

- **User innovation**
  - Users’ discovery of their own needs
  - “Killer apps”

- **Architecture**
  - Experimenting with different relationships among entities
Recapping the argument

- Designs create value
  - Value operates like a force in the economy
  - Changes the structure of industries

- Designs have architectures
  - Modularity and $\sigma$/Option value are the key economic properties of a design architecture
  - $Modularity + High \sigma/Option value => Unmanageable$
    - Why computers are not like autos
    - Why IBM is not Toyota

Next—

How can you capture value in a modular, high-$\sigma$ design architecture?
How can you *capture* value?

- **Architectural/Technological question:**
  - Where/when does *modularization* stop?

- **Business/Strategic question:**
  - How do you *make money* from design architecture?
Where does modularization stop?

Strojwas (2005)
Semiconductor Industry
Top 10 Firms: 1994 and 2004
Modularization stops when integral and modular architectures have essentially equal ROICs

It IS all about ROIC (Return on Invested Capital)
Design Architecture in Competition

- A tale of two industries
  - Computer workstations and PCs
  - Bicycle drive trains
“Footprint” Competition—Apollo

Keeps Design Control of Core

Key:
- x = transfer of material or information from column task to row task;
- T = transaction: sale of good by column owner to row owner;
- O = outsourced task blocks;
- D = downstream or complementary task blocks;
- K = highly interdependent task blocks with many iterations and high within-block mundane transaction costs;

Apollo’s footprint (tasks performed inhouse).
Then Sun came along…

And did even less!

How?
Then Sun came along…

Uses architectural knowledge to redefine Core

Uses public standards for outsourcing (process modularization)

And did even less!

How?
Result: A smaller footprint = less Invested Capital

<table>
<thead>
<tr>
<th></th>
<th>Apollo Computer</th>
<th>Sun Microsystems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Invested Capital Ratios (Annualized)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Working Capital/ Sales (%)</td>
<td>29%</td>
<td>15%</td>
</tr>
<tr>
<td>Ending Net PPE / Sales (%)</td>
<td>24%</td>
<td>13%</td>
</tr>
<tr>
<td>Invested Capital/Sales (%)</td>
<td>57%</td>
<td>31%</td>
</tr>
</tbody>
</table>

**Profitability**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Net Income/Sales</td>
<td>0%</td>
</tr>
</tbody>
</table>

**ROIC**

<table>
<thead>
<tr>
<th></th>
<th>Apollo Computer</th>
<th>Sun Microsystems</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROIC (excl Cash, Annualized)</td>
<td>2%</td>
<td>20%</td>
</tr>
</tbody>
</table>

Sun used its Invested Capital advantage to drive Apollo out of the market

Apollo was acquired by HP in 1989
Compaq vs. Dell

- **Dell did to Compaq what Sun did to Apollo …**

- **Dell** created an equally good machine, and

- Used *design architecture* to reduce its footprint in production, logistics and distribution costs
  - Negative Net Working Capital
  - Direct sales, no dealers

- **Result:** Invested Capital advantage => Higher ROIC
Higher ROIC always wins!

<table>
<thead>
<tr>
<th>Year</th>
<th>Compaq Computer</th>
<th>Dell Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1997</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Invested Capital Ratios (Annualized)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Working Capital/ Sales (%)</td>
<td>-2%</td>
<td>-5%</td>
</tr>
<tr>
<td>Ending Net PPE / Sales (%)</td>
<td>8%</td>
<td>3%</td>
</tr>
<tr>
<td>Invested Capital/Sales (%)</td>
<td>8%</td>
<td>-2%</td>
</tr>
<tr>
<td><strong>Profitability</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Net Income/Sales</td>
<td>8%</td>
<td>7%</td>
</tr>
<tr>
<td><strong>ROIC</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROIC (excl Cash, Annualized)</td>
<td>101%</td>
<td>-287%</td>
</tr>
</tbody>
</table>

Dell started cutting prices; Compaq struggled, but in the end had to exit.

Compaq was acquired by HP in 2002
Bike Drive Trains were a vertically fragmented industry

In the beginning (1980), there were 6 components with mix-and-match modularity

Three major firms, lots of “others”

Then Shimano introduced “index shifting”

– An “integral” product architecture
– Parts had to be aligned “just so”
– All pieces had to fit together, no more mix and match
Index shifting was a “killer app”

Everybody wanted it!
In the beginning (1980)
Introduction of Index Shifting (1985)
In 1990, only Vertical Silos left
In Mountain Bikes… Only Shimano
Remember

◆ Designs create value
  – Value operates like a force in the economy
  – Changes the structure of industries

◆ Designs have architectures
  – Modularity and $\sigma$/Option value are the key economic properties of a design architecture
  – Modularity + High $\sigma$/Option value $\Rightarrow$ Unmanageable
    » Why computers are not like autos
    » Why IBM is not Toyota

◆ You can use design architectures to compete/capture value
  – ROIC RULES—but you can get there in different ways
  – There is no “one best architecture”
Thank you!