Complexity Theory

Some basic concepts

Ole Hanseth

Changing IT landscape

- More applications, more use areas
- More integrations – old and new
- New technologies
  - Internet
  - Cloud
  - Internet of Things …

Growing complexity

- From applications (a few, stand-alone)
- To Platform Ecosystems (platform and apps, platform owner and app developers)
- To Information Infrastructures huge number of interacting components developed by independent actors)
  - Internet
  - Supply chain, bank and financial services, programmatic advertisement, ..
  - Portfolios of numerous (thousands) of integrated applications in large and distr. orgranizations (oil, bank, health care, ..)

Implications of complexity

- Development projects fail
  - ePresecription, Connecting for Health, Flexus, KA
- Reorganizations fail
  - NAV, new penal law, Oslo University Hospital, ..
- Breakdowns – disasters
  - Telenor Mobile, AHUS, ATMs
- Use/data errors
  - Patient data, …
- Security
  - cybercrime
  - From 9/11 to Wikileaks …
  - US presidential election

Why Information Infrastructures?

- Infrastructures last forever, big and heavy
- Evolving ***installed base***, not designed from scratch
- II development
  - Not designing dead material – shaping the evolution
  - Cultivating living organisms

From IS to II: A new paradigm

- From
  - Tool (individual)
  - System (closed)
  - Design (from scratch)
- To
  - Infrastrcuture (shared)
  - Network (open)
  - (Installed base) Cultivation

What is an information infrastructure?

- An info. infra. is a
  - shared,
  - Evolving & open,
  - heterogeneous,
  - **installed base**, which is also
  - (and standardized in one way or another).

  - **No life cycle**

- Opposite of Information/Software systems
  - Stand-alone, simple, designed from scratch, unique for the user group

Complexity

- Complexity: Socio-technical (Internet, globalization)
- Complexity (-ies) = Number of types of components*number of types of links*speed of change
- Key issues: emergence, ***side-effects*** (=history), incomplete knowledge, unpredictability, out-of-control
- Complexity theories
  - Actor network theory:
  - Complexity Science: self-reinforcing processes, driven by side-effects (network externalities)
  - Reflexive Modernization: Self-destructive processes
  - Assemblage Theory: stabilizing and destabilizing processes

Emergence

- Events (car crash, explosion, …)
- “order”
- De-facto standard (TCP/IP, Windows, QWERTY ..)

- New species: Panda with thumb
- Order in a beehive
- Arab spring
- Financial crises
- Climate change

Complexity & Control/Risk

- Complexity = limited knowledge/understanding = risk
- Charles Perrow: Normal Accidents Theory
  - Chemical plants, air traffic control, nuclear power plants, ..
  - Tight couplings
  - Interactive complexity
- Risk management/mitigation = reducing complexity

- Internet resilience

Complexity Science

- Origin: Natural sciences, economic (history)
- Autonomous systems

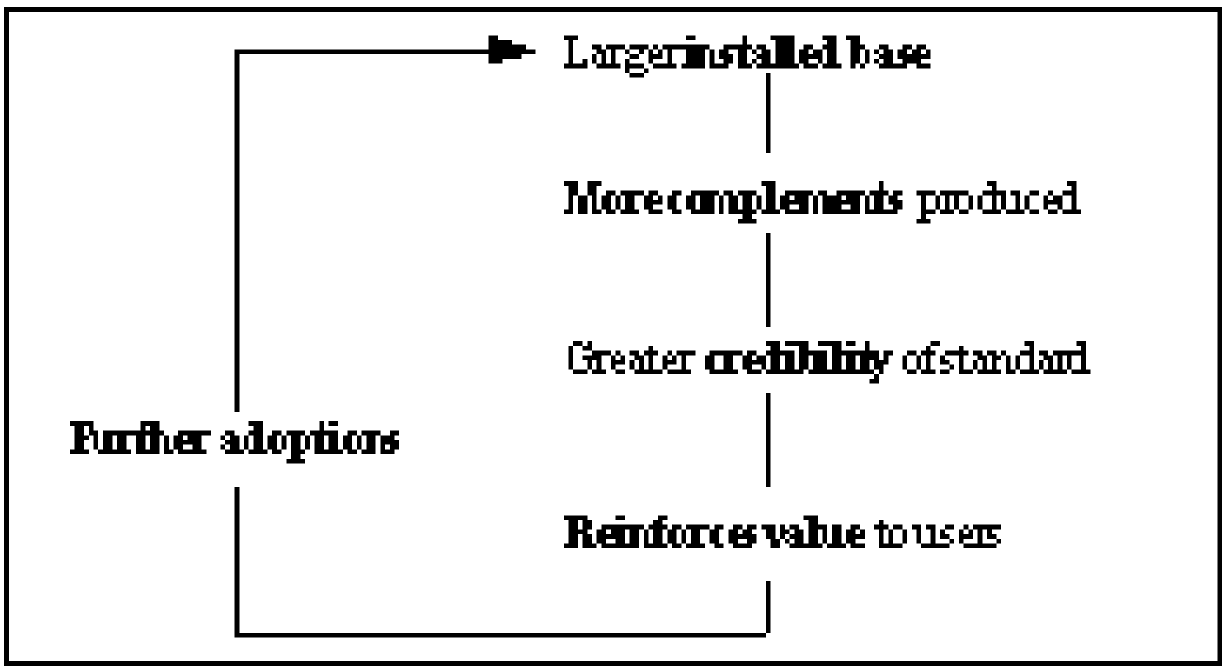
- Emergent order (not designed)
- Non-linearity (ex: History of Microsoft)
- Network externalities
- Increasing returns/Attractor
- Path dependency
    1. Diffusion of standards, competition
    2. Change of standards: Backward compatibility
    3. Chain of events
- Lock-ins

- The 2 laws of historical evolution

- II = Installed base as complex evolving system (=assemblage)

A self-reinforcing installed base

).



’Multidimensional’ critical mass

- Granovetter’s pedestrians: distribution of individual preferences.
- Diversity of users (motivation, knowledge, style, …)
- Heterogeneity of use areas and of technologies.
- Networks of networks

Design dilemmas

- Take-off
- Lock-in

’Bootstrapping’

- Enclocypedia: ’She bootstrapped herself to the top’ – to manage on one’s own
- Lifting yourselves by your hair
- Booting a computer
- Implementing a programming language
- Language learning
- Making a tool/network by means of the tool/network
- ”Deliver a better today, rather than promise a better tomorrow”.
- Late adopters adopt because the others have already
- First adopters must adopt for another reason

Identifying and arranging preferences

- Multi-dimensional
- Personal, individual
- Use areas and situations
- Technological aspects
- Coordination/governance structures
- Arranging preferences and dimensions (dynamically)

Bootstrapping Network Technologies & network of users

- Select motivated and knowledgeable users
- Simple, non-critical, non-complicated use areas where no large organisational changes are required.
- Select simple, relatively cheap and well supported technical solutions.

- Users first, then functionality/technology

Individual/personal preferences

- Motivation, attitudes towards technology
- Knowledge about technology

Aspects of use areas and situations

- Resources
- Benefits of communication within a small network
- Critical/non-critical activities
- Complexity of tasks and work practices
- Organizational changes needed

Aspects of technology

- “Distance” between users and designers/vendors
- complexity
- costs
- flexibility
- “allied with the future”

Design strategy

- Start with
  - simple, cheap, flexible solution
  - small network of users that may benefit significantly from improved com. with each other only
  - simple practices
  - non-critical practices
  - motivated users
  - knowledgeable users

Bootstrapping design principles

1. Design initially for usefulness
2. Draw upon existing installed base
3. Expand installed base by persuasive tactics

Boostrapping algorithm

1. Repeat as long as possible: enrol more users
2. Find and implement more innovative use, go to 1
3. Use solution in more critical cases, go to 1
4. Use solution in more complex cases, go to 1
5. Improve the solution so new tasks can be supported