Store og komplekse informasjonssystemer

Gruppetime uke 35

kribrae@ifi.uio.no

Agenda

- Repetisjon av begrepene fra rammeverket (miniforelesningene)
- Gjennomgang av Hanseth og Lyytinen (2010)
- Gjennomgang av Sommerville et. al (2012)

Repetisjon av begreper

Informasjonsinfrastruktur (II)

- Kompleksitet
- Delt
- Åpen
- Heterogen
- Evolverende
- Installert base

Design theory for dynamic complexity in information infrastructures: the case of building internet

Hanseth og Lyytinen

Design theory for dynamic complexity in information infrastructures: the case of building internet

A design theory that tackles dynamic complexity in the design for Information Infrastructures (IIs)

Tension between two design problems: (1) the bootstrap problem and (2) the adaptability problem

(Hanseth & Lyytinen, 2010)

Definisjon på kompleksitet

Definisjon på kompleksitet:

“Complexity can be defined here as the dramatic increase in the number and heterogeneity of included components, relations, and their dynamic and unexpected interactions in IT solutions”

(Hanseth & Lyytinen, 2010)

The bootstrap problem

The bootstrap problem: IT-løsninger i en II får sin verdi av å ha mange brukere, noe som krever en rask vekst i antall brukere. Men hvordan få de første personene til å bruke løsningen din?

- Må raskt komme opp med løsninger som dekker behovene til de første brukerne, samtidig som man må tenke på helheten = problem

(Hanseth & Lyytinen, 2010)

The adaptability problem

The adaptability problem: Når Ilen utvides som et resultat av stadig flere brukere går den inn i en periode med rask vekst. Designet må være tilpasset mange ulike behov, både teknisk og sosialt. Dette krever stor fleksibilitet i Ilen.

Problem å balansere bootstrapping og adaptability

(Hanseth & Lyytinen, 2010)

Classes of IT solutions

IT capabilities (e.g Text editor)

Applications (e.g Web browser)

Platforms (e.g MS office, Unix)

IIs (e.g Internet)

Økende kompleksitet

(Hanseth & Lyytinen, 2010)

Defining an II

“A *shared, open* (and unbounded), *heterogeneous* and *evolving* socio-technical system (which we call *installed base*) consisting of a set of IT capabilities and their user, operations and design communities”

(Hanseth & Lyytinen, 2010)

Defining an II

Shared across multiple communities in a myriad and unexpected ways

Open:

- New components can be added and integrated with them in unexpected ways and contexts
- No clear boundaries between those that can use the II and those that cannot

IIs become increasingly *heterogeneous* as the number of different kinds of technological components are included, but first of all because IIs include (an increasing number of) components of very different nature: user communities, operators, standardization and governance bodies, design communities, etc.

(Hanseth & Lyytinen, 2010)

Defining an II

Evolving: Because IIs are open, they evolve. IIs are never built in a green field, nor do they die.

- Vokser inkrementelt

The evolution is both enabled and constrained by the *installed base,* that is the existing configuration of II components.

- Whatever is added needs to be integrated and made compatible with this base

(Hanseth & Lyytinen, 2010)

Design theory

Bootstrap problem

Design goal: Generate attractors that bootstrap the installed base

Design principles:

1. Design initially for usefulness
2. Build upon existing installed bases
3. Expand installed base by persuasive tactics to gain momentum

(Hanseth & Lyytinen, 2010)

Design theory

Adaptability problem

Design goal: Make the system maximally adaptive and variety generating as to avoid technology traps

Design principles:

1. Make the IT capability as simple as possible
2. Modularize the II

(Hanseth & Lyytinen, 2010)

Noen viktige punkter

- Definisjon av II
- Bootstrap problem
- Adaptability problem

(Hanseth & Lyytinen, 2010)

Large-Scale Complex IT Systems

Sommerville et al.

Large-Scale Complex IT Systems

The reductionism behind today's software engineering methods breaks down in the face of systems complexity

Coalitions of systems

Systems engineering focuses on developing systems as a whole - vanskelig i "systemer av systemer"

(Sommerville et al., 2012)

System complexity

“Complexity stems from the number and type of relationships between the systems’s components and between the system and its environment”

“When the elements in a system involve many dynamic relationships, complexity is inevitable”

(Sommerville et al., 2012)

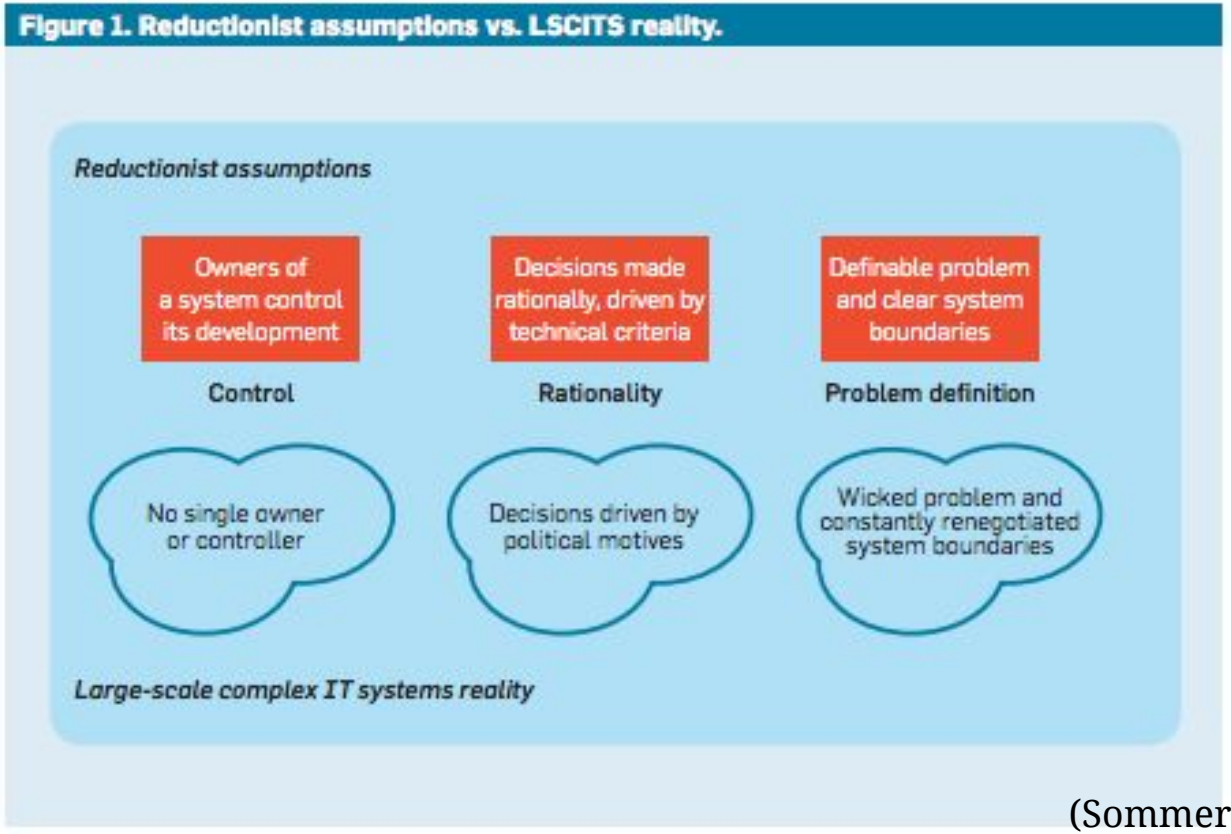
Sosio-Technical Systems

“To reflect the fact they involve evolving and interacting communities that include technical, human, and organizational elements, they are sometimes also called “socio-technical ecosystems” though the term socio-technical systems is more common.”

“Socio-technical systems include people and processes, as well as technological systems”

(Sommerville et al., 2012)

Reductionist assumptions VS reality



(Sommerville et al., 2012)

Key insights

- Coalitions of systems, in which the system elements are managed and owned independently, pose challenging new problems for systems engineering.

- When the fundamental basis of engineering—reductionism—breaks down, incremental improvements to current engineering techniques are unable to address the challenges of developing, integrating, and deploying large-scale complex IT systems.

- Developing complex systems requires a socio-technical perspective involving human, organizational, social, and political factors, as well as technical factors.

(Sommerville et al., 2012)

Plan for neste uke

Forelesning:

IKT, arbeid og organisasjon

Gruppetime (pensum):

- Tverrfaglig teknologiforskning (Aanestad og Olaussen)
- The Integration of Computing and Routine Work (Gasser)