The Process of Innovation in Computing: A Personal 30-Year Perspective

Walt Scacchi
Institute for Software Research
UCIrvine
24 April 2008

http://www.ics.uci.edu/~wscacchi/Presentations/InnovationInComputing.pdf

Overview

- Preliminaries and definitions
- Early computing innovation studies (<1980)
- Developing computing innovation studies (1980-1990)
- Transforming organizations through computing innovation studies (1990-2000)
- Multi-modal computing innovation studies (>2000)

Preliminaries and definitions

- *Process of innovation in computing*
  - A trajectory of situated activities people perform to displace the status quo through a web of socio-technical resources, work situations, and interaction networks
- *Computing innovation*
  - Adds to or redistributes access to the available supply of computing resources, work situations, and interaction networks
- *Unit of analysis*
  - Life history of a computing innovation in a complex organizational setting

Preliminaries and definitions

- *Innovation life cycle stages*
  - Discovery or invention
  - Development
  - Diffusion
  - Adoption
  - Implementation
  - Routinization
  - Evolution (debugging, accretion, tuning, restructuring, merger, migration, retirement, reinvention)

Organizational field sites

- Andersen Consulting LLP
- AT&T and AT&T Bell Laboratories
- Bayfront Technologies, Inc.
- BellCore
- Computer Science Dept., USC
- Daegu Global R&D Collaboration Center
- Defense Acquisition University
- Discovery Science Center
- Eastman Kodak
- EDS
- Enabling Business Technologies LLC
- EON Reality
- Ernst and Young LLP
- Fujitsu
- GDE Systems Inc.
- Hewlett-Packard
- Holosofx Inc. (Active Management Inc.)
- Hughes Aircraft
- IBM
- Intel
- Intelligent Systems Technology Inc.
- Knowledge System Laboratory, Stanford University
- MCC
- McKesson
- Naval Air Warfare Center
- Naval Ocean Systems Center
- Naval Postgraduate School
- New Directions Technologies Inc.
- Northrop-Grumman
- Office of Naval Research
- Pacific Bell
- Pacific Life
- Perceptronics
- Physics Dept., UCI
- Software Engineering Institute, Carnegie-Mellon University
- SUN Microsystems
- TRW
- USC Advanced Biotechnology Center
- USC Entertainment Technology Center
- USC Information Sciences Institute
- *Virtual organizations* (about 20 Web-based projects developing free/open source software)
- WICOR (Wisconsin Gas)
- *Withheld* (another dozen in litigation support)

Early computing innovation studies (<1980)

Consumption of computing innovations

Case study: DoD production of a software innovation (Ada programming language)

- 1973, U.S. Department of Defense reports spending >$3B annually on software development, deployment, and support
  - Upwards of 90% of software costs go to "maintenance"
- 1975-1979 DoD invests $$$ in the invention and development of new programming language for embedded military applications
  - Language called Green, DoD-1, then Ada
- Was this a good idea/approach?

R. Kling and W. Scacchi, The DoD Common High Order Programming Language Effort (DoD-1): What Will the Impacts Be?, *SIGPLAN Notices*, 14(2), 29-41, 1979.

Predicting the future consumption of Ada as a software innovation

- Problematics:
  - Why does software cost so much to produce and consume?
  - How do DoD contractor business practices drive the cost of software?
- Observations:
  - Contractor financial operations treat software development as a liability, and software maintenance as a revenue
  - Financial conditions distort software production costs
  - Be careful for what you hope to realize through computing innovations

Other consequences that followed

- Kling and Scacchi subjected to criticism and career threats by DoD software researchers
- Senior military officials endorse KS79 observations
- Scacchi engaged by DoD Ada Office (1982) to design software production program focus for future Software Engineering Institute
  - Software production process
  - Software technology transfer process (innovation life cycle)
  - SEI-CMU launches in 1985 with these programs
  - MCC also adopts STT process program focus in 1986
- Be careful for what advice you offer for computing innovation processes and practices.

Process of Innovation in Computing (1977-1980)

Innovation / Organization	Hardware	Software	Organizational	Innovation Processes
UCI Physics	Computer graphics terminals	Operating system upgrade	CPU usage charge-back mechanism	Hassling and distracting
Pacific Life	Distributed computing systems (mini-computers)	Database management facilities	User-specialist liaisons	Structuring
Knowledge Systems Lab	Processor memory and disk storage upgrades	Text processing system	Computer-based work environment	Mirroring and pacing
Innovation Processes	Expanding and demanding	Tinkering and layering	Regulating and stabilizing	Fitting, packaging, and cycling

W. Scacchi, *The Process of Innovation in Computing: A Study in the Social Dynamics of Computing*, Unpublished Dissertation, Information and Computer Science Dept., UC Irvine, 1981.

Additional findings

- History of commitments constrains choice
- Narrow incentives and opportunities motivate choice
- Macrostructural patterns influence local computing
- Distribution of computing resources and system configurations is tied to the career contingencies of local participants
- Computing innovations enable new use, and use create demand for computing innovations
- Innovation in computing is a continuous process.

R. Kling and W. Scacchi, The Web of Computing: Computer Technology as Social Organization, in M. Yovits (Ed.), *Advances in Computers*, 21, 2-90, 1982.

Developing computing innovation studies (1980-1990)

Production of computing innovations

Case study: Understanding innovation development teamwork

- Comparative analysis of teamwork practices when formally specifying a software innovation to develop in complex setting.
  - USC System Factory Project (1981-1992)
  - Five graduate student teams, 5-7 members, two-week (part-time) process that incorporates planning, formal notation, automated tool use, reusable examples, documentation tasks, and team shared responsibility.

What to Understand

- Work *Structures* and *Shifts*: Resource arrangements, historical circumstances, division of labor and expertise, etc.
- Work *Processes*: Routine, habitual or emergent patterns of how work flows among people through/ onto work structures
- Work *Practices*: Behavioral discourse and social dynamics enacted through work processes

What to Understand

- Work structures are *domain independent constructs*
  - Prescriptive/descriptive abstractions
- Processes are *classes* of workflow
  - Descriptive and derived
  - Prescriptive and composed
- Practices are *instances*
  - Descriptive, historic and situated

Comparative analysis of software innovation production teamwork

- Six work structure types observed: **N**egotiated, **I**ntegrated, **R**eplicated, **D**elegated, **P**rediscriminated and **S**eparated
- Three types of structural shift observed:
  - anticipated **->**,
  - unanticipated **-->>**,
  - role shift within work structure +.

Work structures and shifts (data)

Team ID	T1	T2	T3	T4	T5
Team Size	6	7	7	7	5
Reusable Exemplar	no	yes	yes	yes	yes
PROCESS					
A. Pre-planning task	N->R->I	N->R->I	N->R->I	N->R->I	N->R->I
B. Planning task	N	N	N	N	N
c.	I	I	I	I -> S	I -> S
d.	I	I+	I	S	S
e.	P (D,I,I)	P (D,I,I)	P (D,I,I)	P (D,S->I,I)	P (D,D,I)
f.	D	D	D	D	D
g.	R	R	R	D	D
h.	D	I	I	D	D
C. Develop preliminary	I -->>	I+	I+	I+ -> S+	I -> S+
(informal) specification	N -->>				
	R -->> I				
D. Develop formal	I+	I+	I+	S+ -->>	S+
(processable) spec.				N -->>	
E. Document	N -->> D	P+	P+	N -->>	P(D,D,I)
write-up				P(D,S->I,D)	
F. Documentation	D -->>	D+	D+	D+	D+
integration	N -->> I				
G. Document review	R	R	R	N -->> R	D
H. Prepare for	D -->>	I+	N -->> I+	N -->> I+	D+
Delivery	N -->> I				

Findings

- Highest (lowest) *quality* product (measured by automated tools): T1 (T5)
- Highest (lowest) *productivity* (self reported time expended): T5 (T1)
- Note the *coincidental* relationship
- Effectiveness of planning, automated tool use, asset reuse *not* clearly associated with high(low) quality or high(low) productivity

Findings

- Teams enacting primarily into Negotiative and Integrative structures had higher quality
- Teams enacting primarily into Delegative, Pre-discriminative or Separative structures had higher productivity
- Computer supported work environments must account for teamwork structures as a usage parameter to be most effective.

Transforming organizations through computing innovation studies (1990-2000)

Going meta

Case study: Designing large-scale production operations

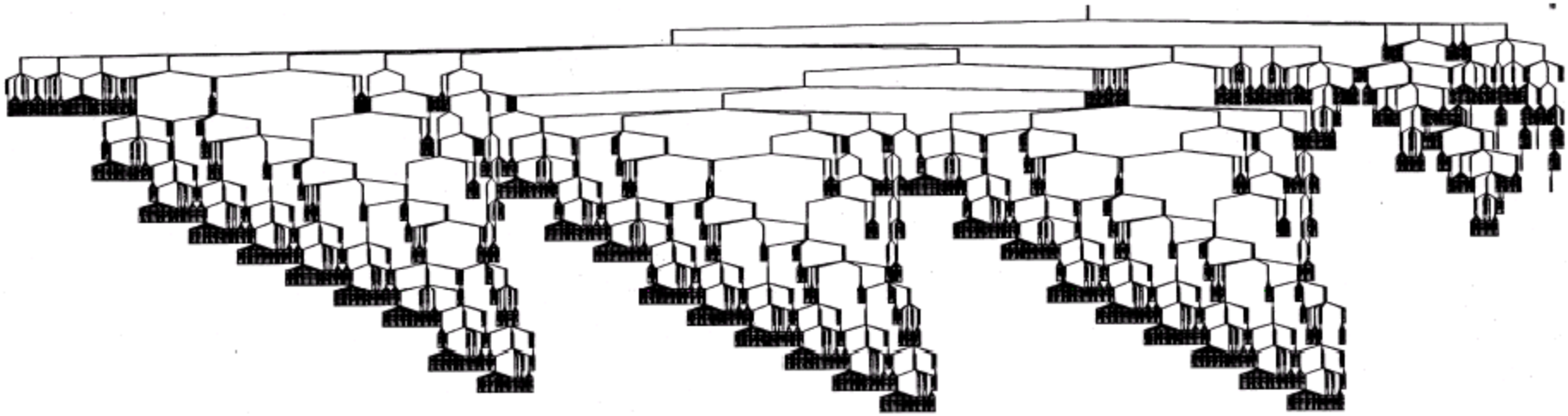
- Major TelCo (AT&T) wants to develop broadband multi-media telecommunications system
- Anticipates $1B development, up to1500 system developers working 2-3 years
- Seeks industrial partners to provide supporting infrastructure to reduce risk
- IT partner (HP) wants to showcase new “process support technology” products as sales lead
- IT partner brings in my academic research team (USC Atrium Lab) to analyze and advise TelCo on “production process design issues”

The story continues

- Team, IT partner, and TelCo jointly elicit, capture, codify (*formalize*) and inter-relate TO-BE system development process.
  - (Re)Design organizational computing innovation
- Team employs IT partner's products to present results of their "process analysis"
- Team view of their effort -- a major computing innovation success for publication (and re-publication)

P.K. Garg, P. Mi, T. Pham, W. Scacchi, and G. Thunquest, The SMART Approach to Software Process Engineering, *Proc. 16th. Intern. Conf. Software Engineering* , IEEE Computer Society, Sorrento, Italy, pp. 341-350, (May 1994). Reprinted in *Process-Centered Software Engineering Environments* , P.K. Garg and M. Jazayeri (eds.), IEEE Computer Society, pp. 131-140, (1996)

A complex organizational process: a *decomposition-precedence* relationship view (19 levels of decomposition, 400+ tasks)



W. Scacchi, Experience with Software Process Simulation and Modeling, *J. Systems and Software*, 46(2/3):183-192,1999.

The story ends

- Team suggests overall process won't succeed -- too complex, too much delegation, problematic hand-offs ("throwing it over the wall")
- TelCo and IT partner dismisses academic team
- Less than one year later, IT vendor abandons innovative process technology product
- Two years later, business press reports TelCo experiences major project failure and losses greater than $200M, and no operational system.

Case Study: Transforming procurement processes at ONR

- Present an approach for how to optimize business processes including software production
- Identify key concepts, techniques, and tools that enable better optimization
- Describe optimization transformations from business process redesign studies
- Describe opportunity areas for exploitation and use

Definitions and Differences

- *Software production*: enterprise processes and resources that produce software
- *Production strategies*: business strategies guiding overall approach to building software systems
- *Production architecture*: configuration of enterprise capabilities to enact strategies
- *Optimizing production*: minimizing enterprise configuration to maximize strategic outcomes

(Re)designing production processes

- Which process first: **to-be** goal vs. **as-is** mess?
  - *If you don't know where you are, any road will do* (proverb)
  - Observation: people at work cannot describe the processes they do with high fidelity (tacit knowledge lemma)
  - Redesign necessitates understanding as-is, to-be, and *here-to-there processes*
- Creating high-performance work groups
  - Empowerment, participation, incentivization (resource sovereignty), and recognition (patronage, status, accomplishment)

W. Scacchi, Redesigning Contracted Service Procurement for Internet-based Electronic Commerce: A Case Study, *J. Information Technology and Management,* 2(3), 313-334, 2001.

Generic Strategies

- Reduce costs
- Reduce cycle time
- Improve cash flow
- Increase customer satisfaction
- Increase sales
- Improve customer service
- Increase productivity
- Open new markets
- Open new “channels”
- Become innovation leader
- Increase market share
- Enable just-in-time service delivery

Enterprise Production Architecture

- A *composite model* that interrelates
  - software system architecture
  - software production architecture
  - development organization architecture
  - information network infrastructure and development tools/environment configuration
  - documentation architecture
  - customer-support knowledge base architecture

P. Mi and W. Scacchi, A Meta-Model for Formulating Knowledge-Based Models of Software Development, *Decision Support Systems*, 17(4):313-330, 1996.

Optimizing Production

- Strategies provide global constraints or opportunities for optimizing production
- Constraints and opportunities realized in production setting
- Constraints and opportunities are distributed across the production architecture

Optimizing Production

- Optimization must address composite production architecture
- Local optimization of any component architecture does not guarantee global optimality of software production
- *Diagnostic analyses* and *transformation heuristics* applied to composite architectural models lead to optimization opportunities

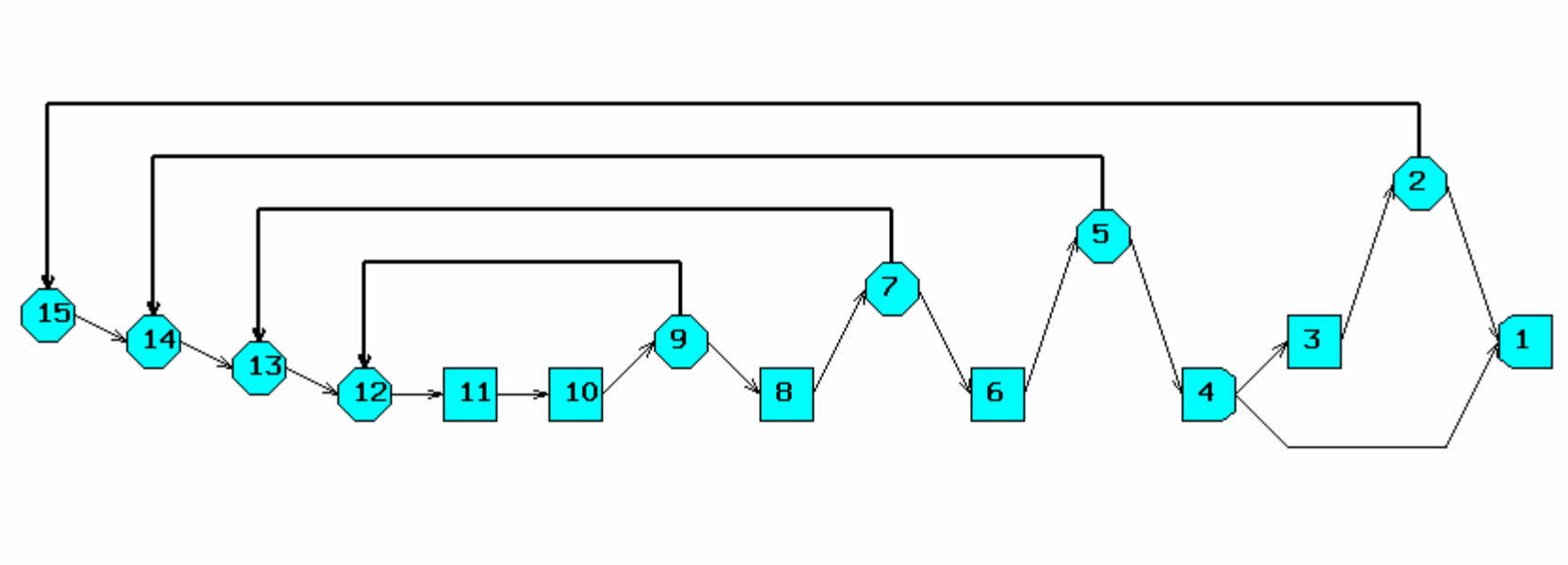
Optimizing Production

- Transformation heuristics classified taxonomically
- Taxonomy classifies *domain-independent* (DI) and *domain-specific* (DS) heuristics
- DI transformations applied in *any* software production setting
- DS transformations applied to specific component architectures

Optimizing Production

- DI transformation classes (sample):
  - Job scope
  - Worker empowerment
  - Organization design
  - Workflow streamlining
  - Information technology (IT)

Research grant justification and approval process at Office of Naval Research (c. 1995)



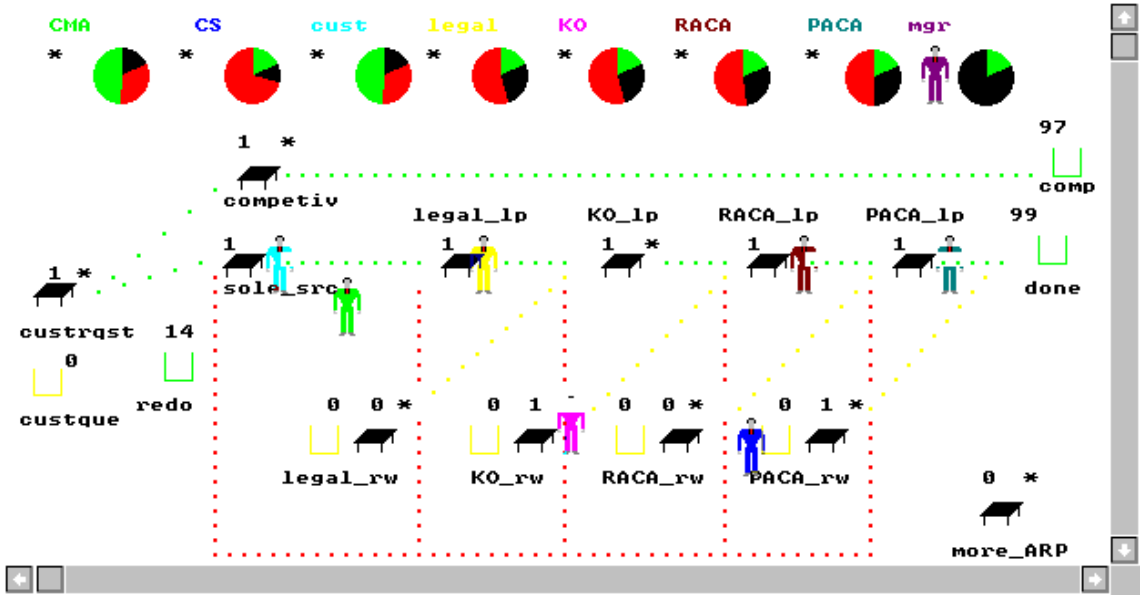
W. Scacchi and J. Noll, Process-Driven Intranets: Life Cycle Support for Process Reengineering , *IEEE Internet Computing*, 1(5):42-49, 1997.

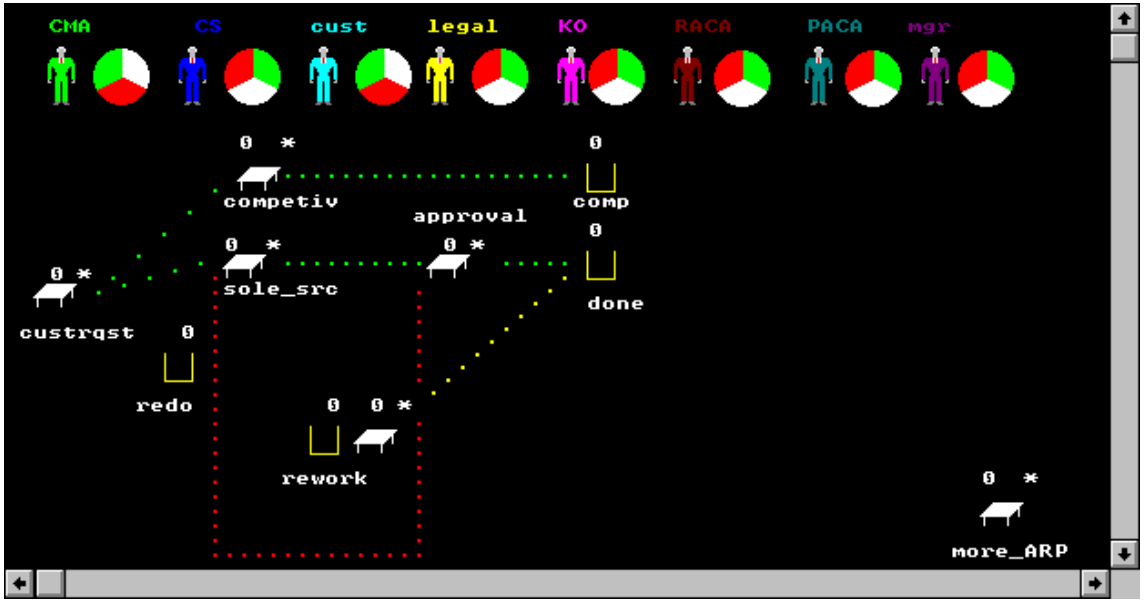
Optimizing Production

- IT transformation sub-classes (sample):
  - Extend IT-based *support* to manual process steps
  - Extend IT-based *communication* facilities to encourage information sharing activities
  - Extend IT-based *automation* to incorporate new kinds of application packages
  - Extend IT-based *integration* to interconnect and interrelate existing "islands of automation"

Example DI process redesign hueristics applied at ONR

As-is Diagnosis	*Applicable Hueristics*	*Expected ROI*
Many manual steps	Consolidate and automate (workflow and IT-support)	Med-High
Many linear step sequences	Identify parallelization opportunities (workflow)	High
Many reviews steps	Joint collaborative reviews (job scope, organization design)	High
Many data validation steps	Rule-based review system (IT-automation)	Med-High
Many data validation steps	Push validation responsiblities upstream (job scope)	Med-High
Manual assembly of compound documents	Rule-based document builder (IT-automation)	Low-Med
Duplicating and circulating documents	Automate distribution and archiving (IT-communication)	Med-Very High
Replace paper documents	Employ electronic proposals and grant documents (IT-communication)	High-Very High
Islands of automation	Intranet with process support, data integration, and product navigation (IT-integration)	Low-High
Wide-area workflow	Internet-based process enactment (IT-support, IT-communication, IT-integration, IT-automation)	Med-High


CMA
CS
cust
legal
KO
RACA
PACA
mgr
competiv
97
comp
legal_lp
KO_lp
RACA_lp
PACA_lp
99
done
sole_src
custrqst
14
redo
custque
legal_rw
KO_rw
RACA_rw
PACA_rw
more_ARP


CMA
CS
cust
legal
KO
RACA
PACA
mgr
competiv
comp
approval
sole_src
done
custrqst
redo
rework
more_ARP

Redesign/Optimization (Innovation) Results

- Reduction in *procurement process* cycle times of 20X, annual operational savings of $10M-15M.
  - Via transformation and realignment of information systems, business processes, corporate strategy, and work practices in a 1-2 year time frame.
  - Participatory design, development and refinement of computational models of new work processes, resource configurations and work practices, together as an *organizational system*.

W. Scacchi, Redesigning Contracted Service Procurement for Internet-Based Electronic Commerce: A Case Study, *Information Technology and Management*, 2(3):313-334, 2001.

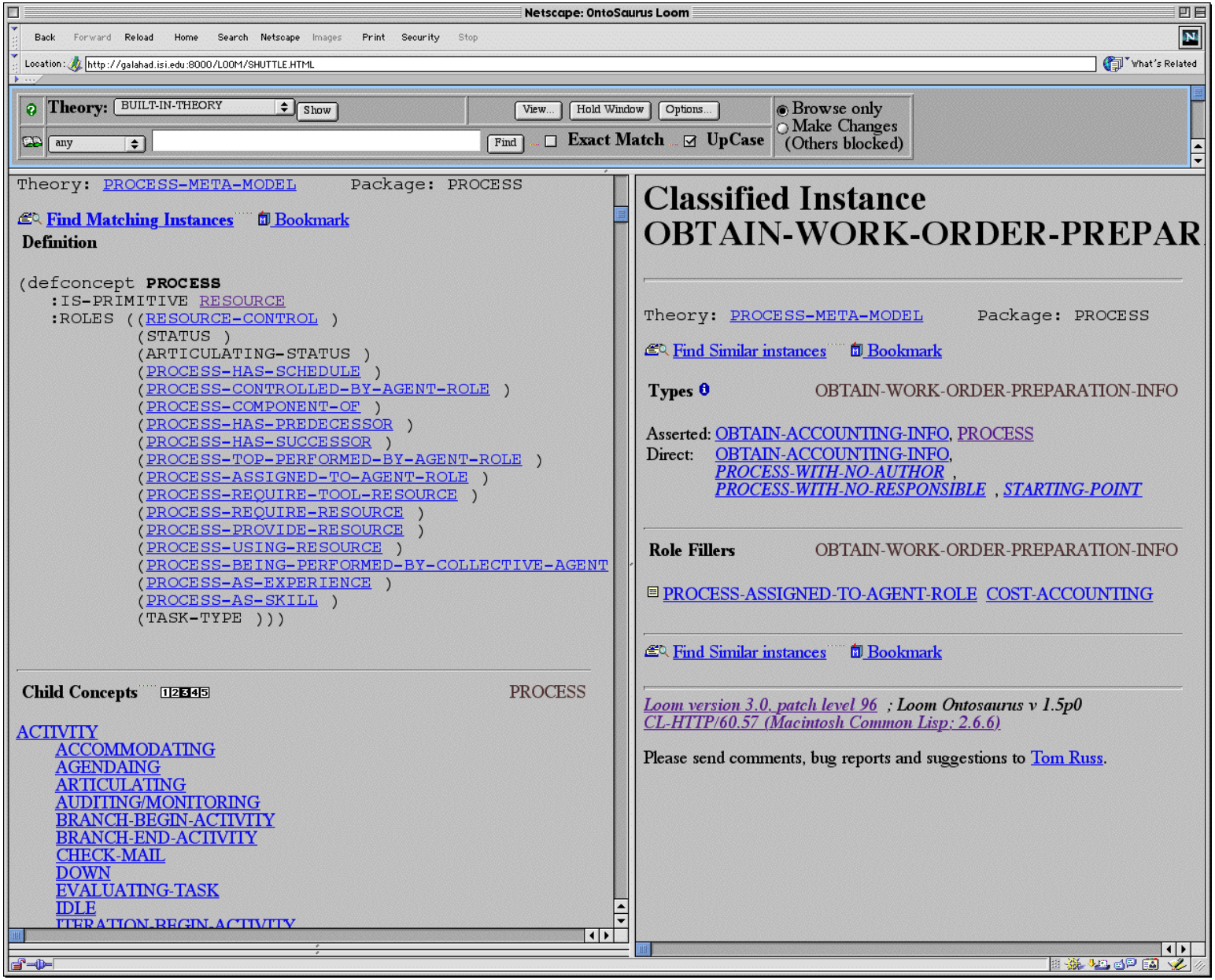
Tools and Techniques

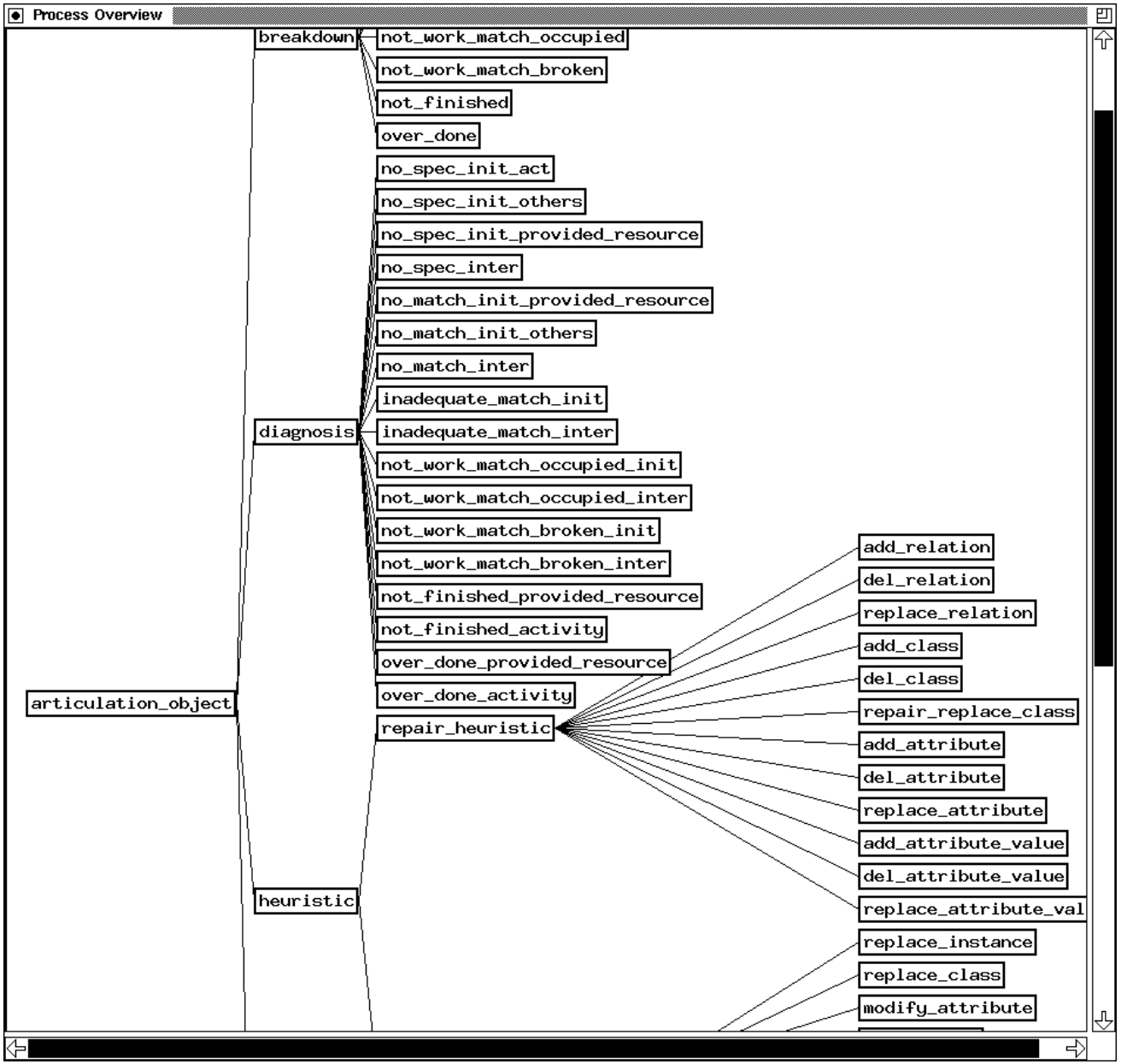
- Software process redesign case web
- Knowledge web for software production
- Process-driven intranets
- Organizational transformation

Computing innovations enabling other computing innovations

- Process redesign (organizational innovation) case web
- Knowledge web for producing and consuming computing innovations:
  - Software production ontology
  - Taxonomy for as-is diagnosis, redesign heuristics
  - Best practices and lessons learned cross-linked
- Engage in organizational transformation projects

W. Scacchi and A. Valente, Developing a Knowledge Web for Business Process Redesign, Presented at the *1999 Knowledge Acquisition Workshop*, Banff, Canada, October 1999.


Netscape: OntoSaurus Loom
Back Forward Reload Home Search Netscape Images Print Security Stop
Location: http://galahad.isi.edu:8000/LOOM/SHUTTLE.HTML
What's Related
Theory: BUILT-IN-THEORY Show
View... Hold Window Options...
Browse only
Make Changes (Others blocked)
any Find Exact Match UpCase
Theory: PROCESS-META-MODEL Package: PROCESS
Find Matching Instances Bookmark
Definition
(defconcept PROCESS
:IS-PRIMITIVE RESOURCE
:ROLES ((RESOURCE-CONTROL)
(STATUS)
(ARTICULATING-STATUS)
(PROCESS-HAS-SCHEDULE)
(PROCESS-CONTROLLED-BY-AGENT-ROLE)
(PROCESS-COMPONENT-OF)
(PROCESS-HAS-PREDECESSOR)
(PROCESS-HAS-SUCCESSOR)
(PROCESS-TOP-PERFORMED-BY-AGENT-ROLE)
(PROCESS-ASSIGNED-TO-AGENT-ROLE)
(PROCESS-REQUIRE-TOOL-RESOURCE)
(PROCESS-REQUIRE-RESOURCE)
(PROCESS-PROVIDE-RESOURCE)
(PROCESS-USING-RESOURCE)
(PROCESS-BEING-PERFORMED-BY-COLLECTIVE-AGENT
(PROCESS-AS-EXPERIENCE)
(PROCESS-AS-SKILL)
(TASK-TYPE)))
Child Concepts 1 2 3 4 5 PROCESS
ACTIVITY
ACCOMMODATING
AGENDAING
ARTICULATING
AUDITING/MONITORING
BRANCH-BEGIN-ACTIVITY
BRANCH-END-ACTIVITY
CHECK-MAIL
DOWN
EVALUATING-TASK
IDLE
ITERATION-BEGIN-ACTIVITY
Classified Instance
OBTAIN-WORK-ORDER-PREPAR
Theory: PROCESS-META-MODEL Package: PROCESS
Find Similar instances Bookmark
Types OBTAIN-WORK-ORDER-PREPARATION-INFO
Asserted: OBTAIN-ACCOUNTING-INFO, PROCESS
Direct: OBTAIN-ACCOUNTING-INFO,
PROCESS-WITH-NO-AUTHOR ,
PROCESS-WITH-NO-RESPONSIBLE , STARTING-POINT
Role Fillers OBTAIN-WORK-ORDER-PREPARATION-INFO
PROCESS-ASSIGNED-TO-AGENT-ROLE COST-ACCOUNTING
Find Similar instances Bookmark
Loom version 3.0, patch level 96 ; Loom Ontosaurus v 1.5p0
CL-HTTP/60.57 (Macintosh Common Lisp; 2.6.6)
Please send comments, bug reports and suggestions to Tom Russ.


Process Overview
articulation_object
breakdown
diagnosis
heuristic
not_work_match_occupied
not_work_match_broken
not_finished
over_done
no_spec_init_act
no_spec_init_others
no_spec_init_provided_resource
no_spec_inter
no_match_init_provided_resource
no_match_init_others
no_match_inter
inadequate_match_init
inadequate_match_inter
not_work_match_occupied_init
not_work_match_occupied_inter
not_work_match_broken_init
not_work_match_broken_inter
not_finished_provided_resource
not_finished_activity
over_done_provided_resource
over_done_activity
repair_heuristic
add_relation
del_relation
replace_relation
add_class
del_class
repair_replace_class
add_attribute
del_attribute
replace_attribute
add_attribute_value
del_attribute_value
replace_attribute_val
replace_instance
replace_class
modify_attribute

Multi-modal computing innovation studies (>2000)

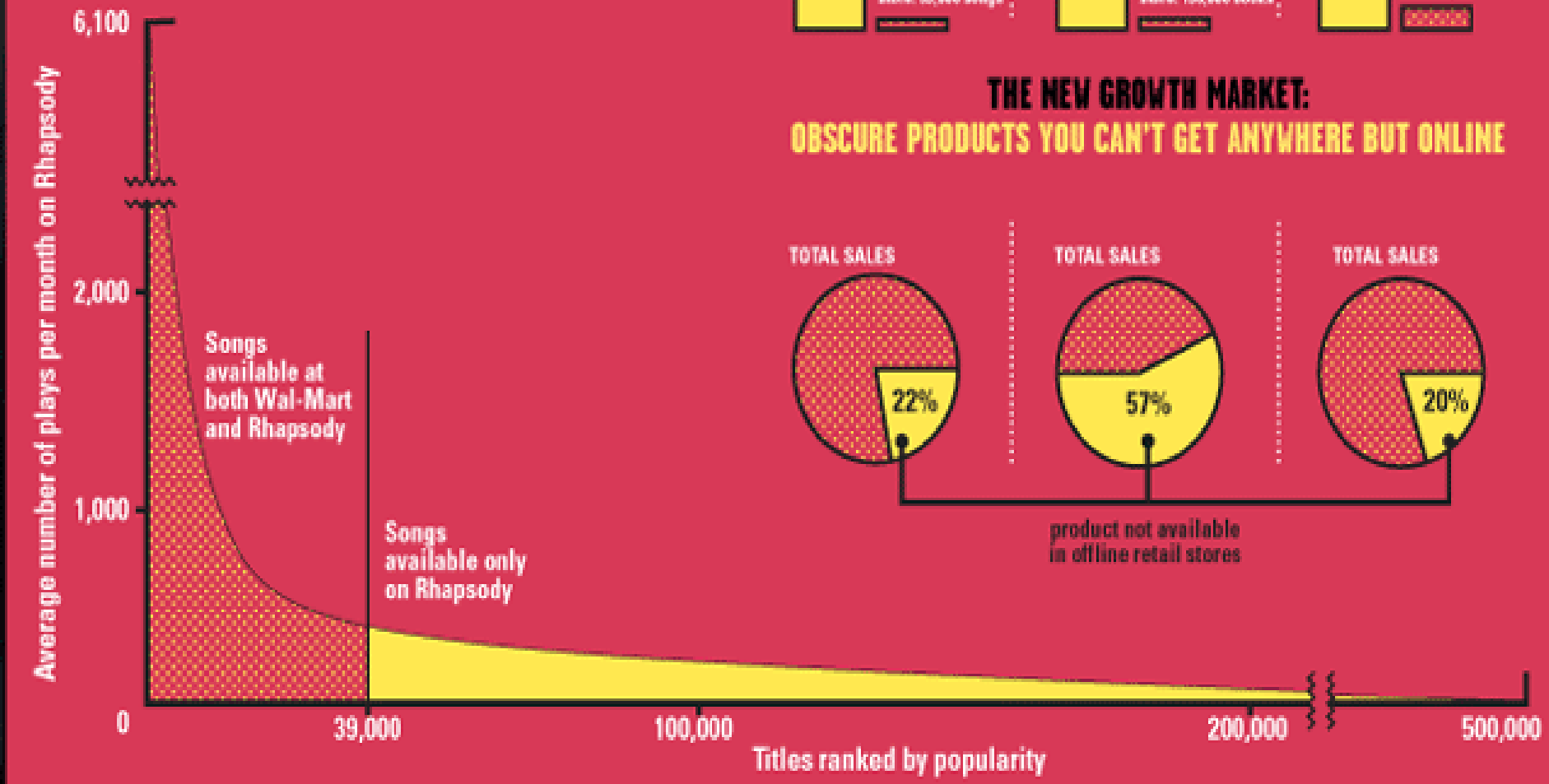
How to go exponential

Current field studies

- Understanding open source software practices and processes in different domains
  - Academic research <---> Commercial development
    - Deep space astronomy and bioinformatics
    - Internet infrastructure
    - Administrative/electronic business computing
    - Networked computer games
- To produce and compare case studies using
  - (informal) ethnographic narratives
  - (semi-structured) hypertext, and
  - (formal) computational models.

ANATOMY OF THE LONG TAIL

Online services carry far more inventory than traditional retailers. Rhapsody, for example, offers 19 times as many songs as Wal-Mart's stock of 39,000 tunes. The appetite for Rhapsody's more obscure tunes (charted below in yellow) makes up the so-called Long Tail. Meanwhile, even as consumers flock to mainstream books, music, and films (right), there is real demand for niche fare found only online.

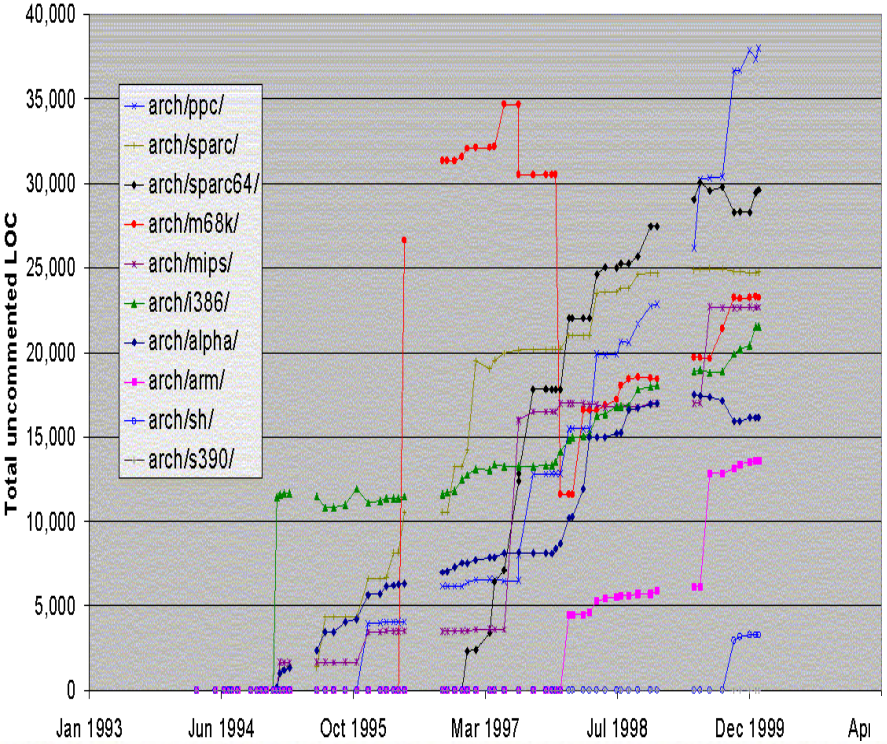


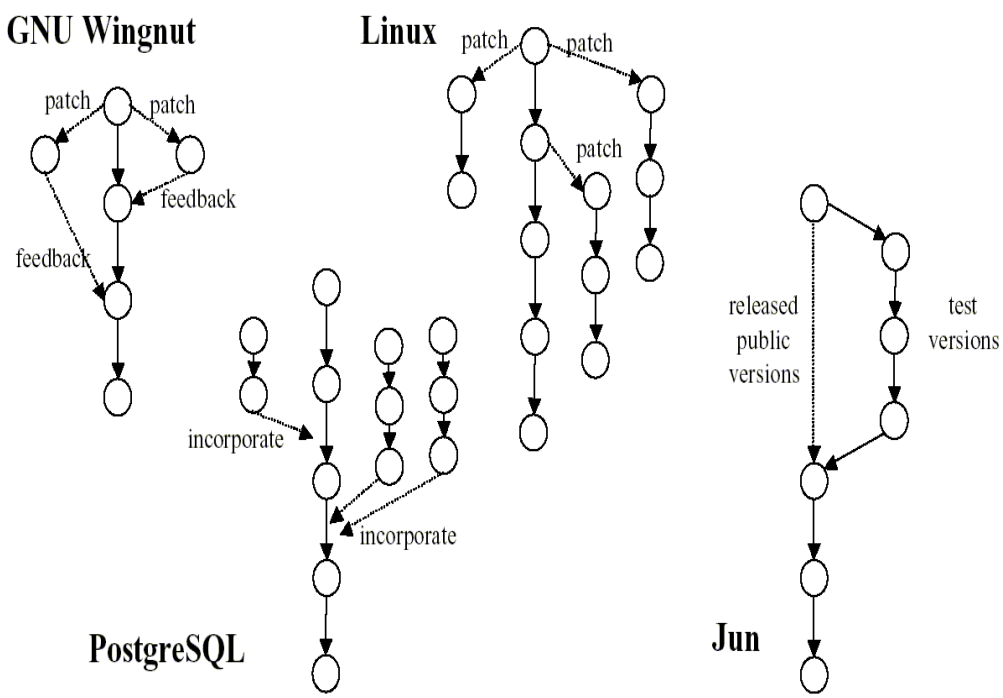
THE NEW GROWTH MARKET:
OBSCURE PRODUCTS YOU CAN'T GET ANYWHERE BUT ONLINE

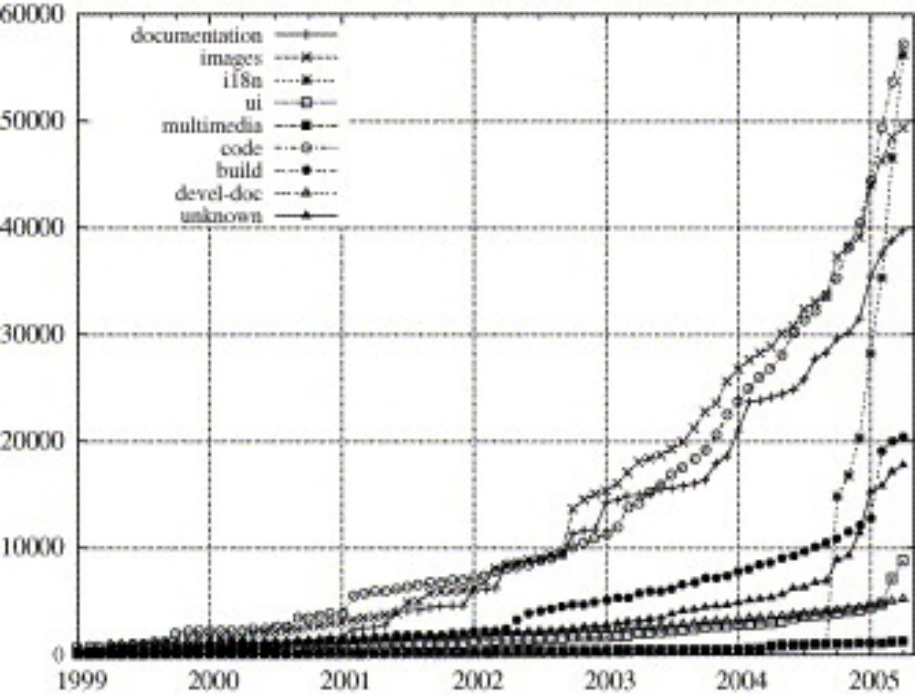
Sources: Erik Brynjolfsson and Jeffrey Hu, MIT, and Michael Smith, Carnegie Mellon; Barnes & Noble; Netflix; RealNetworks

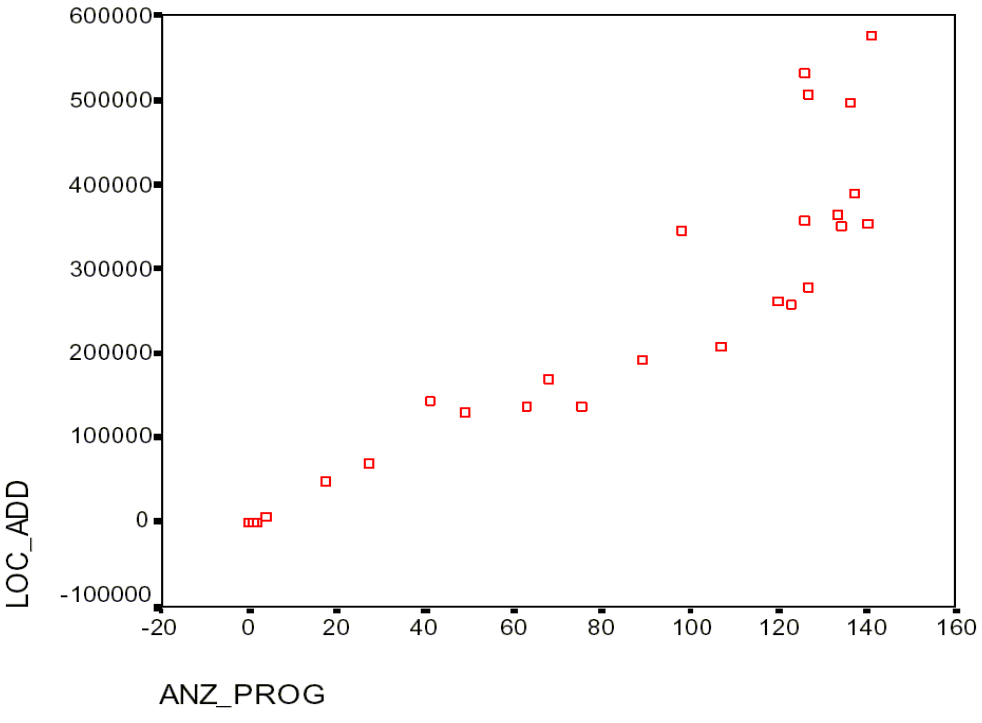
OSSD Projects as innovation engines

- Social/technical innovations in OSSD projects emerge via:
  - Enhancing project resources
    - Inter-project mergers
    - Creating new software development artifacts
  - Tuning adjustments or adaptations
    - Intra-team role migration
    - Incremental product releases (“daily builds”)
  - Restructuring transformations
    - Legal incorporation
    - Code refactoring
  - Reinvention practices
    - Learning from others
    - Commercial product feature replication


Total uncommented LOC
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
0
arch/ppc/
arch/sparc/
arch/sparc64/
arch/m68k/
arch/mips/
arch/i386/
arch/alpha/
arch/arm/
arch/sh/
arch/s390/
Jan 1993
Jun 1994
Oct 1995
Mar 1997
Jul 1998
Dec 1999


GNU Wingnut
patch
patch
feedback
feedback
Linux
patch
patch
patch
PostgreSQL
incorporate
incorporate
Jun
released
public
versions
test
versions


60000
50000
40000
30000
20000
10000
0
documentation
images
i18n
ui
multimedia
code
build
devel-doc
unknown
1999
2000
2001
2002
2003
2004
2005

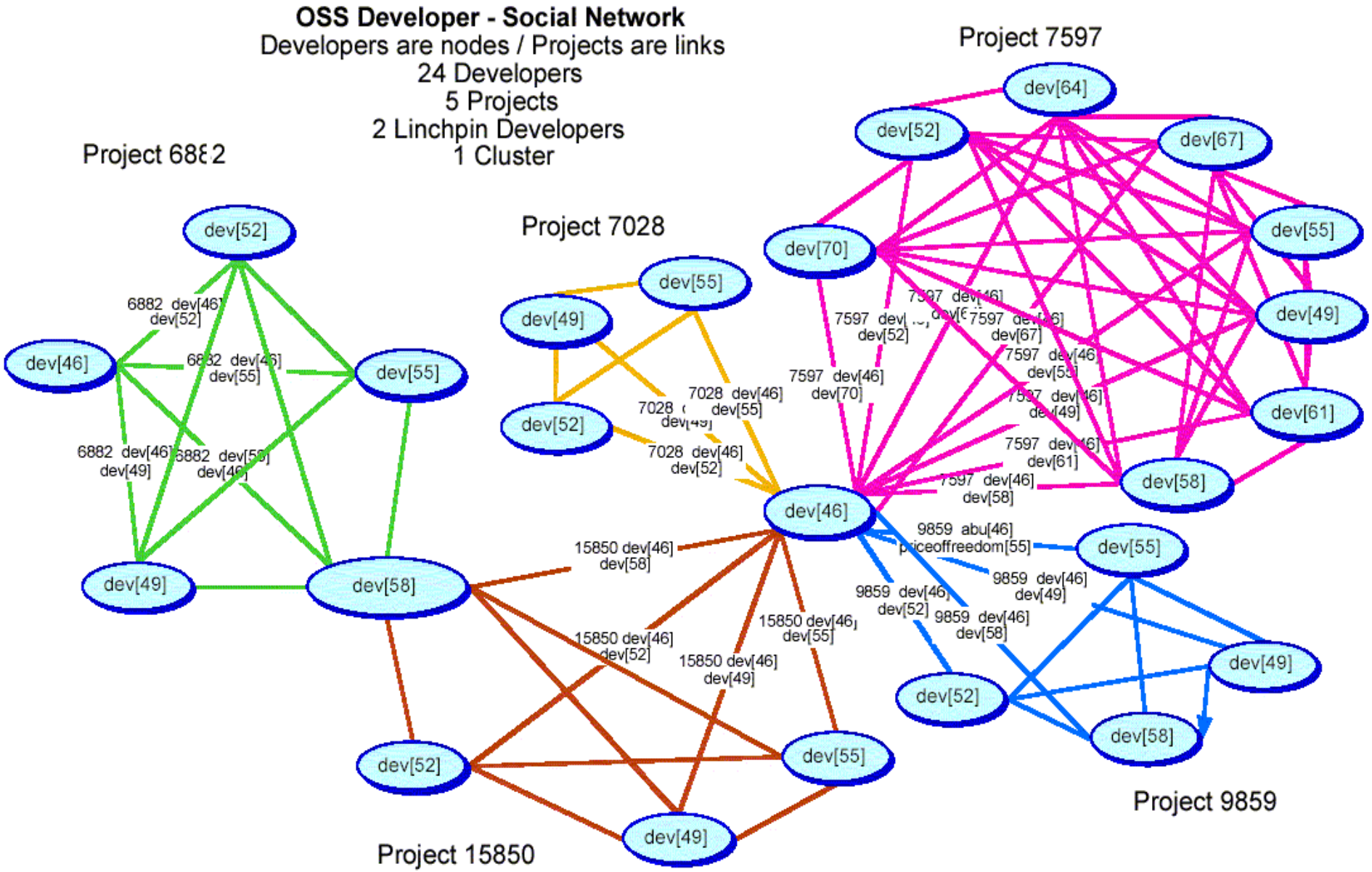

600000
500000
400000
300000
200000
100000
0
-100000
LOC_ADD
-20
0
20
40
60
80
100
120
140
160
ANZ_PROG

OSSD multi-project ecology as an *innovation frontier*

- *OSSD multi-project ecosystem*: a (virtual) enterprise that collectively mobilizes an inter-related group of OSSD projects
  - Barclays Global Investments, Google Summer of Code, Apache Software Foundation, SUN Microsystems, etc.
- *Frontier*: a zone of unsettled land outside the region of existing settlements suitable suitable for exploration and potential development
- *Innovation frontier*: a socio-technical zone for innovation outside of existing enterprise system settlements suitable for exploration and potential development.

Enabling exponential growth for socio-technical innovation frontiers

- The most successful OSSD projects exhibit *sustained exponential growth* via social and technical innovations
- Exponential growth requires critical mass for collective innovation action
  - Critical mass emerges through sufficiently dense socio-technical networks that act as “small worlds”
  - Such networks emerge when participants *enjoy* making social/technical contributions that serve to advance the accumulation of common pool resources


OSS Developer - Social Network
Developers are nodes / Projects are links
24 Developers
5 Projects
2 Linchpin Developers
1 Cluster
Project 7597
Project 7028
Project 6882
Project 15850
Project 9859
dev[64]
dev[52]
dev[67]
dev[55]
dev[70]
dev[49]
dev[61]
dev[58]
dev[46]
dev[52]
dev[55]
dev[49]
dev[58]
dev[46]
dev[55]
dev[49]
dev[52]
dev[58]
dev[52]
dev[55]
dev[49]
7597 dev[46] dev[52]
7597 dev[46] dev[67]
7597 dev[46] dev[55]
7597 dev[46] dev[70]
7597 dev[46] dev[49]
7597 dev[46] dev[61]
7597 dev[46] dev[58]
7028 dev[46] dev[55]
7028 dev[46] dev[52]
6882 dev[46] dev[52]
6882 dev[46] dev[55]
6882 dev[46] dev[49]
15850 dev[46] dev[58]
15850 dev[46] dev[52]
15850 dev[46] dev[55]
15850 dev[46] dev[49]
9859 abu[46] priceoffreedom[55]
9859 dev[46] dev[49]
9859 dev[46] dev[52]
9859 dev[46] dev[58]

Closing remarks

- To be clear, nearly all enterprises and OSSD projects do not obtain exponential innovation growth.
- High, sustained growth OSSD projects do exhibit exponential innovation growth when proper conditions exist.
  - When innovation becomes participatory and self-serving, when innovations contribute to common pool resources, and when sustained collective action emerges as a social movement that transcends enterprise boundaries.
- Possible for enterprises to achieve exponential innovation growth.

Findings

- In addition to those from 1981,
- Innovation in computing is routine
  - Producing computing innovations is “easy”
  - Consuming computing innovation is demanding
- Continuous innovation (in the case of open source software) can enable exponential growth of socio-technical interaction networks
  - What will limit the exponential growth of such innovation?

Details

- W. Scacchi, Understanding Free/Open Source Software Evolution, in N.H. Madhavji, J.F. Ramil and D. Perry (eds.), *Software Evolution and Feedback: Theory and Practice*, 181-206, John Wiley and Sons Inc, New York, 2006.
- W. Scacchi, Emerging Patterns of Intersection and Segmentation when Computerization Movements Interact, to appear in K.L. Kraemer and M. Elliott (eds.), *Computerization Movements and Technology Diffusion: From Mainframes to Ubiquitous Computing*, Information Today, Inc., 2008.
- Funding support through research grants from the National Science Foundation (*no endorsement implied*) #0083075, #0205679, #0205724, #0350754, #0534771, and #0749353.