Ultra-Large-Scale Systems: *Scale Changes Everything*

Linda Northrop
Director, Product Line Systems Program
Software Engineering Institute

SMART Event
March 6, 2008

Software Engineering Institute | Carnegie Mellon

© 2008 Carnegie Mellon University

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
06 MAR 2008		**00-00-2008 to 00-00-2008**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Ultra-Large-Scale Systems: Scale Changes Everything	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Carnegie Mellon University ,Software Engineering Institute (SEI),Pittsburgh,PA,15213**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
presented at the 2008 SMART Ultra-Large-Scale System Forum, Pittsburgh, PA, 6 Mar 2008.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**56**	

Societal Problems

Climate change and the environment

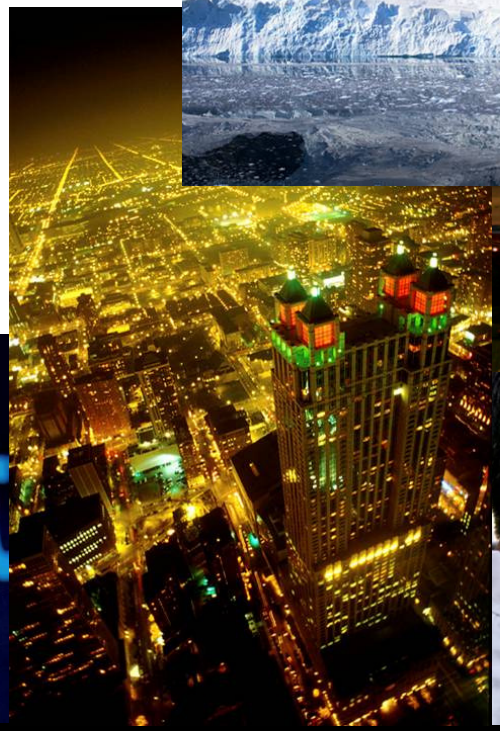
Powering our civilization

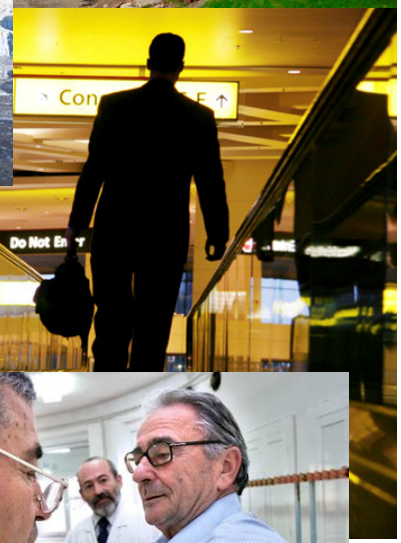
Disease, epidemics, and health care

Livable megacities

Safety and security

Transportation



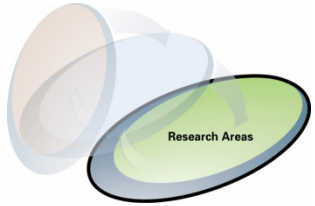


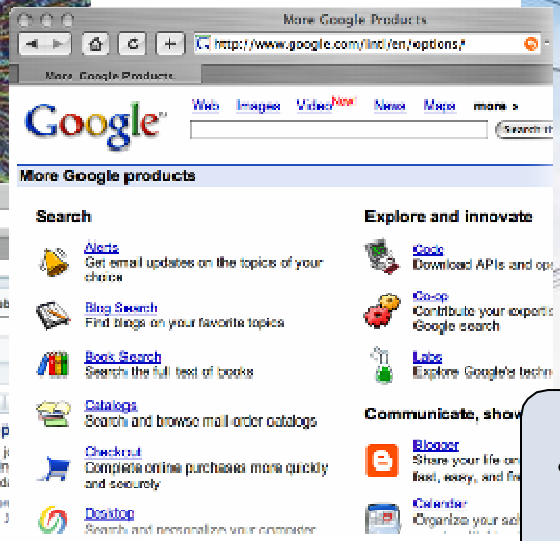
© 2008 Carnegie Mellon University

Society's Dependence on Software

© 2008 Carnegie Mellon University

Trend Toward Increasing Scale-1

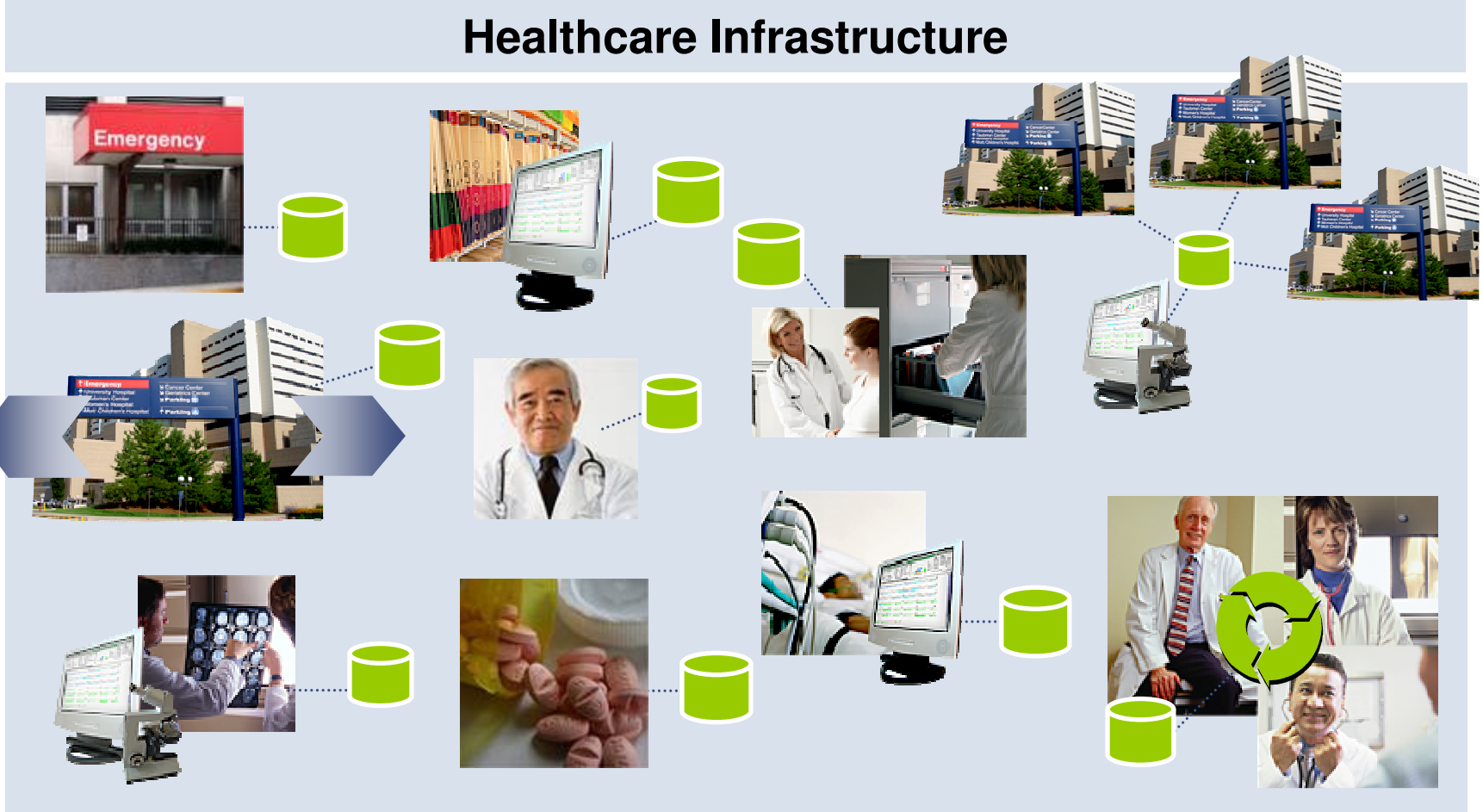




- Enormous web service and computing infrastructure
- Supply chain systems
- Software-based engineering systems

© 2008 Carnegie Mellon University

Trend Toward Increasing Scale-2



© 2008 Carnegie Mellon University

Trend Toward Increasing Scale-3

Homeland Security

© 2008 Carnegie Mellon University

Trend Toward Increasing Scale-4

Networked Automobiles

© 2008 Carnegie Mellon University

Trend Toward Increasing Scale-5

Saving the Environment

© 2008 Carnegie Mellon University

Increasing Scale In Military Systems

Increasingly Complex Systems

- ultra-large, network-centric, real-time, cyber-physical-social systems
  - thousands of platforms, sensors, decision nodes, weapons, and warfighters
  - connected through heterogeneous wired and wireless networks

Goal: *Information Superiority*

- *Transient and enduring resource constraints and failures*
- *Continuous adaptation*
  - changes in mission requirements
  - changes in operating environments
  - changes in force structure
  - perpetual systems' evolution
  - addition of new systems
- *Sustainable - legally, technically, politically*

© 2008 Carnegie Mellon University

The Challenge

The Honorable
Claude M. Bolton, Jr.

"Our soldiers depend on software and will depend more on software in the future.

The Army's success depends on software and the software industry.

We need better tools to meet future challenges, and neither industry nor government is working on how to do things light-years faster and cheaper.

How can future systems, which are likely to be a **billion lines of code**, be built reliably if we can't even get today's systems right?"

— Asst Sec Army Claude Bolton
August 16, 2005

© 2008 Carnegie Mellon University

Ultra-Large-Scale (ULS) Systems Study

Gather leading experts to study:

- characteristics of ULS systems
- challenges and breakthroughs required
- promising research and approaches

Intended outcomes:

- ULS System Research Agenda
- program proposal
- collaborative research network

About the Effort

Funded by the Army (ASA ALT)

Staffing: 9 member SEI team
13 member expert panel

Duration: one year (04/05 -- 05/06)

© 2008 Carnegie Mellon University

SEI Team

Linda Northrop
Software Design and Architecture,
Software Product Lines

Rick Kazman
Adaptive Architectures, Design Methods

Peter Feiler
Methodologies,
Configuration Management

Mark Klein
Real-time Performance Analysis,
Software Architecture Design and Analysis

John Goodenough
Software Reliability,
Safety Assurance

Mark Pleszkoch
Rigorous Software Engineering Methods

Rick Linger
Rigorous Software and
Systems Engineering

Kurt Wallnau
Software Components,
Program Generation, Language Semantics

Tom Longstaff
Security and Survivability Engineering
in Complex Systems

Bill Pollack
Chief Editor

Daniel Pipitone
Chief Graphical Designer

© 2008 Carnegie Mellon University

SEI Team (and Expert Panel)

Gregory Abowd
Georgia Institute of Technology

Peter Neumann
SRI International Computer Science Laboratory

Douglas Schmidt
Vanderbilt University

Carliss Baldwin
Harvard Business School

Mary Shaw
Carnegie Mellon University

Bob Balzer
Teknowledge Corporation

Dan Siewiorek
Carnegie Mellon University

Richard P. Gabriel
Sun Microsystems

Kevin Sullivan
University of Virginia

Gregor Kiczales
University of British Columbia

Ali Mili
New Jersey Institute of Technology

John Lehoczky
Carnegie Mellon University

Jack Whalen
PARC

© 2008 Carnegie Mellon University

Instead: A Different Kind of Study

It presents an overall research agenda -- not just for new tools or a new software method or modest improvements in today's approaches.

It is based on the challenges associated with ultra-large scale.

It focuses on the future.

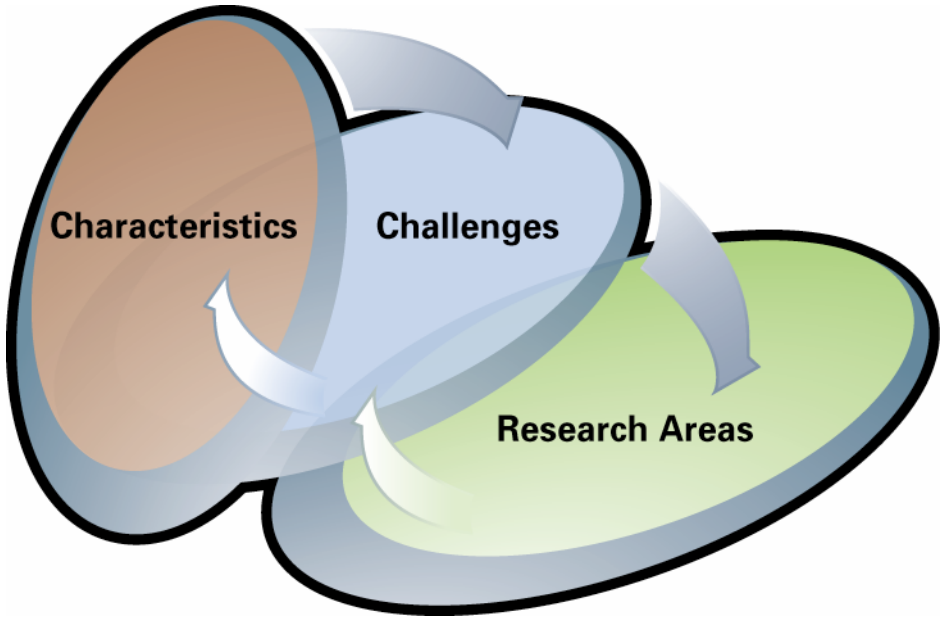
It involves an interdisciplinary base.

It takes a fresh perspective on the development, deployment, operation, and evolution of software-intensive systems.

Germs of these ideas are present today in small research pockets; these efforts are currently too small to have much impact on next-generation ULS systems.

© 2008 Carnegie Mellon University

ULS Systems Research Agenda



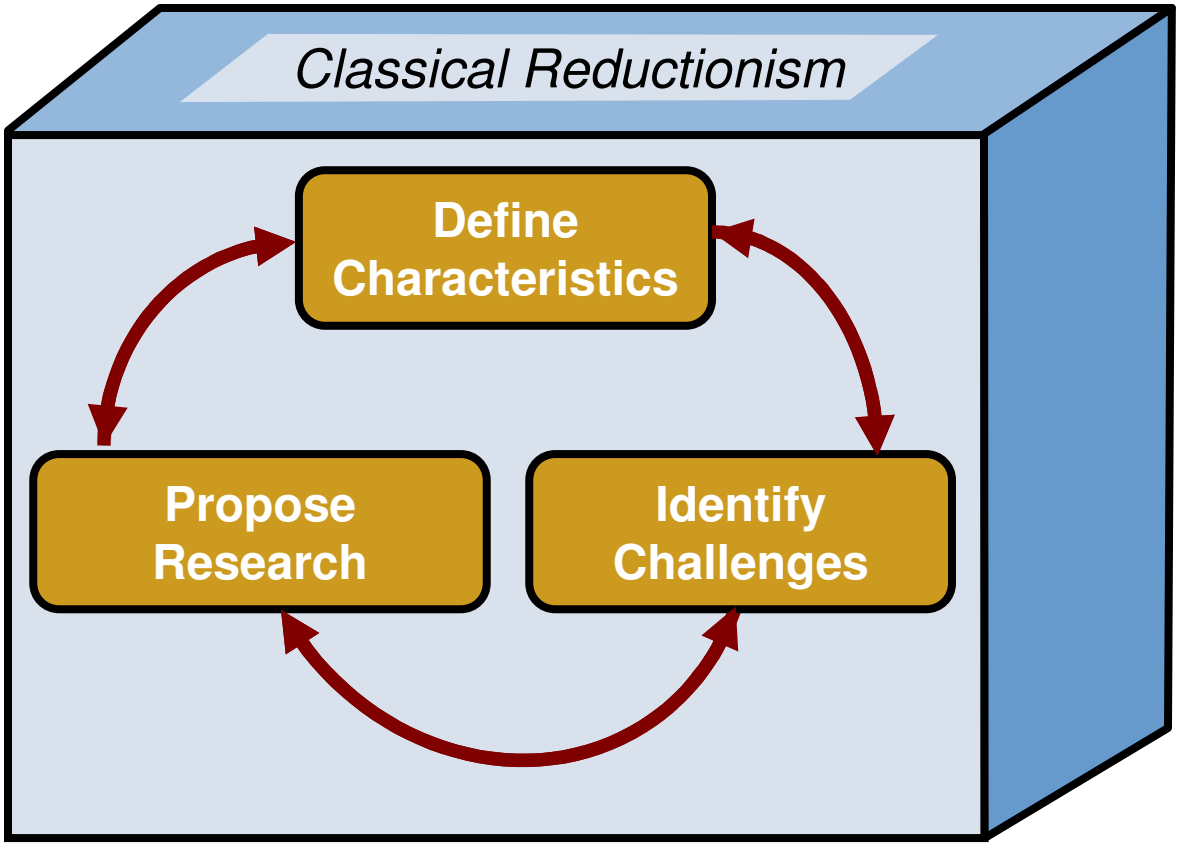
Describes

- the characteristics of ULS systems
- the associated challenges
- promising research areas and topics

Is based on a new perspective needed to address the problems associated with ultra-large-scale systems.

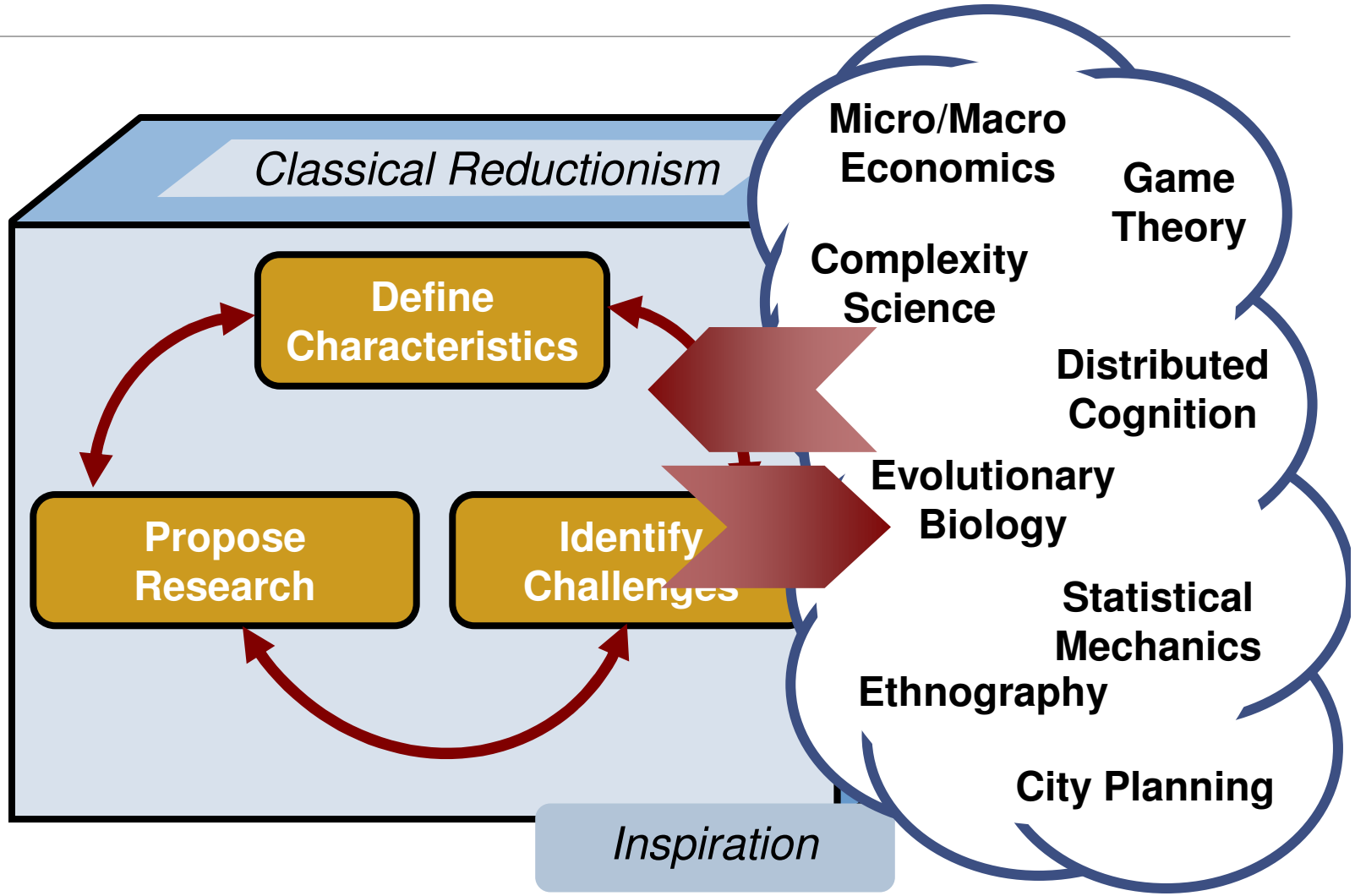
© 2008 Carnegie Mellon University

Working Inside and Outside the Box



© 2008 Carnegie Mellon University

Working Inside and Outside the Box

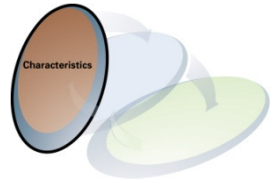


© 2008 Carnegie Mellon University

The Journey

© 2008 Carnegie Mellon University

What Is an Ultra-Large-Scale (ULS) System?



A ULS System has unprecedented scale in some of these dimensions:

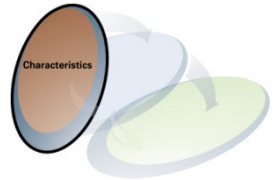
- Lines of code
- Amount of data stored, accessed, manipulated, and refined
- Number of connections and interdependencies
- Number of hardware elements
- Number of computational elements
- Number of system purposes and user perception of these purposes
- Number of routine processes, interactions, and “emergent behaviors”
- Number of (overlapping) policy domains and enforceable mechanisms
- Number of people involved in some way

ULS systems will be interdependent webs of software-intensive systems, people, policies, cultures, and economics.

ULS systems are systems of systems at internet scale.

© 2008 Carnegie Mellon University

Scale Changes Everything



Characteristics of ULS systems arise because of their scale.

- Decentralization
- Inherently conflicting, unknowable, and diverse requirements
- Continuous evolution and deployment
- Heterogeneous, inconsistent, and changing elements
- Erosion of the people/system boundary
- Normal failures
- New paradigms for acquisition and policy

These characteristics may appear in today's systems and systems of systems, but in ULS systems they dominate.

These characteristics undermine the assumptions that underlie today's software engineering approaches.

© 2008 Carnegie Mellon University

Today's Approaches

The Engineering Perspective - for large scale software-intensive systems

- largely top-down and plan-driven
- requirements/design/build cycle with standard well-defined processes
- centrally controlled implementation and deployment
- inherent validation and verification

The Agile Perspective - proven for smaller software projects

- fast cycle/frequent delivery/test driven
- simple designs embracing future change and refactoring
- small teams and retrospective to enable team learning
- tacit knowledge

Today's approaches are based on perspectives that fundamentally do not cope with the new characteristics arising from ultra-large scale.

The mentality of looking backward doesn't scale.

© 2008 Carnegie Mellon University

A New Perspective is Required

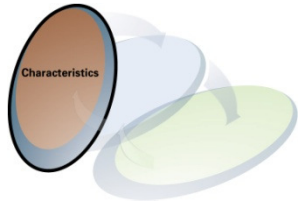
"The older is not always a reliable model for the newer, the smaller for the larger, or the simpler for the more complex…Making something greater than any existing thing necessarily involves going beyond experience."

Henry Petroski

Pushing the Limits: New Adventures in Engineering

© 2008 Carnegie Mellon University

Today We Build “Buildings”



© 2008 Carnegie Mellon University

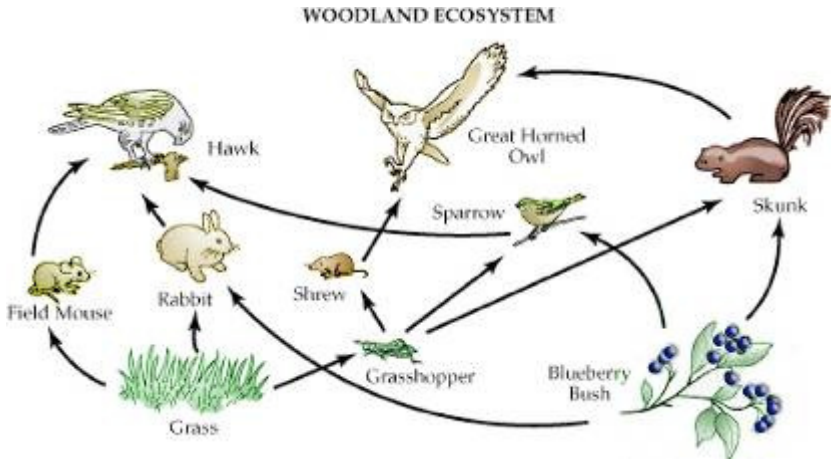
We Need To Think Cities

"Cities are places of massive information flows, networks, and conduits, and myriad transitory information exchanges."

Howard Rheinegold: *Smart Mobs*

© 2008 Carnegie Mellon University

We Need to Think Ecosystem



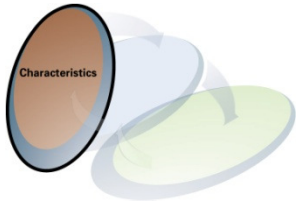
Diverse users with complex networked dependencies and intrinsic adaptive behavior

Has:

- Robustness mechanisms: achieving stability in the presence of disruption
- Measures of health: diversity, population trends, other key indicators

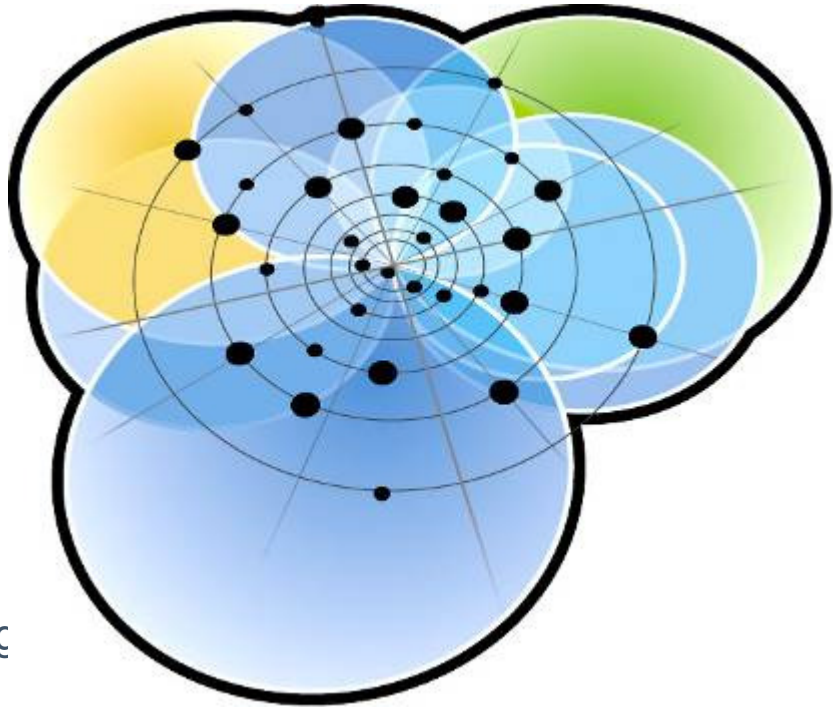
© 2008 Carnegie Mellon University

We Need to Think Socio-Technical Ecosystems



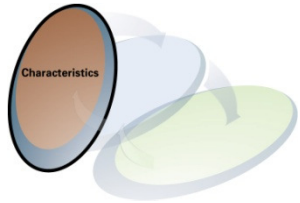
Socio-technical ecosystems include people, organizations, and technologies at all levels with significant and often competing interdependencies.

- There will be competition for resources.
- There will be organizations and participants responsible for setting policies.
- There will be organizations and participants responsible for producing ULS systems.
- There will need to be local and global indicators of health that will trigger necessary changes in policies and in element and system behavior.



© 2008 Carnegie Mellon University

Why a New Perspective?



There are fundamental assumptions that underlie today's software engineering and software development approaches that are ***undermined*** by the characteristics of ULS systems.

There are challenges associated with ULS systems that today's perspectives are very unlikely to be able to address.

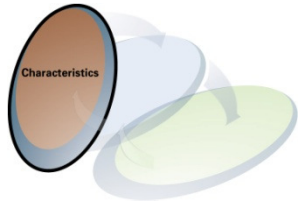
For the last forty years, engineering has been the dominant metaphor for software systems creation.

In ULS systems, we now are dealing with not just software but an ecosystem of people, organizations, governance, social interaction, hardware, and software.

Engineering is no longer the dominant metaphor.

© 2008 Carnegie Mellon University

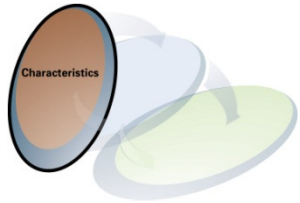
ULS Systems vs Today's Approaches - 1



Characteristics	Today's assumptions undermined
Decentralized control	All conflicts must be resolved and resolved centrally and uniformly.
Inherently conflicting, unknowable, and diverse requirements	Requirements can be known in advance and change slowly. Tradeoff decisions will be stable.
Continuous evolution and deployment	System improvements are introduced at discrete intervals.
Heterogeneous, inconsistent, and changing elements	Effect of a change can be predicted sufficiently well. Configuration information is accurate and can be tightly controlled. Components and users are fairly homogeneous.

© 2008 Carnegie Mellon University

ULS Systems vs Today's Approaches - 2



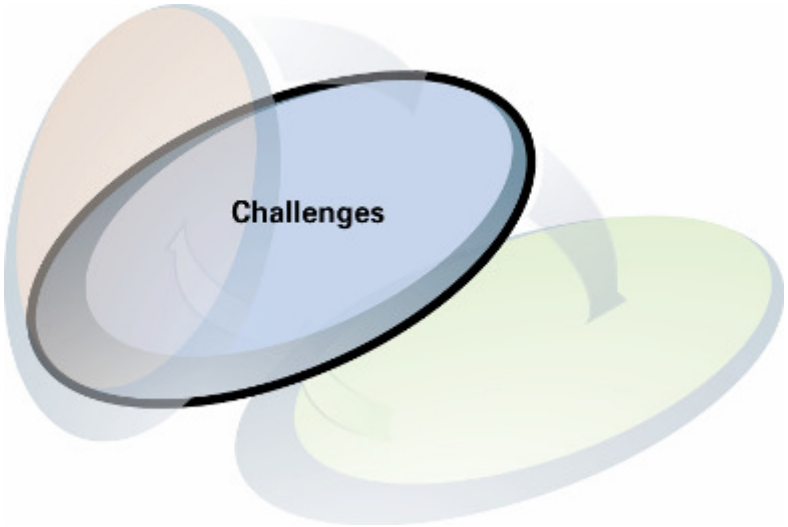
Characteristics	Today's assumptions undermined
Erosion of the people/system boundary	People are just users of the system. Collective behavior of people is not of interest. Social interactions are not relevant.
Normal failures	Failures will occur infrequently. Defects can be removed.
New paradigms for acquisition and policy	A prime contractor is responsible for system development, operation, and evolution.

© 2008 Carnegie Mellon University

Challenges

ULS systems will present challenges in three broad areas:

- Design and evolution
- Orchestration and control
- Monitoring and assessment



© 2008 Carnegie Mellon University

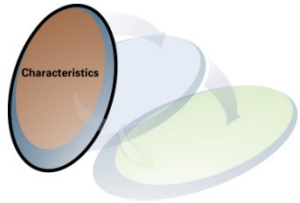
Design and Evolution

Specific challenges in ULS system design and evolution stemming directly from the characteristics of ULS systems:

- Economics and industry structure
- Social activity for constructing computational environments
- Legal issues
- Enforcement mechanisms and processes
- Definition of common services supporting the ULS system
- Rules and regulations
- Agility
- Handling of change
- Integration
- User-controlled evolution
- Computer-supported evolution
- Adaptable structure
- Emergent quality

© 2008 Carnegie Mellon University

Orchestration and Control



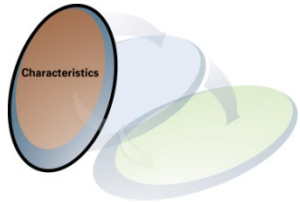
Orchestration and control refers to the set of activities needed to make the elements of a ULS system work together in reasonable harmony to ensure continuous satisfaction of mission objectives.

Orchestration is needed at all levels of ULS systems and challenges us to create new ways for

- Online modification
- Maintenance of quality of service while providing necessary flexibility
- Creation and execution of policies and rules
- Adaptation to users and contexts
- Enabling of user-controlled orchestration

© 2008 Carnegie Mellon University

Monitoring and Assessment



The effectiveness of ULS system design, operation, evolution, orchestration, and control has to be evaluated.

There must be an ability to monitor and assess ULS system state, behavior, and overall health and well being.

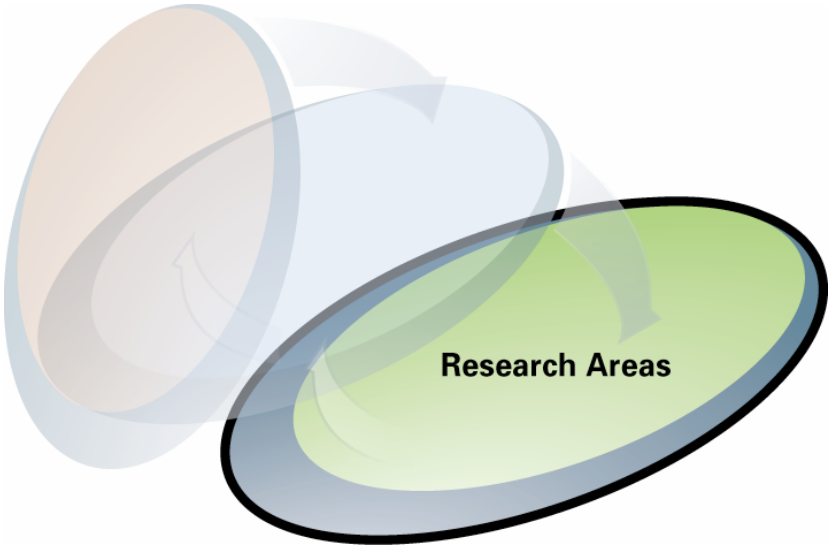
Challenges include

- Defining indicators
- Understanding why indicators change
- Prioritizing the indicators
- Handling change and imperfect information
- Gauging the human elements

© 2008 Carnegie Mellon University

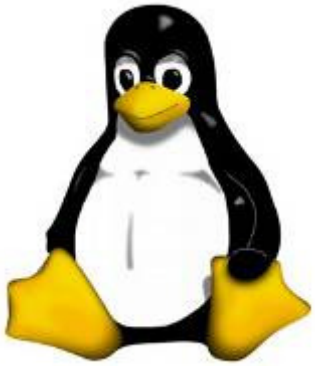
Where Do We Focus Our Research

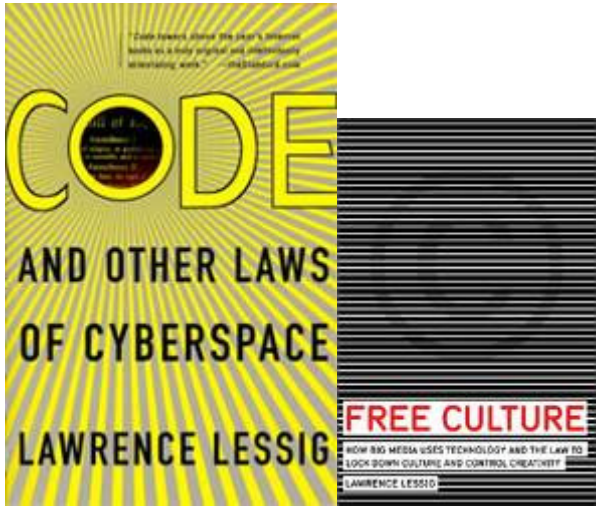
- Address the predominant characteristics of ULS systems and the three challenge categories.
- Look for breakthroughs not incremental improvement in current approaches.
- Take a more expansive view of software research and include its interactions with associated research in the physical and social sciences.



© 2008 Carnegie Mellon University

Inspiration: Open Source and Cooperative Communities



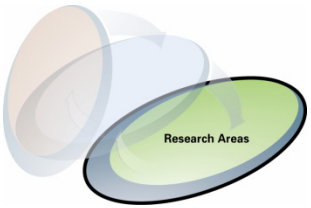


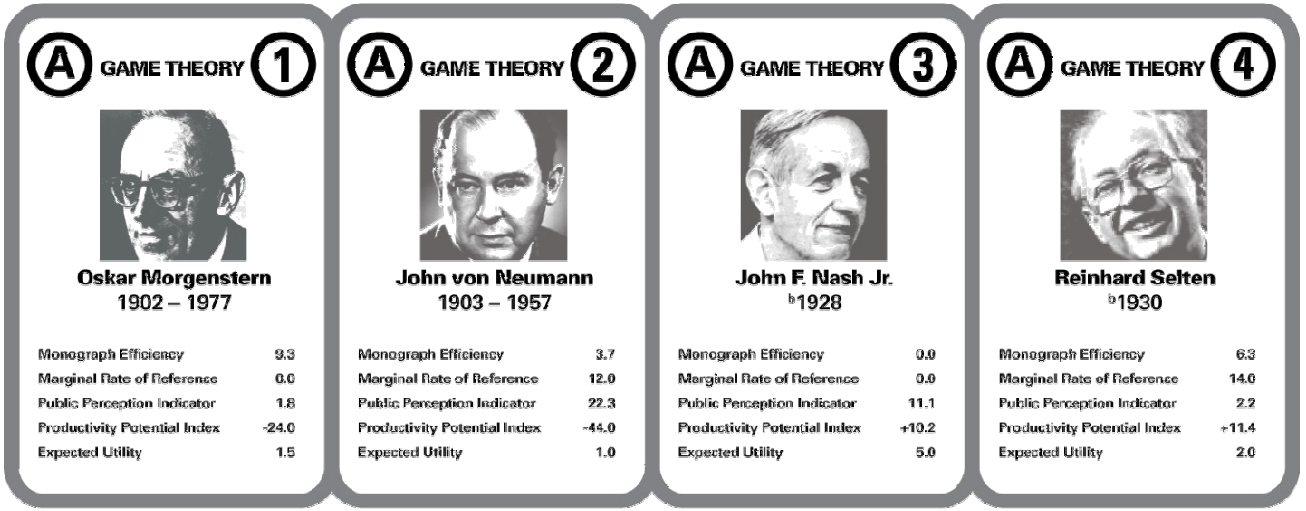
ebay

Google

© 2008 Carnegie Mellon University

Inspiration: Game Theory





Algorithmic Mechanism Design

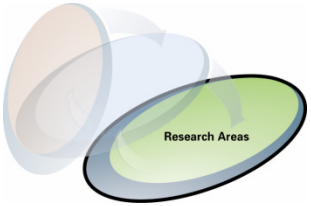
- games + microeconomics + computation
- computational markets for any scarce ULS resource?

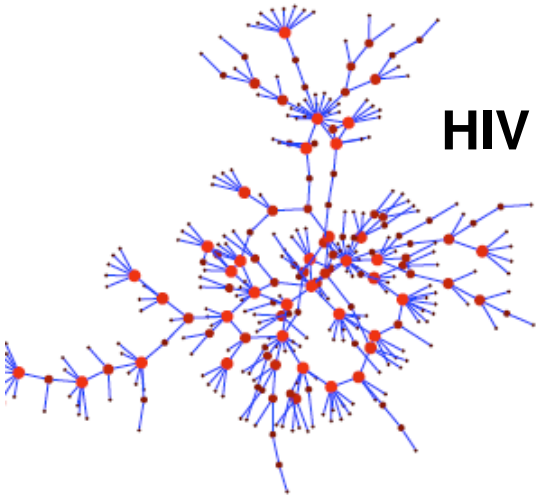
Institution Design

- learning games + self-reinforced expectations + cultural norms
- better formal models of acquisition in non-prime-dominated landscape?

© 2008 Carnegie Mellon University

Inspiration: Networks, Statistical Mechanics, Complexity





HIV Partners

Stability

Networks Are Everywhere

Recurring "scale free" structure

- internet & yeast protein structures

Analogous dynamics?

- epidemiology, robustness and vulnerability

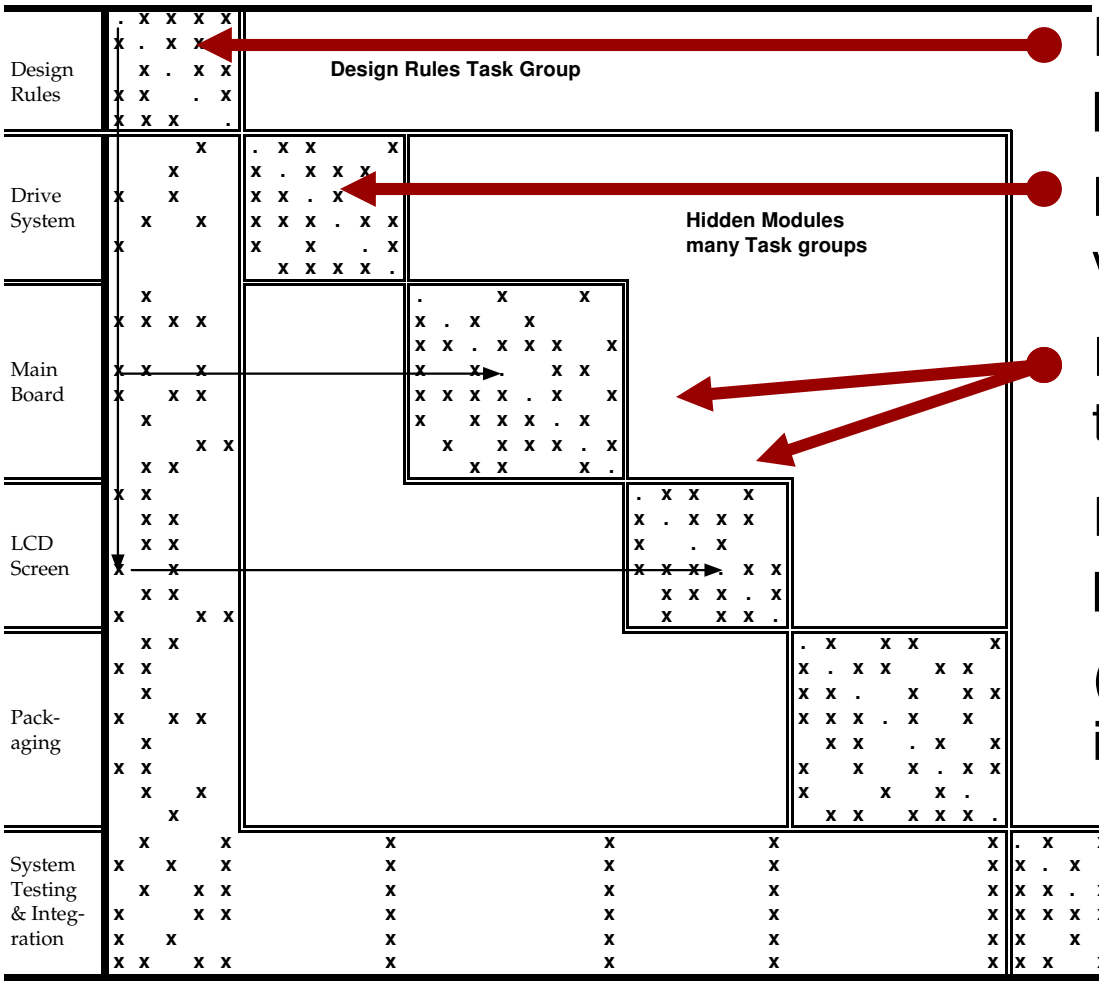
Unstable Equilibrium

How many changes before a system becomes unstable?

What scale and frequency of disruptions can be expected?

© 2008 Carnegie Mellon University

Economics (Finance) As Design Criteria



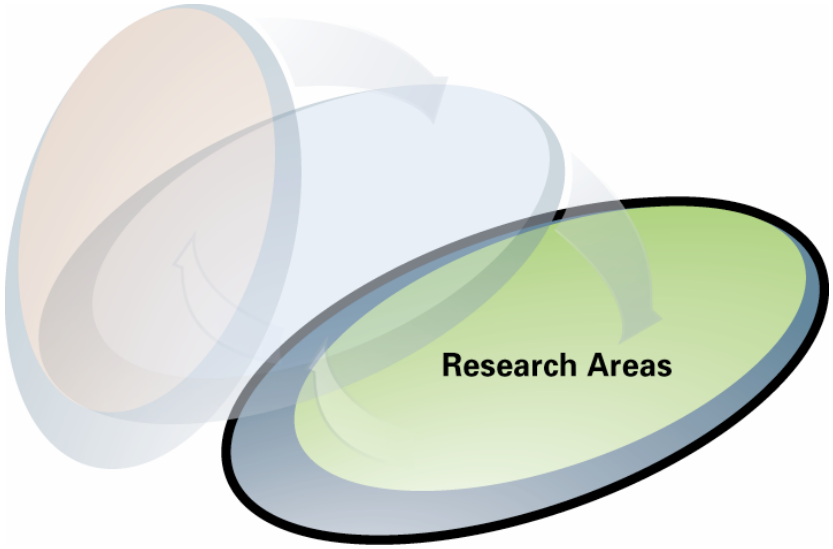
- Design rules (feature parameterization)
- Module maximizes option value
- ROI inversely proportional to module footprint
- Each module a potential point for competition
- (apologies for abuse of ideas to Carliss Baldwin)

© 2008 Carnegie Mellon University

Research Portfolio

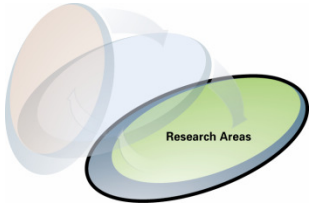
We recommend an interdisciplinary portfolio of seven research areas and suggested topics for breakthrough research needed to meet the challenges associated with ULS systems.

- Is not expressed in terms of today's "hot" technologies.
- Does not supplant current software research.
- Expands today's horizons.



© 2008 Carnegie Mellon University

Research Areas - 1

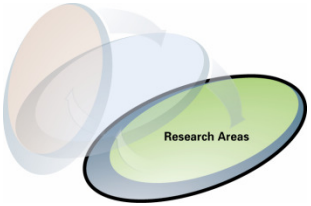


6.1 Human Interaction: Involves anthropologists, sociologists, and social scientists conducting detailed socio-technical analyses of user interactions in the field, with the goal of understanding how to construct and evolve such socio-technical systems effectively.

- Context-Aware Assistive Computing
- Understanding Users and Their Contexts
- Modeling Users and User Communities
- Fostering Non-Competitive Social Collaboration
- Longevity

© 2008 Carnegie Mellon University

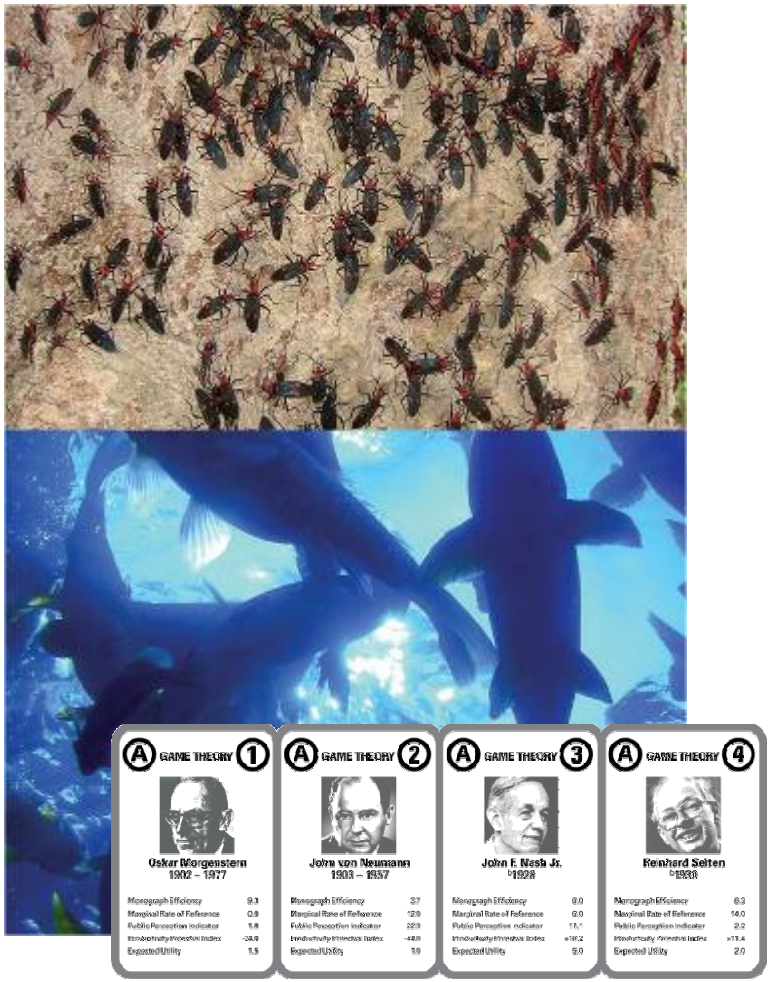
Research Areas - 2



6.2 Computational Emergence:

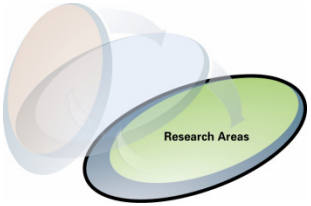
Explores the use of methods and tools based on economics and ***game theory*** (e.g., ***mechanism design***) to ensure globally optimal ULS system behavior by exploiting the strategic self interests of the system's constituencies; explores ***metaheuristics*** and ***digital evolution*** to augment the cognitive limits of human designers.

- Algorithmic Mechanism Design
- Metaheuristics in Software Engineering
- Digital Evolution



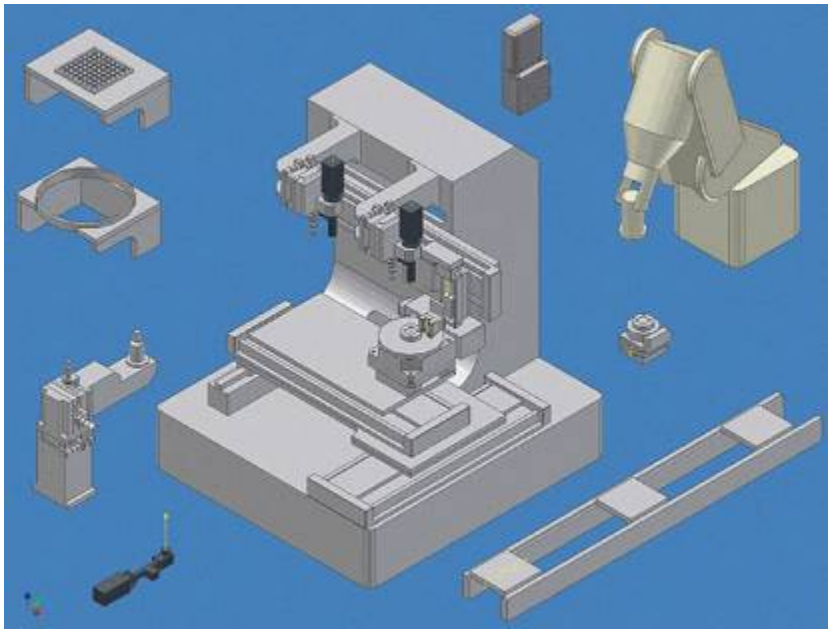
© 2008 Carnegie Mellon University

Research Areas - 3



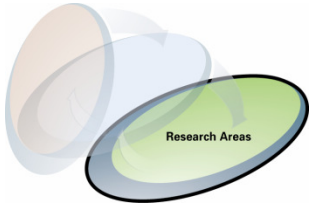
6.3 Design: Broadens the traditional technology-centric definition of design to include people and organizations; social, cognitive, and economic considerations; and design structures such as ***design rules*** and government policies.

- Design of All Levels
- Design Spaces and Design Rules
- Harnessing Economics to Promote Good Design
- Design Representation and Analysis
- Assimilation
- Determining and Managing Requirements



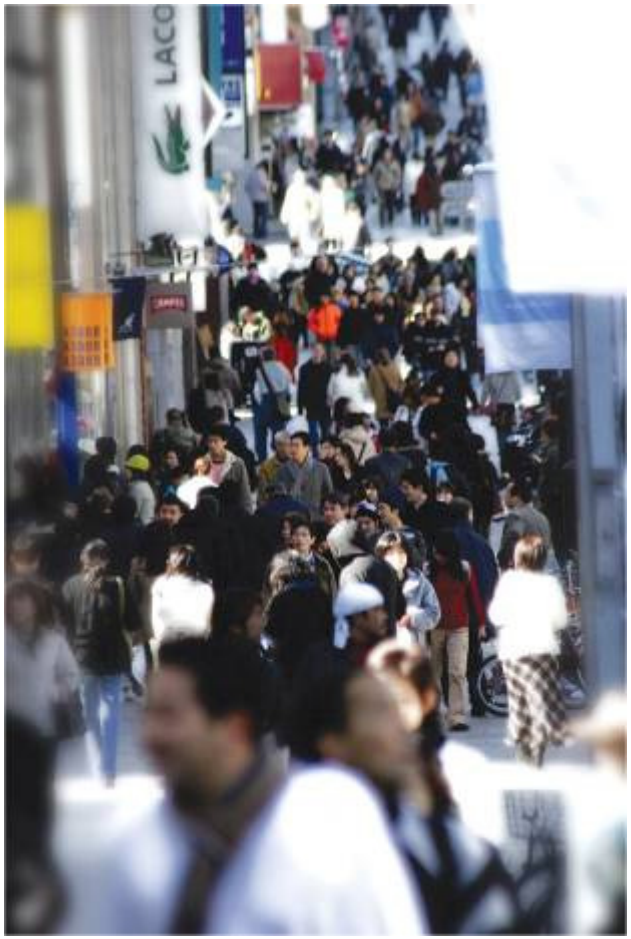
© 2008 Carnegie Mellon University

Research Areas - 4



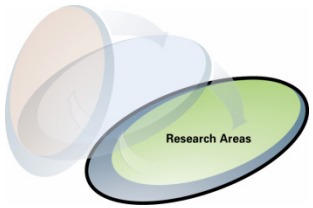
6.4 Computational Engineering: Focuses on evolving the expressiveness of representations to accommodate the semantic diversity of many languages and focuses on providing automated support for computing the evolving behavior of components and their compositions.

- Expressive Representation Languages
- Scaled-Up Specification, Verification, and Certification
- Computational Engineering for Analysis and Design



© 2008 Carnegie Mellon University

Research Areas - 5

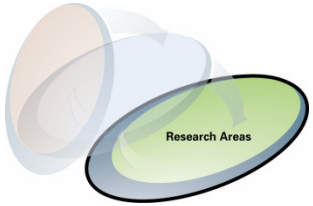


6.5 Adaptive System Infrastructure: Investigates integrated development environments and runtime platforms that will support the decentralized, "always-on," nature of ULS systems as well as technologies, methods, and theories that will enable ULS systems to be developed in their deployment environments.

- Decentralized Production Management
- View-Based Evolution
- Evolutionary Configuration and Deployment
- In Situ Control and Adaptation

© 2008 Carnegie Mellon University

Research Areas - 6

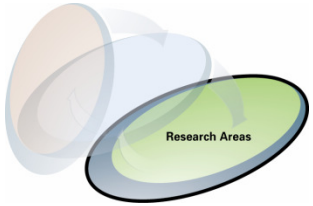


6.6 Adaptable and Predictable System Quality: Focuses on how to maintain quality in a ULS system in the face of continuous change, ongoing failures, and attacks and how to identify, predict, and control new indicators of ***system health*** (akin to the U.S. gross domestic product) that are needed because of the scale of ULS systems.

- Robustness, Adaptation, and Quality Attributes
- Scale and Composition of Quality Attributes
- Understanding People-Centric Quality Attributes
- Enforcing Quality Requirements
- Security, Trust, and Resiliency
- Engineering Management at Ultra-Large Scales

© 2008 Carnegie Mellon University

Research Areas - 7



6.7 Policy, Acquisition, and Management: Focuses on transforming acquisition policies and processes to accommodate the rapid and continuous evolution of ULS systems by treating suppliers and supply chains as intrinsic and essential components of a ULS system.

- Policy Definition for ULS Systems
- Fast Acquisition for ULS Systems
- Management of ULS Systems

© 2008 Carnegie Mellon University

Research Areas and Challenges

Relationship Between Research Areas and Challenges

Research Areas	Design and Evolution	Orchestration and Control	Monitoring and Assessment
Human Interaction	●	●	
Computational Emergence	●	●	
Design	●		
Computational Engineering	●		
Adaptive System Infrastructure		●	●
Adaptable and Predictable System Quality	●	●	●
Policy, Acquisition, and Management	●	●	

© 2008 Carnegie Mellon University

Toward a Roadmap for a ULS Systems Research Program

There are many possible approaches to structuring a research program from the ULS Systems Research Agenda

We provide three possible support structures based on

1. Specific DoD missions and capabilities
2. DoD research funding types required
3. Estimates of the relative starting points of the research

Sponsors with different needs can choose to support different combinations of research.

Table 4: Research Areas and Range of Risk/Reward

Research Areas and Topics	Existing Groundwork	Breaking Ground	New Direction
Human Interaction			
6.1.1 Context-Aware Assistive Computing	●		
6.1.2 Understanding Users and Their Contexts	●	●	
6.1.3 Modeling Users and User Communities		◒	●
6.1.4 Fostering Non-Competitive Social Collaboration		●	◒
6.1.5 Longevity	●	◒	◒
Computational Emergence			
6.2.1 Algorithmic Mechanism Design	●	◒	●
6.2.2 Metaheuristics in Software Engineering	●	◒	
6.2.3 Digital Evolution	●	◒	
Design			
6.3.1 Design of All Levels	●	●	◒
6.3.2 Design Spaces and Design Rules		●	●
6.3.3 Harnessing Economics to Promote Good Design	●	●	●
6.3.4 Design Representation and Analysis		●	◒
6.3.5 Assimilation	●	●	●
6.3.6 Determining and Managing Requirements	●	●	◒
Computational Engineering			
6.4.1 Expressive Representation Languages	◒	●	●
6.4.2 Scaled-Up Specification, Verification, and Certification	◒	●	

The envisioned outcome of the proposed research is a spectrum of technologies and methods for developing ULS systems, with national-security, economic, and societal benefits that far extend beyond ULS systems themselves.

© 2008 Carnegie Mellon University

Study Conclusions

There are fundamental gaps in our current understanding of software development at the scale of ULS systems.

These gaps

- present profound impediments to the technically and economically effective achievement of the DoD goals* based on information superiority and to effective solutions for many of society's vexing problems
- require a broad, fresh perspective and interdisciplinary, breakthrough research

We recommend

- a ULS Systems Research Agenda that includes research areas based on a fresh perspective aimed at challenges arising from increasing scale

** As stated in the Quadrennial Defense Review (QDR) Report, Feb 2006*

© 2008 Carnegie Mellon University

ULS Systems Research Study Report

Acknowledgements

Executive Summary

Part I

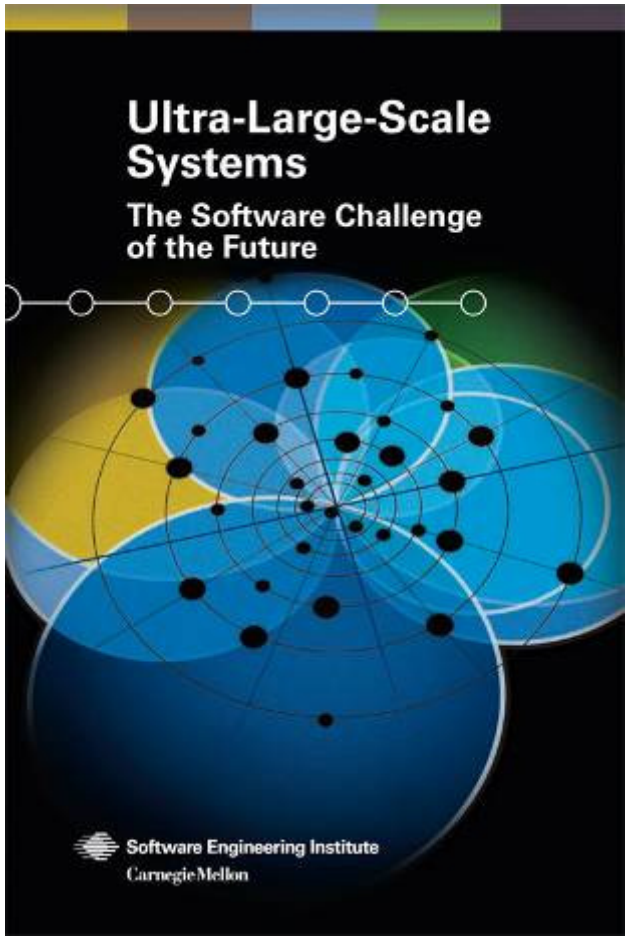
1. Introduction
2. Characteristics of ULS Systems
3. Challenges
4. Overview of Research Areas
5. Summary and Recommendations

Part 2

6 Detailed Description of Research Areas

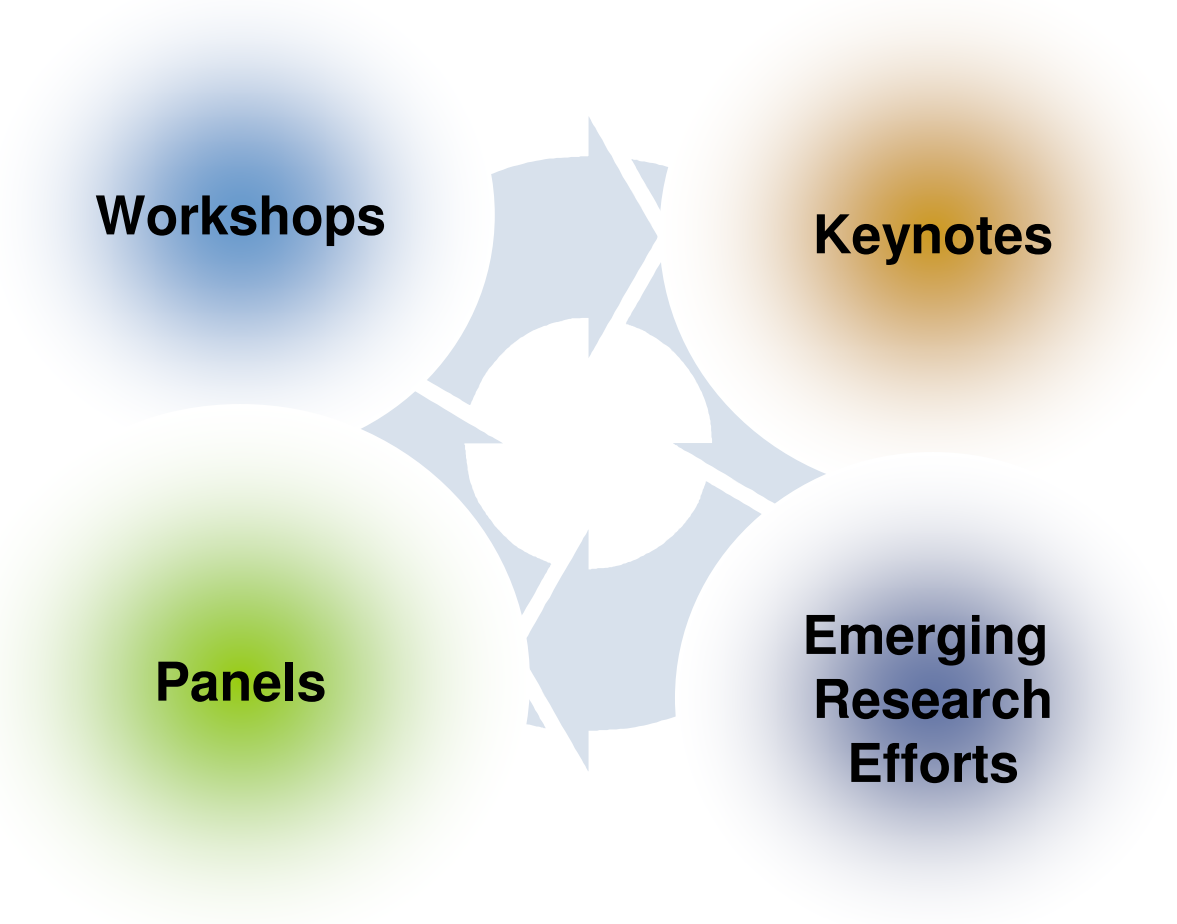
- Glossary

http://www.sei.cmu.edu/uls/



© 2008 Carnegie Mellon University

The Start of a Collaborative Research Network



© 2008 Carnegie Mellon University

Activity Since Report Published in May 06

There is growing community interest and research starts.

Since July 06

- more than 95,000 downloads of the report
- more than 3,000 copies of the report distributed
- more than 14 keynotes and more than 25 presentations by author team
- three press and one industry analyst interviews
- research workshops at OOPSLA 2006 and ICSE 2007
- NSF center established

New SEI Research Activities

- Mechanism design
- Implications of ULS on software architecture

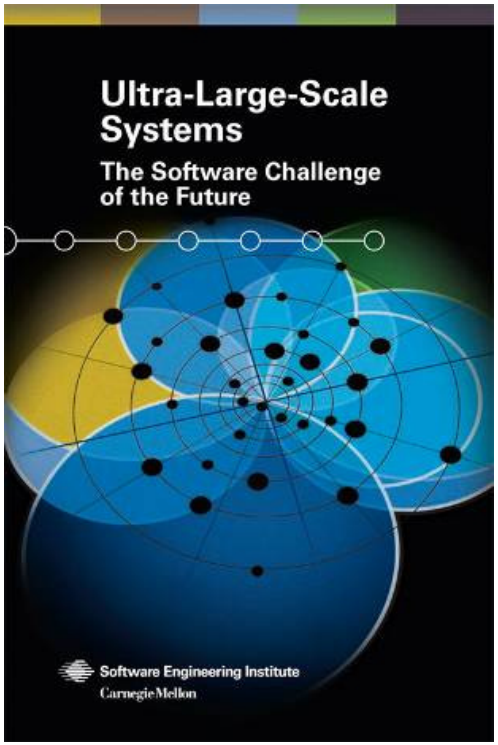
Roadmap Exercise funded by Army organization (CERDEC)

New book from a non-military perspective is underway.

SMART Event

Upcoming event:

- ICSE 2008 Workshop



http://www.sei.cmu.edu/uls

© 2008 Carnegie Mellon University

Redesign of SEI ULS Systems Website

New site features include

- Podcasts and video presentations
- ULS Systems news & events
  - RSS feed
- ULS Systems library
- Online Glossary

http://www-sei.cmu.edu/uls/

© 2008 Carnegie Mellon University

What We Learned That We Want to Share

- There is an unstoppable trend toward increasing scale in many systems important to our society.
- Scale changes everything.
- Manifestations of scale and its attendant complexity arise in many disciplines, and can be understood as a phenomenon in its own right.
- New, interdisciplinary perspective and new research in building ultra-large-scale systems is long overdue.

© 2008 Carnegie Mellon University

thank you

© 2008 Carnegie Mellon University

Contact Information

Linda Northrop
Director
Product Line Systems Program
Telephone: 412-268-7638
Email: lmn@sei.cmu.edu

U.S. Mail:
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213-3890

SEI Fax: 412-268-5758

© 2008 Carnegie Mellon University