Architecture-Based Analysis of Systemility Synergies and Conflicts

Barry Boehm, Jo Ann Lane, USC
Kevin Sullivan, U. Virginia

NDIA Systems Engineering Conference
October 30, 2013

Outline

• Critical nature of the ilities
  – Major source of project overruns, failures
  – Significant source of stakeholder value conflicts
  – Poorly defined, understood
  – Underemphasized in project management
• Challenges for cyber-physical-human systems
• SERC Foundations efforts
  – AFIT, GaTech, MIT, NPS, PennState, USC, Uva, WSU
  – Stakeholder value-based, means-ends hierarchy
  – Formal analysis of ility definitions and relations
  – Architecture strategy synergies and conflicts
Importance of iliity Tradeoffs
Major source of DoD system overruns

- System iliities have systemwide impact
  - System elements generally just have local impact
- iliities often exhibit asymptotic behavior
  - Watch out for the knee of the curve
- Best architecture is a discontinuous function of iliity level
  - “Build it quickly, tune or fix it later” highly risky
  - Large system example below

Role-Based Ilities Value Diversity
Bank of America Master Net; DoD?

Users
- Many features
- Changeable requirements
- Applications compatibility
- High levels of service
- Voice in acquisition
- Flexible contract
- Early availability

Maintainers
- Ease of transition
- Ease of maintenance
- Applications compatibility
- Voice in acquisition

Acquirers
- Mission cost/effectiveness
- Limited development budget/schedule
- Government standards compliance
- Political correctness
- Development visibility and control
- Rigorous contact

Developers
- Flexible contract
- Ease of meeting budget and schedule
- Stable requirements
- Freedom of choice: process
- Freedom of choice: team
- Freedom of choice: COTS/reuse

PC: Process
PD: Product
PP: Property
S: Success
Example of Current Practice

• “The system shall have a Mean Time Between Failures of 10,000 hours”
• What is a “failure?”
  – 10,000 hours on liveness
  – But several dropped or garbled messages per hour?
• What is the operational context?
  – Base operations? Field operations? Conflict operations?
• Most management practices focused on functions
  – Requirements, design reviews; traceability matrices; work breakdown structures; data item descriptions; earned value management
• What are the effects on other –ilities?
  – Cost, schedule, performance, maintainability?

USC: COCOMO II-Based Tradeoff Analysis
Better, Cheaper, Faster: Pick Any Two?

For 100-KSLOC set of features
Can “pick all three” with 77-KSLOC set of features
Outline

- Critical nature of the ilities
  - Major source of project overruns, failures
  - Significant source of stakeholder value conflicts
  - Poorly defined, understood
  - Underemphasized in project management

  Challenges for cyber-physical-human systems

- SERC Foundations efforts
  - AFIT, GaTech, MIT, NPS, PennState, USC, Uva, WSU
  - Stakeholder value-based, means-ends hierarchy
  - Formal analysis of ility definitions and relations
  - Architecture strategy synergies and conflicts

Importance of Cyber-Physical Systems
Major gap in tradespace analysis capabilities

- Current ERS, DARPA tradespace research focused on physical system tradeoffs
  - Range, payload, size, weight, lethality, power and fuel consumption, communications bandwidth, etc.
  - Some focus on physical modularity, composability

- Current cyber tradespace research focused on software, computing, human factors tradeoffs
  - security, safety, interoperability, usability, flexibility, adaptability, dependability, response time, throughput, etc.

- Gaps in capabilities for co-design of hardware, software, and human factors; integration of tradespace analyses
GaTech – FACT Tradespace Tool
Being used by Marine Corps

Configure vehicles from the “bottom up”
Quickly assess impacts on performance

Prioritized JCIDSilities
User View by Combatant Commands: Top priority first

- Intelligence, Surveillance, and Reconnaissance
  - Comprehensive Persistent Survivable Integrated Timely Credible Adaptable Innovative
- Command and Control (note emphasis on Usability aspects)
  - Interoperability Understanding Timeliness Accessibility Simplicity Completeness Agility Accuracy Relevance Robustness Operational Trust
- Logistics: Supply
  - Responsiveness Sustainability Flexibility Survivability Attainability Economy Simplicity
- Logistics: Maintenance
  - Sustainability Responsiveness Attainability Flexibility Economy Survivability Simplicity
- Net-Centric: Information Transport
  - Accessible Capacity Accurate Timely Throughput Expeditionary Latency
Outline

• Critical nature of the ilities
  – Major source of project overruns, failures
  – Significant source of stakeholder value conflicts
  – Poorly defined, understood
  – Underemphasized in project management

• Challenges for cyber-physical-human systems

  ➔ SERC Foundations efforts

  – AFIT, GaTech, MIT, NPS, PennState, USC, Uva, WSU
  – Stakeholder value-based, means-ends hierarchy
  – Formal analysis of ility definitions and relations
  – Architecture strategy synergies and conflicts

SERC Value-Based ilities Hierarchy

Based on ISO/IEC 9126, 25030; JCIDS; previous SERC research

• Individual ilities
  – Resource Utilization: Cost, Duration, Personnel, Scarce Quantities (capacity, weight, energy, ...), Manufacturability, Sustainability
  – Protection: Security, Safety
  – Robustness: Reliability, Availability, Maintainability, Survivability
  – Flexibility: Modifiability, Tailorability, Adaptability
  – Composability: Interoperability, Openness, Service-Orientation

• Composite ilities
  – Comprehensiveness/Suitability: all of the above
  – Dependability: Mission Effectiveness, Protection, Robustness
  – Resilience: Protection, Robustness, Flexibility
  – Affordability: Mission Effectiveness, Resource Utilization
Means-Ends Framework: Affordability

Get the Best from People
- Staffing, Incentivizing, Teambuilding
- Facilities, Support Services
- Kaizen (continuous improvement)

Make Tasks More Efficient
- Tools and Automation
- Work and Oversight Streamlining
- Collaboration Technology

Eliminate Tasks
- Lean and Agile Methods
- Task Automation
- Model-Based Product Generation

Eliminate Scrap, Rework
- Early Risk and Defect Elimination
- Evidence-Based Decision Gates
- Modularity Around Sources of Change
- Incremental, Evolutionary Development
- Value-Based, Agile Process Maturity

Simplify Products (KISS)
- Risk-Based Prototyping
- Value-Based Capability Prioritization
- Satisficing vs. Optimizing Performance

Reuse Components
- Domain Engineering and Architecture
- Composable Components, Services, COTS
- Legacy System Repurposing

Reduce Operations, Support Costs
- Automate Operations Elements
- Design for Maintainability, Evolvability
- Streamline Supply Chain
- Anticipate, Prepare for Change

Value- and Architecture-Based Tradeoffs and Balancing

Architecture Strategy Synergy-Conflict Matrix

<table>
<thead>
<tr>
<th>Reliability</th>
<th>Modifiability</th>
<th>Interoperability</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Non-redundant models enhance reliability, reduces costs.</td>
<td>- Domain architecture eliminates more effective, more modifiable solutions.</td>
<td>- Domain architecture improves reliability, reduces costs.</td>
<td>- Improves reliability, reduces costs.</td>
</tr>
<tr>
<td>- Redundancy increases reliability, reduces costs.</td>
<td>- Domain architecture improves reliability, reducing costs.</td>
<td>- Domain architecture improves reliability, reduces costs.</td>
<td>- Improved reliability, reduced costs.</td>
</tr>
<tr>
<td>- Redundancy increases reliability, reduces costs.</td>
<td>- Domain architecture improves reliability, reducing costs.</td>
<td>- Domain architecture improves reliability, reducing costs.</td>
<td>- Improved reliability, reduced costs.</td>
</tr>
<tr>
<td>- Redundancy increases reliability, reduces costs.</td>
<td>- Domain architecture improves reliability, reducing costs.</td>
<td>- Domain architecture improves reliability, reducing costs.</td>
<td>- Improved reliability, reduced costs.</td>
</tr>
</tbody>
</table>

Interoperability
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.

Synergy
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.

Conflict
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.

Cost
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.
- Domain architecture improves reliability, reducing costs.

Increased reliability increases acquisition costs.
- Increased reliability increases acquisition costs.
- Increased reliability increases acquisition costs.
- Increased reliability increases acquisition costs.
Software Development Cost vs. Quality

![Software Development Cost vs. Quality Diagram]

Software Ownership Cost vs. Quality

![Software Ownership Cost vs. Quality Diagram]