Contributing toward a Prescriptive "Theory of Ilities"

Dr. Adam M. Ross, MIT

SE 410 Graduate Seminar
April 2, 2014
Missouri University of Science & Technology
Rolla, MO

Goals for the Research

Design for –ilities Handbook

Now with Metrics!

Table of Contents

Acknowledgements

Foreword

Introduction and Motivation
The need for ilities, history, current use

Defining "ilities"
What it can mean, how it can be represented

Organizing Ilities: Semantic Fields
Change type, architecture type, new ability type

Defining Ilities: Semantic Bases
Change-type, architecture-type, and new ability type

"Design for "ility"
Grouped by some scheme (alphabetical or by semantic field)

- "Definition"/description (also using semantic basis)
- Semantic basis example
- Related ilities
- Enablers/principles
- Examples (with and without the ility; light and detailed)
- Metrics (exist, evaluation, valuation)
- Considerations/concerns/challenges/potential confusion

Open Research Questions

Maps and "Tools"
Hierarchies

Glossary

Index

Wouldn't it be great to have this on your desk?

One of the key objectives for this work is to stimulate a new conversation on a theory of ilities

A Linguistic Approach to Ilities

Linguistic meaning vs. speaker meaning

Literal vs. nonliteral

*"Rep. John Mica called on the agency to "reform" and "become...a thinking, risk-based, **flexible** agency that analyzes risks, sets security standards and audits security performance."*

*"ULA says the network augments 'more **robust** and **flexible** execution of Command and Control, Communications Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR)…"*

*To Axe, there is a "clear" need for the K-MAX because operations in the country are "highly dependent on **flexible**, **reliable** and secure logistics…"*

*"Defense Secretary Panetta: "The US joint force will be smaller and it will be leaner. But it will be more **agile**, more **flexible**, ready to deploy quickly, innovative and technologically advanced."*

*… "the Defense Department and the Office of the Director of National Intelligence pledged to foster an industrial base that is **'robust**, competitive, **flexible**, healthy, and delivers **reliable** space capabilities on time and on budget.'"*

- Linguistics is the scientific study of human natural language, including semantics
- Semantics is the study of "meaning" and is a promising area for clarifying the "ilities"
- Meaning arises from interplay of "use" (i.e. speech) and "prescription" (i.e. dictionaries)
- Technical and political leaders are using "ilities" so we need to understand them well enough to ensure systems predictably display these properties

Semantic field
"a group of words with related meanings, for example kinship terms or color terms"
Akmajian et al 2001, p. 587

Quotes from AIAA Daily Launch, 20 Jul 2011 – 13 Feb 2012

Semantic Challenges for Ilities

- Fundamental "ambiguity" in terms
  - Many of these terms are use colloquially and therefore inherit meaning
  - Polysemy – "The property of [a term] having multiple meanings that are semantically related"
    - Flexibility (able to be changed) and flexibility (able to satisfy multiple needs)
  - Synonymy – "The property of multiple terms having similar meaning"
    - Flexibility (able to be changed) and changeability (able to be changed or change itself)
- Problem partly stems from considering one ility at a time

  e.g. flexibility:
  - Saleh, J. H., Mark, G.T., and Jordan, N.C. (2009). "Flexibility: a multi-disciplinary literature review and a research agenda for designing flexible engineering systems." *Journal of Engineering Design.*
  - Nilchiani, R. (2005). Measuring Space Systems Flexibility: A Comprehensive Six-element Framework. *PhD in Aeronautics and Astronautics*, MIT.
  - De Neufville, R. and Scholtes, S. (2011). *Flexibility in Engineering Design,* MIT Press: Cambridge MA.
- Some work done on sets

  e.g. changeability:
  - Fricke and Schulz (2005). "Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout their Entire Lifecycle." *Systems Engineering*
  - Ross, Rhodes and Hastings (2008). "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value." *Systems Engineering*

If challenge can be addressed by looking at sets of ilities, how do we select members of the set?

Ility Set Potential Sources (1)

Academic Opinion: ESD Symposium Committee (2001)

ESD Symp. Cmtee, "ESD Terms and Definitions (ver 12)" *ESD-WP-2002-1*, 2001.

This list was developed by 11 MIT faculty working as a committee

The list is reflective of the interests of the committee members

ID Number	Name
P1	Flexibility
P2	Agility
P3	Robustness
P4	Fail-safe
P5	Adaptability
P6	Scalability
P7	Modularity
P8	Safety
P9	Durability
P10	Sustainability
P11	Quality
P12	Reliability
P13	Repairability
P14	Maintainability

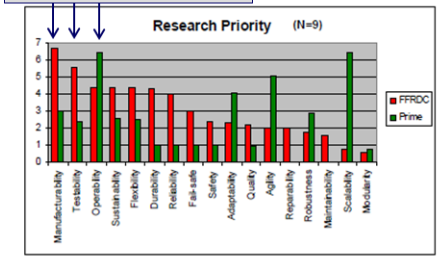
"Requirements of systems, such as flexibility or maintainability, often ending in 'ility;' requirements of systems that are not necessarily part of the fundamental set of functions or constraints" p. 4

Asking Practitioners: Interviews for Ilities Knowledge Gap

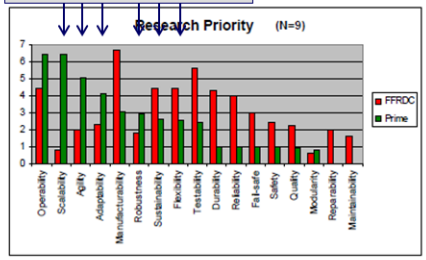
Ross, A.M., "Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration" *Engineering Systems Division*, Ph.D., 2006.

14 ESD-defined ilities + 3 (manufacturability, testability, operability)

Top 3 are non-ESD



Top 6 ESD



Primes are practitioners; FFRDCs are more like academia

From Prime: The top 6 (ESD) ilities research priorities are

1. **Scalability**
2. Agility
3. **Adaptability**
4. **Robustness**
5. Sustainability
6. **Flexibility**

" 'ilities' are not part of the lexicon of even the most successful project managers"
-Select quote from interviews (p. 37)

From Standards

1. Internal and External Quality
   a. Functionality: Suitability, Accuracy, Interoperability, Security
   b. Reliability: Maturity, Fault Tolerance, Recoverability
   c. Usability : Understandability, Learnability, Operability, Attractiveness
   d. Efficiency: Time Behavior, Resource Utilization,
   e. Maintainability: Changeability, Stability, Testability
   f. Portability: Adaptability, Installability, Co-Existence, Replaceability
2. Quality in Use
   a. Effectiveness: Productivity, Safety, Satisfaction

ISO/IEC, 2007; Boegh, 2008

Ility Set Potential Sources (2)

Crowd Sourced: Wikipedia "ilities" List*

accessibility
accountability
accuracy
adaptability
administrability
affordability
agility
auditability
autonomy
availability
credibility
process capabilities
compatibility
composability
configurability
correctness
customizability
debugability
degradability
determinability
demonstrability
dependability
deployability
discoverability
distributability
durability
effectiveness
efficiency
evolvability
extensibility
failure transparency
fault-tolerance
fidelity
flexibility
inspectability
installability
Integrity
interchangeability
interoperability
learnability
maintainability
manageability
mobility
modifiability
modularity
operability
orthogonality
portability
precision
predictability
producibility
provability
recoverability
relevance
reliability
repeatability
reproducibility
resilience
responsiveness
reusability
robustness
safety
scalability
seamlessness
self-sustainability
serviceability (a.k.a.
supportability)
securability
simplicity
stability
standards compliance
survivability
sustainability
tailorability
testability
timeliness
traceability
ubiquity
understandability
upgradability
usability

Within systems engineering, quality attributes are non-functional requirements used to evaluate the performance of a system. These are sometimes named "ilities" after the suffix many of the words share.

Missing from this list: changeability, manufacturability, quality, reconfigurability, versatility

*There were **61** "-ilities" on Wikipedia on April 24, 2008, **71** on March 3, 2010, **78** on February 8, 2011, **79** on February 13, 2012

Nonfunctional Requirements

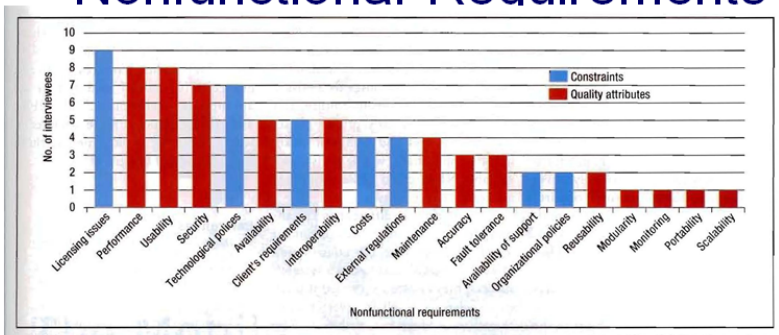


FIGURE 1. Relevance of nonfunctional requirement types from surveyed architects' perspective. This shows the number of interviewees that mentioned each type of NFR as relevant.

Buschmann et al., 2012

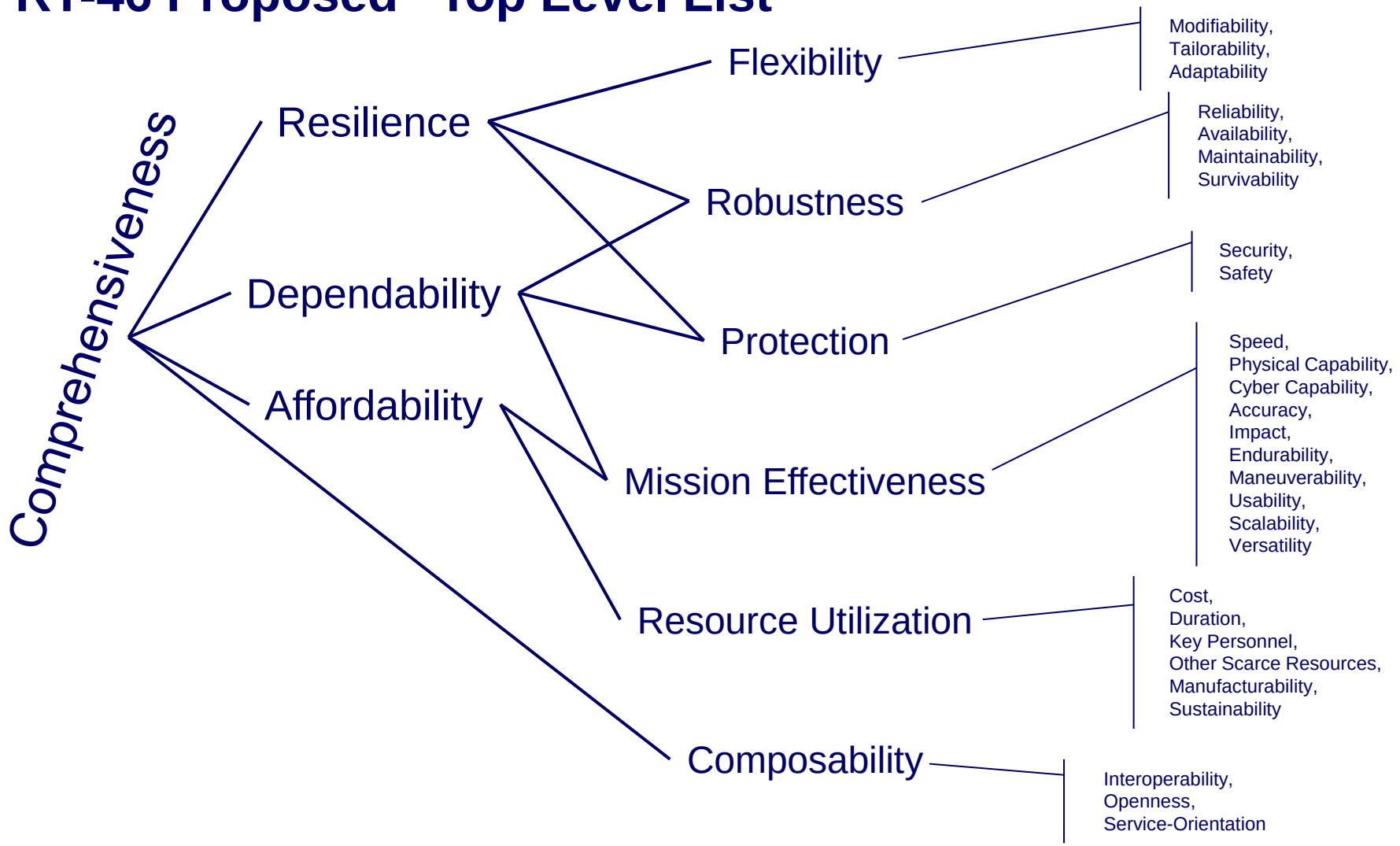
Quality Requirements

Ility	Effect on DoD Operational System
Quality of Service	Stakeholders-satisfactory balance of Performance, Accuracy, Usability, Scalability, Versatility
Performance	Stakeholders-satisfactory balance of desired performance ilities: speed, range, payload, response time, etc.
Accuracy	Closeness to target
Usability	Ease of learning, ease of use, difficulty of misuse
Scalability	Sustainability of system capability across a range of system or environmental scales
Versatility	Range of capabilities provided
Resource Utilization	Smallest percent remaining of distance to system constraint
Cost	Percent remaining of distance to system cost constraint
Duration	Percent remaining of distance to system duration constraint
Key Personnel	Percent remaining of distance to system key personnel constraint
Other Constraints	Percent remaining of distance to other system constraint (size, weight, energy, bandwidth, data storage, etc.)
Protection	Stakeholder loss of value due to natural causes, adversary actions, or compromise of personal information
Safety	Stakeholder loss of value due to natural causes,
Security	Stakeholder loss of value due to adversary actions
Privacy	Stakeholder loss of value due to compromise of personal information
Robustness	Ability of the system to continue to deliver stakeholder-desired other-ility levels
Reliability	Probability that the system will continue to deliver stakeholder-desired other-ility levels
Availability	Fraction of the time that the system will deliver stakeholder-desired other-ility levels
Maintainability	Expected amount of time required to restore system capability
Survivability	Ability of the system to continue to deliver partial stakeholder-desired other-ility levels
Flexibility	Expected ROI in total ownership cost from investments in Modifiability, Tailorability, and Adaptability
Modifiability	Expected ROI in total ownership cost from investments in Modifiability
Tailorability	Expected ROI in total ownership cost from investments in Tailorability, including Extendibility
Adaptability	Expected ROI in total ownership cost from investments in Adaptability
Composability	Expected ROI in total ownership cost from investments in Interoperability, Openness, and Service-Orientation
Interoperability	Expected ROI in total ownership cost from investments in Interoperability, including Portability
Openness	Expected ROI in total ownership cost from investments in Openness, including Standards Compliance
Service-Orientation	Expected ROI in total ownership cost from investments in Service-Orientation
Composite ilities	
Comprehensiveness	All of the above
Dependability	Quality of Service, Protection, Robustness
Resilience	Protection, Robustness, Flexibility
Affordability	Quality of Service, Resource Utilization

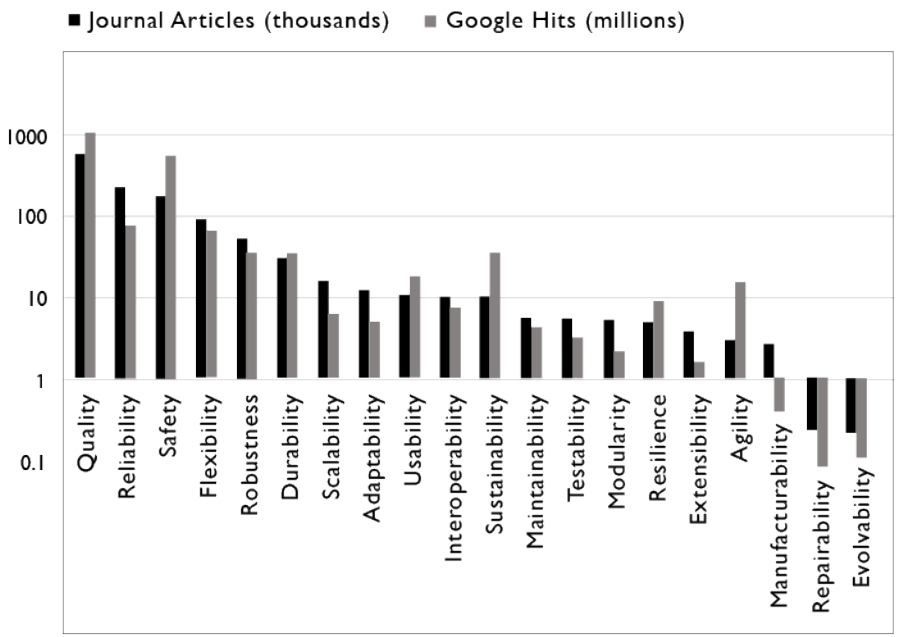
USC technical report, "The Nature of Information System Dependability: A Stakeholder/Value Approach." as USC-CSE-2004-520 at http://csse.usc.edu/csse/TECHRPTS/2004/2004_main.html

Ility Set Potential Sources (3)

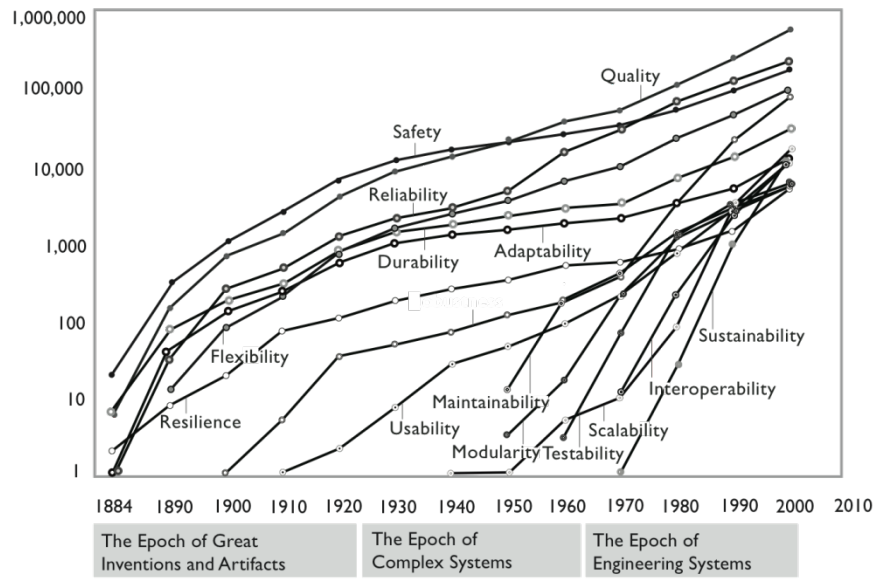
RT-46 Proposed "Top Level List"



Descriptive Approach: Ilities Mentioned in Literature



Frequency of ilities mentioned in journal articles and Google hits (de Weck, Roos, Magee 2011, p. 67)

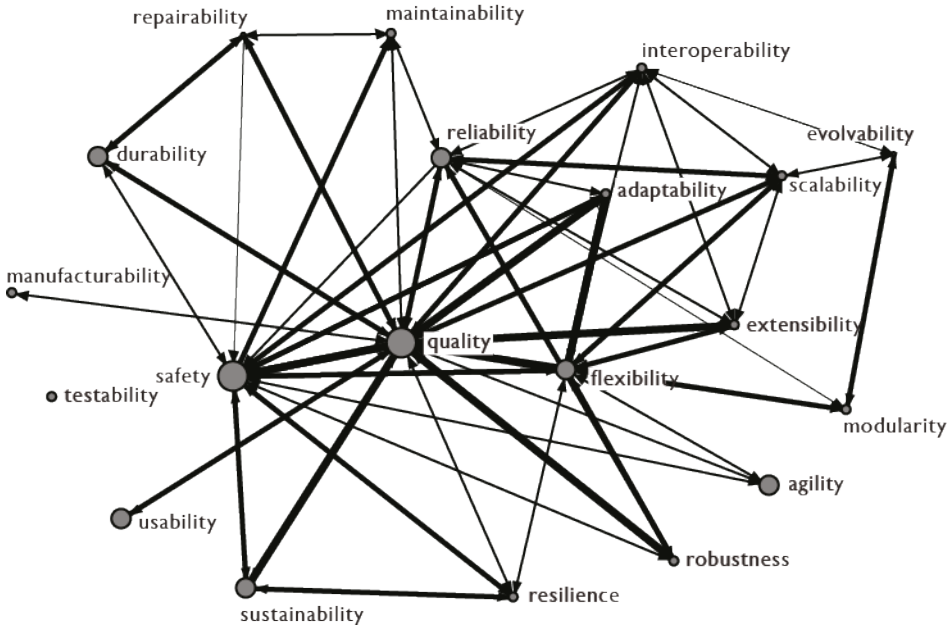


Frequency of ilities mentioned in literature across time (de Weck, Roos, Magee 2011, p. 69)

- This work shows the frequency of written usage of various ilities in a snapshot, and as a function of time (Q: Does frequency correlate with importance or relevance?)
- The challenge is that these results do not capture relationships amongst ilities or reasons behind the frequencies

de Weck, O.L., Ross, A.M., and Rhodes, D.H., "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)," 3rd International Conference on Engineering Systems, TU Delft, the Netherlands, June 2012.

Descriptive Approach: Derivation of Ilities Relationships



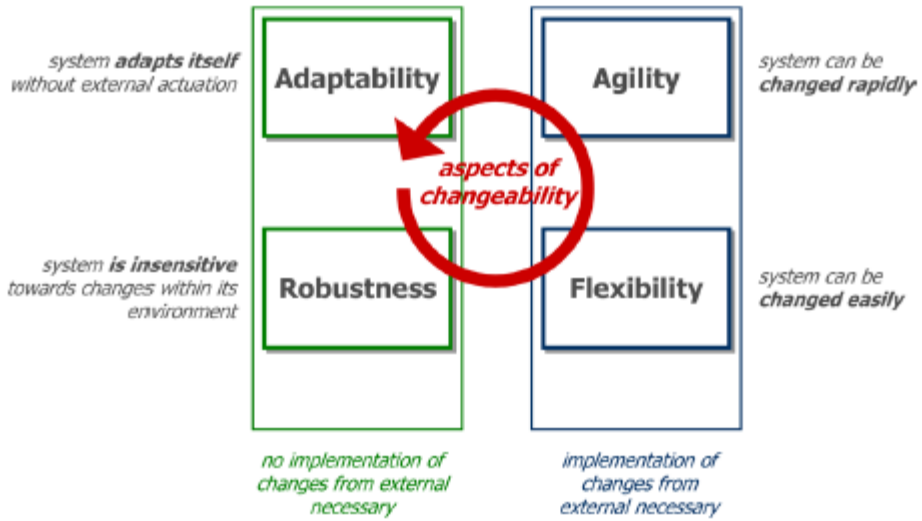
Ility term co-occurrence in the literature with implied dependence (de Weck, Roos, Magee 2011, p. 83)

- This work shows the co-occurrence of written usage of various ilities in a snapshot, (Q: What is the nature of the co-occurrence? Complementary? Substitution? +/-?)
- These results are a good first step for proposing deeper inquiry into the nature of the relationships amongst ilities, but why this particular list of ilities?

de Weck, O.L., Ross, A.M., and Rhodes, D.H., "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)," 3rd International Conference on Engineering Systems, TU Delft, the Netherlands, June 2012.

Prescriptive Approach: Fricke and Schulz (2005)

"Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout their Entire Lifecycle" *Systems Engineering*, Vol. 8, No. 4, 2005.



This work is based on the authors' PhD research and experiences in German Product Development (e.g. BMW)

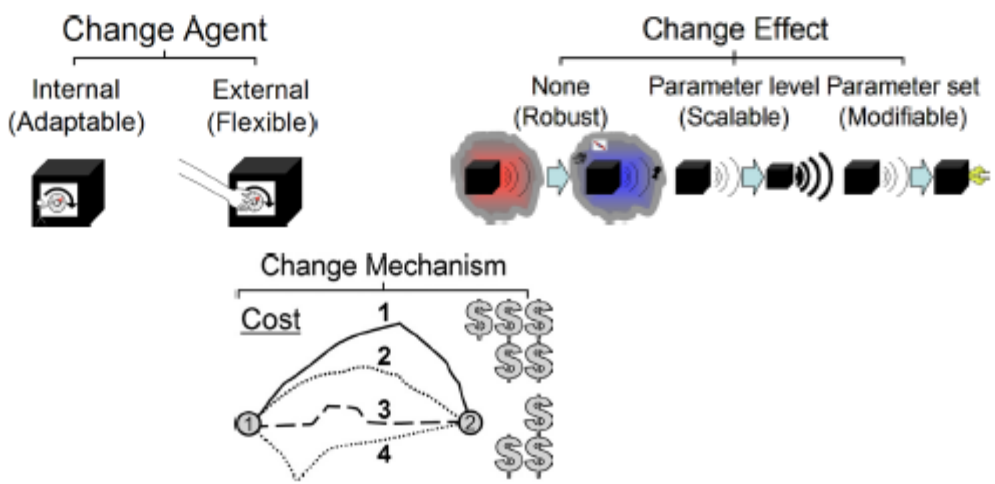
Explicit ilities: changeability = {adaptability, robustness, agility, flexibility}, architecture principles = {simplicity, independence, modularity, integrability, autonomy, scalability, non-hierarchy, decentralization, redundancy

Implicit ilities: evolvability,

Related, but distinct: platforming

Prescriptive Approach: Ross, Rhodes and Hastings (2008)

"Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining System Lifecycle Value" *Systems Engineering*, Vol. 11, No. 3, 2008.



This work attempted to build upon Fricke and Schulz, ESD, and others to create a more rigorous basis for specifying and quantifying several ilities

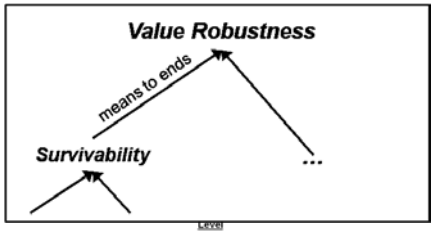
Explicit ilities: changeability = {adaptability, flexibility, scalability, modifiability, robustness}

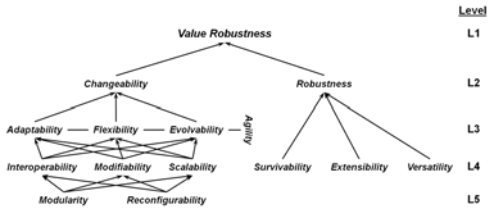
The system shall be	_____	_____	in	_____	for less than	_____.
	(change agent type) flexibly or adaptably	(change effects) scalable, modifiable		(system parameter) with range		(resources)

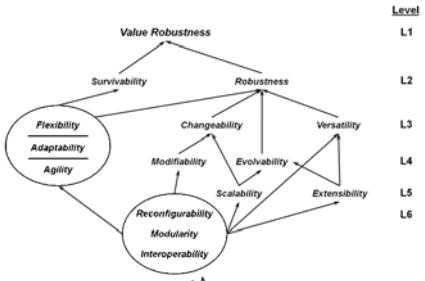
Verifiable changeability statement

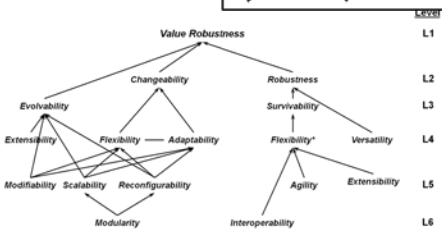
Means-Ends Hierarchy from Prescriptive Definitions

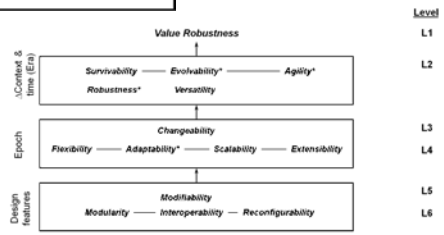
Using SEAri definitions, groups constructed means-ends hierarchies from a given set of ilities

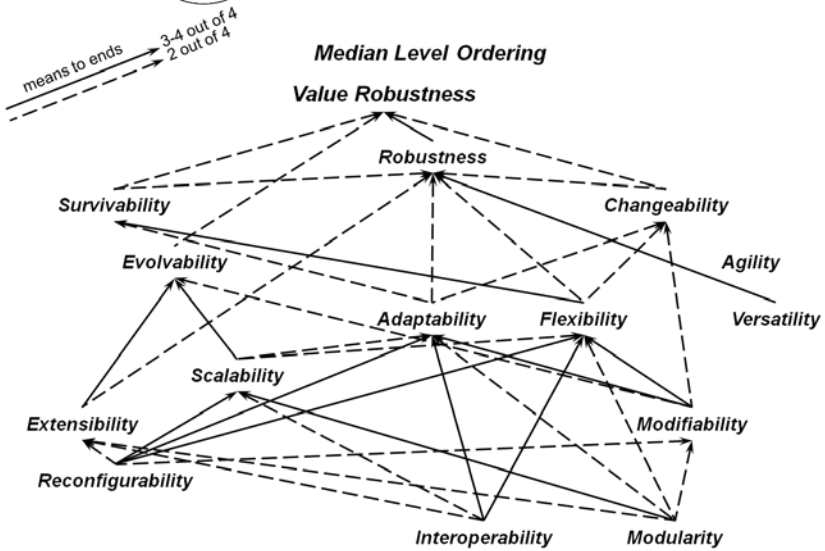












Lack of consensus and emergent “depth” criterion suggested more than “means-ends” relationships exist among ilities

de Weck, O.L., Ross, A.M., and Rhodes, D.H., "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)," 3rd International Conference on Engineering Systems, TU Delft, the Netherlands, June 2012.

Many Student Theses on Ilities

Beesemyer, J.C., *Empirically Characterizing **Evolvability** and **Changeability** in Engineering Systems*, SM, Aero/Astro, MIT, June 2012.
Fitzgerald, M.E., *Managing Uncertainty in Systems with a Valuation Approach for Strategic **Changeability***, SM, Aero/Astro, MIT, June 2012.
Friedel, A., *Investigating the Management of Uncertainty in Product **Platform** Lifecycles*, Dipl., TUM, January 2011.
Fulcoly, D.O., *A Normative Approach to Designing for Evolvability: Methods and Metrics for Considering **Evolvability** in Systems Engineering*, SM, Aero/Astro, MIT, June 2012.
Koo, C.K.K., *Investigating Army Systems and Systems of Systems for **Value Robustness***, SM, SDM, MIT, February 2010.
Mekdeci, B., *Managing the Impact of Change through **Survivability** and **Pliability** to Achieve **Viable** Systems of Systems*, PhD, ESD, MIT, February 2013.
Nilchiani, R.N., *Measuring the Value of Space Systems **Flexibility**: A Comprehensive Six-element Framework*, PhD, Aero/Astro, MIT, September 2005.
Richards, M.G., *Multi-Attribute Tradespace Exploration for **Survivability***, PhD, ESD, MIT, June 2009.
Richards, M.G., *On-Orbit **Serviceability** of Space System Architectures*, SMx2, Aero/Astro and TPP, MIT, June 2006.
Roark, III, H.H., *Value Centric Approach to Target System **Modularization** Using Multi-Attribute Tradespace Exploration and Network Measures of Component Modularity*, SM, SDM, MIT, June 2012.
Roberts, C.J., *Architecting **Evolutionary** Strategies using Spiral Development for Space Based Radar*, SM, TPP, MIT, June 2003.
Ross, A.M., *Managing Unarticulated Value: **Changeability** in Multi-Attribute Tradespace Exploration*, PhD, ESD, MIT, June 2006
Saleh, J.H., *Weaving Time into System Architecture: New Perspectives on **Flexibility***, Spacecraft Design Lifetime, and On-orbit Servicing, PhD, Aero/Astro, MIT, June 2002.
Shah, N.B., ***Modularity** as an Enabler for **Evolutionary** Acquisition*, SM, Aero/Astro, MIT, June 2004.
Viscito, L., *Quantifying **Flexibility** in the Operationally Responsive Space Paradigm*, SM, Aero/Astro, MIT, June 2009.
Wilds, J.M., *A Methodology for Identifying **Flexible** Design Opportunities*, SM, TPP and Aero/Astro, MIT, September 2008.

Have led to the following observation:

There are at least three types of ilities

1. "Change-related"
2. "Architecture-related"
3. New ability related

These are just some of the theses from our group explicitly addressing ilities… their literature reviews uncovered many, many more theses outside of MIT, in addition to MIT ESD's recent theses

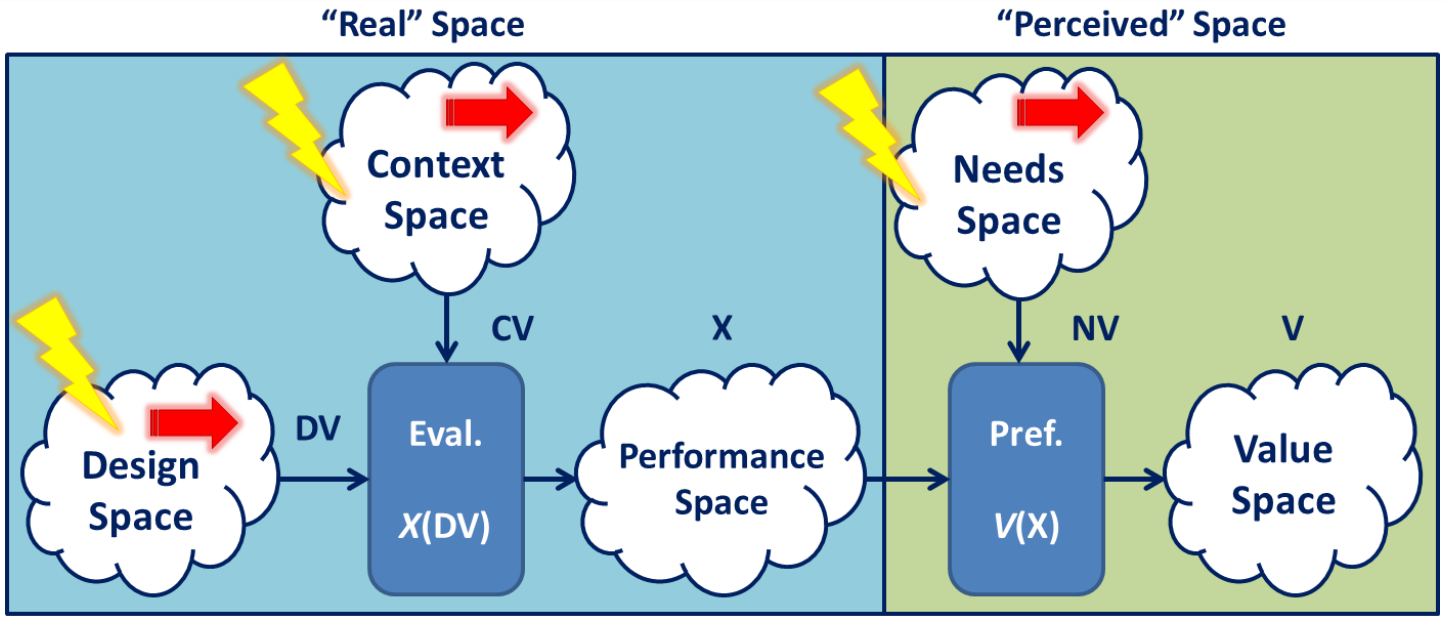
Change-Related Ilities	Architecture-Related Ilities
Adaptability	Accessability
Agility	Controllability
Changeability	Decentralization
Evolvability	Independence
Extensibility	Interoperability
Flexibility	Integrability
Modifiability	Modularity
Reconfigurability	Protectability
Scalability	Readability
Survivability	Redundancy
Versatility	Simplicity

For now, we will focus on "change-related" ilities

Our working hypothesis* is that "architecture-related" ilities are enablers for "change-related" ilities

* this is supported by our research on "design principles" (e.g. for survivability, evolvability, etc.)

Theoretical Framework for the System Value Design Problem



Real Space
Abstraction
Aspect
Phase
Context

Perturbation	Change
Disturbance	Parameter
	Agent
Shift	Effect
	Destination

Perceived Space
Benefit
Cost
Reaction
Span

Beesemyer, J.C., *Empirically Characterizing Evolvability and Changeability in Engineering Systems*, Master of Science Thesis, Aeronautics and Astronautics, MIT, June 2012.

10-D Semantic Basis

- Drawing analogy from linear algebra, a basis describes a spanning set that defines a space
  - Can we decompose the “change-related” semantic field into distinct basis vectors?
  - This would enable direct representation of how ilities are related to each other
  - It would also help to avoid definition wars one ility at a time
- Dimensions can be used to differentiate between considered ilities
- Basis can be used to generate change statements for description or prescription

Dimension	Name	Source
1	Perturbation	Chapter 3
2	Context	Chapter 3
3	Phase (Change Mechanism)	Ross and Rhodes 2011
4	Change Agent	Ross, Rhodes, Hastings 2008
5	Change Effect	Ross, Rhodes, Hastings 2008
6	Parameter (Change Effect)	Ross, Rhodes, Hastings 2008
7	End States (Change Mechanism)	Ross and Rhodes 2011
8	Aspect	Chapter 3
9	Abstraction	Chapter 3
10	Valuable	Ross and Rhodes 2011

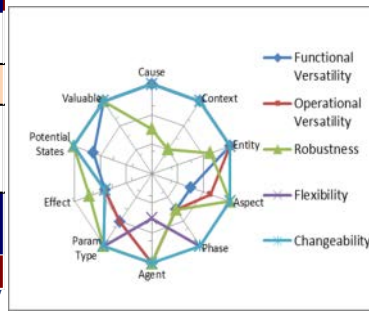
Prescriptive Semantic Basis for Change-type Ilities

In response to “cause” in “context”, desire “agent” to make some “change” in “system” that is “valuable”

Cause	Context	Phase	Agent	Change	System	Valuable

In response to "perturbation" in “context”, desire “agent” to make some "effect" to the "parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in “reaction”, “span”, “cost” and "benefits"

Perturbation	Context	Phase	Agent	Effect	Parameter (Type)	Destination	Aspect	Abstraction	Reaction	Span	Cost	Benefit
					"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
1	2	3	4	5	6	7	8	9	10			



Generalizing the Change-related Statement: A Prescriptive Basis

The system shall be	_____	_____	in	_____	for less than	_____.
	(change agent type)	(change effects)		(system parameter)		(resources)
	flexibly or adaptably	scalable, modifiable		with range		

(From Ross, Rhodes, and Hastings 2008)

Prescriptive Semantic Basis for Change-related Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	System			Agent	Change				Valuable			
							(choose one)						
Why	**Where**	**What**	**What**	**When**	**Who**	**What**	**What**	**What**	**What**	**When**	**When**	**For What**	**For What**
Cause	Context	Entity	Aspect	Phase	Agent	Param Change Type	Effect (Scale)	Effect (Amount)	Potential States	Timing	Span	Resources	Benefit
perturbation	specificity	abstraction	aspect	LC phase	executes	param type	level	set	target range	reaction	duration	cost	utility
disturbance	circumstantial	architecture	form	pre-ops	internal	level	bigger	more	one	sooner	shorter	more	more
shift	general	design	function	ops	external	set	smaller	less	few	later	longer	less	less
none	any	system	operations	inter-LC	either	any	not-same	not-same	many	always	same	same	same
any		any	any	any	none		same	same	any	any	any	any	any
					any		any	any					
1	2	3	4	5	6	7	8		9	10			

10 category basis for specifying "change-related" ilities

Using the Basis to Map Ility Labels Early Application

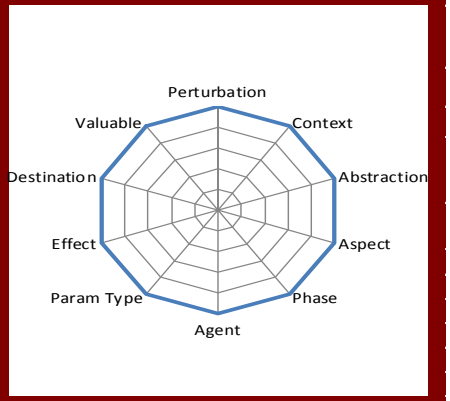
Prescriptive Semantic Basis for Change-type Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	Phase	Agent	Change	System	Valuable

In response to "perturbation" in "context", desire "agent" to make some "effect" to the "system parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Effect	Parameter (Type)	Destination	Aspect	Abstraction	Reaction	Span	Cost	Benefit
					"parameter"	*"state"*			*"threshold"*	*"threshold"*	*"threshold"*	*"threshold"*
disturbance	circumstantial	pre-ops	internal	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	any	any	any	any	any
			any	any								



													Ility Name	Publish
shift	any	any	any	same	"Value"	few	any	any	any	any	any	any	Value Robustness	
disturbance	any	any	any	same	"Value"	few	any	any	any	any	any	any	Value Survivability	
shift	any	ops	any	same	any	few	any	any	any	any	any	any	Robustness	
shift	any	ops	none	same	level	few	form	system	any	any	any	any	Classical Passive Robustness	
disturbance	any	ops	any	same	any	few	any	any	any	any	any	any	Survivability	
any	any	any	either	not-same	any	any	any	any	any	any	any	any	Changeability	
shift	general	inter-LC	any	not-same	any	any	any	architecture	any	any	any	any	Evolvability	
any	any	any	internal	not-same	any	any	any	any	any	any	any	any	Adaptability	
any	any	any	external	not-same	any	any	any	any	any	any	any	any	Flexibility	
any	any	any	any	not-same	level	any	any	any	any	any	any	any	Scalability	
any	any	any	any	not-same	set	any	any	any	any	any	any	any	Modifiability	
any	any	ops	either	increase	set	any	any	any	any	any	any	any	Extensibility	
any	any	any	any	not-same	any	any	any	any	any	shorter	any	any	Agility	
any	any	any	any	not-same	any	any	any	any	sooner	any	any	any	Reactivity	
any	any	ops	any	same	"Element set"	one	form	any	any	any	any	any	Form Reconfigurability	
any	any	ops	any	not-same	"Link set"	any	form	any	any	any	any	any		
any	any	ops	any	same	"Element set"	one	operations	any	any	any	any	any	Operational Reconfigurability	
any	any	ops	any	not-same	"Order set"	any	operations	any	any	any	any	any		
any	any	ops	any	same	any	one	form	any	any	any	any	any	Functional Versatility	
any	any	ops	any	not-same	set	few	function	any	any	any	any	any		
any	any	ops	any	same	any	one	form	any	any	any	any	any	Operational Versatility	
any	any	ops	any	not-same	set	few	operations	any	any	any	any	any		
any	any	ops	any	same	any	one	fnct/ops	any	any	any	any	any	Exchangeability	
any	any	ops	any	not-same	set	few	form	any	any	any	any	any		

Using the Basis Descriptively: **iPhone App Example**

Change #	21
System	iPhone 4
Domain	Consumer Electronics
Preliminary Ility Type	Flexible, Extensible
Mapped Ility Type	Flexible, Modifiable, Versatile, Value Robust, Changeable, Extensible
Parameter	Applications Set
Path Enabler	AppStore, Software, Internet
Why (Cause)	Shift
Where (Context)	General
What (Entity)	System
What (aspect)	Function
When (Phase)	E (ops)
Who (Agent)	External
What (Param. Type)	Set
What effect (Scale)	--
What effect (Amount)	More
What (Potential States)	Many
When (Timing)	Always
When (Span)	
For what (Resources)	Cheaper
For what (Benefit)	More

Change Description	In response to user preference change in mobile phone capabilities desire user to change/add an application to the iPhone to improve functionality. The App store allows users to choose from thousands of apps that add functionality and the ability to personalize the iPhone to be used in many different situations

Explanation

Perturbation	User preference shift
Specificity	A general change
Abstraction	Change in system, within design
Aspect	The app is determined to be part of the functions
LC Phase	During operations of the phone
Executes	User
Param. Type	In the set of applications
Level	--
Set	Adding a parameter to the set
Target Range	Many different configurations
Start	Easy to implement whenever you want to
Duration	
Costs	Easier to add apps than most competitor's phones w/ appstore
Utility	More functionality

Image

Manufacturer	Apple
Production	Mass Produced
Potential Variety	Many
Actual Variety	Many
Population	Many

Preliminary Information

The iPhone 4 is a touchscreen smartphone developed by Apple. It is the fourth generation of iPhone, and successor to the iPhone 3GS. It is particularly marketed for video calling (marketed by Apple as FaceTime), consumption of media such as books and periodicals, movies, music, and games, and for general web and e-mail access. It was announced on June 7, 2010, at the WWDC 2010 held at the Moscone Center, San Francisco,[6] and was released on June 24, 2010, in the United States, the United Kingdom, France, Germany and

Contact Info

Adaptable	Robust	Agile	Interoperable
Flexible	Classically Robust	Survivable	Extensible
Scalable	Versatile	Reconfigurable	Value Robust
Modifiable	Evolvable	Modular	Changeable

Adaptability	
Flexibility	1
Scalability	
Modifiability	1
Robustness	
Classical Robustness	
Versatility	1
Evolvability	
Agility	
Survivability	
Reconfigurability	
Modularity	
Interoperability	
Extensibility	1
Value Robustness	1
Changeability	1

Source: http://www.apple.com/iphone

Cost:

Cost Context:

Related Change:

Multiple ility term labels apply to the same change description

Using the Basis Descriptively: **John Deere Engine Example**

Change #	54
System	PowerTech PSX Interim Tier 4/Stage IIIB
Domain	Commercial Vehicle
Preliminary Ility Type	Evolvable, Flexible, Changeable
Mapped Ility Type	Changeability, Flexibility, Modifiability, Evolvability, Value Robustness
Parameter	PM reducing method
Path Enabler	Diesel Oxidation Catalyst, Diesel Particulate Filter, Exhaust Temperature Management
Why (Cause)	Shift
Where (Context)	General
What (Entity)	Architecture
What (aspect)	Form
When (Phase)	Inter-LC
Who (Agent)	External
What (Param. Type)	Set
What effect (Scale)	--
What effect (Amount)	Not-Same
What (Potential States)	One
When (Timing)	
When (Span)	
For what (Resources)	
For what (Benefit)	More

Change Description	In response to "more stringent emissions regulations" by the EPA in "off-road vehicles", desire "John Deere" to add "catalyzed exhaust filter" in new "PowerTech PSX Engines" in order to minimize Particulate Matter (PM) that is "within code". The move to Interim Tier 4/Stage III B emissions regulations is the most significant to date. The regulations call for a 90 percent reduction in particulate matter (PM) along with a 50 percent drop in nitrogen oxides (NOx)

Explanation

Perturbation	Shift in emissions regulations
Specificity	Applies to all off-road vehicles
Abstraction	Outside design pliable range
Aspect	Change in physical component
LC Phase	Between lifecycles
Executes	John Deere engineers
Param. Type	Change in design method
Level	--
Set	Different design method
Target Range	Only one filter option
Start	--
Duration	--
Costs	--
Utility	Less Particulate Matter

Image

Manufacturer	John Deere
Production	Mass Produced
Potential Variety	Few
Actual Variety	Few
Population	Many
Avg Lifecycle	5-15 Years

Preliminary Information

PowerTech PSX 9.0L engines feature series turbochargers that deliver more boost pressure than single turbocharging configurations which results in higher power density, improved low-speed torque, and improved high altitude operation. These high-performing engines use cooled exhaust gas recirculation (EGR). This technology cools and mixes measured amounts of cooled gas with incoming fresh air to lower peak combustion temperatures, thereby reducing NOx. These engines utilize a catalyzed exhaust filter that contains a

Contact Info

Adaptable	Robust	Agile	Optional
Flexible	Classically Robust	Survivable	Extensible
Scalable	Operationally Versatile	Reconfigurable	**Value Robust**
Modifiable	Functionally Versatile	**Evolvable**	**Changeable**

Adaptability	0
Flexibility	1
Scalability	0
Modifiability	1
Robustness	0
Classical Robustness	0
Operational Versatility	0
Functional Versatility	0
Agility	0
Survivability	0
Exchangeability	0
Evolvability	1
Reconfigurability	0
Extensibility	0
Value Robustness	1
Changeability	1

Source	http://www.deere.com/en_US/rg/emissionsinfo/tier4/index.html http://www.deere.com/en_US/rg/servicesupport/publications_and manuals/engine_literature/interim_tier4_broch
Relative Change Cost	Less
Relative Change Time	Lifecycle
Related Change	
Cluster ID	1
Interoperability	
Modularity	
ClusterOfInterest	6

Multiple ility term labels apply to the same change description

Basis-Formulated Change Statements

- Simple statement that represents only the change and system information:
- ***Desire some "change" in "system."***
  - **e.g. Desire hospital power source to switch from power grid to gas generator.**

- ***In response to "perturbation" in "context", desire some "parameter change" in "system" that is "valuable."***
  - **e.g. In response to a power outage in a severe winter storm, desire power source to be switched from grid to generator in the hospital to maintain operation of life-critical equipment lighting.**

- More information, will yield a higher level of differentiation amongst ilities.
- As more dimensions are expressed, more detail about each change can express higher variation.
- First statement may only relate to system *changeability*, where the second statement may relate to *survivability* (since perturbation is now defined) as well.
- The most complete statement, using all ten categories and the four sub categories in the value section gives the most complete change requirement:

- ***In response to "perturbation" in "context", desire "agent" to make some "effect" to the "parameter" in the "aspect" of the "abstraction" during "phase" with "destination(s)" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits."***
  - **e.g. In response to a power outage in a severe winter storm, desire power control box to automatically switch the power source from grid to generator in the operations of the county hospital during daily ops with destination of full generator use that is valuable with respect to reacting within 1 minute of perturbation, change spanning less than 2 minutes, without losing any life-critical operations or equipment in order to maintain hospital care.**

Example Insights Using the Basis: **Reactivity / Agility / Responsiveness**

Reactivity

The ability of a system to react in a timely fashion

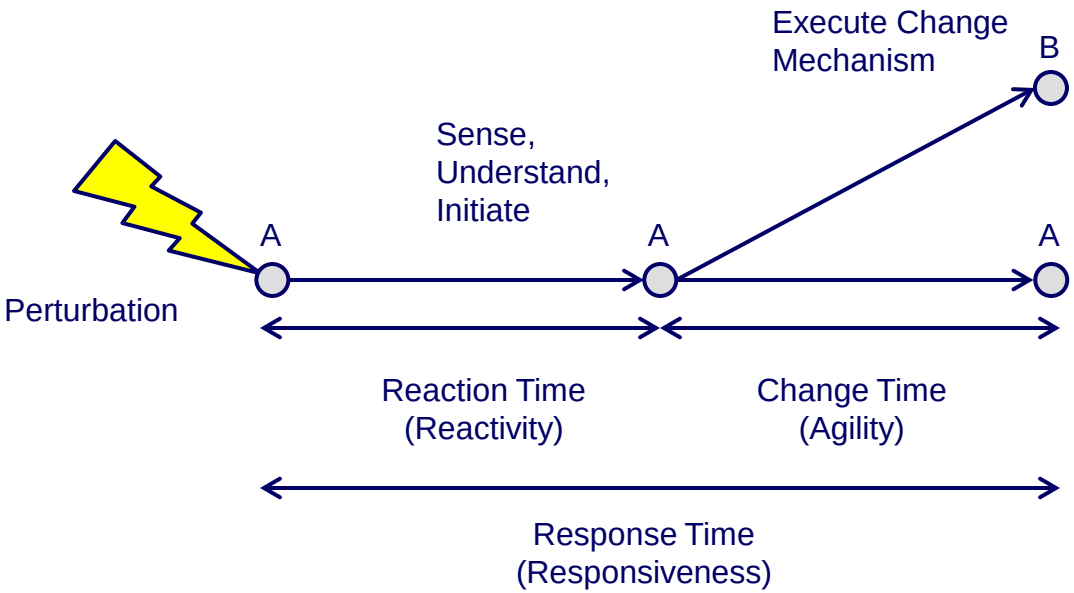
Agility

The ability of a system to change in a timely fashion

Responsiveness

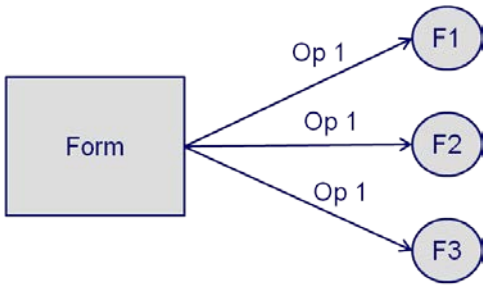
The ability of a system to respond (react and change) in a timely fashion

What potential states	**When** timing	**When** span	**For what** resources	**For what** benefit
target range	reaction	duration	cost	utility
one	sooner	shorter	cheaper	more
few	later	longer	expensiver	less
many	always	same	same	same
any	any	any	any	any



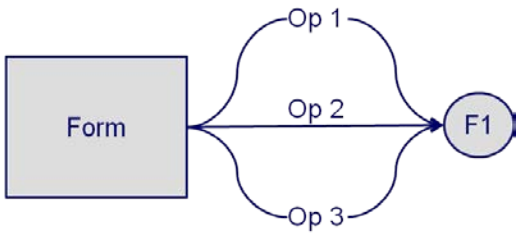
Example Insights Using the Basis: **Versatility**

- Discussion in system 'aspect' lead to clarification on types of versatility
- Different "flavors" of versatility
  - The ability of a system to satisfy diverse needs for the system without having to change form (measure of latent value).
- Can accomplish this two different ways
  - Change in function equates to functional versatility
  - Change in operations equates to operational versatility
- Possibly where there are many regulations of operations are dynamic (rules of engagement)



Functional Versatility

Achieving multiple functions with similar form and operations.

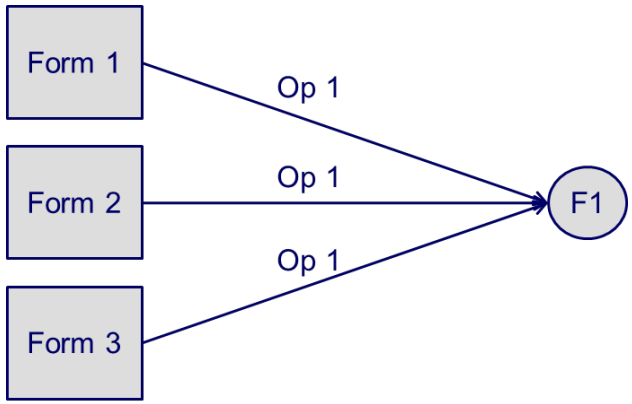


Operational Versatility

Achieving similar function with multiple operations, but similar form.

Example Insights Using the Basis: **Substitutability**

- Discovered as the counterpart to versatility in the ility framework
  - Framework allows clarification and exploration of system properties
- Substitutability then is using multiple forms to accomplish the same function with same operations
  - Example would be components in a desktop computer (cd drives, hard drives, monitors, keyboards)



Substitutability

Using multiple forms to achieve similar function using similar operations.

Definition Re-Examination

Value Robustness	ability of a system to maintain value delivery in spite of changes in contexts or needs
	ability of an entity to maintain value delivery in spite of **shifts** in contexts, needs, or **design**
Robustness	ability of a system to maintain its level and set of specification parameters in the context of changing system external and internal forces
	ability of an entity to maintain a specified parameter during **ops** in spite of **shifts** in context, needs, or **design**
Changeability	ability of a system to alter its form, and consequently possibly its function , or operations, at an acceptable level of resource expenditure
	ability of an entity to alter form, function, or ops
Flexibility	ability of a system to be changed by a system-external change agent with intent
	ability of an entity to be changed by a system-external change **agent**
Adaptability	ability of a system to be changed by a system-internal change agent with intent
	ability of an entity to be change by a system-external change **agent**
Evolvability	ability of a architecture to be inherited and changed across generations (over time)
	ability of an architecture to be **changed** between generations in response to general shifts in context or needs
Survivability	ability of a system to minimize the impact of a finite duration disturbance on value delivery
	ability of an entity to maintain a **specified parameter** during **ops** in spite of disturbances in **context, needs, or design**
Versatility	ability of a system to satisfy diverse needs for the system without having to change form (measure of latent value)
	ability of an entity to change its **set of functions** or **operations** while maintaining original form during **ops**
Scalability	ability of a system to change the current level of a system specification parameter
	ability of an entity to change the level of a parameter
Modifiability	ability of a system to change the current set of system specification parameters
	ability of an entity to change the set of a parameter
Reconfigurability	ability of a system to change its configuration (component arrangement and links)
	ability of an entity to change the **link set of form/operations while maintaining the original element set of** [illegible]**ng ops**
	[illegible] in a timely fashion
	[illegible] in a shorter time **span** with respect to a **threshold value**
Extensibility	ability of a system to accommodate new features after design
	ability of an entity to increase **a parameter set during ops** by internal or external change agents

Basis was used to audit definitions to uncover implicit assumptions

14-D Prescriptive Semantic Basis

**key change*: addition of “impetus” and “outcome” rather than just “outcome” (triggered by verbose versatility in 10D)

Prescriptive Semantic Basis for Change-type Ilities

In response to “cause” in “context”, desire “agent” to make some “change” in “system” that is “valuable”

Cause	Context	Phase	Agent	Impetus Change	System	Outcome Change	System	Valuable

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the design "parameter" with "destination(s)" in the "aspect" to have an "effect" to the outcome "parameter" with "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in “reaction”, “span”, “cost” and "benefits"

Perturbation	Context	Phase	Agent	Impetus				Outcome				Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Destination	Aspect	Effect	Parameter	Destination	Aspect					
					"parameter"	"state"			"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
1	2	3	4	5	6	7	8	9	10	11	12	13	14			

Prescriptive Semantic Basis for Change-type Ilities

In response to “cause” in “context”, desire “agent” to make some “change” in “system” that is “valuable”

Cause	Context	Phase	Agent	Impetus Change	System	Outcome Change	System	Valuable

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the design "parameter" with "destination(s)" in the "aspect" to have an "effect" to the outcome "parameter" with "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in “reaction”, “span”, “cost” and "benefits"

Perturbation	Context	Phase	Agent	Impetus				Outcome				Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Destination	Aspect	Effect	Parameter	Destination	Aspect					
					"parameter"	"state"			"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
disturbance	circumstantia	pre-ops	internal	increase	level	one	form	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	same		any	any	any	any	any	any	any
			any	any				any								

Reconciling Challenges in 10-D Version: 14-D Semantic Basis

Prescriptive Semantic Basis for Change-type Ilities

In response to "cause" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Cause	Context	Phase	Agent	Impetus Change	System	Outcome Change	System	Valuable

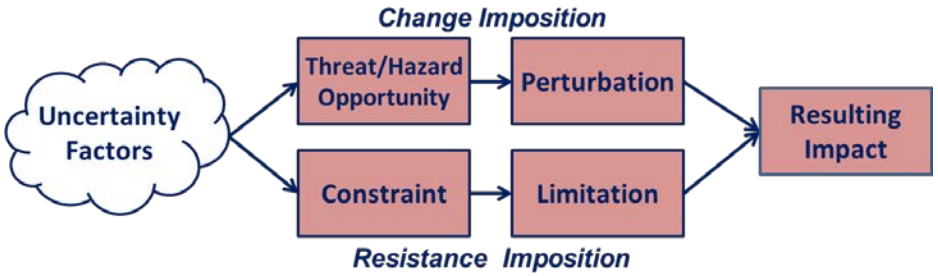
In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the design "parameter" with "destination(s)" in the "aspect" to have an "effect" to the outcome "parameter" with "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Impetus				Outcome				Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Destination	Aspect	Effect	Parameter	Destination	Aspect					
					"parameter"	"state"			"parameter"	"state"			"threshold"	"threshold"	"threshold"	"threshold"
disturbance	circumstantia	pre-ops	internal	increase	level	one	form	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	same		any	any	any	any	any	any	any
			any	any				any								
shift		ops						same	"Value"	few						
disturbance		ops						same	"Value"	few						
shift		ops						same		few						
shift		ops		not-same				same		few						
shift		ops		same		few		same		few						
shift		ops	none	same		few		same	level	few	form	system				
disturbance		ops						same		few						
			either	not-same				not-same								
shift	general	inter-LC		not-same				not-same				architecture				
			internal	not-same				not-same								
			external	not-same				not-same								
				not-same				not-same	level							
				not-same				not-same	set							
		ops	either	not-same				increase	set							
				not-same				not-same	any					shorter		
				not-same				not-same	any				sooner			
		ops		same	"Element set"	one	form	not-same	"Link set"		form					
		ops		same	"Element set"	one	operations	not-same	"Order set"		operations					
		ops		same		one	form/ops	not-same	set	few/many						
		ops		same		one	form/ops	not-same	set	few/many	function					
		ops		same		one	form/fnct	not-same	set	few/many	operations					
		ops		same		one	fnct/ops	not-same	set	few/many	form					

Ility Label
Value Robustness
Value Survivability
Robustness
Active Robustness
Passive Robustness
Classical Passive Robustness
Survivability
Changeability
Evolvability
Adaptability
Flexibility
Scalability
Modifiability
Extensibility
Agility
Reactivity
Form Reconfigurability
Operational Reconfigurability
Versatility
Functional Versatility
Operational Versatility
Substitutability

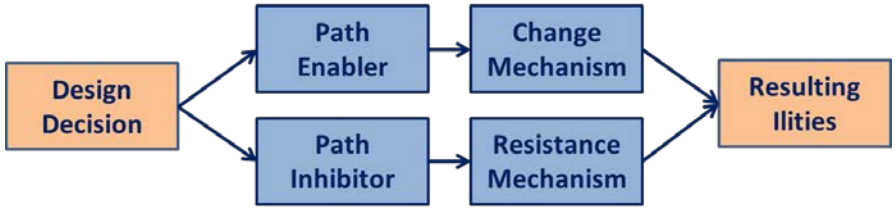
Ilities as Responses to Uncertainties

Uncertainties

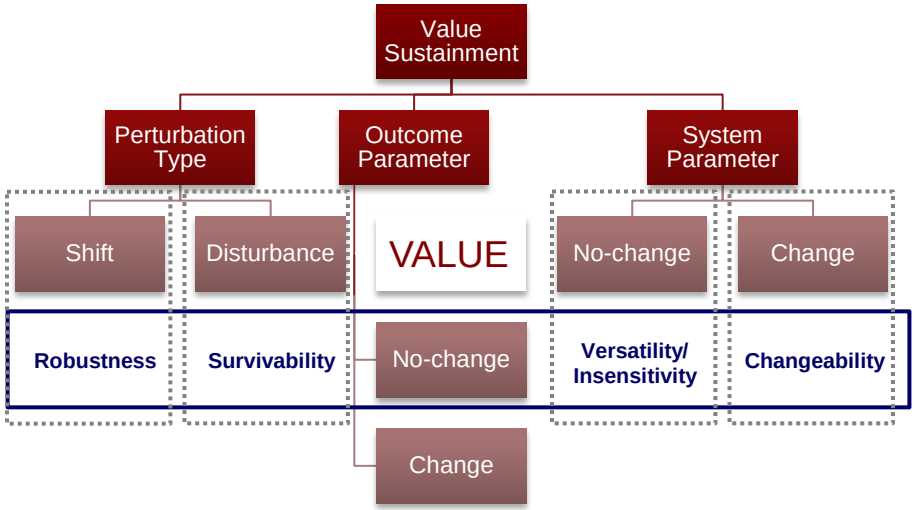


Perturbations and limitations impact value

Responses

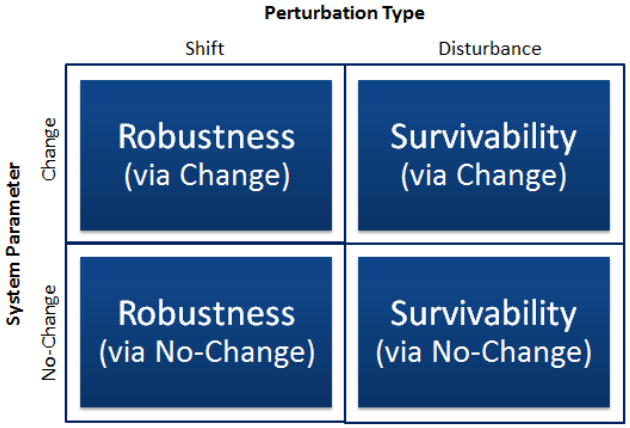


Changes and resistances maintain value



Suppose we want to maintain value
(i.e. no-change in outcome parameter *value*)

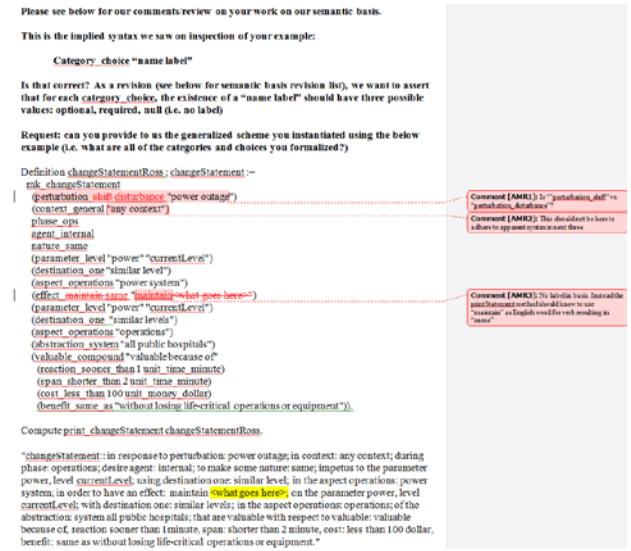
There are four high level ility responses



Having a basis allows us to quickly derive responses

Work with UVA

- In addition to iterating internally, we've been working with UVA on refining the basis
- UVA formalized the basis in Coq by specifying:
  - Abstract syntax
  - Pretty printing
  - A type assignment function that assigns zero or more ility term labels to a given change statement
- Feedback from UVA revealed some implicit assumptions in the basis
- Also triggered realization of different use cases for the basis that should be explicated



Please see below for our comments/review on your work on our semantic basis.

This is the implied syntax we saw on inspection of your example:

Category_choice "name label"

Is that correct? As a revision (see below for semantic basis revision list), we want to assert that for each category_choice, the existence of a "name label" should have three possible values: optional, required, null (i.e. no label)

Request: can you provide to us the generalized scheme you instantiated using the below example (i.e. what are all of the categories and choices you formalized?)

```
Definition changeStatementRoss : changeStatement :=
mk_changeStatement
  (perturbation_shall_disturbance "power outage")
  (context_general "any context")
  phase_ops
  agent_internal
  nature_same
  (parameter_level "power" "currentLevel")
  (destination_one "similar level")
  (aspect_operations "power system")
  (effect_maintain_same "maintain <what goes here>")
  (parameter_level "power" "currentLevel")
  (destination_one "similar levels")
  (aspect_operations "operations")
  (abstraction_system "all public hospitals")
  (valuable_compound "valuable because of"
    (reaction_sooner_than 1 unit_time_minute)
    (span_shorter_than 2 unit_time_minute)
    (cost_less_than 100 unit_money_dollar)
    (benefit_same_as "without losing life-critical operations or equipment")).

Compute print_changeStatement changeStatementRoss.
```

"changeStatement : in response to perturbation: power outage; in context: any context; during phase: operations; desire agent: internal; to make some nature: same; impetus to the parameter power, level currentLevel; using destination one: similar level; in the aspect operations: power system; in order to have an effect: maintain <what goes here>; on the parameter power, level currentLevel; with destination one: similar levels; in the aspect operations: operations; of the abstraction: system all public hospitals; that are valuable with respect to valuable: valuable because of, reaction sooner than 1 minute, span: shorter than 2 minute, cost: less than 100 dollar, benefit: same as without losing life-critical operations or equipment."

Comment [AMR1]: Is "perturbation_shall" vs. "perturbation_disturbance"?

Comment [AMR2]: This should not be here to adhere to apparent syntax as set three.

Comment [AMR3]: No label in basis. Instead the printStatement method should know to use "maintain" as English word for verb resulting in "same"

Latest Version of Basis (20D)

Prescriptive Semantic Basis for Change-type Ilities

In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Perturbation	Context	Phase	Agent	Impetus Change	Mech	Outcome Change	System	Valuable* (this category is not complete)

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the system "parameter" from "origin(s)" to "destination(s)" in the "aspect" using "mechanism" in order to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Impetus* (optional)					Mech	Outcome					Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Origin	Destination	Aspect	Mechanism	Effect	Parameter	Origin	Destination	Aspect					
optional	circumstantial: required; general: optional	null	optional	null	required	optional	optional	null* (this is implied by "parameter")	Optional	null	required	optional	optional	null*	optional	required	required	required	required
"name"	"name(s)"		"name(s)"		"parameter"	"state(s)"	"state(s)"		"name"		"parameter"	"state(s)"	"state(s)"		"name"	"threshold units"	"threshold units"	"threshold units"	"threshold units"
none	circumstantial	pre-ops	none	decrease	level	one	one	form		decrease	level	one	one	form	architecture	sooner	shorter	less	more
disturbance	general	ops	internal	same	set	few	few	function		same	set	few	few	function	design	later	longer	same	same
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations		increase	<empty>	many	many	operations	system	always	same	more	less
<empty>		<empty>	either	not-same		<empty>	<empty>	<empty>		not-same		<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>
			<empty>	<empty>						<empty>									

The semantic basis would be used differently in different use cases

Full basis:

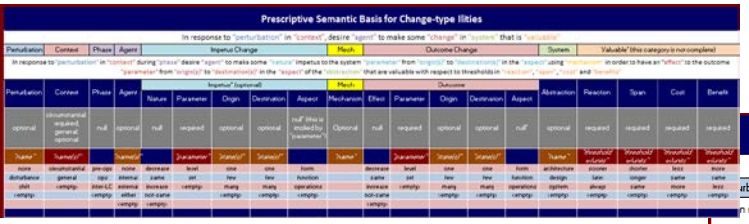
- When trying to write a very specific requirement statement (should not occur until AFTER analysis to determine what should be done)

Subset of basis:

- Early in the design phase, leave out the "valuable" categories (these are subjective, depend on outside factors)
- If one is trying to avoid fixating on a solution-centric approach, leave out change mechanism (allow engineers to propose own alternatives)

Different Use Cases of the Basis (1)

Full basis: 20 columns



(19 columns)

Prescriptive Semantic Basis for Change-type Ilities

In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the system "parameter" from "origin(s)" to "destination(s)" in the "aspect" in order to have an "effect" to the outcome "parameter" from "origins" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Impetus* (optional)					Outcome					Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Origin	Destination	Aspect	Effect	Parameter	Origin	Destination	Aspect					
optional	circumstantial: required; general: optional	null	optional	null	required	optional	optional	null* (this is implied by "parameter")	null	required	optional	optional	null*	optional	required	required	required	required
"name"	"name(s)"		"name(s)"		"parameter"	"state(s)"	"state(s)"			"parameter"	"state(s)"	"state(s)"		"name"	"threshold w/units"	"threshold w/units"	"threshold w/units"	"threshold w/units"
none	circumstantia	pre-ops	none	decrease	level	one	one	form	decrease	level	one	one	form	architecture	sooner	shorter	less	more
disturbance	general	ops	internal	same	set	few	few	function	same	set	few	few	function	design	later	longer	same	same
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations	increase	<empty>	many	many	operations	system	always	same	more	less
<empty>		<empty>	either	not-same		<empty>	<empty>	<empty>	not-same		<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>
			<empty>	<empty>					<empty>									

When to use: before engineering design/analysis has determined the best mechanism for achieving the change via impetus to achieve outcome.

(14 columns)

Prescriptive Semantic Basis for Change-type Ilities

In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"

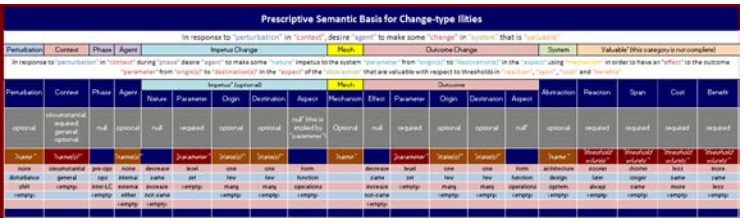
In response to "perturbation" in "context" during "phase" desire "agent" to have an "effect" to the outcome "parameter" from "origins" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Outcome					Abstraction	Reaction	Span	Cost	Benefit
				Effect	Parameter	Origin	Destination	Aspect					
optional	circumstantial: required; general: optional	null	optional	null	required	optional	optional	null*	optional	required	required	required	required
"name"	"name(s)"		"name(s)"		"parameter"	"state(s)"	"state(s)"		"name"	"threshold w/units"	"threshold w/units"	"threshold w/units"	"threshold w/units"
none	circumstantia	pre-ops	none	decrease	level	one	one	form	architecture	sooner	shorter	less	more
disturbance	general	ops	internal	same	set	few	few	function	design	later	longer	same	same
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations	system	always	same	more	less
<empty>		<empty>	either	not-same		<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>
			<empty>	<empty>									

When to use: to be OUTCOME oriented (i.e., focused on the "effects") as well as ensuring the change is "valuable" relative to defined dimensions.

Different Use Cases of the Basis (2)

Full basis: 20 columns



(10 columns)

Prescriptive Semantic Basis for Change-type Ilities

In response to "perturbation" in "context", desire "agent" to make some "change" in "system"

Perturbation	Context	Phase	Agent	Outcome Change					System
In response to "perturbation" in "context" during "phase" desire "agent" to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable									
				Outcome					
Perturbation	Context	Phase	Agent	Effect	Parameter	Origin	Destination	Aspect	Abstraction
optional	circumstantial: required; general: optional	null	optional	null	required	optional	optional	null*	optional
"name"	"name(s)"		"name(s)"		"parameter"	"state(s)"	"state(s)"		"name"
none	circumstantial	pre-ops	none	decrease	level	one	one	form	architecture
disturbance	general	ops	internal	same	set	few	few	function	design
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations	system
<empty>		<empty>	either	not-same		<empty>	<empty>	<empty>	<empty>
			<empty>	<empty>					

When to use: early in design in order to not over specify the change mechanism (allow engineers to propose/evaluate alternatives), or impetus (i.e. this is OUTCOME oriented).

Note: Leaving out "valuable" part of statement supports exploration. Later, when implications of ility statement are better understood, one can specify (differently across stakeholders, if desired) subjective thresholds on what makes the change "valuable."

(11 columns)

Prescriptive Semantic Basis for Change-type Ilities

In response to "perturbation" in "context", desire "agent" to make some "change" in "system"

Perturbation	Context	Phase	Agent	Mech	Outcome Change				System
In response to "perturbation" in "context" during "phase" desire "agent" using "mechanism" to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction"									
				Mech	Outcome				
Perturbation	Context	Phase	Agent	Mechanism	Effect	Parameter	Origin	Destination	Aspect	Abstraction
optional	circumstantial: required; general: optional	null	optional	optional	null	required	optional	optional	null*	optional
"name"	"name(s)"		"name(s)"	"name"		"parameter"	"state(s)"	"state(s)"		"name"
none	circumstantial	pre-ops	none		decrease	level	one	one	form	architecture
disturbance	general	ops	internal		same	set	few	few	function	design
shift	<empty>	inter-LC	external		increase	<empty>	many	many	operations	system
<empty>		<empty>	either		not-same		<empty>	<empty>	<empty>	<empty>
			<empty>		<empty>					

When to use: if there is a constraint to make use of an existing/inherited mechanism, for example.

Note: This version is OUTCOME oriented, leaving open the "valuable" specification, but leaves in the "mechanism" category to constrain how the change should occur.

Prescriptive Semantic Basis for Change-type Ilities

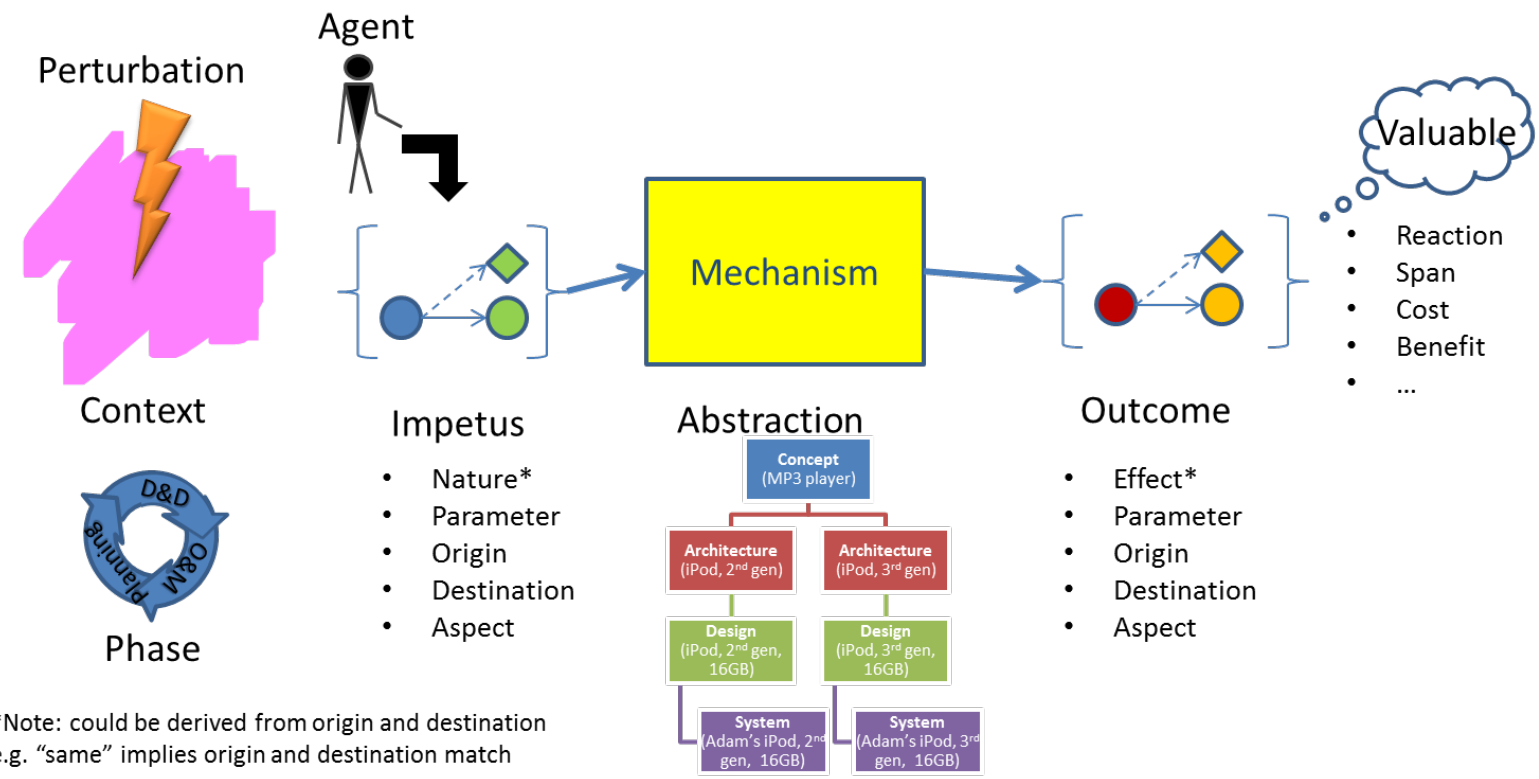
In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Perturbation	Context	Phase	Agent	Impetus Change	Mech	Outcome Change	System	Valuable* (this category is not complete)

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the system "parameter" from "origin(s)" to "destination(s)" in the "aspect" using "mechanism" in order to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Impetus* (optional)					Mech	Outcome					Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Origin	Destination	Aspect	Mechanism	Effect	Parameter	Origin	Destination	Aspect					

Some change in inputs Results in Some change in outputs



*Note: could be derived from origin and destination
e.g. "same" implies origin and destination match

Toward a Theory of Ilities

What are the semantic fields that span the general set of ilities?
e.g. “change-type”, “architecture-type”, “new ability-type”

Prescriptive Semantic Basis for Change-type Ilities

In response to “cause” in “context”, desire “agent” to make some “change” in “system” that is “valuable”

Cause	Context	Phase	Agent	Impetus Change	System	Outcome Change	System	Valuable

In response to “perturbation” in “context” during “phase” desire “agent” to make some “nature” impetus to the design “parameter” with “destination(s)” in the “aspect” to have an “effect” to the outcome “parameter” with “destination(s)” in the “aspect” of the “abstraction” that are valuable with respect to thresholds in “reaction”, “span”, “cost” and “benefits”

				Impetus				Outcome								
Perturbation	Context	Phase	Agent	Nature	Parameter	Destination	Aspect	Effect	Parameter	Destination	Aspect	Abstraction	Reaction	Span	Cost	Benefit
					“parameter”	“state”			“parameter”	“state”			“threshold”	“threshold”	“threshold”	“threshold”
1	2	3	4	5	6	7	8	9	10	11	12	13	14			

Generated ility “labels”

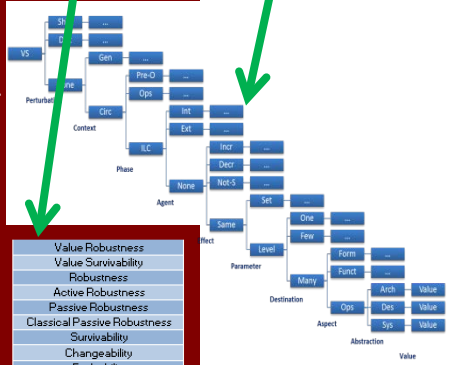
Basis

Derived ility “hierarchies”

We do not want more definitions, but rather, unambiguous, verifiable, standardized representations of desired system properties

Perturbation	Context	Phase	Agent	Impetus				Outcome				Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Destination	Aspect	Effect	Parameter	Destination	Aspect					
					“parameter”	“state”			“parameter”	“state”			“threshold”	“threshold”	“threshold”	“threshold”
disturbance	circumstantia	pre-ops	internal	increase	level	one	form	increase	level	one	form	architecture	sooner	shorter	less	more
shift	general	ops	external	decrease	set	few	function	decrease	set	few	function	design	later	longer	more	less
none	any	inter-LC	either	not-same	any	many	operations	not-same	any	many	operations	system	always	same	same	same
any		any	none	same		any	any	same		any	any	any	any	any	any	any
			any	any				any								
shift	any	ops	any	any	any	any	any	same	“Value”	few	any	any	any	any	any	any
disturbance	any	ops	any	any	any	any	any	same	“Value”	few	any	any	any	any	any	any
shift	any	ops	any	any	any	any	any	same	any	few	any	any	any	any	any	any
shift	any	ops	any	not-same	any	any	any	same	any	few	any	any	any	any	any	any
shift	any	ops	any	same	any	few	any	same	any	few	any	any	any	any	any	any
shift	any	ops	none	same	any	few	any	same	level	few	form	system	any	any	any	any
disturbance	any	ops	any	any	any	any	any	same	any	few	any	any	any	any	any	any
any	any	any	either	not-same	any	any	any	not-same	any	any	any	any	any	any	any	any
shift	general	inter-LC	any	not-same	any	any	any	not-same	any	any	any	architecture	any	any	any	any
any	any	any	internal	not-same	any	any	any	not-same	any	any	any	any	any	any	any	any
any	any	any	external	not-same	any	any	any	not-same	any	any	any	any	any	any	any	any
any	any	any	any	not-same	any	any	any	not-same	level	any	any	any	any	any	any	any
any	any	any	any	not-same	any	any	any	not-same	set	any	any	any	any	any	any	any
any	any	ops	either	not-same	any	any	any	increase	set	any	any	any	any	any	any	any
any	any	any	any	not-same	any	any	any	not-same	any	any	any	any	any	any	any	any
any	any	any	any	not-same	any	any	any	not-same	any	any	any	any	sooner	any	any	any
any	any	ops	any	same	“Element set”	one	form	not-same	“Link set”	any	form	any	any	any	any	any
any	any	ops	any	same	“Element set”	one	operations	not-same	“Order set”	any	operations	any	any	any	any	any
any	any	ops	any	same	any	one	form/ops	not-same	set	few/many	function	any	any	any	any	any
any	any	ops	any	same	any	one	form/fnct	not-same	set	few/many	operations	any	any	any	any	any

Value Robustness
Value Survivability
Robustness
Active Robustness
Passive Robustness
Classical Passive Robustness
Survivability
Changeability
Evolvability
Adaptability
Flexibility
Scalability
Modifiability
Extensibility
Agility
Reactivity
Form Reconfigurability
Operational Reconfigurability
Functional Versatility
Operational Versatility



Ultimate Goal: develop the basis/bases to be a prescriptive instrument(s) for spanning the semantic fields whose union encompass all “ilities”

Next Steps & Research Questions

- High level feedback
  - Does this approach make sense?
- Applicability to Semantic Fields
  - Does this basis only apply to “change-type” semantic field?
  - What are the members of this field?
  - What other semantic fields may exist?
  - Can a different basis be used for each semantic field?
- Refinement of basis
  - What are appropriate basis categories?
  - What are appropriate choices within a category?
- Refinement of ility labels
  - Are there consensus patterns in matching SEAri ilities to basis?
  - Are there consensus patterns for given ility terms without provided definitions?
  - How do other definitions for ility labels map to basis?
- Prescriptive use
  - Can someone use the basis to generate change statements, which will automatically label with the appropriate ilities?
  - How useful is the change statement for supporting verifiable requirements?

Ultimately we do not want more definitions, but rather, unambiguous, verifiable, standardized representations of desired system properties

Ultimate Goal: develop the basis/bases to be a prescriptive instrument(s) for spanning the semantic fields whose union encompass all “ilities”

References

Akmajian, A., Demers, R.A., Farmer, A.K., and Harnish, R.M. (2001), *Linguistics*. MIT Press: Cambridge, MA.

Boegh, J., "A New Standard for Quality Requirements," *IEEE Software,* March/April 2008, pp. 57-63.

Boehm, B., Port, D, and Al Said, M., Avoiding the Model-Clash Spiderweb, *IEEE Computer,* November 2000, pp. 120-122.

Boehm, B., Huang, L., Jain, A., and Madachy, R., "The Nature of Information System Dependability: A Stakeholder/Value Approach," USC-CSSE TR USC-CSE-2004-520; http://csse.usc.edu/csse/TECHRPTS

Buschmann, F., Ameller, D., Ayala, C., Cabot, J., and Franch, X., "Architecture Quality Revisited," *IEEE Software,* July/August 2012, pp. 22-24.

de Neufville, R. and Scholtes, S. (2011), *Flexibility in Engineering Design*, MIT Press: Cambridge MA.

de Weck, Roos, and Magee (2011), "Life-cycle Properties of Engineering Systems," in *Engineering Systems*, MIT Press: Cambridge, MA.

de Weck, O.L., Ross, A.M., and Rhodes, D.H., "Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities)," 3rd International Conference on Engineering Systems, TU Delft, the Netherlands, June 2012.

Fricke, E. and Schulz, A.P., (2005), "Design for Changeability (DfC): Principles to Enable Changes in Systems Throughout their Entire Lifecycle," *Systems Engineering*, Vol. 8, No. 4, pp. 342-359.

ISO/IEC 9126-1:2001, Software Engineering—Product Quality—Part 1: Quality Model, Int'l Organization for Standardization, 2001.

ISO/IEC 25030:2007, Software Engineering—Software Product Quality Requirements and Evaluation (SQuaRE)—Quality Requirements, Int'l Organization for Standardization, 2007.

Nilchiani, R. (2005), "Measuring Space Systems Flexibility: A Comprehensive Six-element Framework," *PhD in Aeronautics and Astronautics*, MIT, Cambridge MA.

Ross, A.M. (2006), "Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration," *PhD in Engineering Systems*, MIT, Cambridge, MA.

Ross, A.M., Rhodes, D.H., and Hastings, D.E. (2008), "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, Modifiability, and Robustness for Maintaining Lifecycle Value," *Systems Engineering*, Vol. 11, No. 3, pp. 246-262.

Ross, A.M., Beesemyer, J.C., and Rhodes, D.H. (2011), A Prescriptive Semantic Basis for System Lifecycle Properties. SEAri Working Paper Series, WP-2011-2-1, pp. 1-16. http://seari.mit.edu/documents/working_papers/SEAri_WP-2011-2-2.pdf. (*last accessed on 20 February 2014*)

Saleh, J. H., Mark, G.T., Jordan, N.C., (2009), "Flexibility: a multi-disciplinary literature review and a research agenda for designing flexible engineering systems," *Journal of Engineering Design*, Vol. 20, No. 3. pp. 307-323.