"An Empirical Investigation of System Changes to Frame Links between Design Decisions and Ilities"

J. Clark Beesemyer, Adam M. Ross, and Donna H. Rhodes
Massachusetts Institute of Technology

CSER 2012
St. Louis, 19-22 March 2012

Motivation

"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."*

"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.'"*

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

Ross, Beesemyer, and Rhodes 2012

© 2012 Massachusetts Institute of Technology

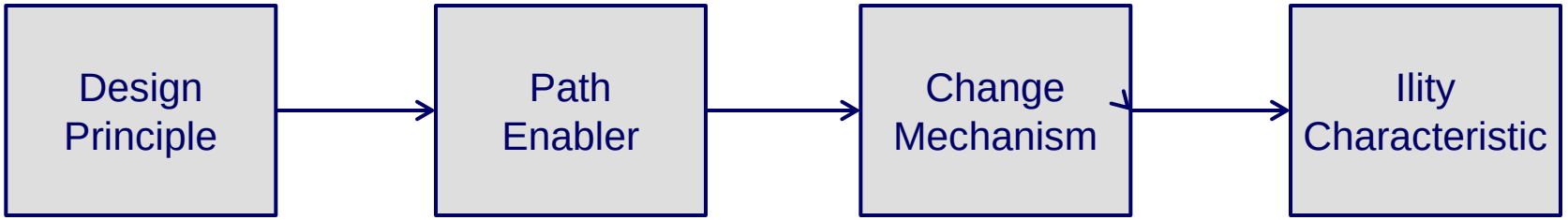
Motivation

- High impact early decisions usually made with incomplete system knowledge
- Decision makers may improve their capacity to discriminate between system concepts and design choices by measuring a system's "ilities" such as changeability, scalability, and survivability
  - These system properties (ilities) can be ambiguous, imprecise, and hard to validate or verify
- Increased interest in ilities requires better understanding of the change mechanisms and more precise language

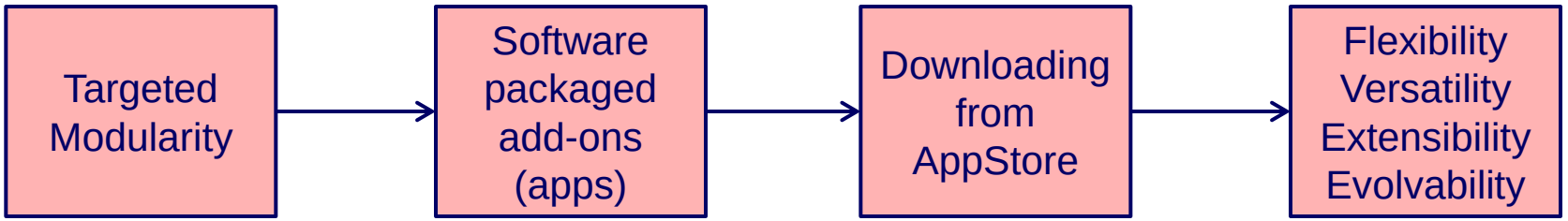
Looking at empirical cases of system changes and associated ilities in a structured manner may lead to more effective design approaches.

© 2012 Massachusetts Institute of Technology

Ilities as Outcomes

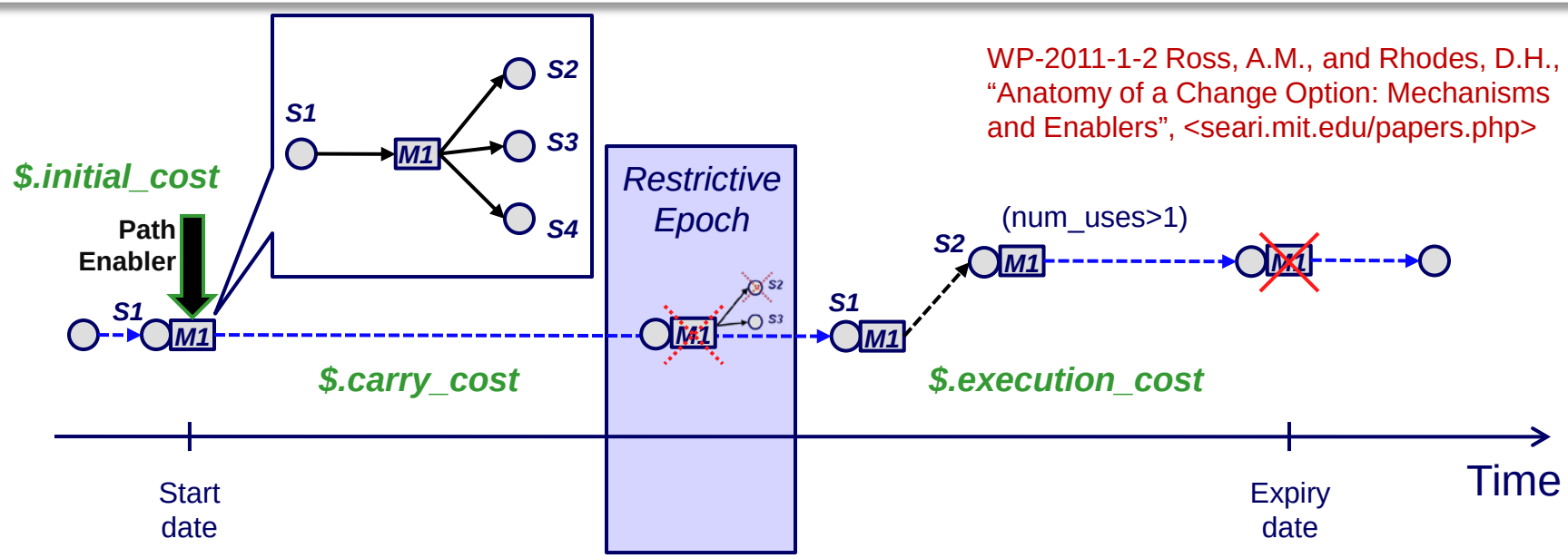


- Ilities are performance characteristics made possible ultimately by the design principles used in the early phases of design
- Descriptive research aims to look at many of these cases for trends and insight that may be used in future designs



© 2012 Massachusetts Institute of Technology

The Change Option



The following factors characterize a change option:

- Name
- Start date
- Expiry date
- Possible end states = f(start state)
- Initial cost = cost of path enabler?
- Carry cost = f(now, execution date, expiry date)
- Execution cost = f(end state, epoch)
- Reusability = number of times it can be executed
- Valid epochs
- Valid lifecycle phase
- Pre-requisites for execution (e.g. agent)
- Change "type" (e.g. flexible, adaptable, etc.)

Change options can be compared and valuated on this basis

© 2012 Massachusetts Institute of Technology

Generalizing the Change 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 Changeability-type 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	less	more
shift	general	design	function	ops	external	set	smaller	less	few	later	longer	more	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					

(choose one)

1 2 3 4 5 6 7 8 9 10

10 dimensional basis for specifying "changeability"-related ilities

WP-2011-1-2 Ross, A., Beesemyer, J., and Rhodes, D., "A Prescriptive Semantic Basis for System Lifecycle Properties", seari.mit.edu/papers.php

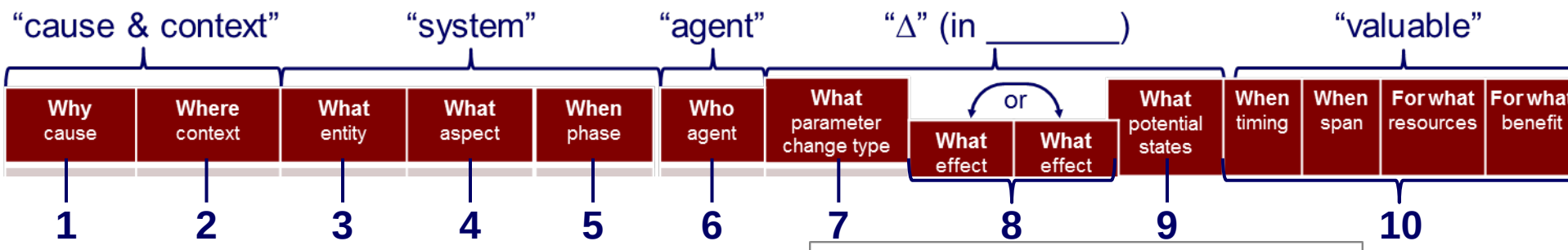
© 2012 Massachusetts Institute of Technology

Generalizing the Change Statement: A Prescriptive Basis

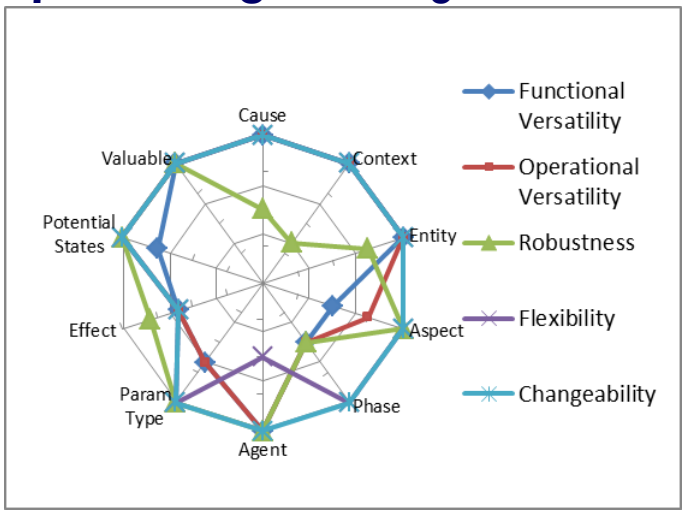
10 dimensional basis for specifying "changeability"-related ilities

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

Prerequisite: What "parameter" are you specifying?



Dim #	Category Name	# Levels
1	Cause	4
2	Context	3
3	Entity	4
4	Aspect	4
5	Phase	4
6	Agent	4
7	Parameter Type	3
8	Effect	5
9	Potential States	4
10	Valuable	256



WP-2011-1-2 Ross, A., Beesemyer, J., and Rhodes, D., "A Prescriptive Semantic Basis for System Lifecycle Properties", seari.mit.edu/papers.php

© 2012 Massachusetts Institute of Technology

Captured Ilities

Prescriptive Semantic Basis for Changeability-type 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	less	more
shift	general	design	function	ops	external	set	smaller	less	few	later	longer	more	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					
(choose one)													

Ility Name

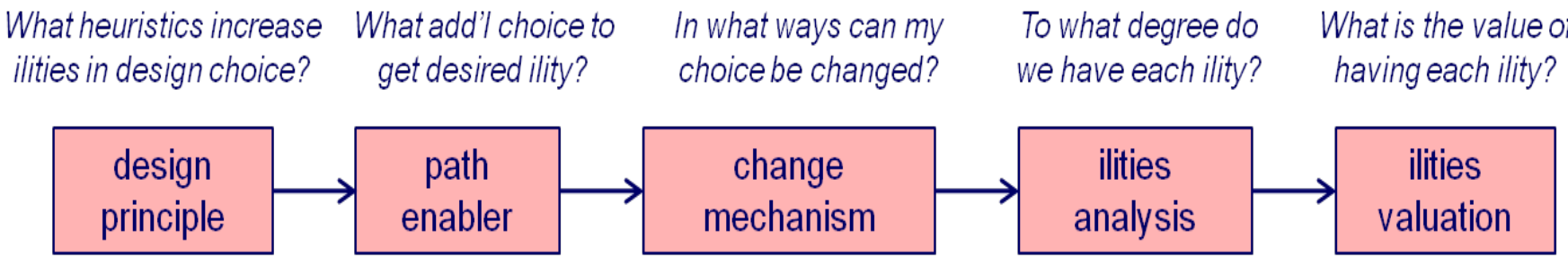
any	any	any	function	ops	any	set	any	not-same	many	any	any	any	any
any	any	any	operations	ops	any	set	any	not-same	many	any	any	any	any
shift	circumstantial	system	any	ops	any	any	same	same	any	any	any	any	any
shift	circumstantial	system	form	ops	none	level	same	any	few	any	any	any	any
shift	general	architecture	any	inter-LC	any	any	any	any	any	any	any	any	any
any	any	any	any	any	internal	any	not-same	not-same	any	any	any	any	any
any	any	any	any	any	external	any	not-same	not-same	any	any	any	any	any
any	any	any	any	any	any	level	not-same	any	any	any	any	any	any
any	any	any	form	ops	any	set	any	not-same	many	any	any	any	any
any	any	any	any	any	any	set	any	not-same	any	any	any	any	any
disturbance	circumstantial	any	any	ops	any	any	any	any	any	any	any	any	any
any	any	any	form	ops	any	any	any	any	any	any	any	any	any
any	any	any	any	any	any	any	not-same	not-same	any	any	shorter	any	any
any	any	any	any	any	any	any	not-same	not-same	any	sooner	any	any	any
any	any	any	any	ops	any	set	any	more	any	any	any	any	any
any	any	any	any	any	either	any	not-same	not-same	any	any	any	any	any
shift	any	any	any	any	any	any	any	any	any	any	any	any	same

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

© 2012 Massachusetts Institute of Technology

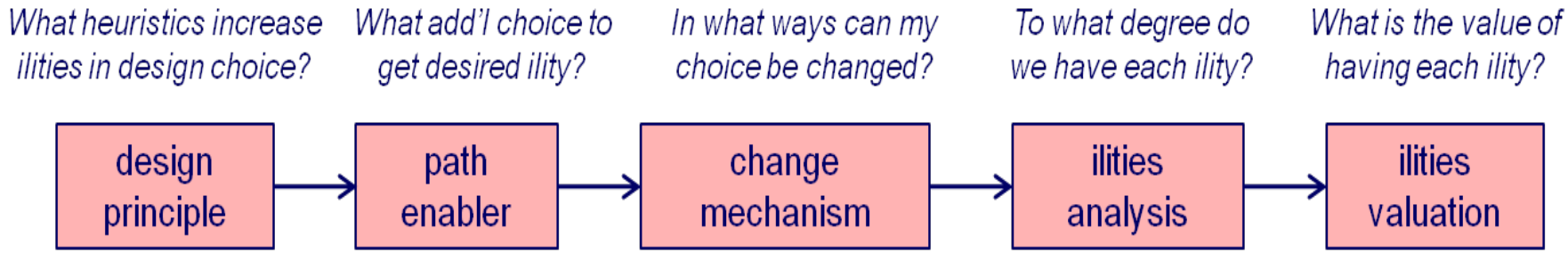
Change Database

- To capture changes from many different systems and various domains
  - Currently holding ~100 changes from ~45 systems
- Developed from Ility Framework previously discussed
- Takes system and change information in a categorical manner to determine ilities present in changes



© 2012 Massachusetts Institute of Technology

Change Database



This research aims to find a better means of determining which ilities are present in different system changes and map those ilities to various design principles. When stakeholders identify an ility as a desired property of a design, the ultimate goal of added value to the system from having this ility represents one end of this relationship.

© 2012 Massachusetts Institute of Technology

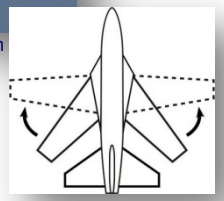
Change Database

What heuristics increase ilities in design choice? → **design principle** → *What add'l choice to get desired ility?* → **path enabler** → *In what ways can my choice be changed?* → **change mechanism** → *To what degree do we have each ility?* → **ilities analysis** → *What is the value of having each ility?* → **ilities valuation**

Design Principles are the heuristics designers follow when making design choices.

Path Enablers are instantiations of DPs that facilitate CEs in a system.

www.militaryfactory.com



Change Mechanisms give the system options to change if necessary given conditions and performance

Ilities are properties shown during these changes that equate to more or less value for stakeholders.

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

© 2012 Massachusetts Institute of Technology

Change Database

Change #	54
System	PowerTech PSX Interim Tier 4/Stage IIIB
Domain	Consumer Vehicle
Preliminary Ility Type	Evolvable, Flexible, Changeable
Mapped Ility Type	Evolvable, Flexible, Changeable, Modifiable, Value Robust
Parameter	PM reducing method
Path Enabler	Diesel Oxidation Catalyst, D 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 e. The or a 90 percent culate matter a 50 percent oxides (NOx)
Perturbation	Shift in emissions regulations
Specificity	Applies to all off-road vehicles
Abstraction	Outside design pl
Aspect	Change in physic
LC Phase	Between lifecycles
Executes	John Deere engir
Param. Type	Change in design
Level	--
Set	Different design n
Target Range	Only one filter opti
Start	
Duration	
Costs	
Utility	Less PM

Image

Manufacturer	John Deere
Production	Mass Produced
Potential Variety	Few
Actual Variety	Few
Population	Many

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

XXXXXXXXXX

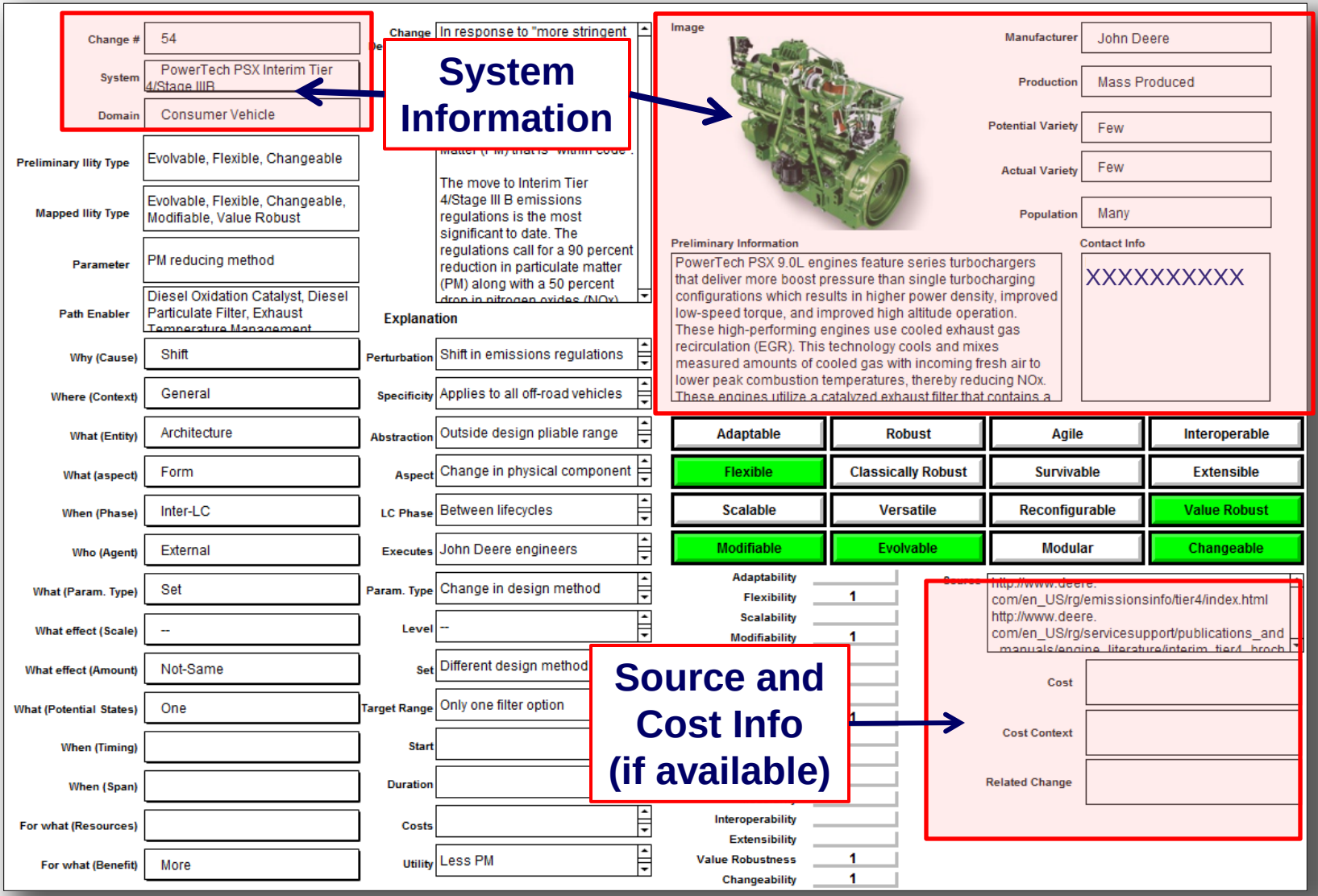
Change database

Change #	System	Context	Entity	Aspect	Phase	Agent	Param_T...	Scale	Amount	Potenti...	Preliminary Ility Type
1	HMMWV	General	Design	Form	Inter-LC	External	Level	Bigger	--	Few	Evolvable, Flexible
2	B-52	General	Architecture	Form	E (ops)	External	Set	--	More	Few	Evolvable, Flexible
3	F-16	General	Architecture	Operations	E (ops)	Internal					Evolvable, Flexible,
4	F-18 (Swiss)	Circumstantial	Design	Form	E (ops)	External	Level	Bigger	--	One	Evolvable, Flexible
5	S-92	General	Architecture	Form	Inter-LC	External	Set	--	Not-	Few	Evolvable, Flexible
6	HMMWV	General	System	Form	E (ops)	External	Level	Bigger	--	Few	Flexible
7	HMMWV	General	Design	Form	Inter-LC	External	Level	Bigger		Few	
8	A-10	Circumstantial	Design	Operations	E (ops)	None	Level	Same	--	Few	
9	A-10	Circumstantial	System	Form	E (ops)	Internal	Level	Same	--	Few	
10	F-14	Circumstantial	System	Form	E (ops)	Internal	Level	Not-Same	--	Many	Reconfigurable,
11	Evolved Expendable Launch	General	Architecture	Form	Inter-LC	External	Set	--	Less	Few	Evolvability
12	M1126 Stryker	General	Architecture	Form	Inter-LC	External	Set	--	Not-	Few	Evolvability,
13	iPhone 4	General	Design	Form	E (ops)	External	Set		Not-	Many	Flexibility
14	F-14	Circumstantial	System	Form	E (ops)	External	Level	Not-Same	--	One	Reconfigurable,
15	Evolved Expendable Launch	General	Architecture	Form	Inter-LC	External	Set	--	Less	Few	Evolvability
16	Evolved Expendable Launch	General	Architecture	Form	Inter-LC	External	Set	--	Less	Few	Evolvability
17	Evolved Expendable Launch	Circumstantial	Design	Form	Inter-LC	External	Set	--	Less	Few	Evolvability
18	Evolved Expendable Launch	Circumstantial	Design	Form	Inter-LC	External	Set	--	Less	Few	Evolvability
19	F-14	Circumstantial	System	Form	E (ops)	Internal	Level	Same	--	Many	Reconfigurable,
20	iPhone 4	General	Architecture	Form	Inter-LC	External	Set		Not-	One	Evolvability,
21	iPhone 4	General	System	Function	E (ops)	External	Set		Not-	Many	Flexibility,
22	iPhone 4	Circumstantial	System	Form	E (ops)	Internal	Any			Few	Adaptability
23	M1126 Stryker	General	Architecture	Form	Inter-LC	External	Set		Not-	Few	Evolvability,
24	F-35	Circumstantial	System	Form	E (ops)	External	Set		Not-	Few	Changeability
25	F-35	Circumstantial	System	Form	E (ops)	External	Set	--	Not-	Few	Changeability
26	B-52	General	Architecture	Form	E (ops)	External	Set	--	More	Few	Evolvable, Flexible
27	Swiss Army Knife	Circumstantial	System	Form	E (ops)	External	Set	--	Not-	Many	Reconfigurable
28	Swiss Army Knife	Circumstantial	System	Operations	E (ops)	External	Set	--	Not-	Many	Reconfigurable
29	V-22	Circumstantial	System	Form	E (ops)	Internal	Set		Not-	Many	Changeability
30	Climate Manager	Circumstantial	System	Form	E (ops)	Internal	Level		Same	Few	Adaptability,
31	Climate Manager	Circumstantial	System	Form	E (ops)	Internal	Level		Same	Few	Adaptability,
32	Climate Manager	Circumstantial	System	Form	E (ops)	Internal	Level		Same	Few	Adaptability,
33	Solar Tracking System	Circumstantial	System	Form	E (ops)	Internal	Level	Same		Many	Adaptable
34	A-10	Circumstantial	System	Operations	E (ops)	None	Level	Same	--	Few	
35	HMMWV	Circumstantial	Design	Form	E (ops)	External	Set	Not-Same		Many	
36	8R Series Tractor	Circumstantial	System	Operations	E (ops)	Internal	Level	Not-Same	--	Few	Adaptable
37	8R Series Tractor	General	System	Operations	E (ops)	Internal	Set	--	More	Many	Adaptable, Versatile
38	8R Series Tractor	Circumstantial	System	Form	E (ops)	External	Set	--	Not-	Many	versatile, flexible,
39	8R Series Tractor	Circumstantial	System	Form	E (ops)	External	Set	--	Not-	Few	Flexible

Changeability

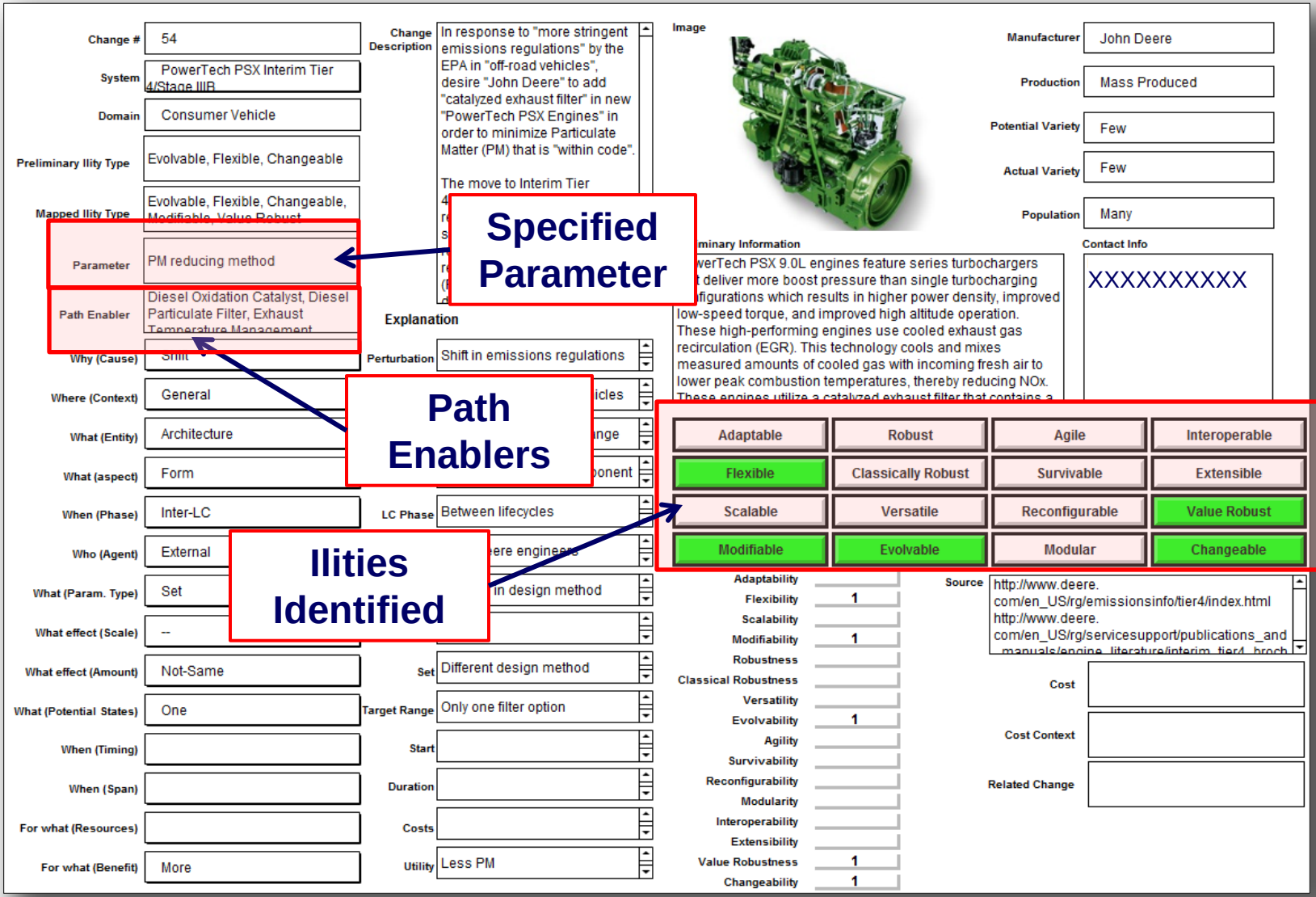
© 2012 Massachusetts Institute of Technology

Change Database



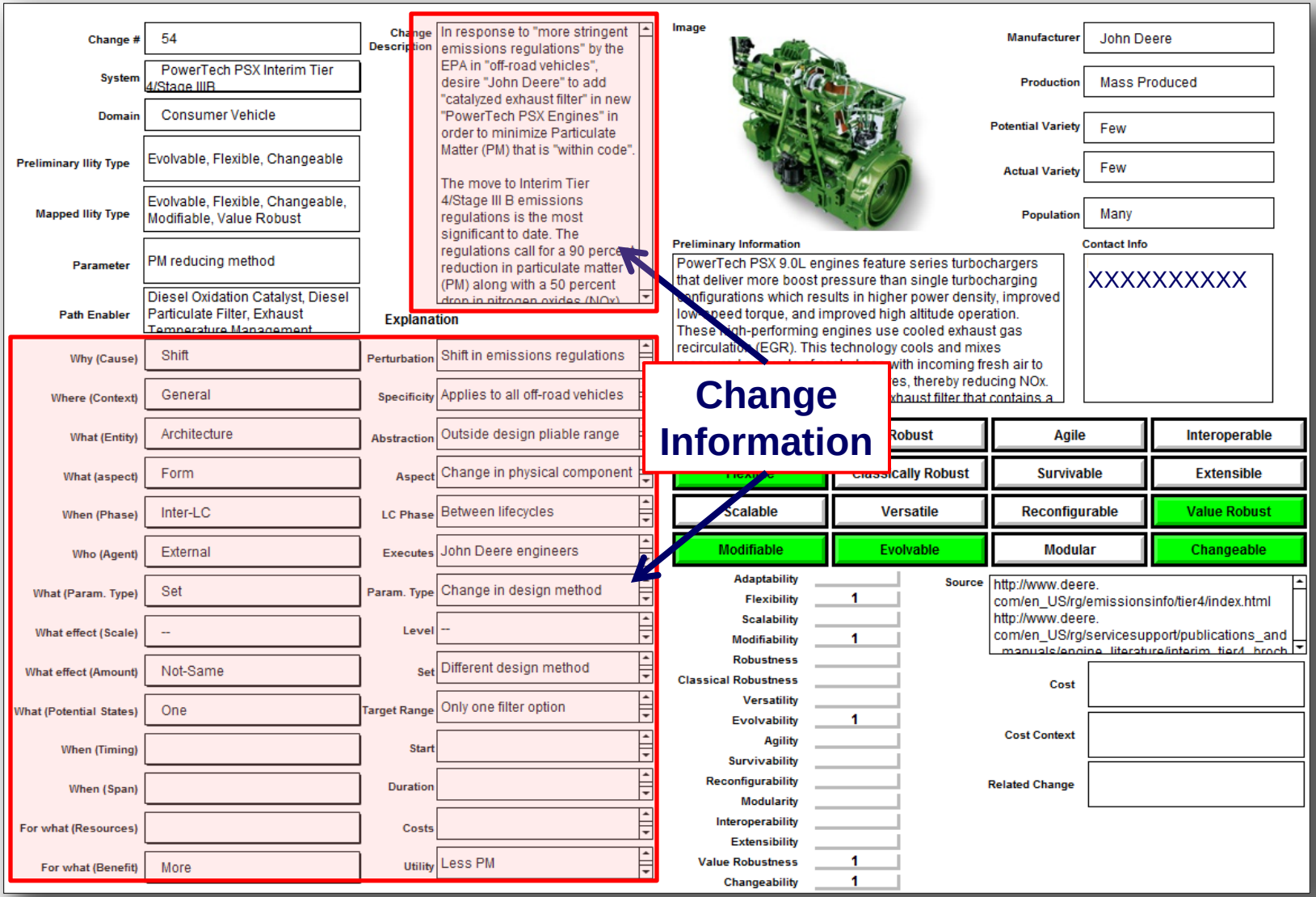
© 2012 Massachusetts Institute of Technology

Change Database



© 2012 Massachusetts Institute of Technology

Change Database

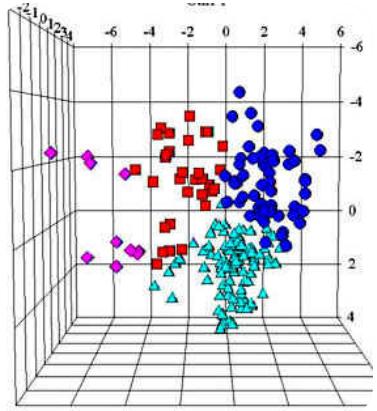


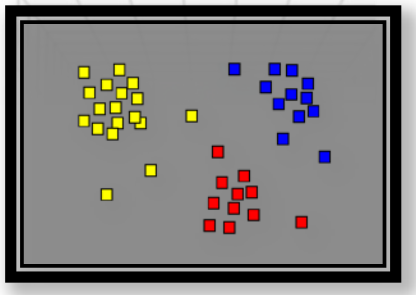
© 2012 Massachusetts Institute of Technology

Cluster Analysis

COOLCAT

- Clustering algorithm made to handle categorical data, (Fu et al.)
- Model based clustering (entropy model)
  - Where entropy is a measure of dissimilarity
- Basic Method
  - Turn categorical data into nominal dataset
  - Specify number of clusters
  - Run piecewise entropy to initialize first 2 clusters w/ highest entropy (most different)
  - Max(min(entropy)) of remaining records for each new cluster to specified number of clusters
  - Fill remaining records into clusters by minimizing system entropy for each record

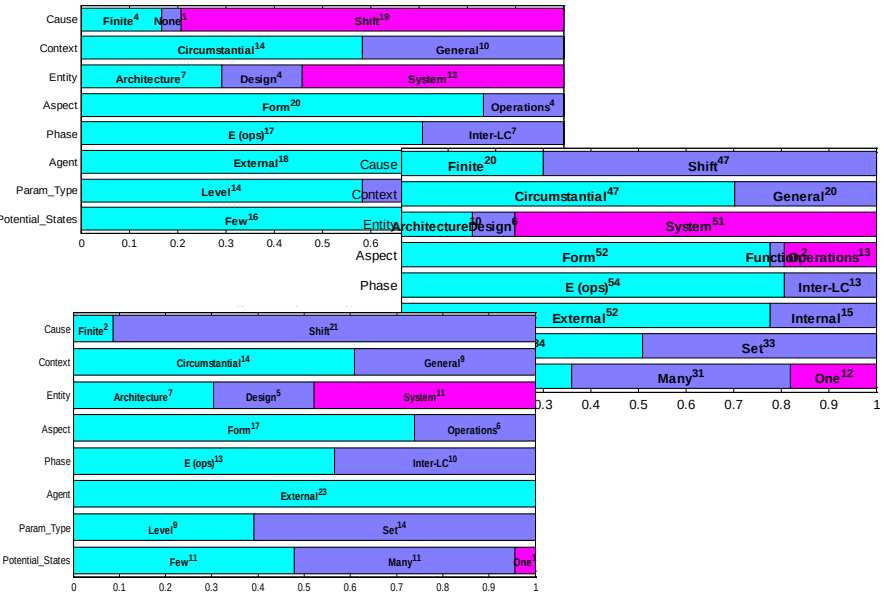




© 2012 Massachusetts Institute of Technology

Preliminary Insights

Heuristic-based clusters



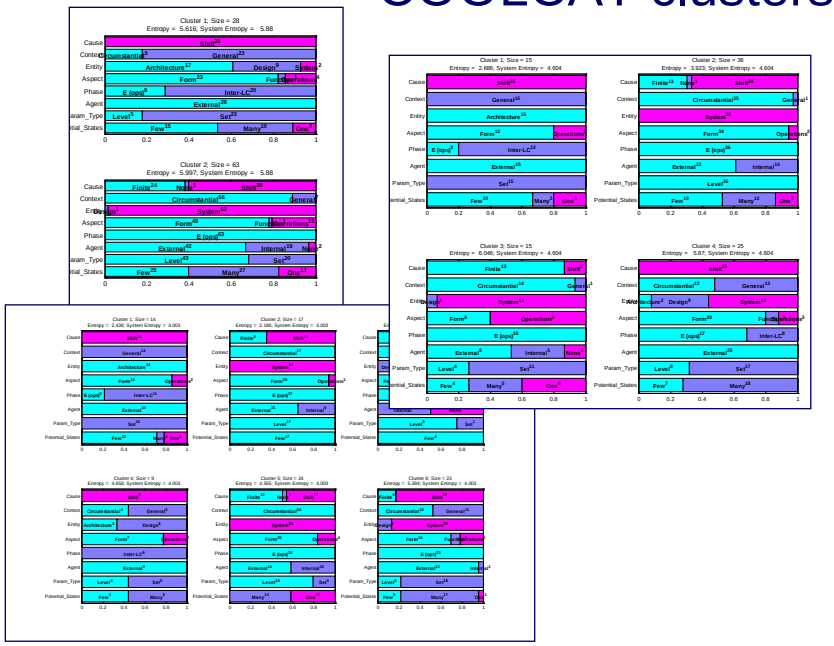
Military vs non-military

Large population vs small population

Mass produced vs unique

Long lifespan vs short

COOLCAT clusters



Control the number of clusters (2-n)

Can compare mental clusters to COOLCAT clusters.

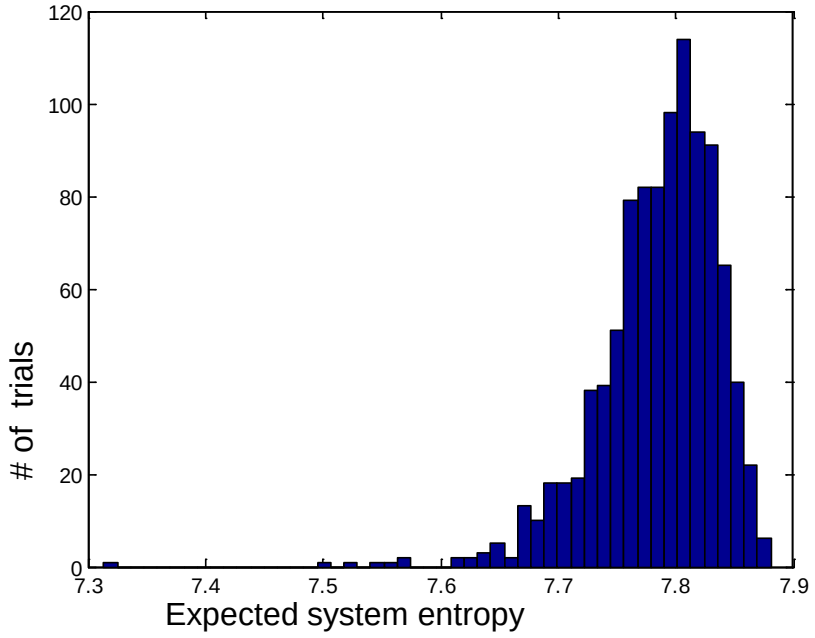
COOLCAT will find similarities more complex than simple data mining.

For example, the mental models show that shorter lifecycles tend to be more evolvable. COOLCAT models yield an evolvable cluster that is almost equal lifecycles but much more military and space oriented.

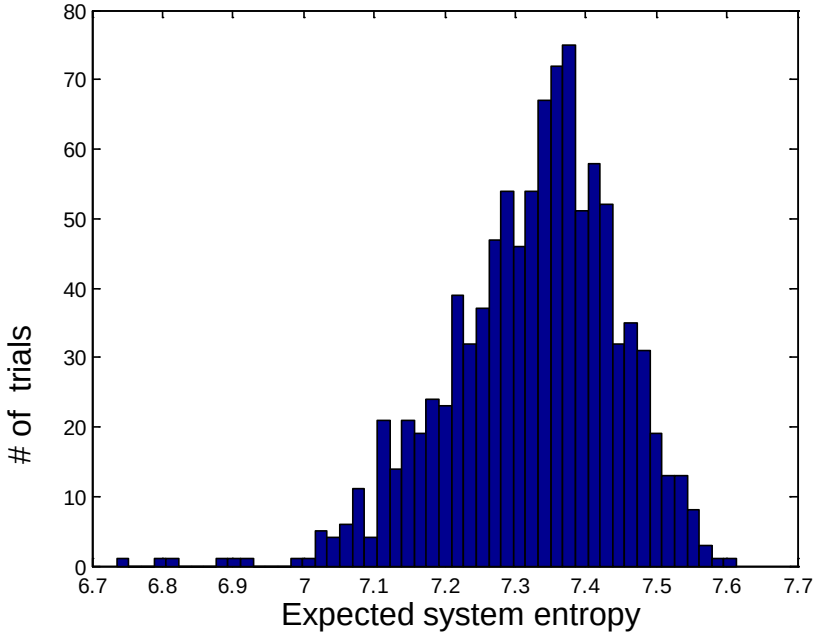
© 2012 Massachusetts Institute of Technology

Significance

- 2 clusters
  - Expected system entropy 5.8508 using COOLCAT
  - Exp Sys Ent 1000 trials using random clustering:



- 6 clusters
  - Expected system entropy 3.6795 using COOLCAT
  - Exp Sys Ent 1000 trials using random clustering:



© 2012 Massachusetts Institute of Technology

Conclusion

- Systems can use similar path enablers to achieve many different states, and these changes may be characterized by multiple ilities
- Despite desire to implement ilities in designs, a clear method is still lacking for identifying “good” designs with respect to preferred ilities
- Possible to classify / decompose system changes in a 10-dimensional semantic basis to describe the change mechanism
- Ilities are attributed not *a priori*, but as an outcome to the change
- Enables decision makers to create better requirements that are clear, concise, and verifiable.

Clarity in language is important as we look for design insights in groups of change mechanisms present in many systems.

© 2012 Massachusetts Institute of Technology

Questions?

© 2012 Massachusetts Institute of Technology