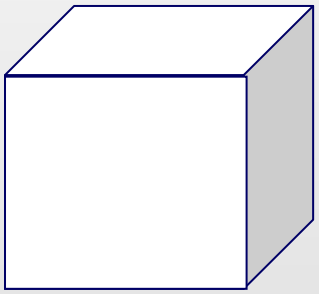
Using Attribute Classes to Uncover Latent Value during Conceptual Systems Design

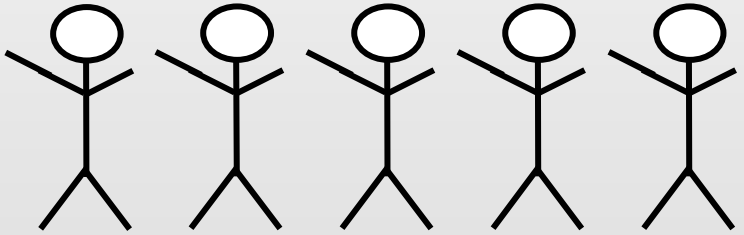
Adam M. Ross and Donna H. Rhodes
Massachusetts Institute of Technology

2nd Annual IEEE Systems Conference
Montreal, Canada
April 8, 2008

Designing a “Good” Box

No one likes a blank box…

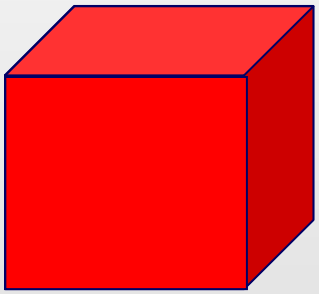


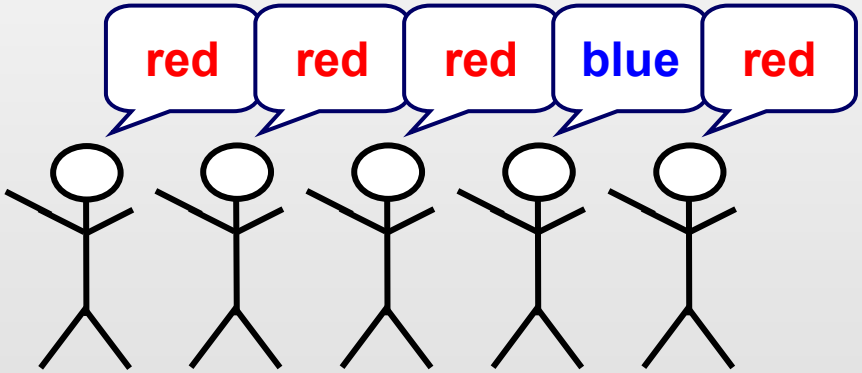


Let’s ask our stakeholders what color the box should be…

Revealed Preferences

Articulated: **Color** (red v. blue)

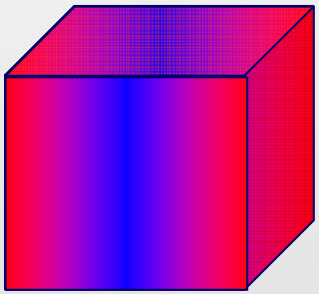


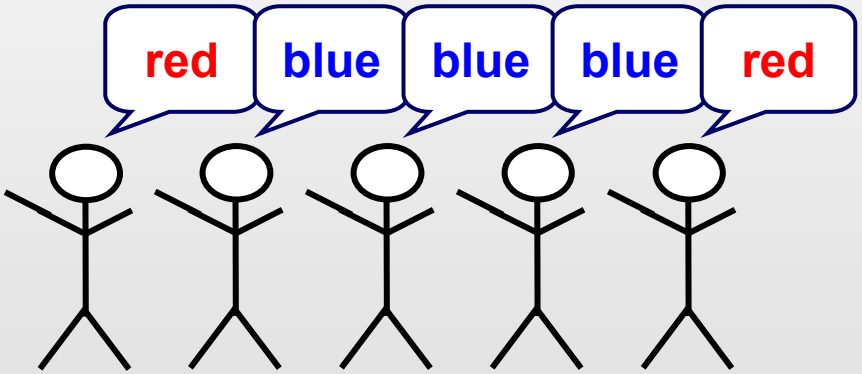


Should box be red or blue?

Listening to Changing Needs

Articulated: **Color** (red v. blue)



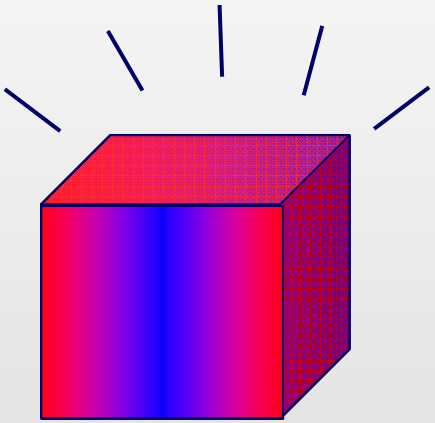


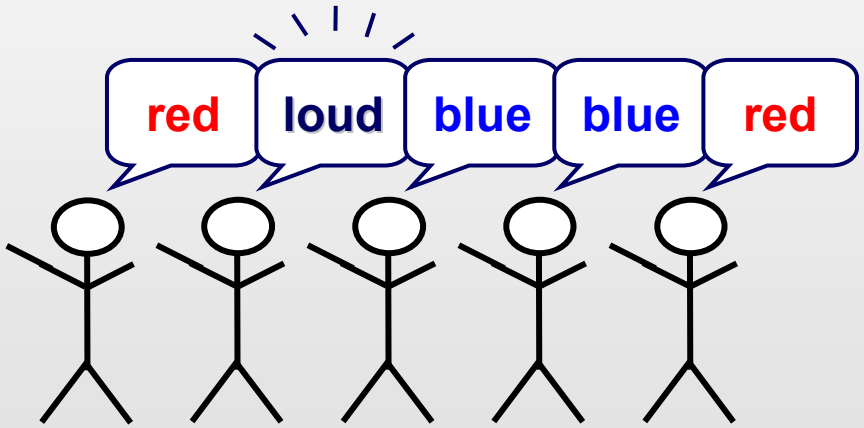
Now, should box be red or blue?

Listening to Revealed Needs

Articulated: **Color** (red v. blue), **Sound** (**loud**)

Unarticulated: It was **Sound**… how do others feel?



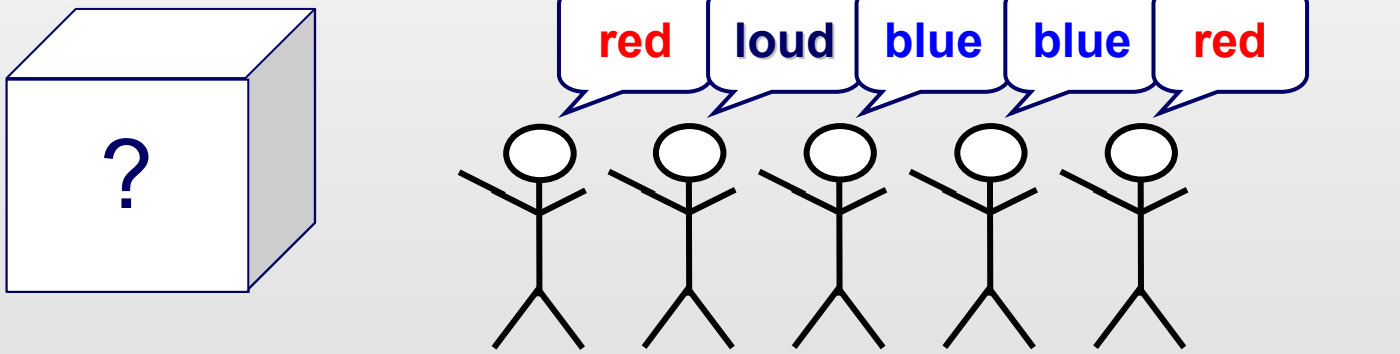


Now, what? Should box be red or blue, and **loud**?

Realizing Unexpressed Needs

Articulated: **Color** (red v. blue), **Sound** (**loud**)

Unarticulated: **Color** (red), **Sound** (*quiet*)

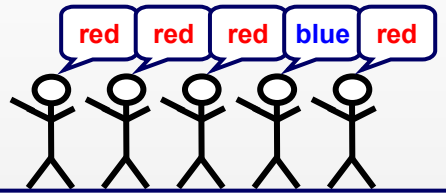


Now, what? Should box be red or blue, and **loud** or *quiet*?

Listening to the Need

Articulated: **Color** (red v. blue)

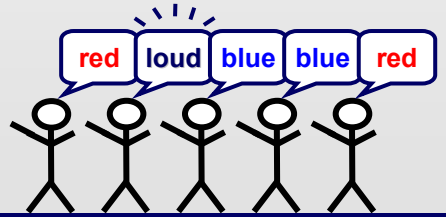
Unarticulated: ?



Should box be red or blue?

Articulated: **Color** (red v. blue), **Sound** (loud)

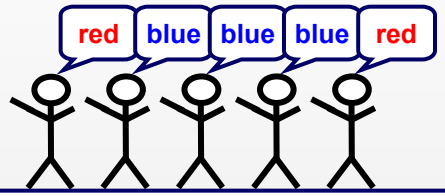
Unarticulated: It was **Sound**… how do others feel?



Now, what? Should box be red or blue, and loud?

Articulated: **Color** (red v. blue)

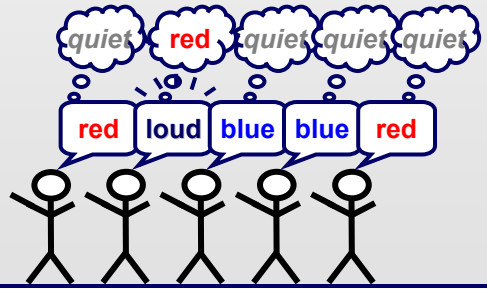
Unarticulated: ?



Now, should box be red or blue?

Articulated: **Color** (red v. blue), **Sound** (loud)

Unarticulated: **Color** (red), **Sound** (*quiet*)

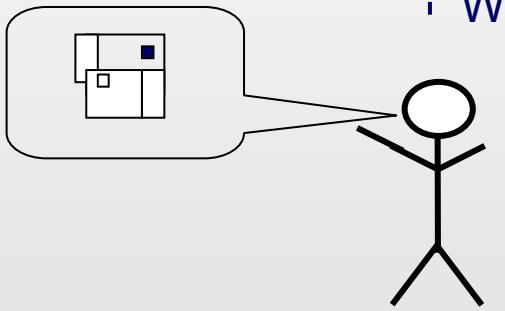


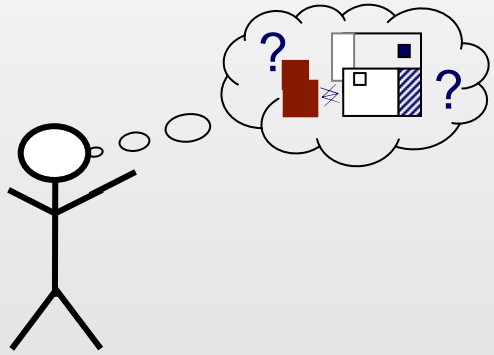
Now, what? Should box be red or blue, and loud or *quiet*?

How can system deal with articulated *and* unarticulated need?

Perceived Value Spectrum

Perceived Value Spectrum				
Articulated	Unarticulated			
Objectives	Can't say:	Forgot	Don't know yet	Intangible
Requirements	Don't say:	Assumed		
Attributes	Won't say:	Secret		





Elucidators

Personal reflection (time)

Conversations with mediators (facilitation)

Experience with system (learning by doing)

Interactions with system context (competition, test-driving)

Change in context (change of the "rules")

How to deliver value in the face of various value perceptions?

Articulated Value: Attributes as Decision Metrics

Attribute: **A decision maker-perceived metric that measures how well a *decision maker-defined* objective is met**

In the limit ranges converge to a point, the attributes become requirements

$\text{Attribute}^i \equiv X^i$

Attribute Characteristics

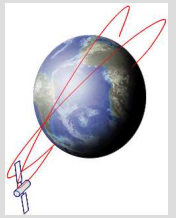
- Definition
- Units
- Range (least-most acceptable)

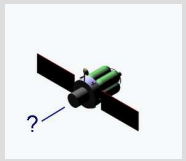
Space Tug Example:

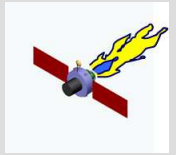
1) **Delta-V**: How much velocity can the vehicle impart on itself and/or the target? (km/sec) [>0→12]

2) **Interaction Capability**: What can the vehicle do to the target? (kg of equipment carried) [>0→5000]

3) **Speed**: Can the Space Tug change orbits in days? Months? (binary) [0→1]







Attributes as Decision Metrics

Attribute: **A decision maker-perceived metric that measures how well a *decision maker-defined* objective is met**

A set of attributes should be*:

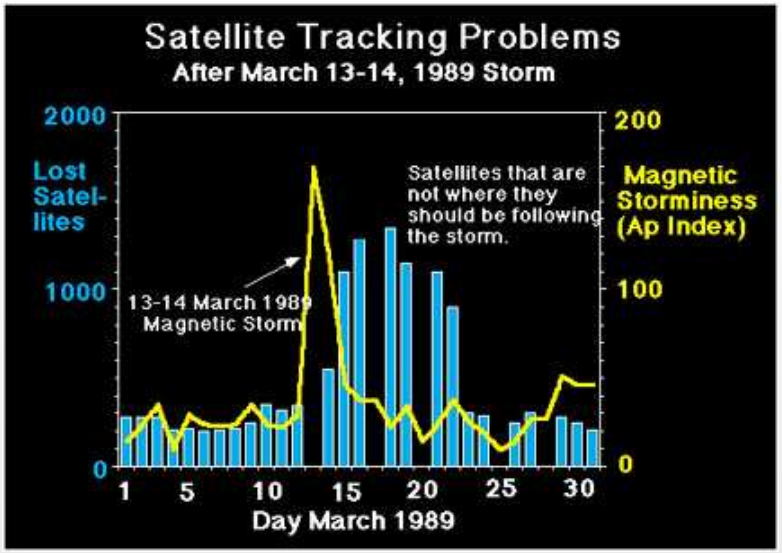
- Complete
- Operational
- Decomposable
- Non-redundant
- Minimal
- Perceived Independent**

In practice, the "Rule of 7" applies: Human mind limited to roughly 7 (7 ± 2) simultaneous concepts***

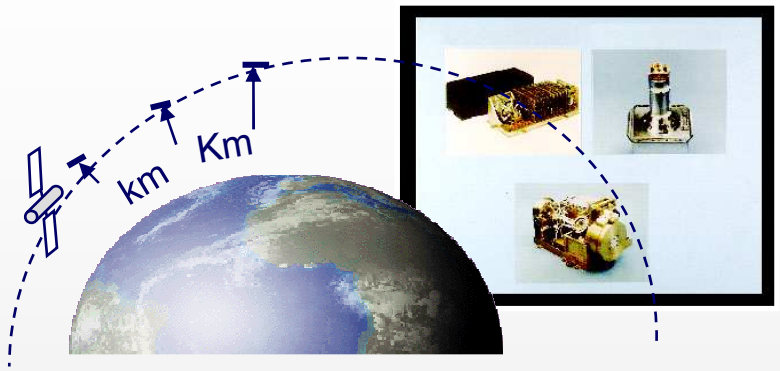
Attribute Set ≡ $\{X^i\}$

*Keeney, R.L. & Raiffa, H. *Decisions with Multiple Objectives--Preferences and Value Tradeoffs*. 2nd ed. Cambridge: Cambridge University Press, 1993.
**Not strictly necessary, but reduces interview time and complexity.
***Miller, G. A. "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information." *The Psychological Review* 63 (1956).

Example: Project X-TOS



- Problem: Inadequate drag models cause low-orbit objects to become "lost"
- Need: Better information on atmospheric drag
- Concept: In-situ vehicle carrying known instrument suite



DESIGN VARIABLES

- Mission Scenarios
  - Single satellite, single launch
  - Two satellites, sequential launch
  - Two satellites, parallel
- Orbital Parameters
  - Apogee altitude (km) 150-1100
  - Perigee altitude (km) 150-1100
  - Orbit inclination 0, 30, 60, 90
- Physical Spacecraft Parameters
  - Antenna gain
  - communication architecture
  - propulsion type
  - power type
  - delta_v

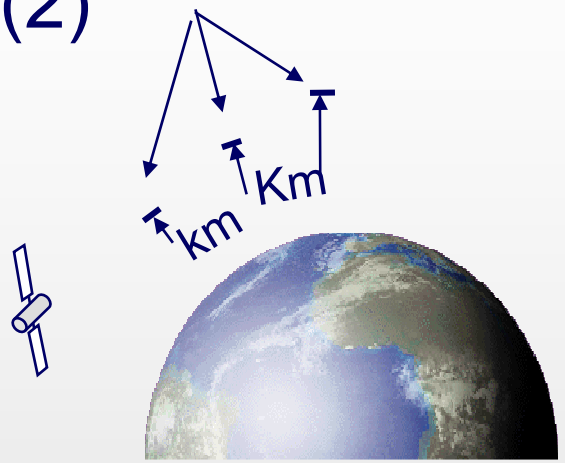
Number of Designs Explored: 50488

X-TOS User Attributes

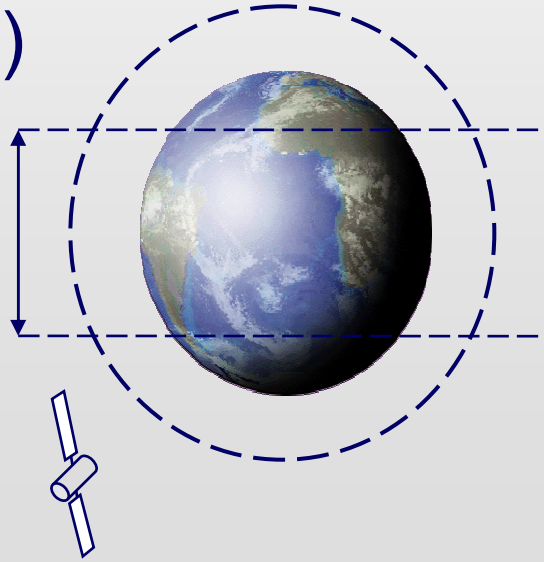
1) Data Life Span
2) Data Altitude
3) Maximum Latitude
4) Time Spent at Equator
5) Data Latency

(1)

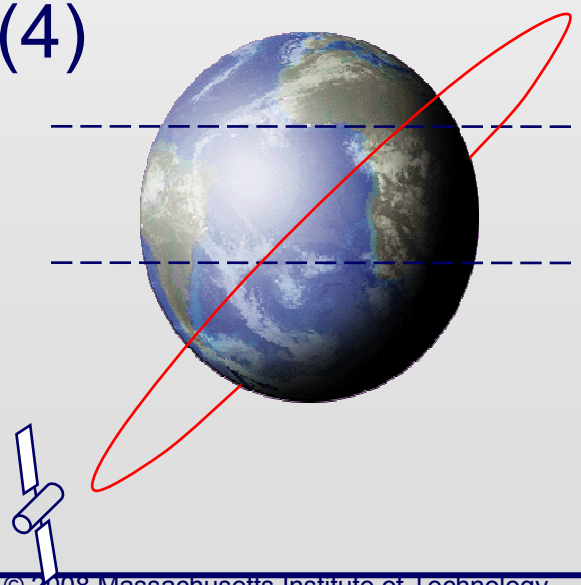
(2)



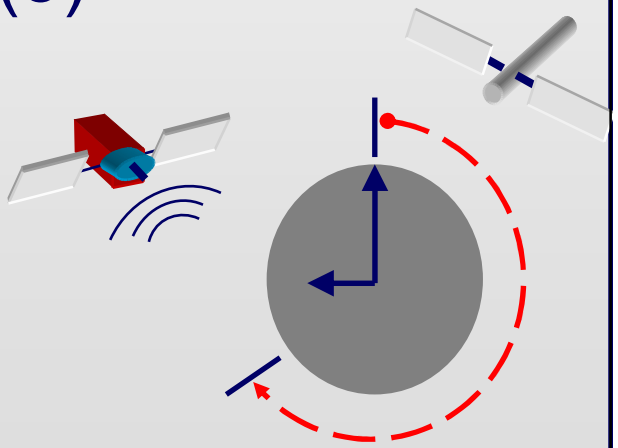
(3)



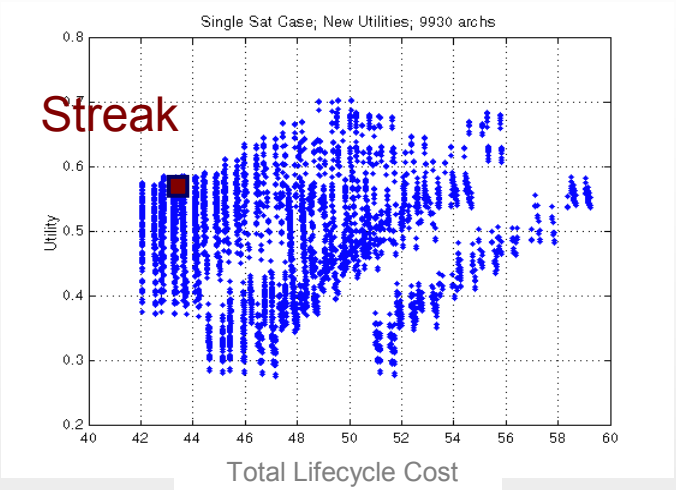
(4)

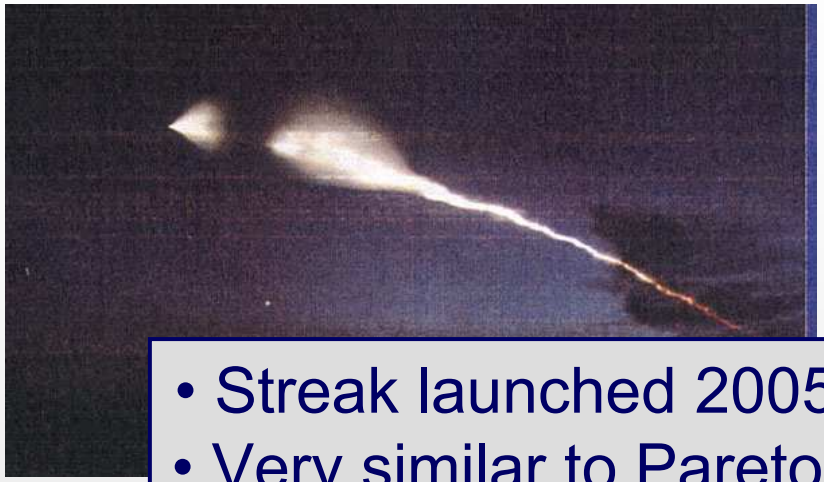


(5)



Streak - Successful System





- Streak launched 2005
- Very similar to Pareto X-TOS design

X-TOS vs. Streak

Inclination different

A user preference for data at the terminator unexpressed at time of X-TOS

	XTOS (2002 study)	Streak (Oct 2005 launch)
Wet Mass kg	325 - 450	420
Lifetime (yrs)	2.3 - 0.5	1
Orbit	300 -185 km @ 20°	321a-296p -> 200 @ 96°
Launch Vehicle	Minotaur	Minotaur
Utility	0.61 - 0.55	0.57 - 0.54*
Modified Utility**	0.56 - 0.50	0.59
Cost $M	75 - 72	75***
Instruments	Three (?)	"Ion gauge and atomic oxygen sensor"

Extra fuel allowed orbit changes to increase value

Streak values from information in published sources (Aviation Week)
*Calculated with X-TOS preferences
**Modified to account for changed preferences
***Estimate using X-TOS model

Making Decisions

At a fundamental level, design is about constrained "choice"

- Designers: choice of tools, concepts, colleagues, work hours, technology, etc.
- Users: CONOPS, reflected needs, anticipated needs, risk aversion and gaming, etc.
- Customers: benefit at cost, whose benefit, time value of money, etc.

Question: What makes a good "choice"?

Answer: Perceived benefit?

- Complicating factors: time, uncertainty, relative vs. absolute costs and benefits, distribution of costs and benefits…

How can design be improved through a "choice" point of view?

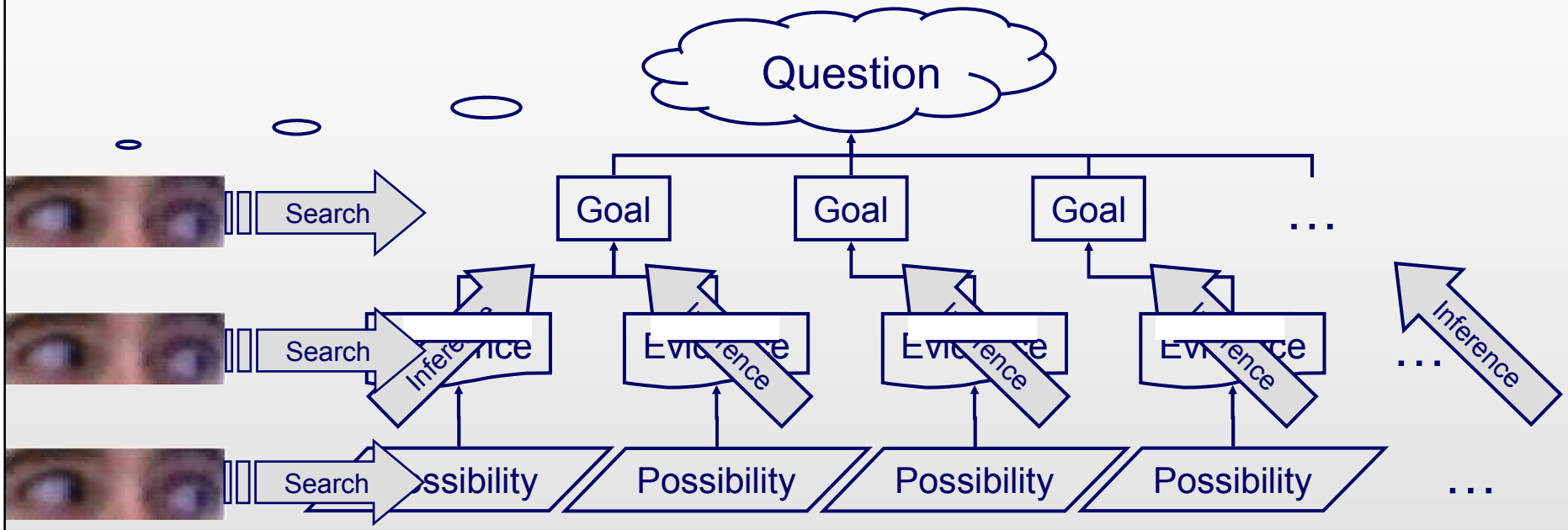
An Example Decision

Objectives	Goals	Possibilities
Buy a good house	Low price/sq ft	1124 Elm St, Newton, MA
	Be near a supermarket	326 Harvard St, Cambridge, MA
	Good schools nearby	477 Main St, Cambridge, MA
	Large square footage	37 Spruce Ter, Somerville, MA
	Good prospect for property value increasing	63 Lloyd Pl, Brookline, MA
	Short commute to work	455 Patterson Blvd, Quincy, MA
	Low utilities cost	75 Lowell Hwy, Jamaica Plain, MA
	…	…

With so many possibilities, how to make sure choice is a “good” one?

Thinking and Deciding: Search-Inference Framework

A Model of Thinking: Search-Inference



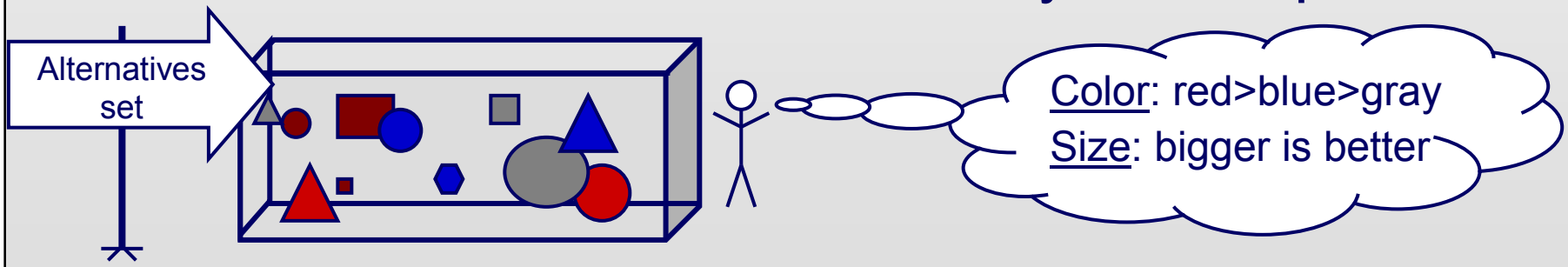
"For any choice there must be purposes or goals, and goals can be added to or removed from the list. …search for goals is *always* possible"*

People *will* change their minds

*Baron, Jonathan. *Thinking and Deciding: "Chapter 1—What is Thinking?"* Cambridge: Cambridge University Press, 2000. p7-9.

So design is about choice?

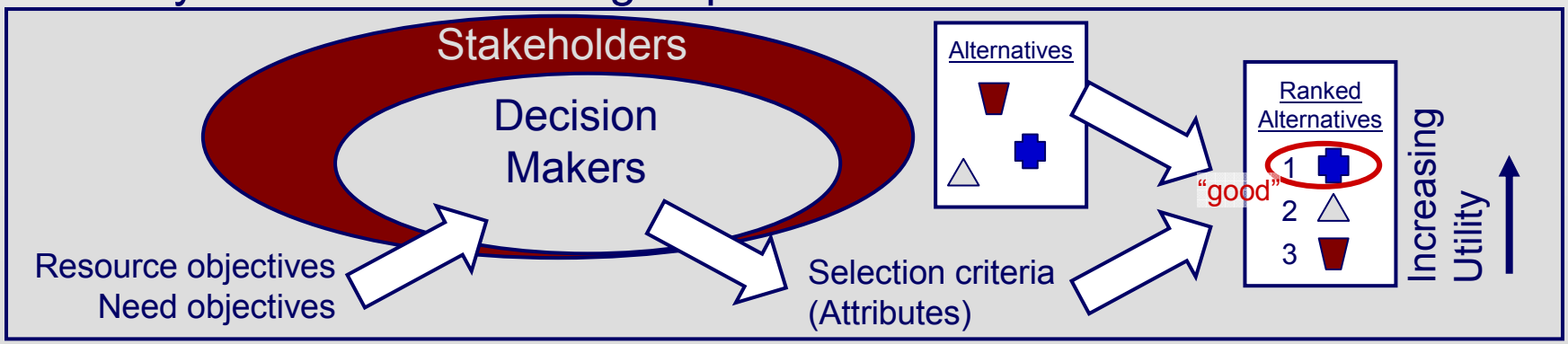
- "Choice" making a selection from a set of alternatives
- Classical decision theory concerns this problem
- Design encompasses a special class of decision problems: "wicked"
  - Open set of alternatives (infinite(?) possibilities)
  - Multi-criteria selection rule (multiple goals)
- Not a well-defined, theoretically solved problem…



Actually "Design" is about creating "good" alternatives

Stakeholders-Attributes-Utilities

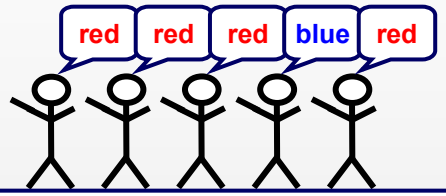
- In order to ensure a successful mission, the implied value proposition must be fulfilled
- Each system **stakeholder** has a value proposition—what they want to "get out" of the mission
- Decision makers are stakeholders with influence over the mission objectives for needs and/or resources
- Meeting the objectives for each decision maker can be assessed in terms of "**attributes**"
- An alternative that scores well in a set of attributes gives a decision maker value, or "**utility**"
- The goal for the selection of a good alternative is to maximize the utility for individuals and groups



"Listening" to the Need

Articulated: **Color** (red v. blue)

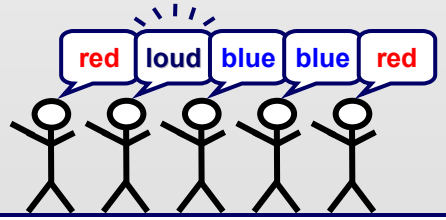
Unarticulated: ?



Should box be red or blue?

Articulated: **Color** (red v. blue), **Sound** (loud)

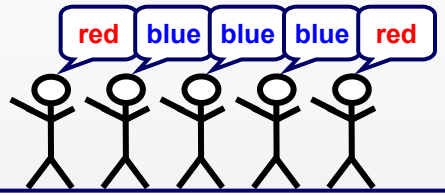
Unarticulated: It was **Sound**… how do others feel?



Now, what? Should box be red or blue, and loud?

Articulated: **Color** (red v. blue)

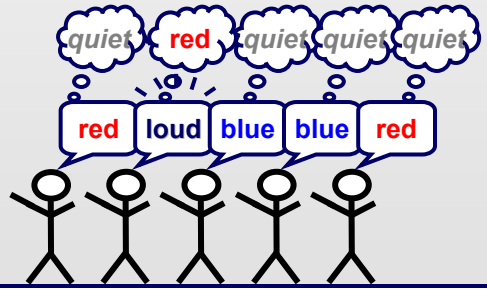
Unarticulated: ?



Now, should box be red or blue?

Articulated: **Color** (red v. blue), **Sound** (loud)

Unarticulated: **Color** (red), **Sound** (*quiet*)



Now, what? Should box be red or blue, and loud or *quiet*?

How can system deal with articulated *and* unarticulated value?

A Tough Design Question

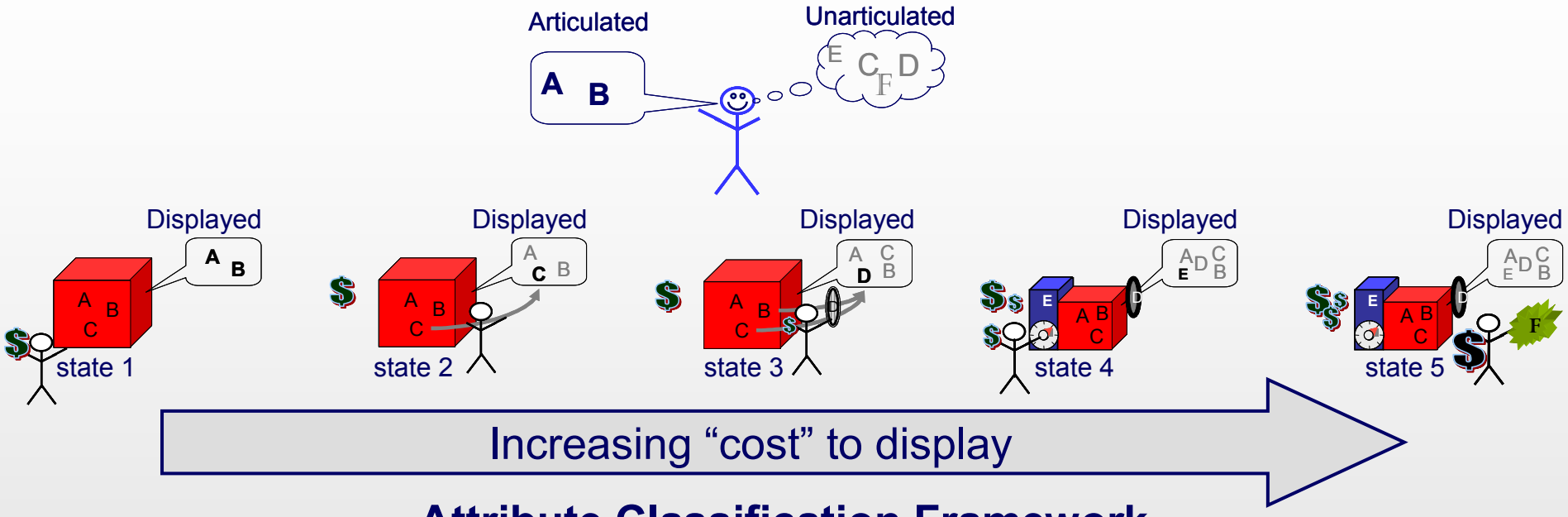
How can a system deal with articulated *and* unarticulated value?

Answer 1: Design a **versatile** system,
one with extra value already built-in*

Answer 2: Design a **changeable** system,
one with value accessible at reasonable cost

*But beware of value-detracting “extra” features…

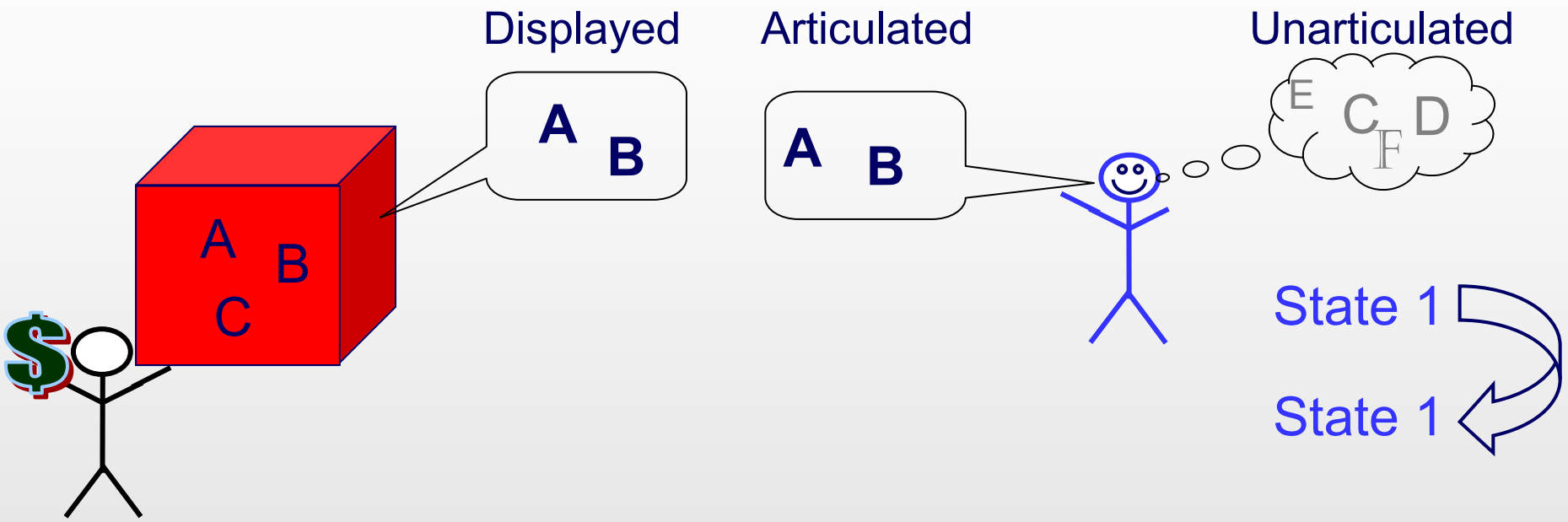
Designing for Unarticulated Value



Attribute Classification Framework

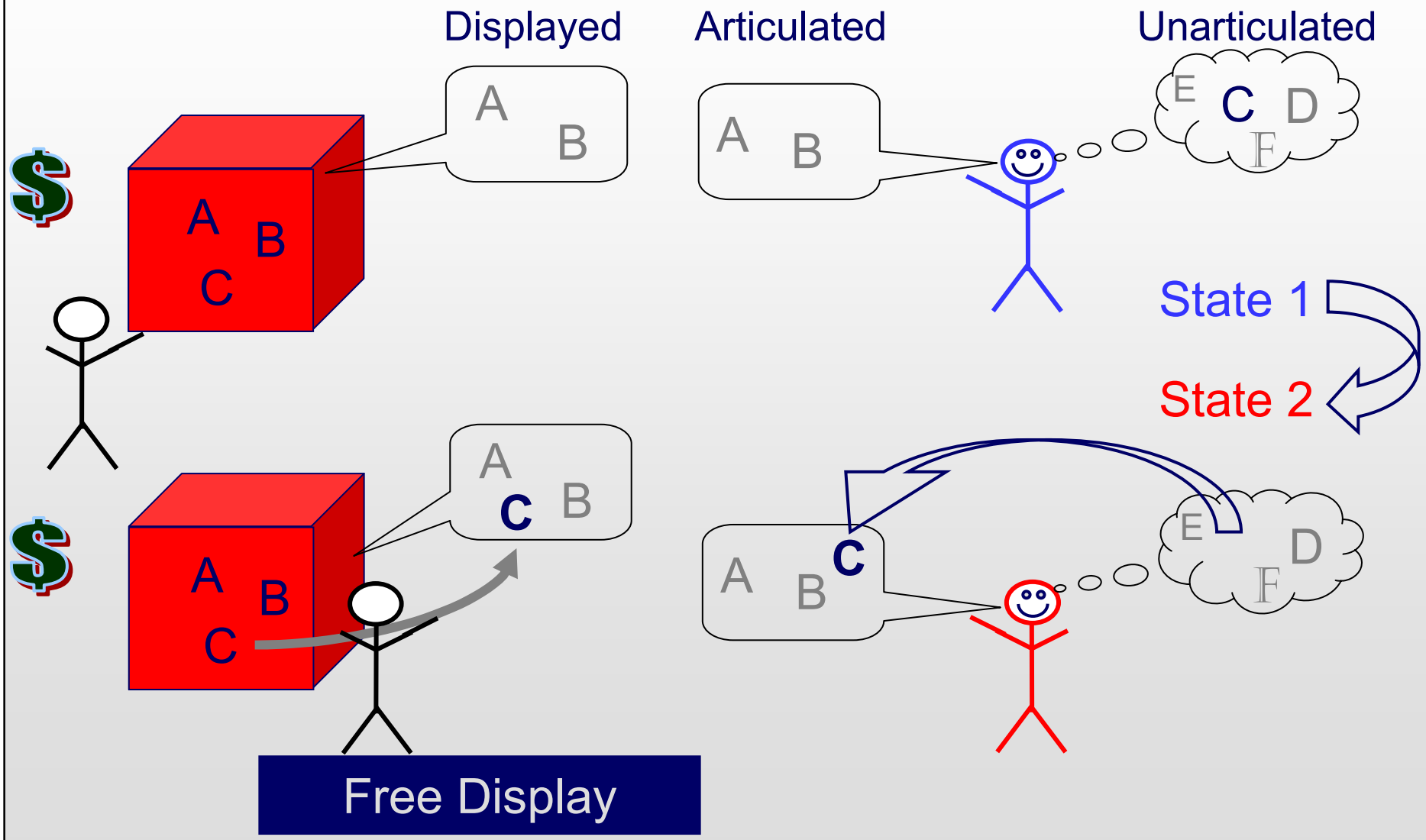
Class	Name	Property of Class	Cost to Display
0 (state 1)	Articulated Value	Exist and assessed	0
1 (state 2)	Free Latent Value	Exist, not assessed	0
2 (state 3)	Combinatorial Latent Value	Can exist by recombining class 0 and/or 1	Small
3 (state 4)	Accessible Value	Can be added through changing system (scale or modify)	Small → large
4 (state 5)	Inaccessible Value	Cannot be added through changing system (system too rigid)	Large → infinite

Class 0: Articulated Value

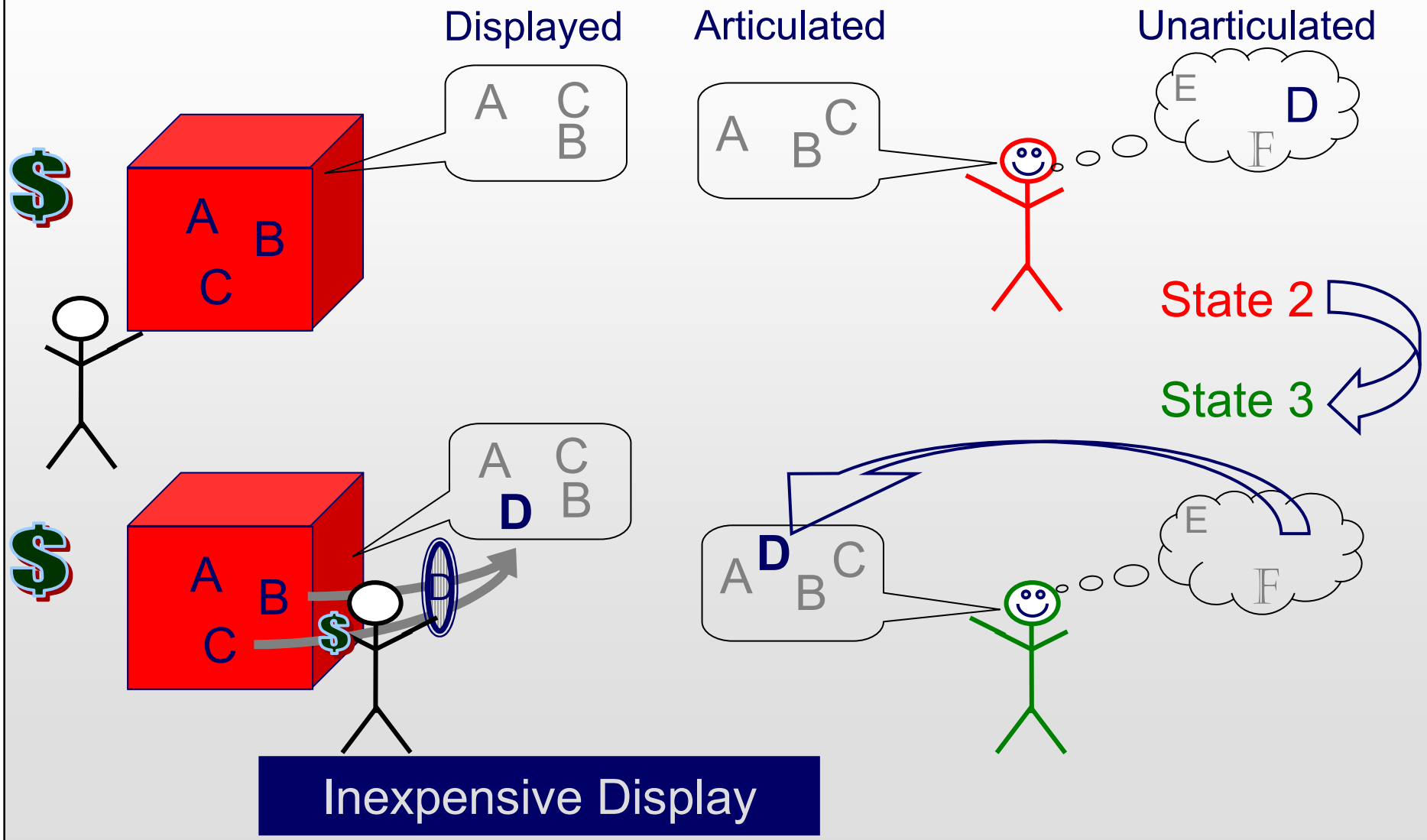


Static Value

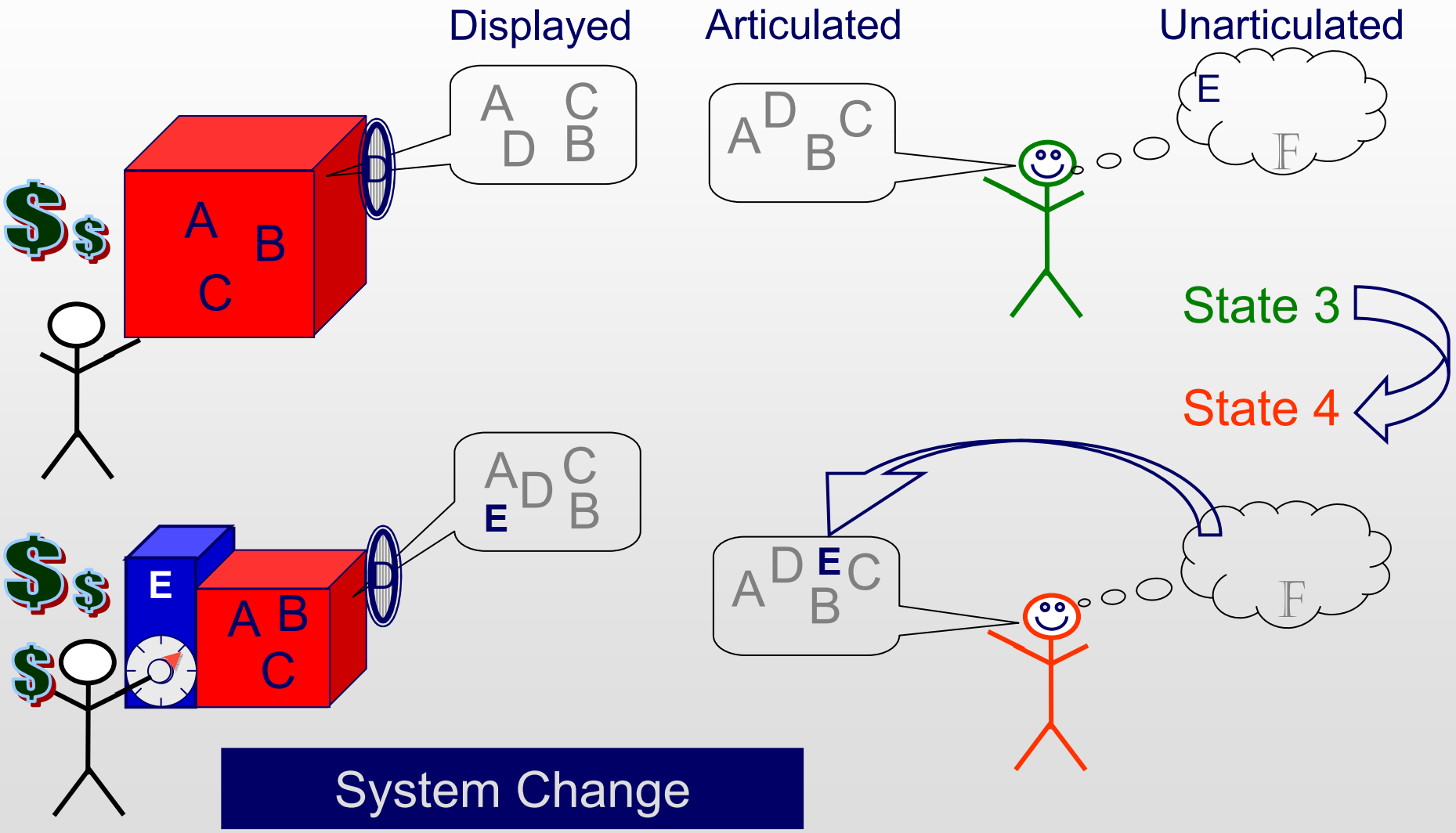
Class 1: Free Latent Value



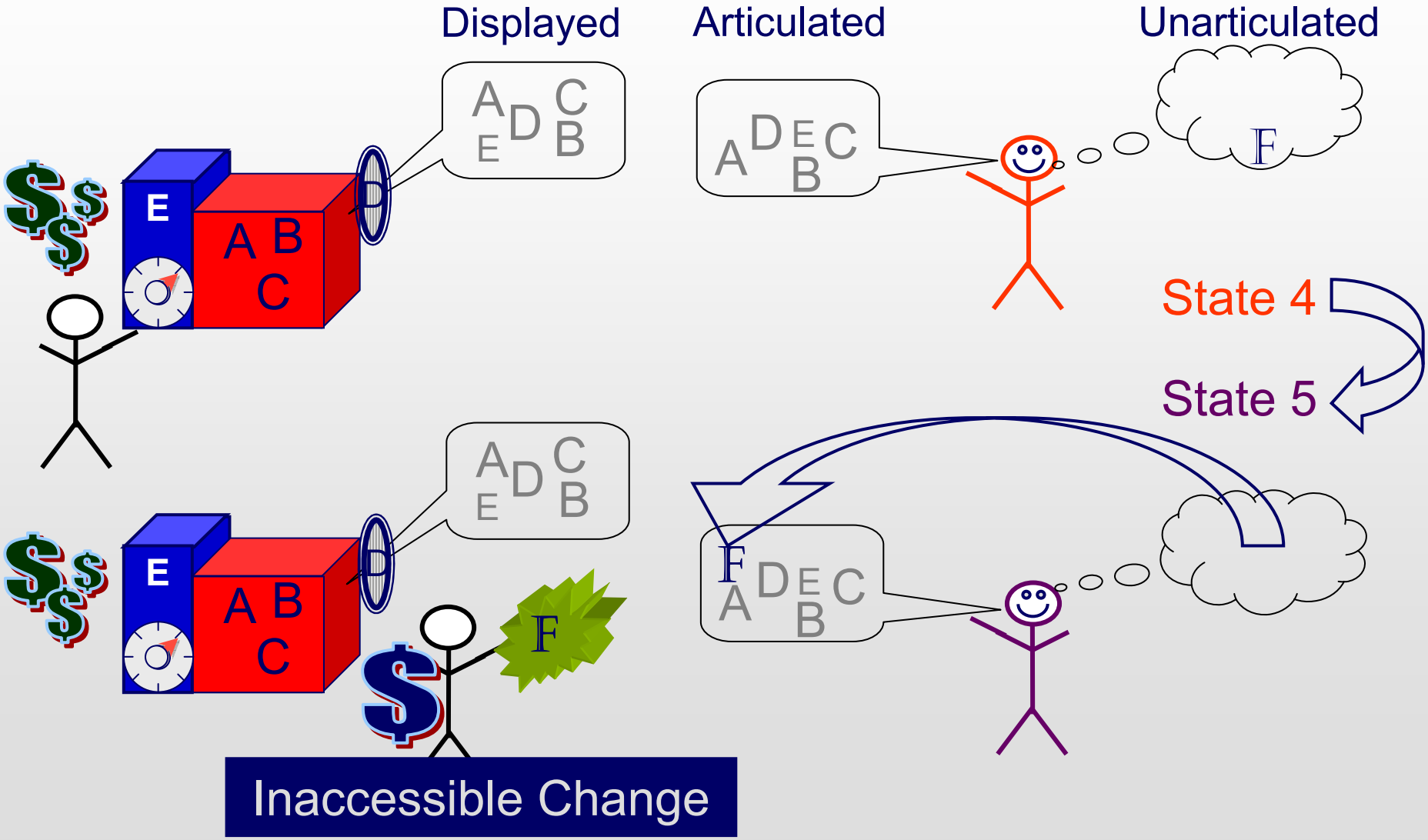
Class 2: Combinatorial Latent Value



Class 3: Accessible Value



Class 4: Inaccessible Value

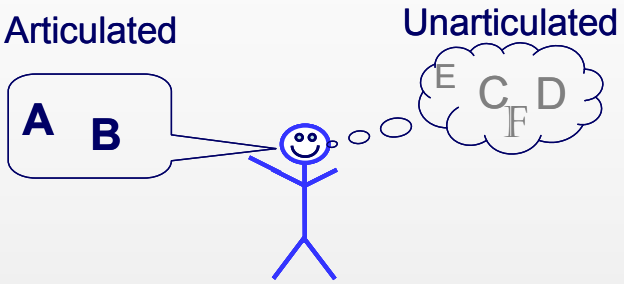


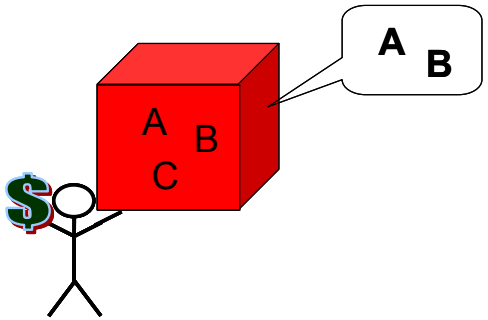
Needs Articulation Relates to Dynamic Preferences

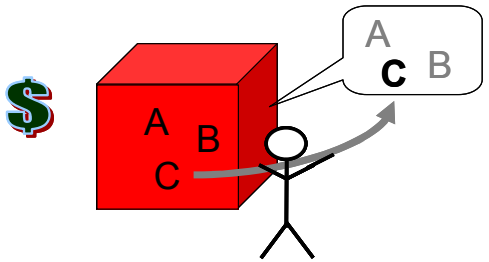
- Current preferences include current articulated and unarticulated needs
- Current preferences are a snapshot in a dynamic stream
- Causes of apparent dynamic preferences
  - Personal drift
  - Changing context
  - Articulation of unarticulated needs
- Maximizing delivered value requires matching dynamic preferences

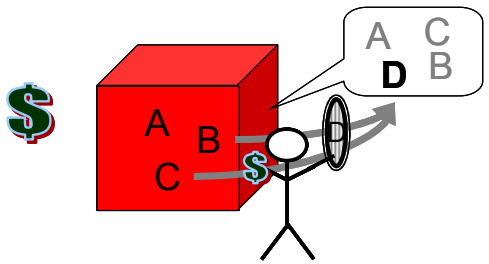
What about a framework for organizing preferences, both known and unknown?

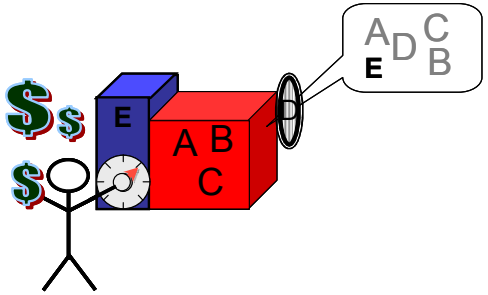
Sources of Value: Attribute Classes

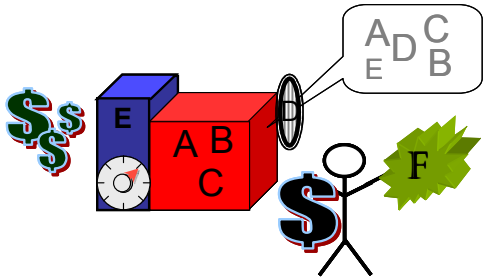












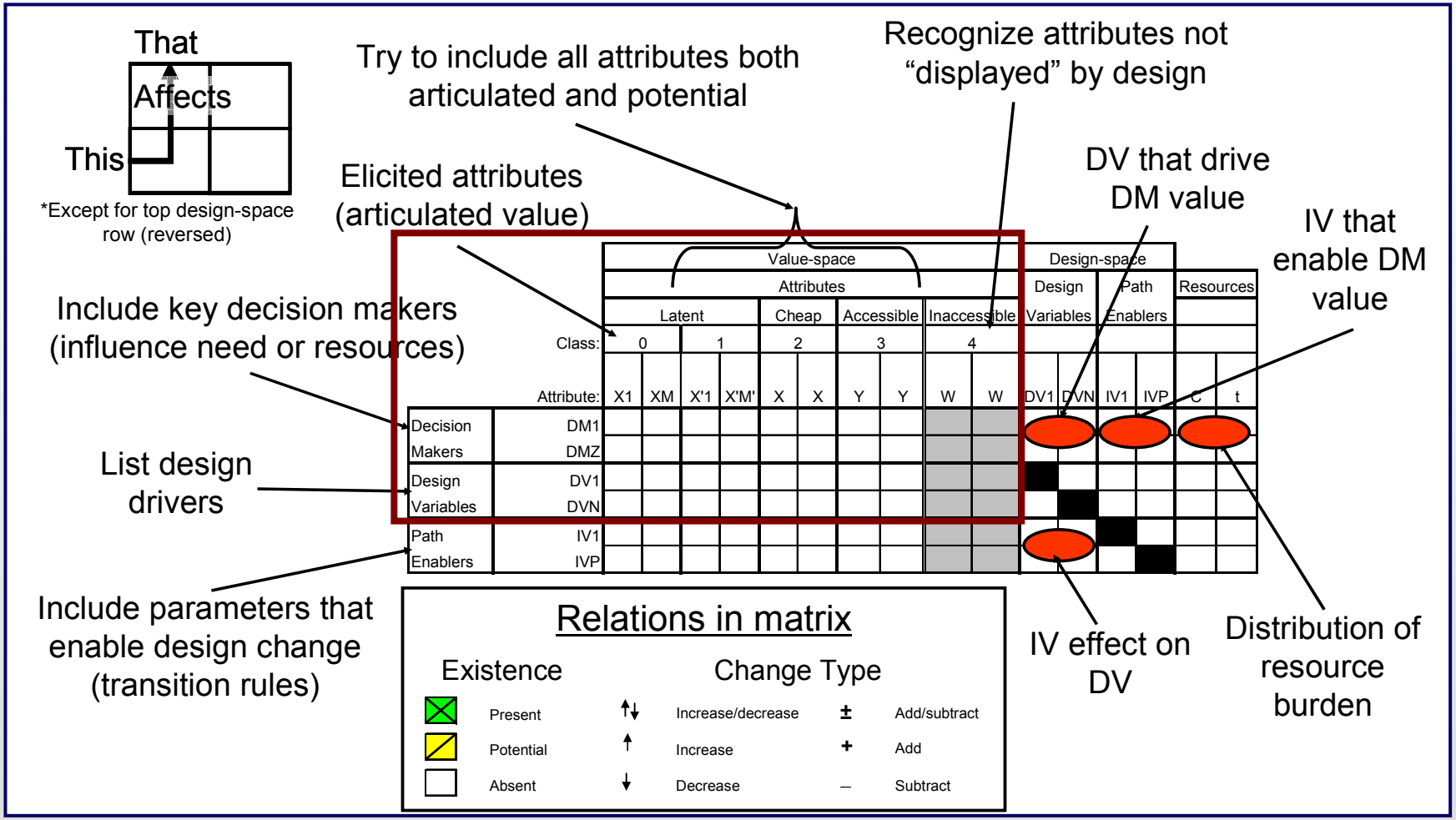
Attributes classified in terms of "cost" to display

Innovation from Observed Latent Value

A few years ago one of Nokia's designers visited China and noticed that people there used the light from their mobile phone screens to illuminate dark hallways so they could more easily unlock their doors. After he discussed his observation with other Nokia designers, Nokia added a penlight to some models.

Hoping to Make Phone Buyers Flip, NY Times, Feb 29, 2008

The Design-Value Matrix

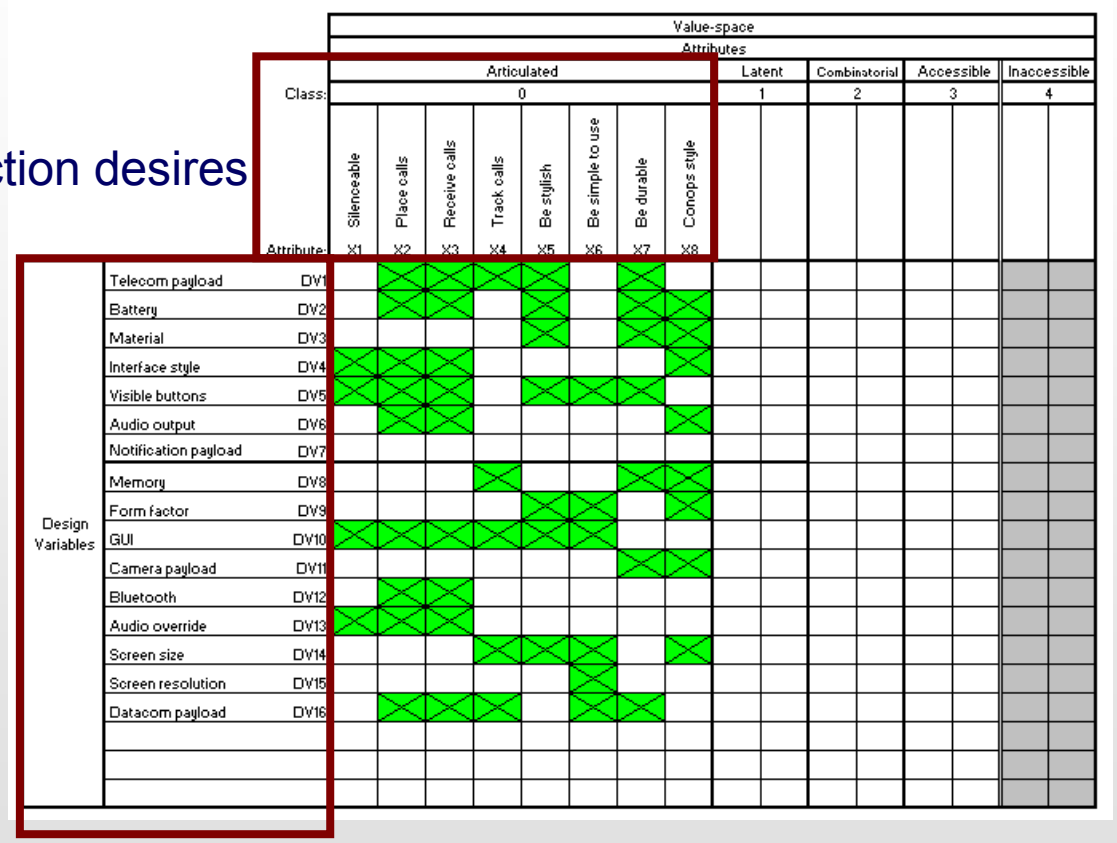


*The DVM is described at length in Ross, A.M., *Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration*, PhD in Engineering Systems, MIT, 2006.

Class 0: Articulated Value

Stated form/function desires

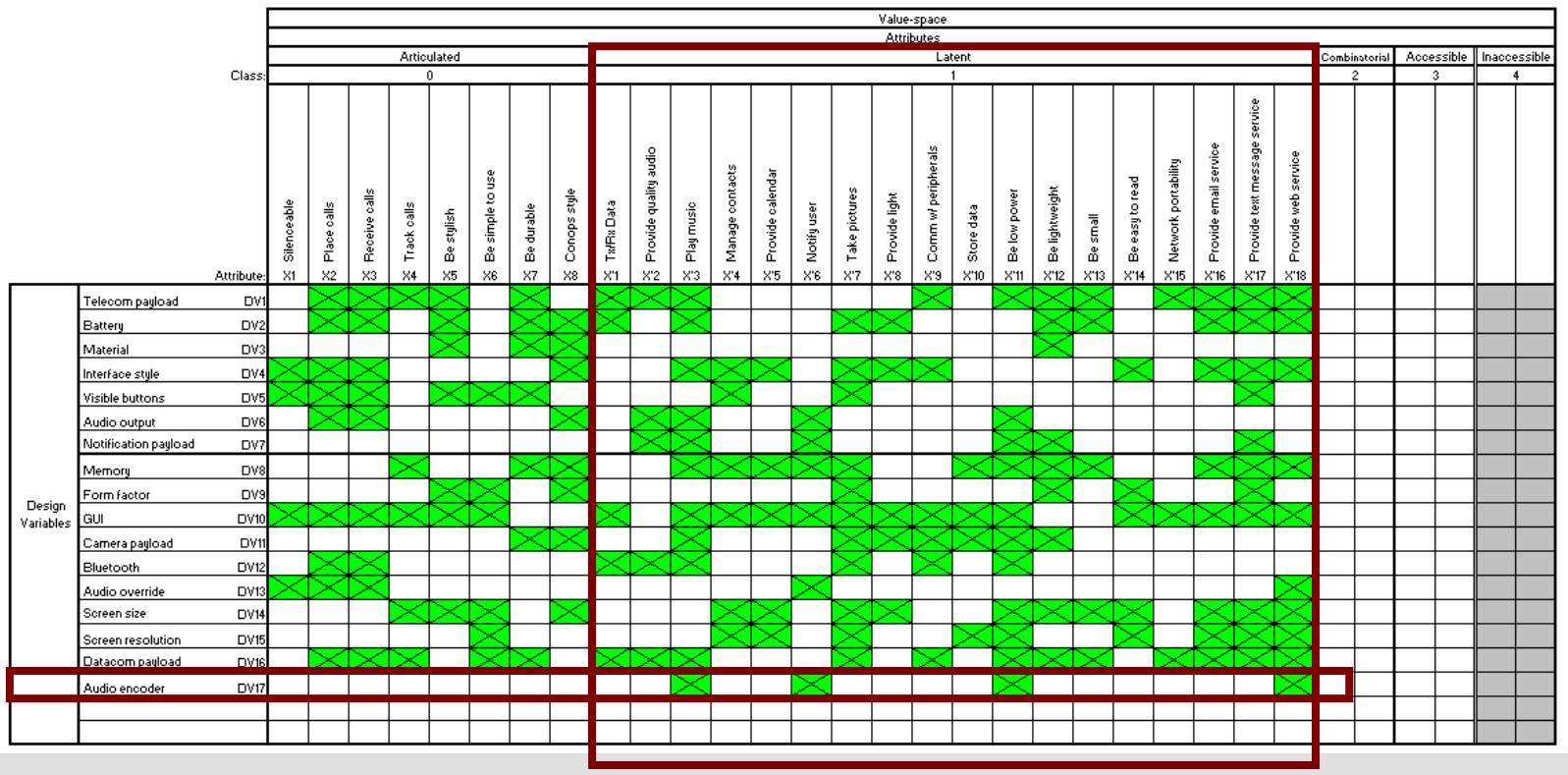
Value-drivers within "concept"



Designer-controlled parameters to drive value for the user

Class 1: Latent Value

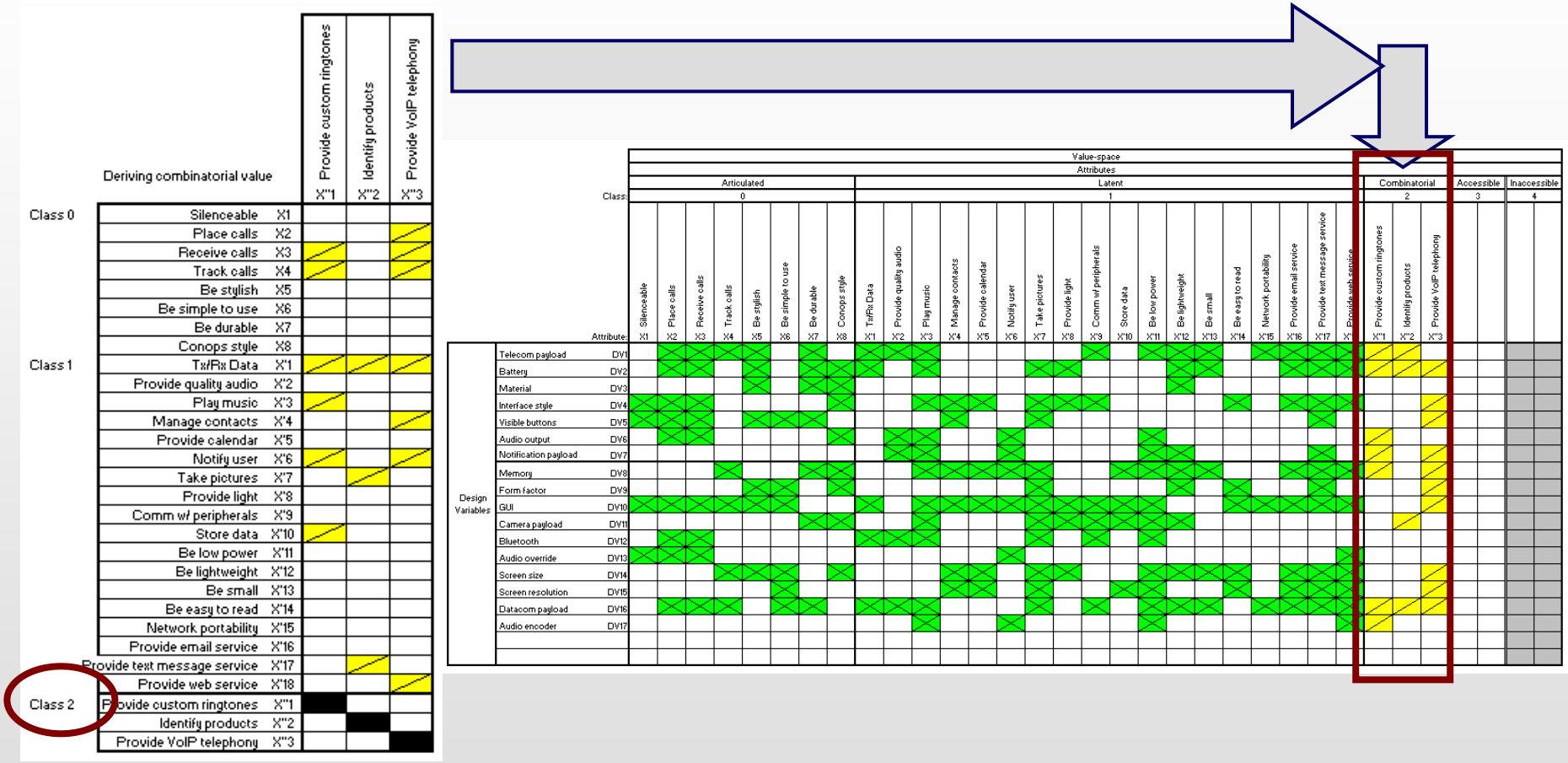
Given the design parameters proposed by designer, what other attributes are latent in the design?



Extra design parameter

For a system with multiple stakeholders, some latent attributes may be articulated by others

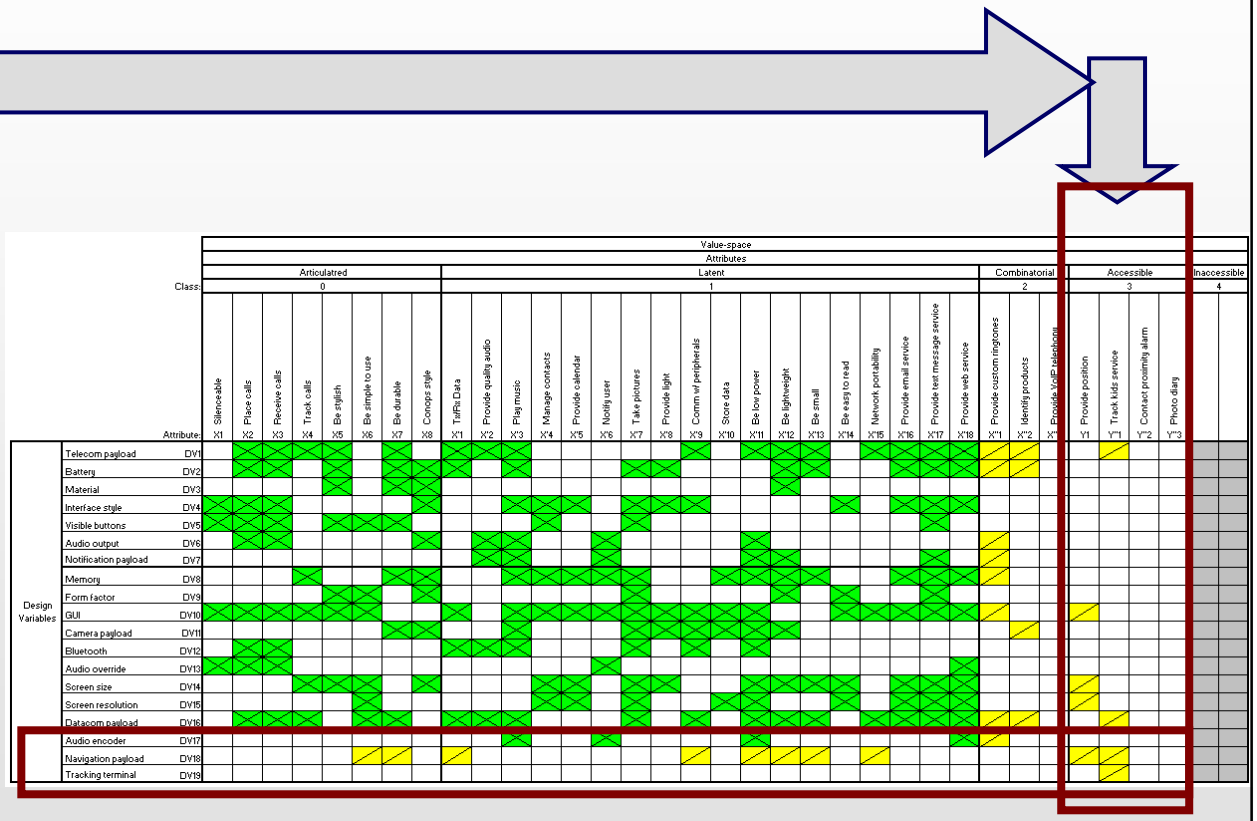
Class 2: Combinatorial Value



Using class 0 and 1, recombine to form “higher order” attributes

Class 3: Accessible Value

	Deriving combinatorial value		Provide custom ringtones	Identify products	Provide VoIP telephony	Track kids service	Contact proximity alarm	Photo diary
			X''1	X''2	X''3	Y''1	Y''2	Y''3
Class 0	Silenceable	X1						
	Place calls	X2			◢			
	Receive calls	X3	◢		◢			
	Track calls	X4	◢		◢			
	Be stylish	X5						
	Be simple to use	X6						
	Be durable	X7						
	Conops style	X8						
Class 1	TxRx Data	X'1	◢	◢	◢	◢	◢	◢
	Provide quality audio	X'2						
	Play music	X'3	◢					
	Manage contacts	X'4			◢		◢	
	Provide calendar	X'5						
	Notify user	X'6	◢		◢		◢	
	Take pictures	X'7		◢				◢
	Provide light	X'8						
	Comm w/ peripherals	X'9						
	Store data	X'10	◢					
	Be low power	X'11						
	Be lightweight	X'12						
	Be small	X'13						
	Be easy to read	X'14						
	Network portability	X'15						
	Provide email service	X'16						
	Provide text message service	X'17		◢		◢	◢	◢
	Provide web service	X'18			◢			◢
Class 2	Provide custom ringtones	X''1	■					
	Identify products	X''2		■				
	Provide VoIP telephony	X''3			■			
Class 3	Provide position	Y1				◢	◢	◢
	Track kids service	Y''1				■		
	Friend proximity alarm	Y''2					■	
	Photo diary	Y''3						■

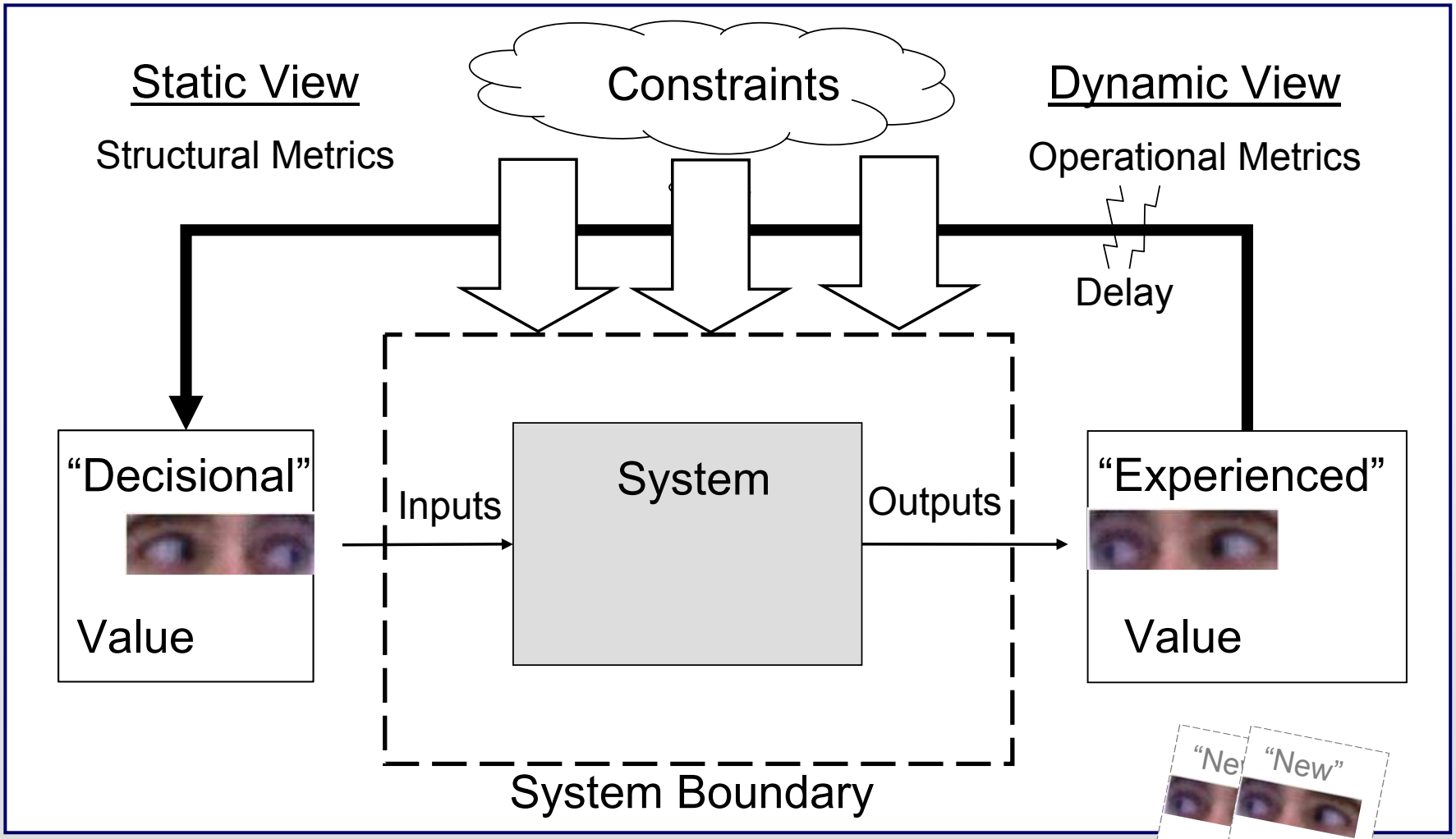


A system change (here, adding a "navigation payload") can enable new attributes and new combinatorial value

Addressing Changing Needs

- Building in “latent” value reduces the cost to display attributes as expectations change
- Process provides structured method for generating system latent value
- Cost for carrying latent value must be addressed
- Four operators for altering system “display” of attributes:
  1. Add attribute
     - Example: computer expansion through peripherals
  2. Reveal attribute
     - Example: software upgraded through purchased license key
  3. Remove attribute
     - Example: uninstalling software from new computer
  4. Hide attribute
     - Example: hiding extra buttons on universal remote

Dynamic System Context: Value Lenses



*"Decision", "Experienced", and "Remembered" Utility from (Kahneman 2000)

**Discussion of "structural" versus "operational" metrics in (Giachetti et al. 2003)

System Mask as a Filter to Hide Undesired Attributes

User desires “A”

User sees “A”

Success!

User desires “AB”

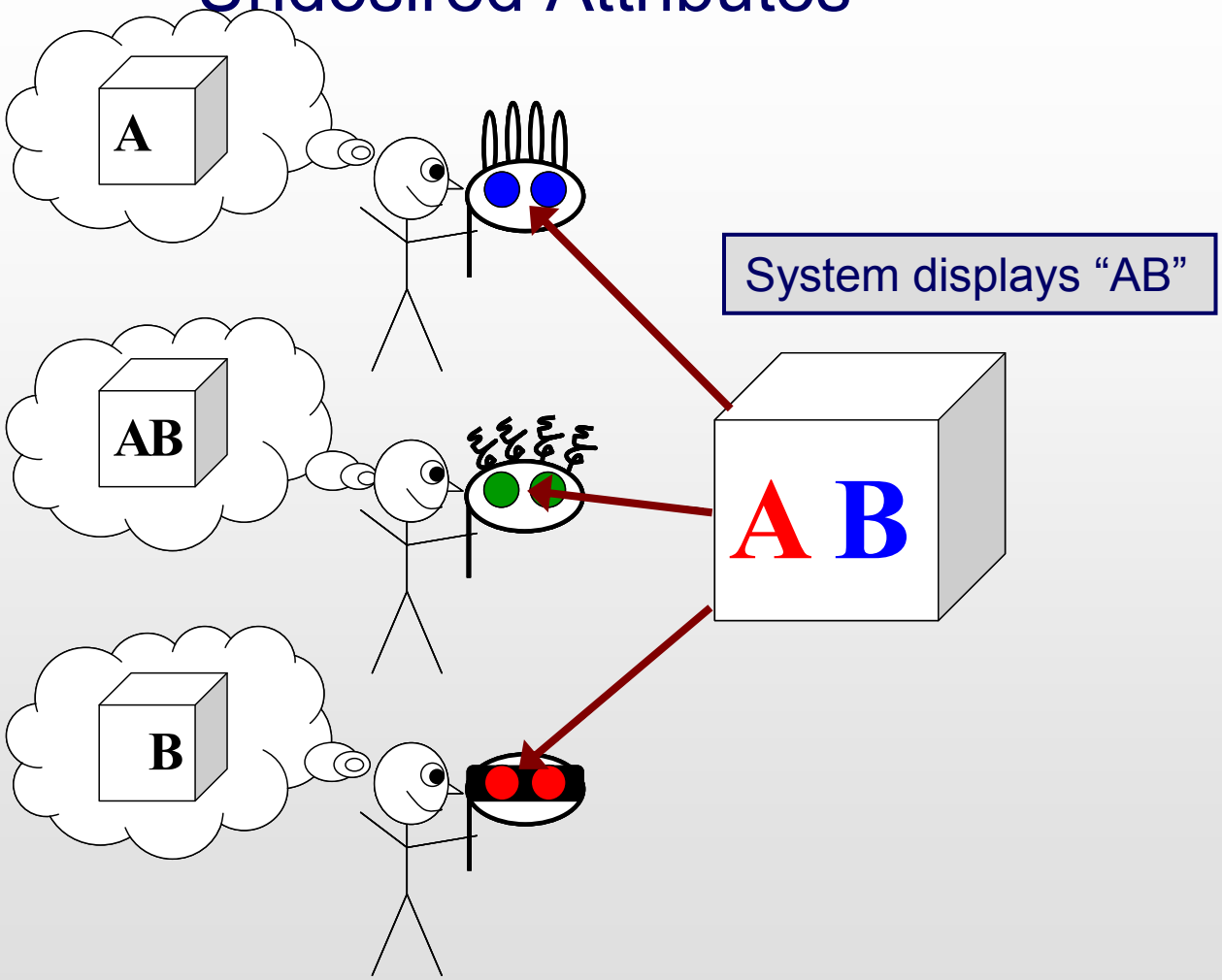
User sees “AB”

Success!

User desires “B”

User sees “B”

Success!



System Mask customizes “experience” to meet expectations

From Ross, A.M. and Rhodes, D.H., “The System Shell as a Construct for Mitigating the Impact of Changing Contexts by Creating Opportunities for Value Robustness,” 1st Annual IEEE Systems Conference, Honolulu, HI, April 2007.

Examples: Mask

Cellular Phone "Appearance"

GPS System "Appearance"

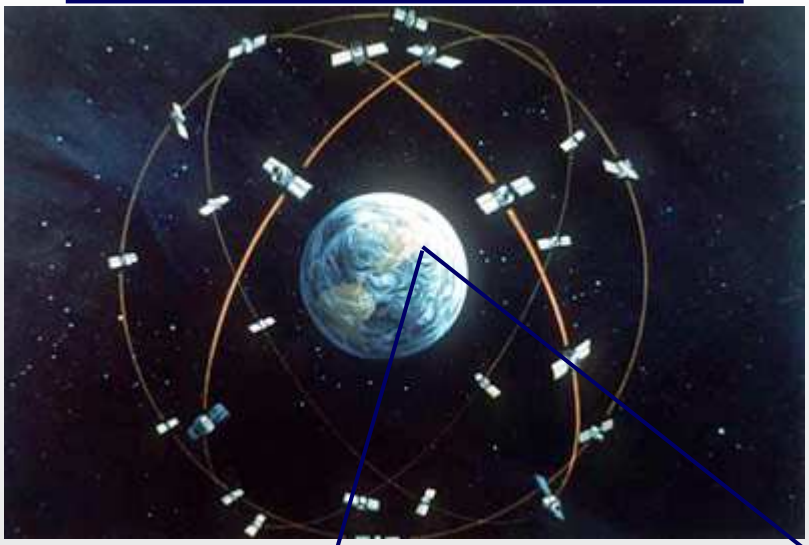


Photo courtesy U.S. Department of Defense

Solution: Faceplates

Solution: Various interpretive receivers

Photo courtesy Garmin

Images from http://www.akcessories.com/index.html?lmd=39141.647419

Masks filter "appearance" of system to context more cheaply than modification of system itself

From Ross, A.M. and Rhodes, D.H., "The System Shell as a Construct for Mitigating the Impact of Changing Contexts by Creating Opportunities for Value Robustness," 1st Annual IEEE Systems Conference, Honolulu, HI, April 2007.

Sustaining Value through Attribute Classes

The fundamental trade-off:

Build system with large latent value, or build custom systems for each need

- Both cost and time for revealing “new” system must be considered
- Large, long-development time systems may benefit from maximizing latent value
- Balance add/reveal attributes and remove/hide attributes to manage development and “display” costs

Attribute classes help designers manage unarticulated value by allowing for explicit consideration of designing in extra value in a system

QUESTIONS?

http://seari.mit.edu

adamross@mit.edu