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 9-12, 2007

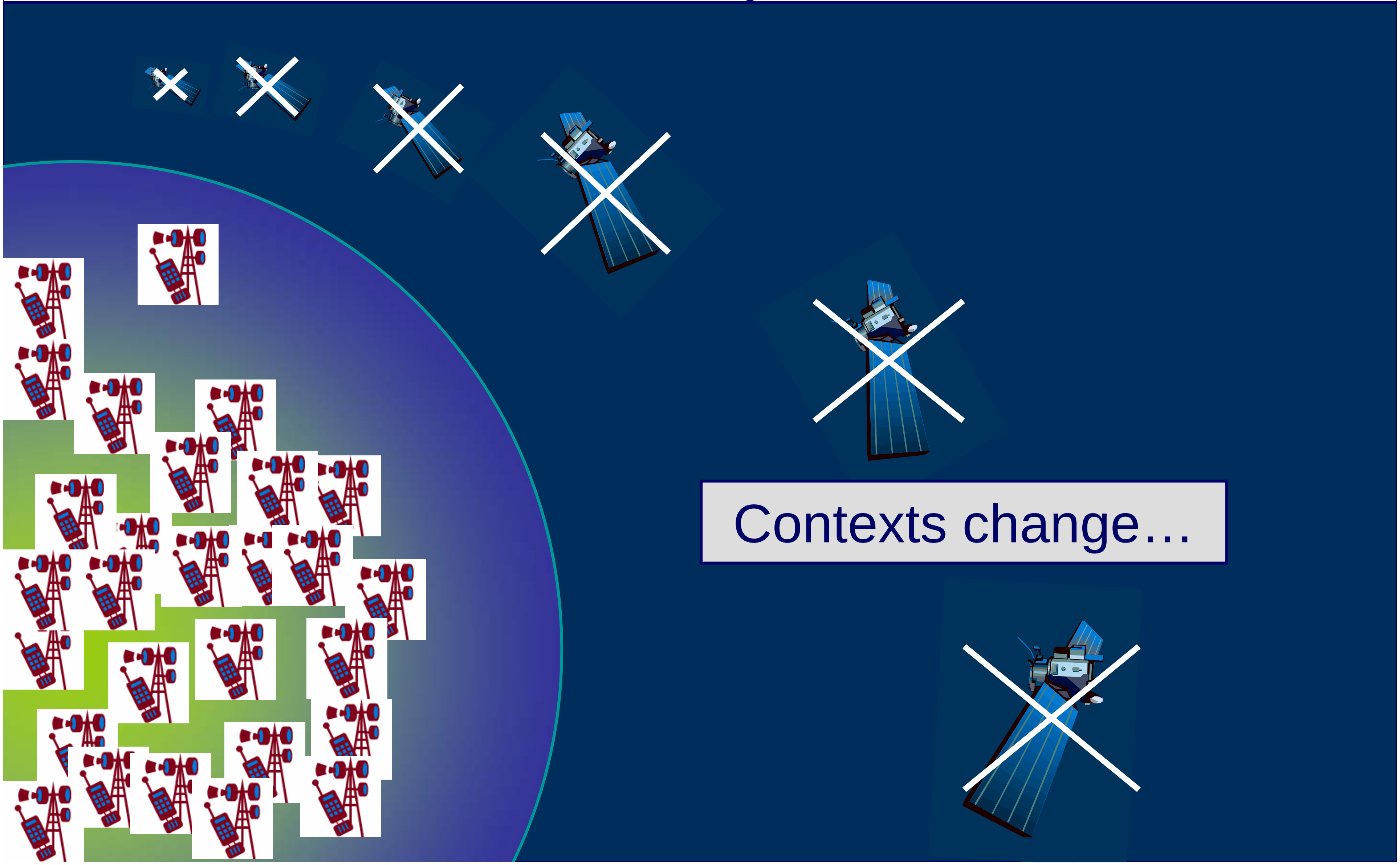
Presented by Adam Ross, Ph.D.

April 11, 2007

Achieving System Success

- Goal of design is to create value (profits, usefulness, voice of the customer, etc…)
- Requirements capture a mapping of needs to specifications to guide design

Deploying a "Valuable" System…



Achieving System Success (cont.)

- Goal of design is to create value (profits, usefulness, voice of the customer, etc…)

- People change their minds; the world changes…
- To continue to deliver value, systems must cope with context change…

What is Context?

- Context includes forces exogenous to system
  - Stakeholder expectations
  - Operating environment
  - Policy constraints
  - Available technologies
  - Competitive market
  - Etc…
- System success depends on system performance within a given *context*
- In order to ensure success, designers must consider context beyond traditional “operating environment” (classical robust design)

What is System Success?

Success is defined across multiple perspectives and multiple time periods

System success, Ψ, across N decision makers at time t

Net "experience" Net "expectations"

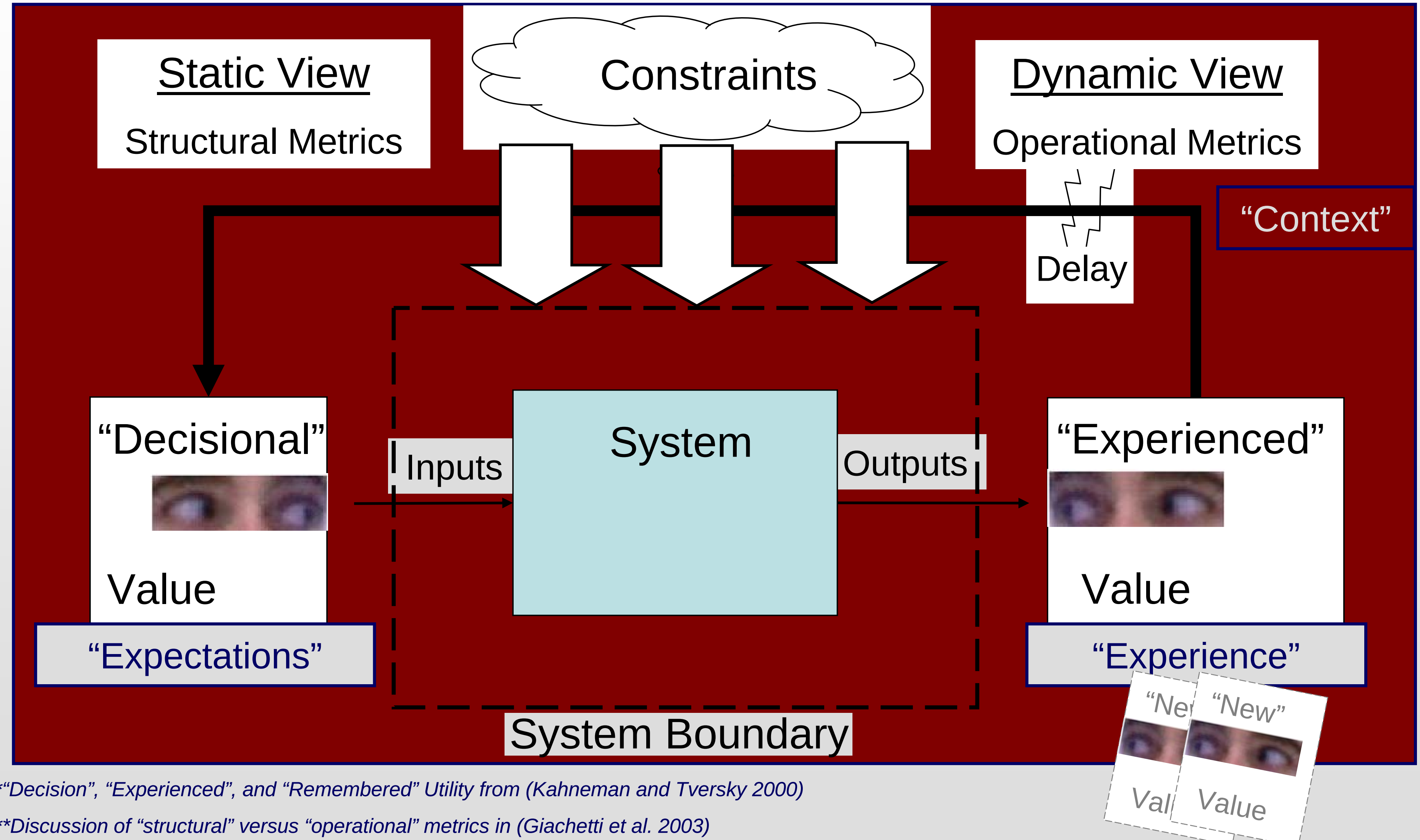
$$\Psi(t) = \sum_{i=1}^{N} \left[X_{DMi}(t) + \varepsilon_C^{X_{DMi}}(t) \geq Y_{DMi}(t) + \varepsilon_C^{Y_{DMi}}(t) \right]$$

$$0 \leq \Psi(t) \leq N$$

$X_{DMi}(t)$	Decision maker i system "experience" at time t	$\varepsilon_C^{X_{DMi}}(t)$	Context effect on decision maker i "experience" at time t
$Y_{DMi}(t)$	Decision maker i system "expectation" at time t	$\varepsilon_C^{Y_{DMi}}(t)$	Context effect on decision maker i "expectation" at time t

System Success: Net "experience" must meet or exceed net "expectations"

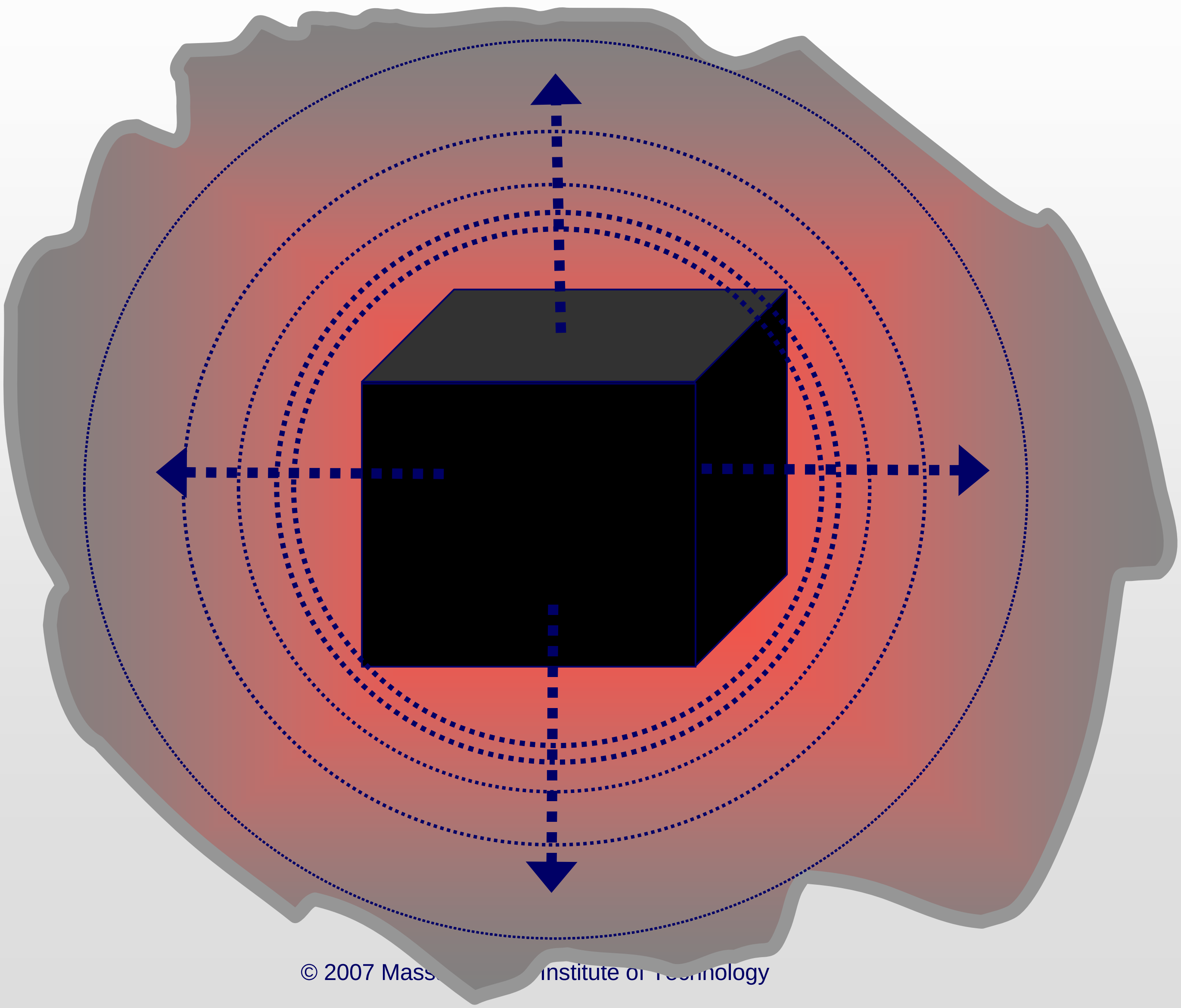
Dynamic System Context: Value Lenses



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

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

Construct: Black Box System

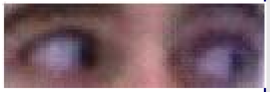


Black Box System

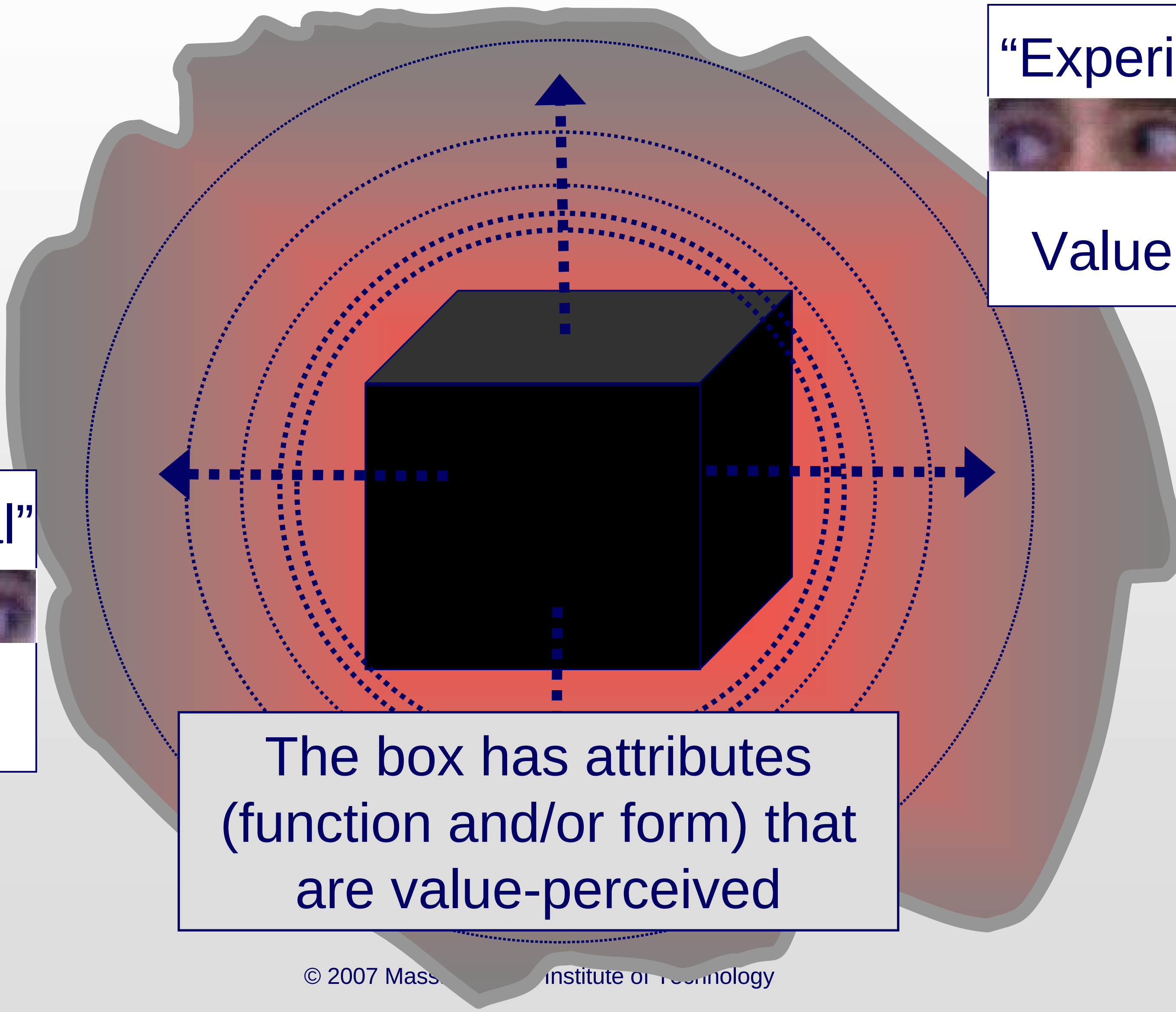
“Experienced”

Value

“Decisional”

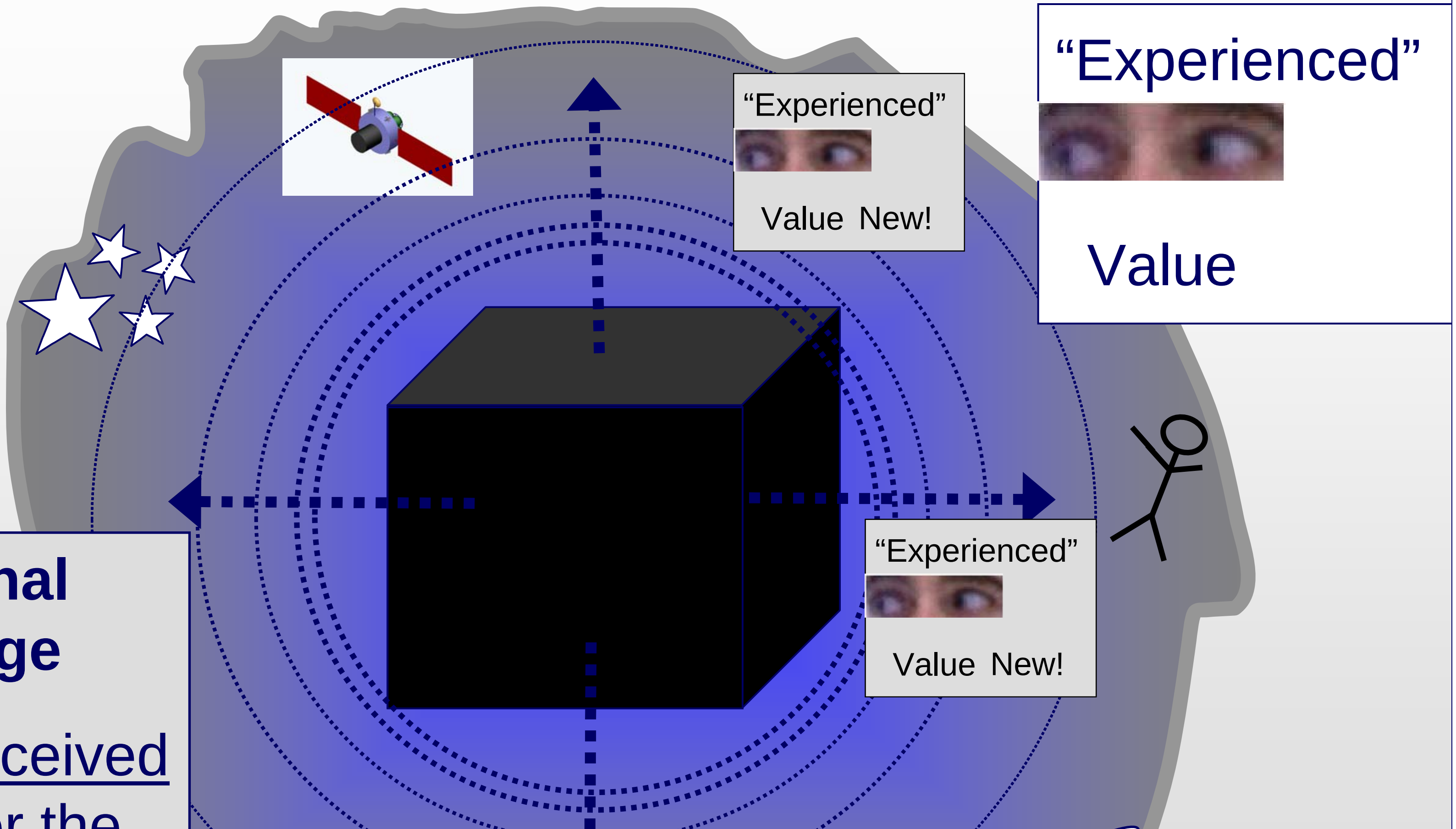


Value



The box has attributes (function and/or form) that are value-perceived

Value Robust Black Box

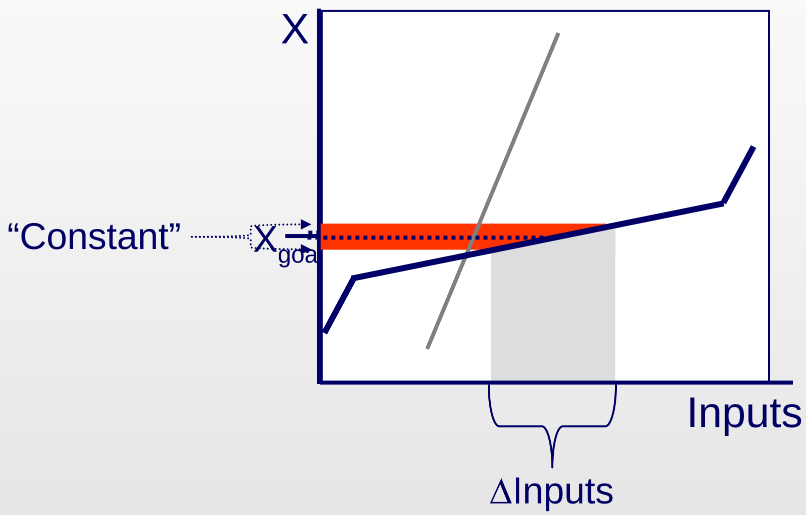


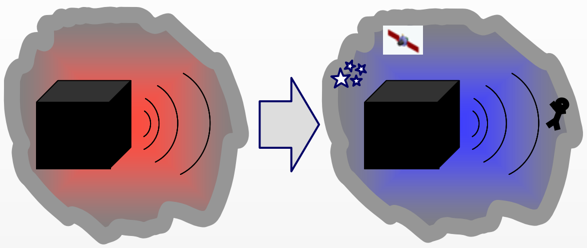
External Change

Box is perceived to deliver the same value

The box is robust if it can maintain its value delivery as external "context" changes

Robustness Defined





No change in perceived value

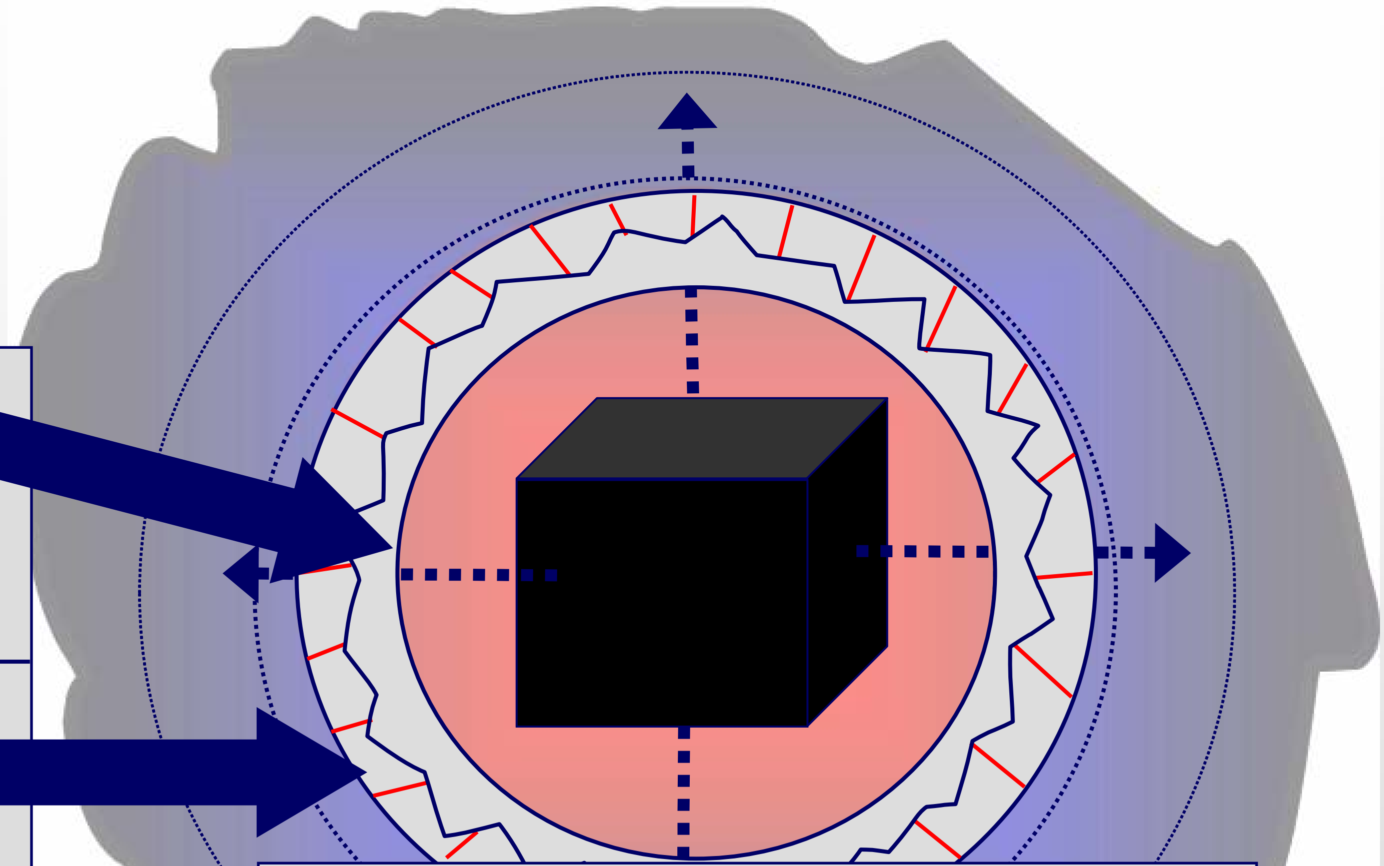
"Robust"

A box can be quantified in terms of *robust in X^i to "Input" change*

(i.e., can X^i remain "constant" over range of "Input"?)

Level of attribute performance is function of inputs (and constraints including environment), so robustness is an insensitivity to the inputs (and constraints)

System Shell Illustrated



Inner Shell
(The Shelter)

Change Context as seen by Box

Outer Shell
(The Mask)

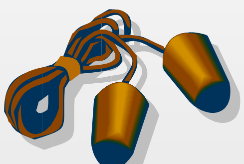
Change Box as seen by Context

The shell protects the box from changes in its context

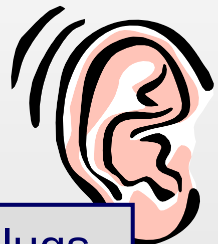
The shell filters the appearance of the box to its context

Examples: Shelter

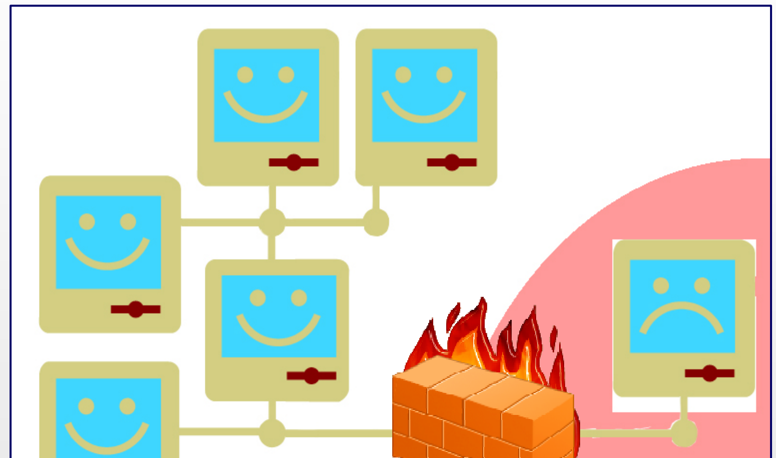
Context: Noisy environment



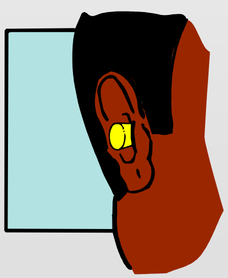
Solution: Earplugs



Context: Hostile external network



Solution: Firewall



Shelters “protect” system from context more cheaply than modification of system itself

System Mask as a Filter

User desires "A"

User sees "A"

Success!

User desires "AB"

User sees "AB"

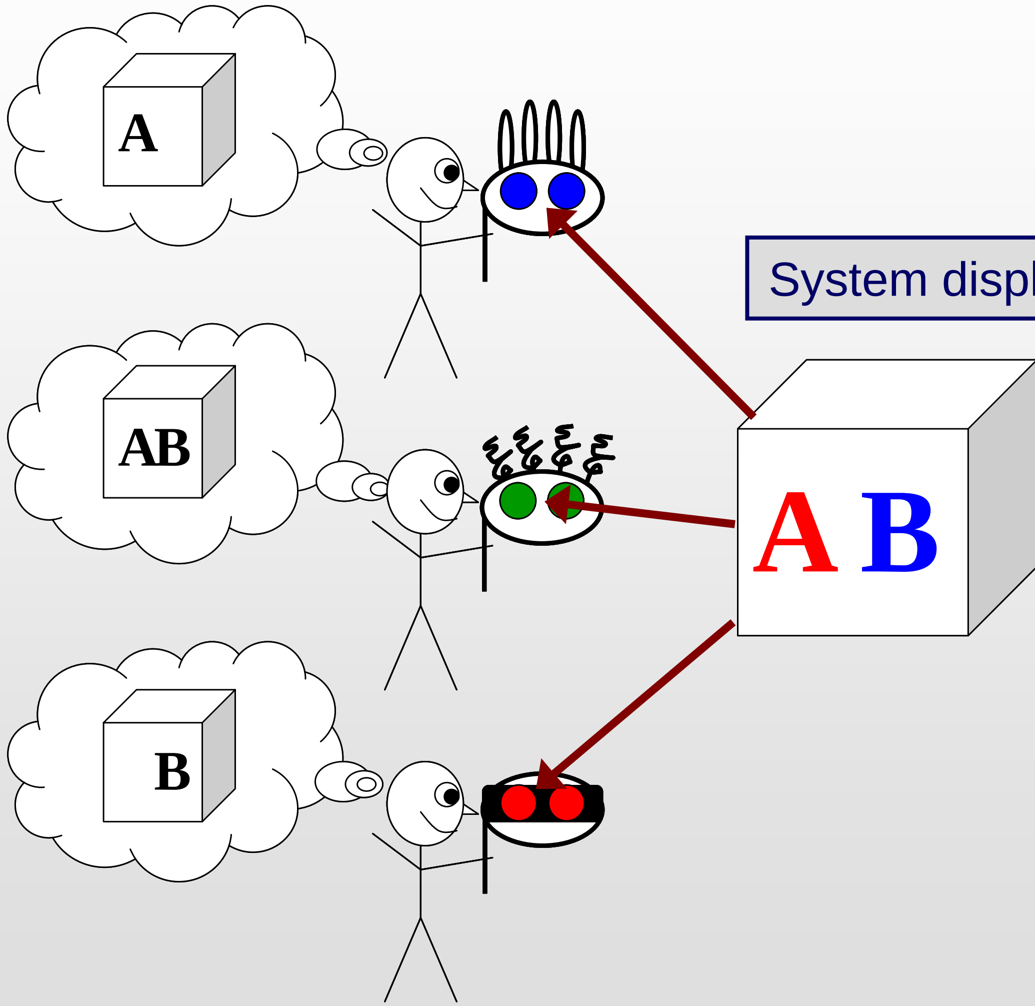
Success!

User desires "B"

User sees "B"

Success!

System displays "AB"



System Mask customizes "experience" to meet expectations

Examples: Mask

Cellular Phone “Appearance”

Solution: Faceplates

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

GPS System “Appearance”

Photo courtesy U.S. Department of Defense

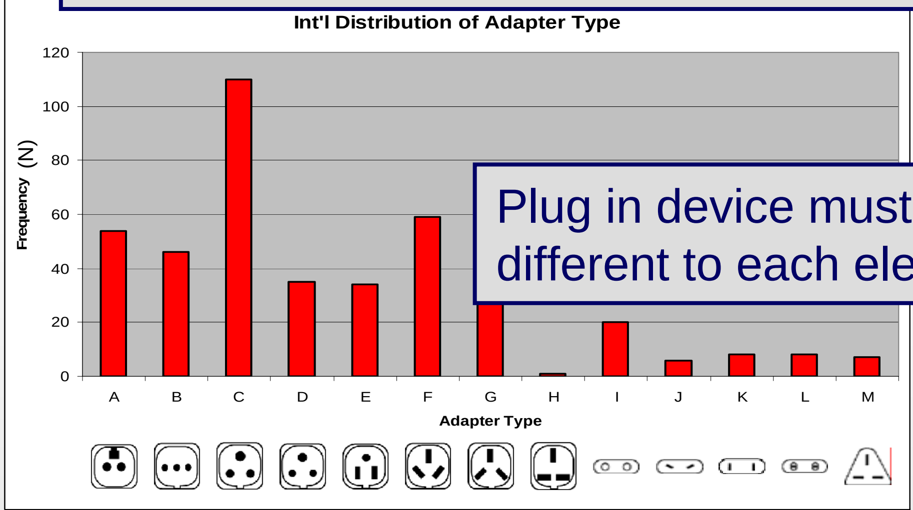
Solution: Various interpretive receivers

Photo courtesy Garmin

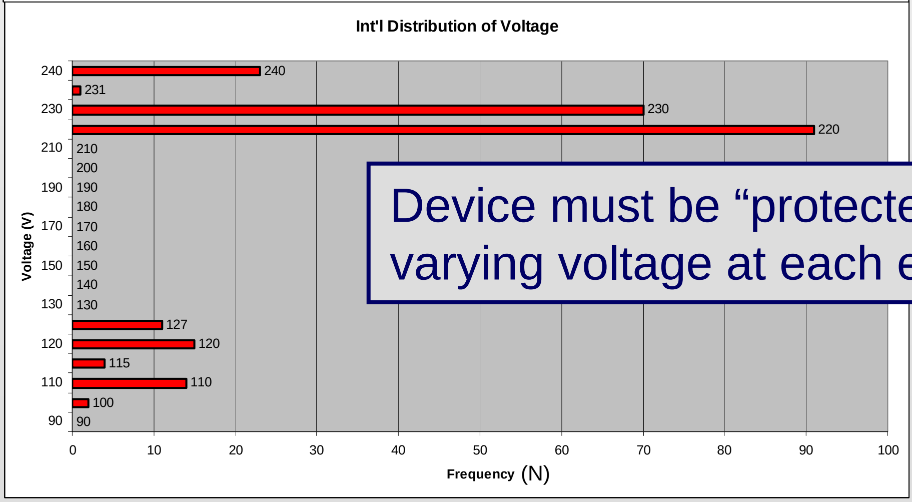
Masks filter “appearance” of system to context more cheaply than modification of system itself

Examples: Shell

Changing Context: International Electrical Power Sockets



Plug in device must "look" different to each electric socket



Device must be "protected" from varying voltage at each electric socket

Source: http://www.kropla.com/electric2.htm

Solution: Adapter (Mask) and Converter (Shelter)

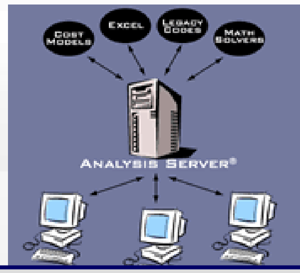
Image from Amazon.com, Samsonite Converter/Adaptor Plug Kit with Pouch

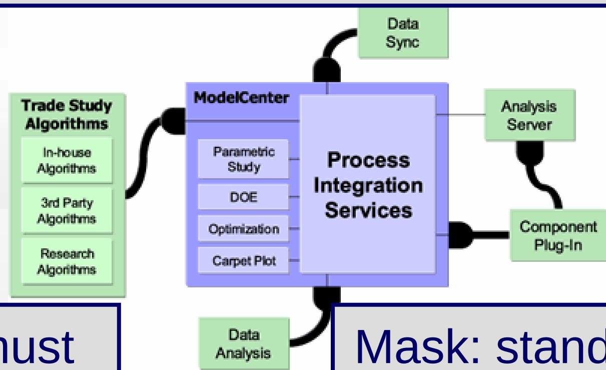
Examples: Software Shell

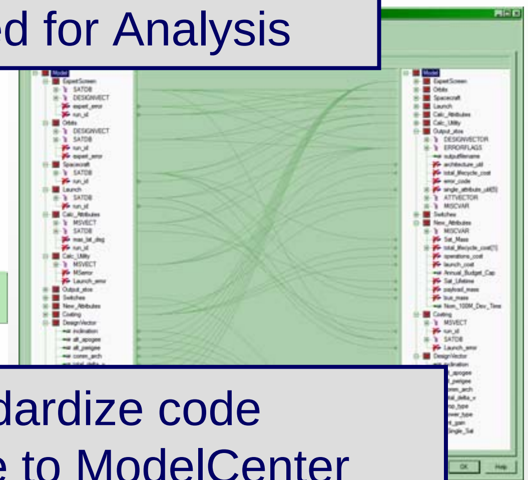
Systems Engineering Advancement Research Initiative

Changing Context: Software Models to be Linked for Analysis

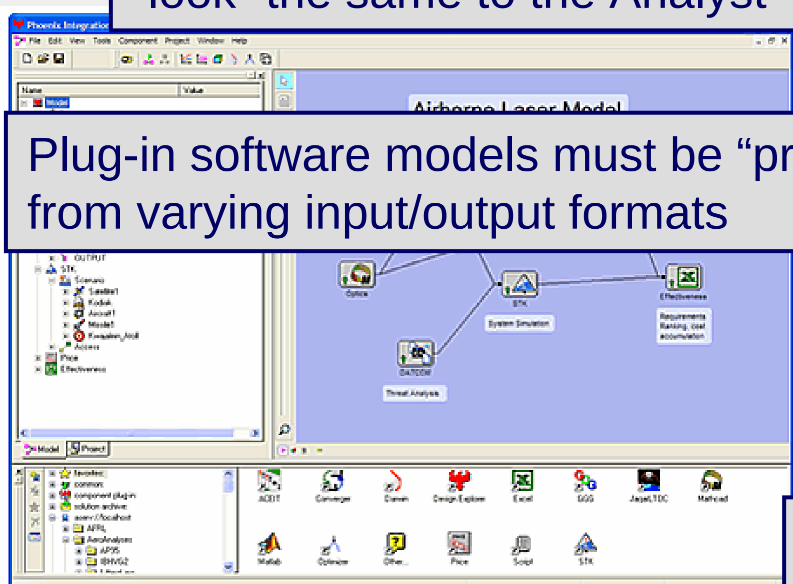
PHOENIX INTEGRATION

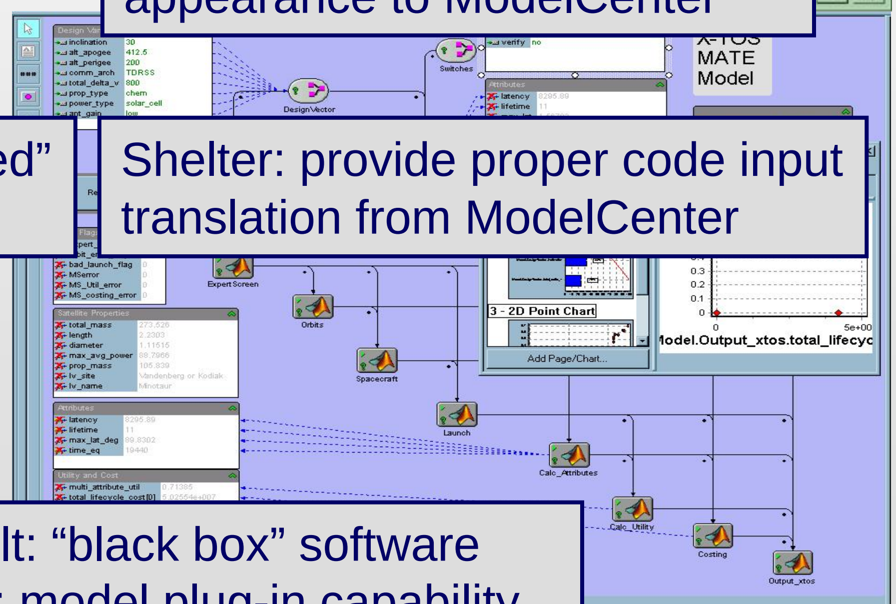






- Plug-in software models must “look” the same to the Analyst
- Mask: standardize code appearance to ModelCenter

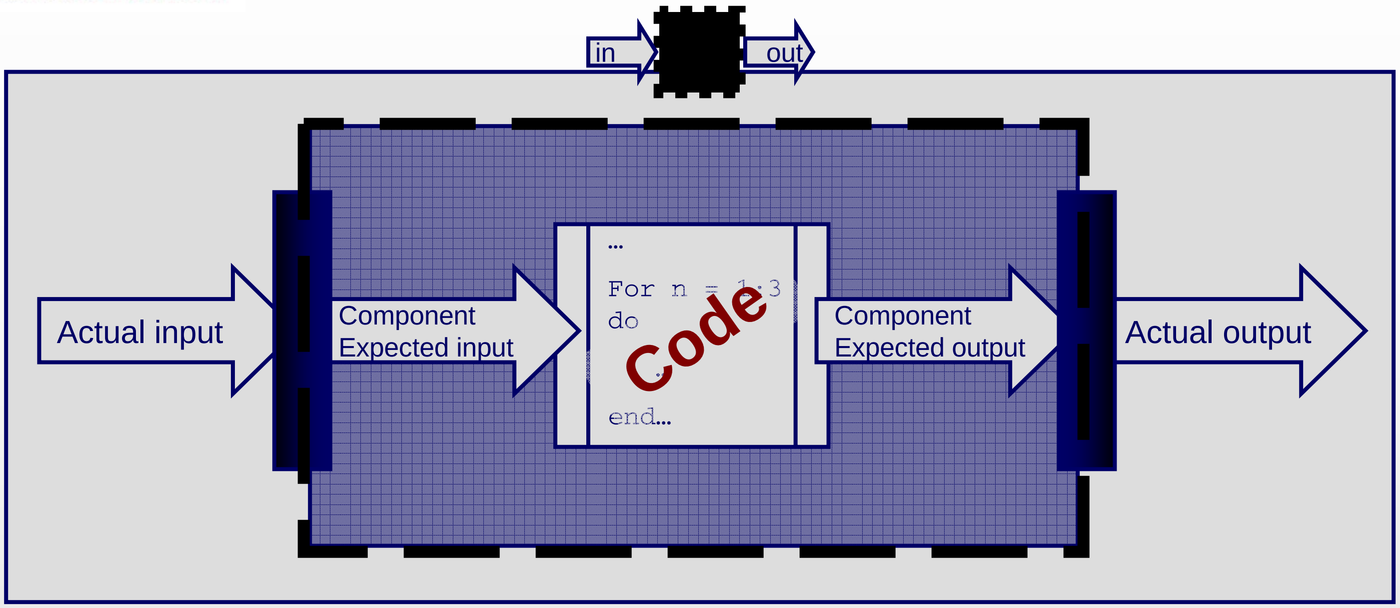




- Plug-in software models must be “protected” from varying input/output formats
- Shelter: provide proper code input translation from ModelCenter
- Result: “black box” software code; model plug-in capability

Source: http://www.phoenix-int.com/products/modelcenter.php

Example: Software/SoS Shell

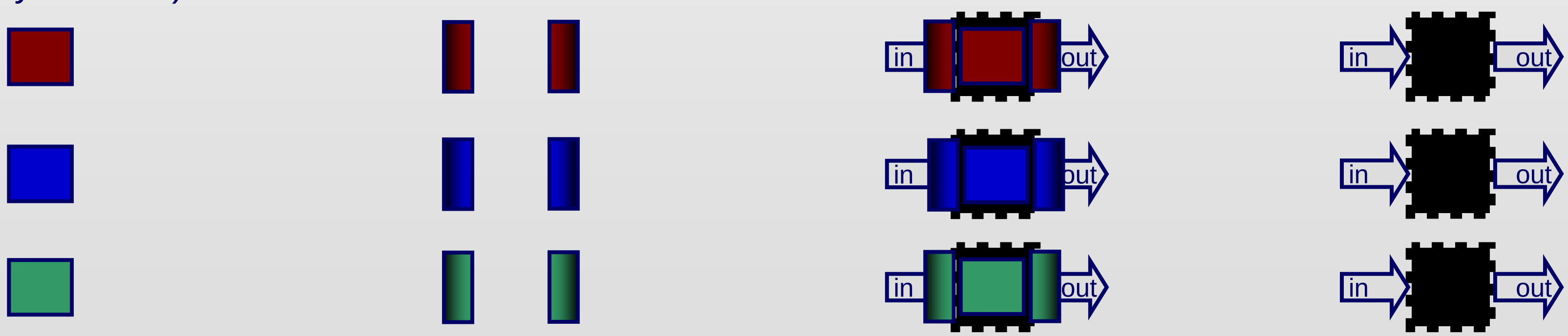


<u>Component set</u>
(virtually limitless)

<u>Shelter</u>
custom translation code

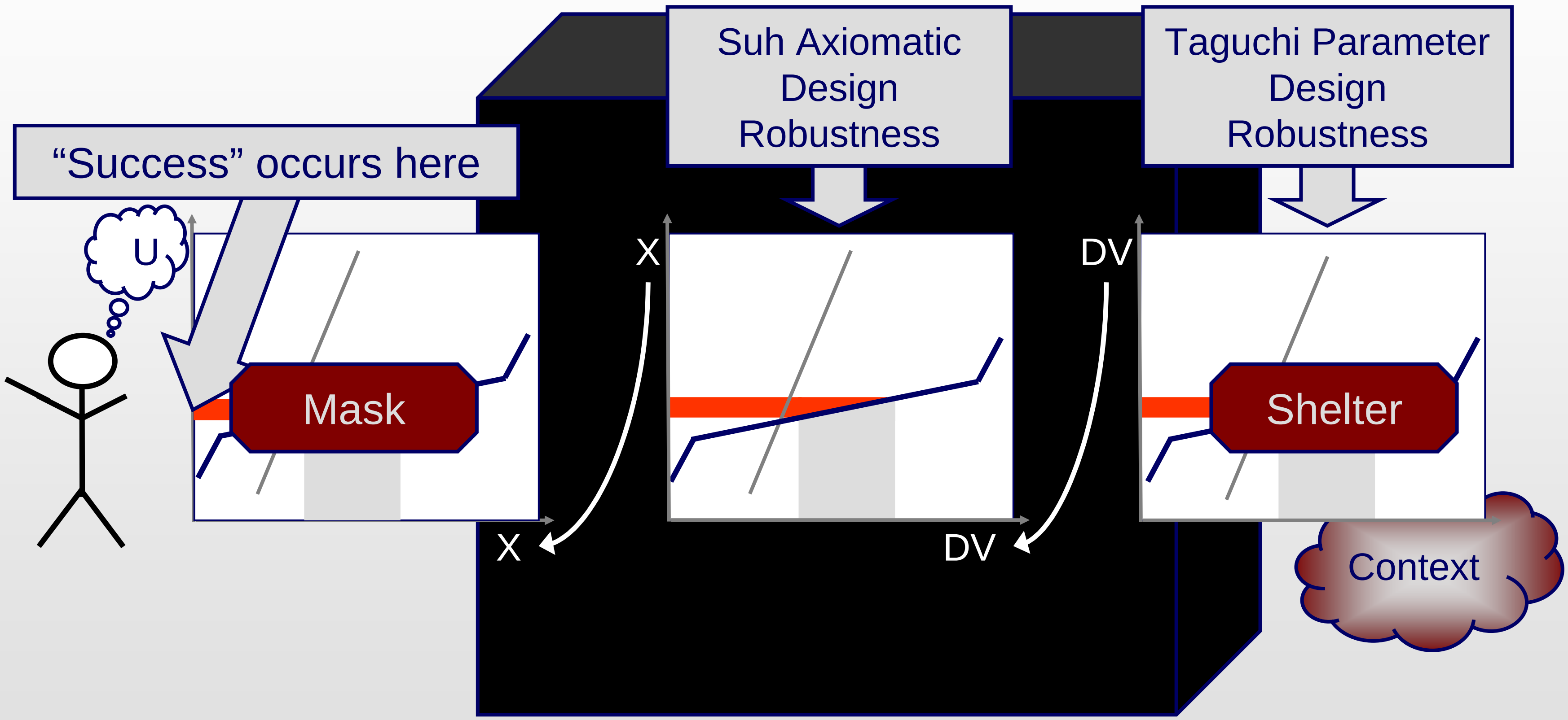
<u>Mask</u>
standardized format

Appearance to SoS



Result: “system of system” design with standard component appearance

Flow of Value Robustness



Robustness strategies can be pursued at various locations in value flow from context to value perception

Discussion

- When should a shell be used?

$$\text{Benefit}_{old\_sys+shell} - \text{Cost}_{shell} > \text{Benefit}_{new\_sys} - \text{Cost}_{new\_sys}$$

i.e., Net Benefit of Shell > Net Benefit of New System

- Implications for design
  - Extend operating ranges (augmented robustness)
  - Customized user experiences (multi-stakeholder satisfaction)
  - Distribution of costs for “shell”
- System of System design
- Separating changeable system parts

Conclusion

- System Shell construct decouples system from context through “protection” and “filtering”
- System Shell can be part of system or layered on top
- May be a cost-effective and/or timely solution to multiple perspective, multiple time period context

Using a System Shell is an effective technique for achieving Value Robustness

Systems Engineering Advancement Research Initiative

Thank you for your attention!

Any questions?

For further details on topic please see:

Ross, Adam M., Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration. Cambridge, MA: MIT. PhD in Engineering Systems. 2006.