SEAri Working Paper Series

Title: *Ility Space*

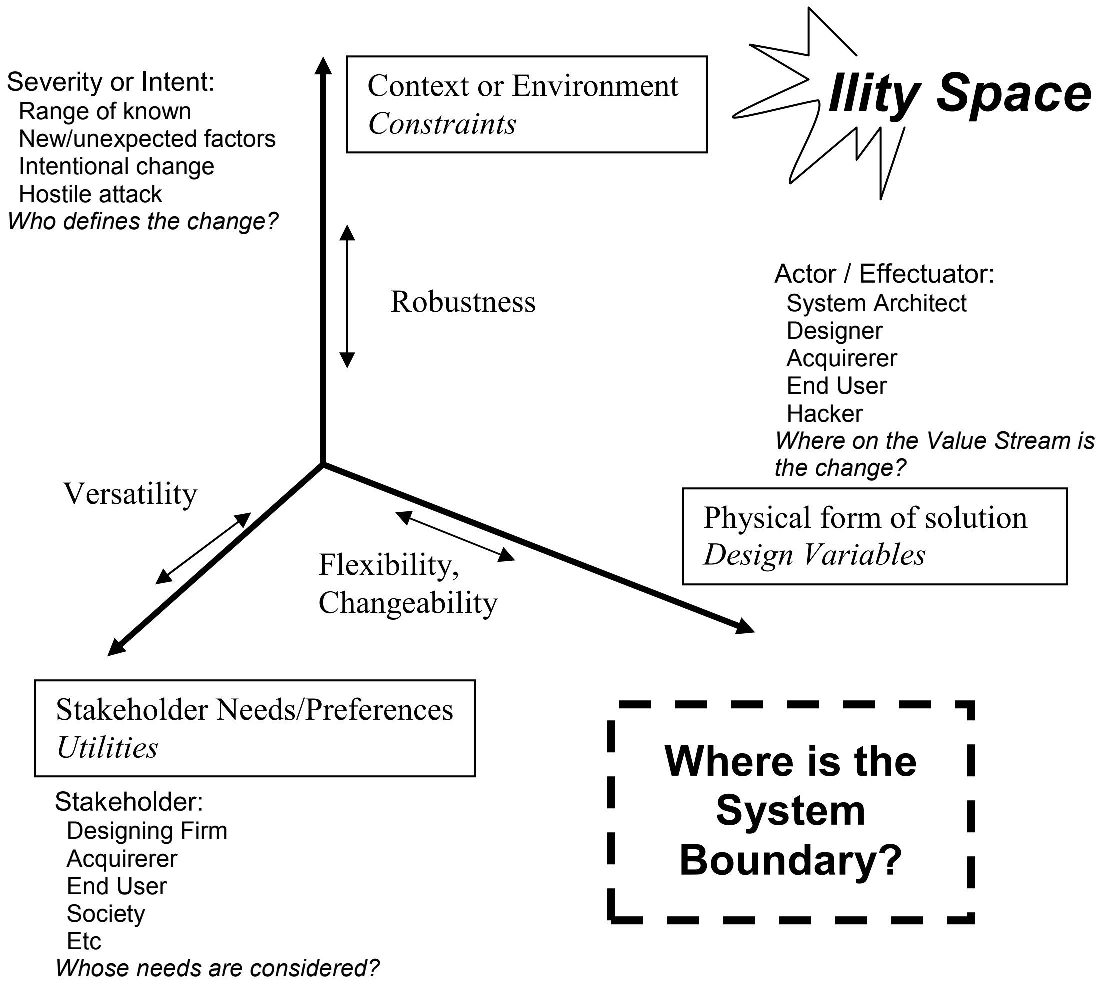
Author: Hugh McManus

Paper Number: **WP-2007-2-2**

Revision Date: September 7, 2008

Notes on "Ility Space"

A SEAri Working Paper by Dr. Hugh McManus, 11/7/06, Updated 9/7/08



The point of ilities is that the system can maintain or even improve function in the presence of change. The diagram above is a framework for thinking about the kinds of change that can be / need to be considered. The point is similar to the "mad-lib" diagram in the uncertainties paper[1] – one can take a given approach, and trace it through the above space, as a way of organizing ones thoughts and developing a holistic view of how the various ilities relate to one another and to other aspects of the design process.

Note that this is all a brainstorm level process – it is probably non-rigorous in a variety of ways, could use a lot more thought, and may ultimately prove non-useful, but it is the product of a group consensus so far.

[1] McManus, H., and Hastings, D., "A Framework for Understanding Uncertainty and its Mitigation and Exploitation in Complex Systems," *IEEE Engineering Management Review,* Vol. 34, No. 3, 2006, pp. 81-94.

Features of Ility Space:

Axes: these are the things that change. The three axes are, as far as we can see, fully orthogonal at least for a given choice of system boundary – more on that later. The three things that can change over a system lifetime are the system itself (an object or group of objects in physical space, represented in MATE by the design variables); the needs/desires of stakeholders (a perceived abstraction in the minds of the stakeholders, represented in MATE by the Utilities), and the environment (most abstractly, physical and financial space outside of the system boundary, represented by MATE as constraints, both explicit and embedded in the modeling of the physical reality in which the system operates). Note these are not single-dimension axes *per se*; only in an extremely simple case (one monotonic variable on each axis) could you actually plot anything quantitatively in this space.

They are consistent with Adam Ross's framework[2] (the noisy colorful box) in that you can reduce them to the same simple analogy. The physical form axis represents the color and noisiness, as well as other, perhaps initially unspecified, design variables. The needs axis contains articulated and perhaps unarticulated needs. The environment axis contains things that affect the users perception of value from the boxes (lighting, acoustic tile...) as well as things that may affect funciton (temperature?) and even threats such as fire or people who hate colorful objects that emit sound.

Note that statically, this is basically a tradespace representation with an environment axis, and is not very useful. The point is things *move* in this space; various types of movement represent various sorts of ilities. Next to each axis is a description of what happens if a system can move on that axis. An unchanging system that can face various environments and retain value delivery, is *robust*. An unchanging system that satisfies a variety of needs is *versatile*. A system which can change is, well, *changeable*;[3] what is interesting is how the system changes in response to changes on the other axes. For example:

- If needs of stakeholders change and the system can be changed by some agent to meet this need, it is *flexible*. On the space, the system has moved on both the need and design dimensions
- If the need or environment changes and the system can change itself to satisfy the needs, it is *adaptable*. On the space, the system moves in two or three dimensions.
- We are still working on what *survivable* means. An answer might be if the environment is severely disturbed (large motion in the environment dimension) it loses some function (drop on need axis) but does not pass below a minimum threshold, and in the new state has some path (through internal or external adaptation – motion on the design dimension) back to full functionality.

---

[2] Ross, A., Rhodes, D., and Hastings, D., "Defining Changeability: Reconciling Flexibility, Adaptability, Scalability, and Robustness for Maintaining Lifecycle Value," INCOSE International Symposium 2007, San Diego, CA, June 2007.

[3] We used *flexible* as the root ility on this axis, but that is probably not right, as flexibility assumes change in response to some change in need or environment

Definition:
Some obvious questions come up with the picture. The first is "what is the system boundary?" The chose of system boundary will dominate how one considers the ilities. For example, the axes are only orthogonal if we answer this question carefully – i.e. the pilot could be a stakeholder with a need (considered outside the system) or an effector of adaptive change (considered inside the system). Lots of other examples could be brought up but his would belabor the hopefully obvious point.

Each dimension also has some key information associated with any change that may happen in it:

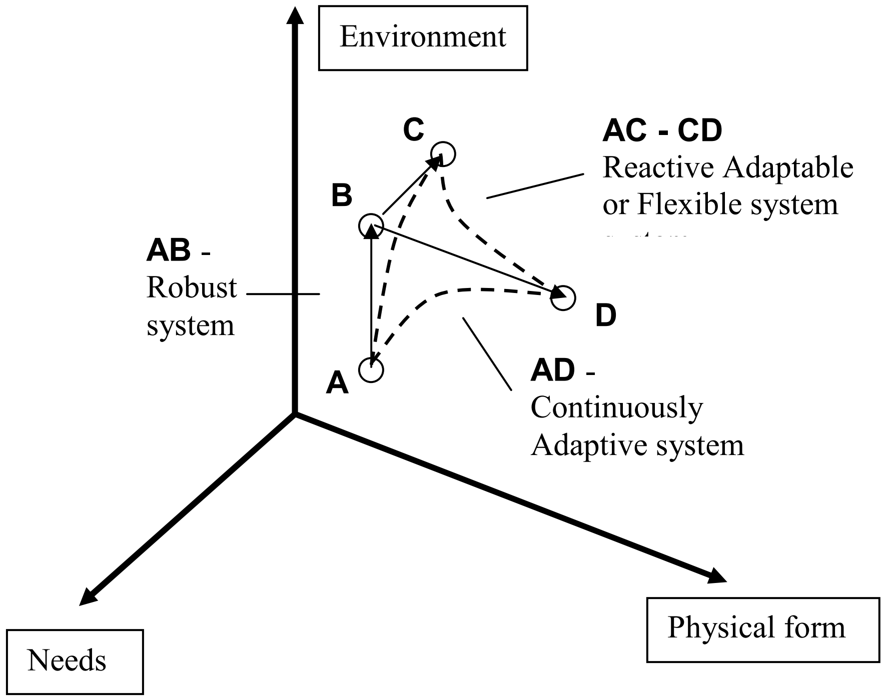
- The system may more in the environment dimension by changes in values of know variables (temperature increase); the introduction of new issues (rain?); intentional change (use in the desert – sand?) or even hostile attack. These are examples, not a complete list. The key question is "who (or what) is driving the change?
- The system may change in the design dimension by a variety of mechanisms. We could ask who effects the change – architect, designer, purchaser, user, hacker,[4] etc. A related question would be where in the lifecycle value stream does the change occur – architectural concept, detailed design, purchase, customized service live, or life extension/modification?
- The utility dimension is defined by the stakeholder defining the utility. Each stakeholder's needs may change through a variety of mechanisms, including the articulation of previously unarticulated value, changes in the perceived or actual use situation, and the adding or subtracting of stakeholders from a multi-stakeholder set. The key question – who defines the need?

Using the Framework

If the key questions are answered we can use the framework to formalize and visualize an instance of "ility" behavior. Let's define a system defined by a spacecraft, and its command communications system and operator. It exists in the natural environment of some orbit. It satisfies the needs for a service (e.g. communications) for some stakeholders. Statically, it initially occupies a point **A** in ility space.

The environment changes (a solar storm creating a noisy signal, for example). The system moves "up" in the environment dimension. Does this degrade its performance? If not, it occupies point **B**, and is *robust* to this environment change. If performance is affected by the environment change, the system moves to some point **C** (along curve **AC**), which may not satisfy the user. However, we have noted that the operator is part of the system; if the operator commands the vehicle to take measures to counteract the environment change (more power?) then the system moves, by internal mechanisms, to point **D**, which again satisfies the user. By our definitions, the system has shown itself to be *adaptable*. Typically, an adaptive system might follow the curve **AD**, continuously adapting to the environment to maintain user satisfaction.

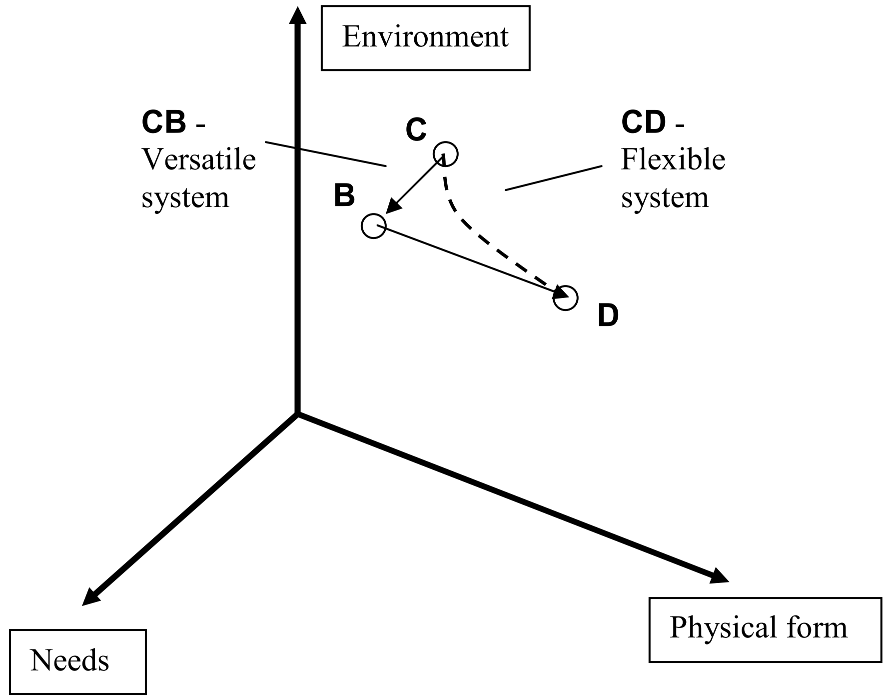
[4] post-user or alternate-user stakeholder who changes the product to unexpected ends



If the operator did not have the authority to alter the system sufficiently to react to the change, an outside actor (the company?) might arrange a more drastic change (reshuffling transponders between customers?). This would be an example of a flexible system, and might follow the more reactive path **AC - CD**.

A few other notes on the figure. The surface **ACD** is a patch of the surface of possible states for the system. It is more likely a volume, but ideally you would want to keep the system on the optimum surface, where maximum needs were satisfied for a given environment.[5] As drawn, point **B** is outside the "envelope" for this system. A robust system, perhaps one with extra margin, would move the surface up and to the left, such that point **B** is on or inside the envelope. Note that in this case, point **A**, which is the normal operation point, would be suboptimal (not on the surface) – typical of a system that obtains robustness by the crude mechanism of extra margin.

[5] This could get complicated if the changes were expensive – you would then want to optimize the navigation through the space to get maximum value for money!

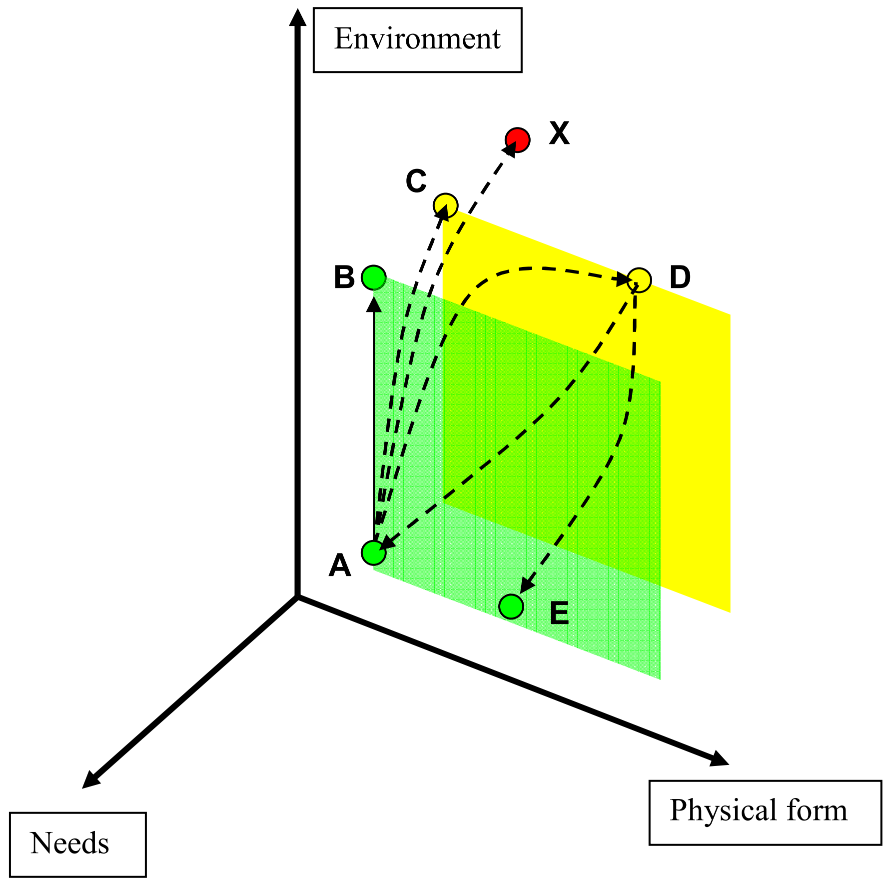


The above figure illustrates another ility story. If the system is initially at point **C**, and user needs change such that more performance is desired, a versatile system could occupy point **B**, satisfying the needs without change in physical form. A flexible system (assuming here a change effected by an outside actor) could change its form to satisfy the need at point **D**. Note the use of the word flexible in both of the figures, driven by the definition that flexibility is change driven by some actor outside the system. This suggests a possible rethinking – should we define flexible in terms of changing to satisfy needs, and adaptable in terms of changing in response to environments? Not sure of the answer, but interesting that the question is raised by playing with the framework.

The framework brings up issues:

- Complexity – the system boundaries, and the definitions required on each axis, show that a given (sub) system may have many ility spaces. This is not too bad though – there are many possible "cuts" at a MATE space as well, and that does not negate the usefulness of the tradespace representation as an analysis and communication tool.
- Still, it is likely that this concept will remain an idea-generating tool, and possibly a presentation tool for a sophisticated audience.
- It does appear useful for framing the problem – just the process of drawing the above diagrams clarifies things. Suggest we adopt this as a framing concept and see how our various ideas fit.

Appendix: *Survivability*:



System starts at point **A**. It is on the green plane, which is defined by a constant "operational" performance level. An attack (drastic shift in environment happens. A system that can exist at point **B** is robust to this attack. A system **C** that has performance degraded to no worse than a defined minimum level, defined by the yellow plane, has "survived" the attack. If performance is degraded more than that (Point **X**) the system cannot recover and is "dead." Note **C** survived passively; a system such as the one that can move to point **D** adapts to the attack by some change (safe mode?) and can also survive. After the attack, Matt Richard's definition[6] requires recovery of full functionality; that could be accomplished either by returning to the original state **A** (see line **DA**) or by attaining some new "battle damage repair" state that is also fully functional, such as **E**.

[6] Richards, M., Ross, A., Shah, N., and Hastings, D., "Metrics for Evaluating Survivability in Dynamic Multi-Attribute Tradespace Exploration," AIAA-2008-7882, Sept. 2008.