Developing Methods to Design for Evolvability: Research Approach and Preliminary Design Principles

J. Clark Beesemyer

Massachusetts Institute of Technology,
77 Mass Ave, E38-550, Cambridge, MA 02139

jaycb@mit.edu

Daniel O. Fulcoly

Massachusetts Institute of Technology
77 Mass Ave, E38-550, Cambridge, MA 02139

fulcoly@mit.edu

Adam M. Ross

Massachusetts Institute of Technology
77 Mass Ave, E38-574, Cambridge, MA 02139

adamross@mit.edu

Donna H. Rhodes

Massachusetts Institute of Technology
77 Mass Ave, E38-573, Cambridge, MA 02139

rhodes@mit.edu

Abstract

Evolvability is a design characteristic that facilitates more manageable transitions between system generations via the modification of an inherited design. This paper pursues parallel descriptive (empirical-based) and normative (theory-based) approaches to determine initial design principles and measurable metrics for evolvability. Contrasting biological and technological evolutionary processes was used to yield insight into possible design principles. Several starting points for evolvability metrics have been identified, including metrics for interface complexity, visibility, and changeability. Once these metrics reach a level of maturity that allows for simulation of case studies, it is expected that additional data-based design principles will be identified to augment the existing list of proposed design principles that include modularity and redundancy.

Motivation

The early phases of conceptual design require careful consideration as early decisions will have substantial influence on the new system, ultimately enabling or limiting success of the system over time. Measuring a system's "ilities" such as changeability, adaptability, flexibility, and survivability, gives stakeholders and decision makers an enhanced basis for differentiating between design alternatives. In the face of changing contexts or needs, called *epochs*, systems can be designed to change in response, or remain robust, in order to retain useful functionality to avoid suffering deficiencies and even failure. Designing an evolvable system may reduce the long term cost of system upgrades or replacements in the presence of epoch shifts over its lifespan. Evolutionary design starts from an existing design, rather than a blank slate, and is an increasingly common trend; for instance nearly 85% of GE's products are modifications of previous products (Holtta-Otto 2005). In industries where redesign is the norm, evolvability clearly is a desirable trait. This paper describes a research effort to characterize and quantify system evolvability and ultimately to provide a set of prescriptive design principles for designing for evolvability.

Research Approach

The initial thrust of this research has been to precisely define evolvability. After fully exploring the state of evolvability literature, the research will draw on two approaches, descriptive and normative, and the results of these will be combined into prescriptive measures for designing for evolvability (Figure 1).

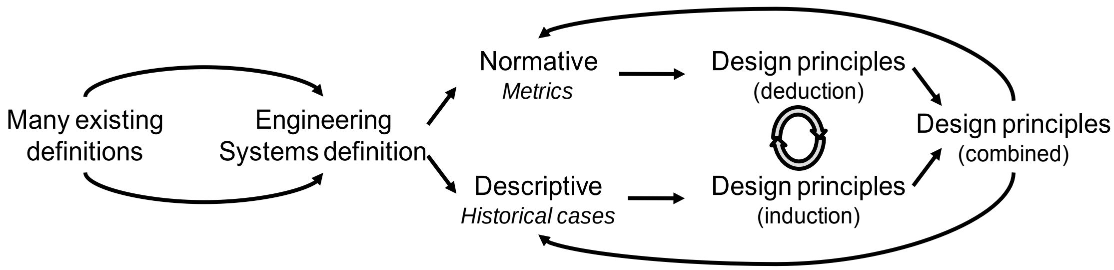

Figure 1. Dual approach to develop evolvability design principles

In the descriptive approach, initial design principles of evolvability are derived using purported principles in literature and knowledge gathered from interviews with systems engineers. The validity of these design principles may be tested by inductively mapping the evolvable characteristics of existing systems to the set of preliminary design principles, similar to what was done in recent survivability design principle research (Richards 2008). Initial principles will be revised with the insight gained from these empirical test cases. If these test cases show patterns that cannot be mapped to the working design principle set or if certain principles are not found in practice, the working design principle set may be revised to reflect these insights.

The normative approach focuses on extracting relevant ideas from existing literature and looking for trends, which will form the basis of a theoretical model of design evolvability. The relevant ideas have included definitions of evolvability, suggestions for when to design for evolvability, evolvability metrics, and principles for designing for evolvability. Several candidate metrics are explored with the intent of building on them to form a more comprehensive evolvability metric. An analysis of applications of this metric to selected case studies will result in theory-based evolvability design principles.

Design Evolvability Defined

The first task in this research has been to establish a meaningful definition of evolvability that is broad enough to apply to systems in multiple domains, yet free of unnecessary ambiguities. The proposed definition was initially informed by the biological perspective, the originating domain of evolution-related concepts. Additionally, applications from other fields such as systems and computational engineering were also considered. The current proposed definition of evolvability at this stage of the research is:

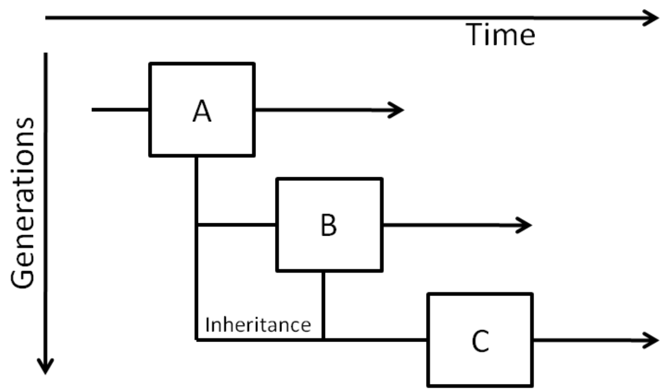

Figure 2. The Evolution of Systems Over Time

The ability to change an inherited design across generations [over time].

The key aspects of the definition include: some threshold amount of change in the system has occurred, and the new system is based upon or has 'inherited' part of its design from a prior 'generation.' This change between generations will occur through some mechanism of variation and selection. In Figure 2, the inheritance from prior generations of the system are shown as vertical connecting lines going into the design of the new system generation. It should be noted that the older generation system may continue to operate in parallel with newer generations of the system, and that inheritance may come from any prior generation.

Contrasting Biological and Technological Evolution

As the definition for evolvability was initially informed by the biological perspective, exploring characteristics of biological evolution may be beneficial in developing initial design principles for engineering systems.

Darwinian evolution, as summarized by Maynard Smith, requires populations of entities to have three properties: (1) the ability to multiply; (2) variation of characteristics within the population; and (3) some level of heritability with the variations. Natural selection is the emergent behavior of populations with such principles. Variations are passed to future generations through genetic code in an entity's DNA, or genotype, and their specific realizations, or phenotypes, are selected on by the fitness function of their environment (Ziman 2000). These variations are generally created through some means of recombination and mutation. If these variations turn out to give some sort of competitive advantage to an entity, it is more likely that the new genotype will be carried into future generations. Thus over time, favorable characteristics are passed on, and the species evolves. It is important to note that these variations are random, and generally not induced by their environment, and that the variation-selection mechanism acts on the population (Ziman 2000).

While there are currently much greater levels of complexity to the biological basis for evolution, the Darwinian view of variation and selection is remarkably still applicable to the natural evolution that has been going on for millennia. This basic framework of biological evolution seems to relate to the engineering systems' design process of concept creation and selection.

Over time, life has developed and refined alternative means for implementing variation and selection. Most notably, the emergence of sexual reproduction, a much more complex means to reproduce, has increased evolvability due to its effective capacity to create increased variation. Applying the concepts of evolution to engineering systems can be informed by and improved by these mechanisms of variation and selection that have emerged in nature. Kevin Kelly has done a large amount of work comparing the mechanisms of biological evolution to those of technological evolution (Kelly 2010). As biological evolution shapes how “living systems” change over time across generations, technological evolution shapes how “engineering systems” can change over time across generations. Kelly’s study of technological evolution provides a descriptive basis for this comparison of natural “systems” evolution to engineering systems evolution.

Forces for Evolution

If the goal of designing for evolvability is to improve how a system can evolve through generations, understanding what forces drive evolution in biology and technology is an important first step. Figure 3 is a representation of Kelly’s triad of forces that impact evolution in life and in technology. Each domain has three legs on which evolution stands. The Historical/Contingent leg is the ‘luck’, or the happenstance of a species. Since variation is random, speciation comes from improbable triggers in the past that leads a species down a contingent path (Kelly 2010). The second leg of the triad is the Structural/Inevitable. This force drives the macro-level structures in biology and technology. Where the contingent force is historical, this force is ahistorical. Corresponding to the theory of convergent evolution, this force would be responsible for similar forms independently developed seen today. This force is based on the laws of physics in the world that yield general truths like air density affecting flight or fluid displacement affecting marine life and technology. In biology an example of this might be the convergent evolution of the wing, where technology might show this principle in the form of hemispherical boats. While the causes of convergent evolution are debatable and controversial in biology, it is more important to see it as an emergent behavior of evolution and a useful tool in analyzing the evolvability of systems.

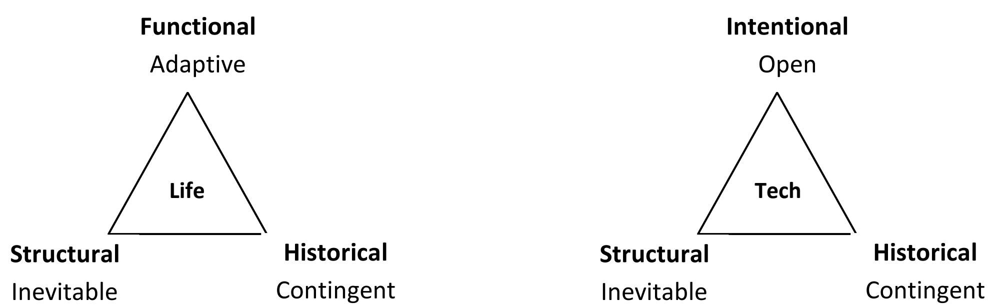

Figure 3. Kevin Kelly’s Triad of Forces Driving Evolution in Biology and Technology

The Adaptive/ Functional leg is the fundamental, orthodox force discussed earlier. Those stronger species that are able to reproduce are naturally selected to continue. This force however, is where life and technology may differ. While biology relies on the mechanism of BVSR (Blind Variations Selectively Retained) (Ziman 2000), technology, and more specifically, a system, can be deliberately designed and implemented by the Intentional/Open force.

In the third force, possibilities are expanding in time. This intentional part of Kelly’s triad is the optimization engine of technology. The intentional mechanism differs from life’s natural selection engine in that it may be completely conscious, driven by free will and choice. This is why technology, once given the tool of the human mind, or an intelligent designer, has evolved so much faster than life on Earth (Kelly 2010).

Kelly’s construct is a good starting point in comparing natural evolution to technological evolution. As Kelly intended to inform modern innovation by following evolutionary trends in the past, engineering systems may find value in early concept and design phases by effectively implementing principles derived from natural evolution. Designing an evolvable system requires the daunting task of predicting trends, or evolutionary pathways, and designing for future contexts and needs in order to maintain value to system stakeholders over time (Ross 2008).

Characteristics for Technological Evolution

In exploring different domains' ideas on the relationship between technological and biological evolution, common characteristics of evolutionary mechanisms emerged for technology that biology does not utilize as well, if it all. These observed aspects of technology may lead to initial design principles that could be applied to engineering systems in order to lower the required effort to move from one generation to the next.

Non-sequential Inheritance. Biological evolution requires sequential improvements passed down from parents to children (the next generation). Small changes that improve fitness are passed down to offspring and so on. DNA may be seen as a transcript of evolution over millions of years; however, each new population can only pull traits from their respective parents. Populations cannot directly pull from other species or their distant relatives, many generations earlier. The evolution of technology, however, does not have this chronological constraint. Since most technology is recorded in patents and journals, for example, any new system can pull from any generation before itself, all the way back to the inception of the technology.

Evolutionary Leaps. Living systems have slowly and incrementally evolved to improve fitness over time. Since more extreme variations generally lead to life that is not viable, biological evolution favors small changes between generations. In technology, however, there is no requirement of small, incremental steps in evolution. Transistors did not slowly evolve from vacuum tubes one small piece at a time. Technology is often characterized by fewer, but larger evolutionary steps between generations.

Conscious Variation and Selection. Technology has the benefit of having an 'intelligent designer'. While biology relied on natural selection to slowly find optimal solutions over millions of years, technology may produce systems that never could have naturally evolved with biological rules for reproduction. While technology may benefit from some level of random variations, the ability to channel or 'push' these variations into useful system concepts improves evolvability. Biology limits the inheritance of offspring not only to the previous generation, but also to the same species (the parent species). Technology, however, can pull from any domain or 'species' in creating new 'life' or a new system. A principle or design from aerospace can be applied to automobiles or trains or vice versa. Any one system's success can be applied to another system in the future. Figure 4 shows how systems may use this principle in their evolutionary path between generations.

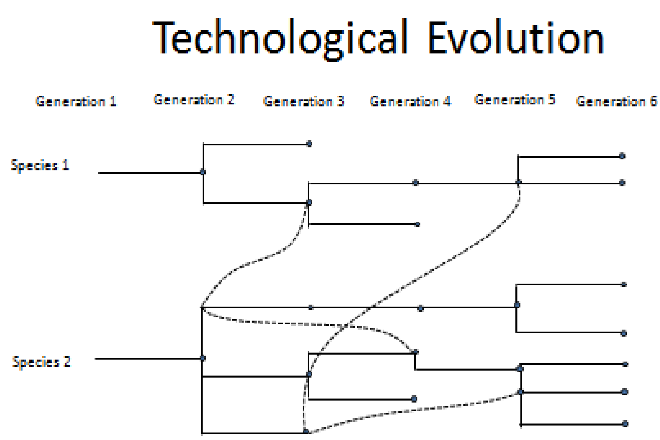

Figure 4. Inheritance structure of technological systems

Technology is not vertical, as in the family trees found in biological lineage. It can use lateral jumps across 'species' or generations to reuse or repurpose old ideas. Inheritance may occur from different systems or different generations at anytime in development. Successful solutions may be shared across time and domains.

Exaptations. Exaptations are inadvertent inventions that give value in a repurposed role. These are rare in biology. For example, feathers were first used in biology as a means of thermal control, but in time, became very useful in another task, flight, adding another competitive advantage to participating species. In technology, exaptations are common and very useful (Kelly 2010). Inadvertent discoveries in one domain can translate to more utility in another domain.

The goal of future research is to use these characteristics to inform possible biology-informed design principles for evolvability. While these principles would be derived from the biological comparison, other principles may be derived directly from engineering systems concepts.

Designing for Evolvability

Whereas biological evolution is more of a constant emergent behavior of populations, choosing to use an evolvable engineering process to modify an existing design or incorporating evolvability in a new design is a conscious decision of the designer. Ideally, a set of design principles should be accompanied by considerations for when and under what conditions to use them.

Deciding to Design for Evolvability

Why should a design team bother to design for evolvability? Several considerations go into this decision based on timing and needs: (1) how often will available technologies change?, (2) how often will requirements change?, (3) what is the life cycle of the system?, and (4) is the system part of a larger system of systems? The idea of comparing a technology development cycle to a system development cycle is not a new one. In Axiomatic Design, (Suh 1990, p 376) explains that "in some industries, products become obsolete so rapidly that the product development cycle – the lead time for product development – must become shorter and shorter to keep up with competition and customer demand." An evolvable design ideally accommodates faster change (requiring less effort and resources) and thus is better suited for rapid modification in the presence of new requirements and/or available technologies.

Ideas from an approach called Epoch-Era Analysis can be used to form a better context for looking at this problem. A system era is the full lifespan of a system and is made up of one or more epochs. An epoch is a period of time where the system experiences fixed context and value expectations. Further characteristics of an epoch include static constraints, available design concepts, available technology, and articulated attributes (Ross and Rhodes 2008). This approach can be abstracted a further level to compare system eras and context frequencies. A system era captures the lifetime of a system and additional information can indicate the frequency at which new systems will be developed, the redesign frequency. A context frequency would encompass the rate at which technologies associated with the system are changing as well as how often requirements change. The context frequency is a function of many component frequencies associated with epoch variables, which are the va-

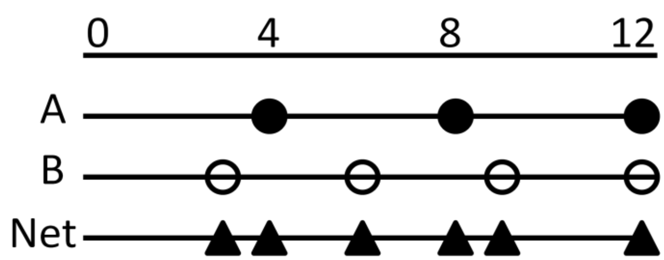

Figure 5. Net Context Frequency Derived from Epoch Variable Component Frequencies

riables representing key exogenous uncertainties and are used to characterize an epoch (If epoch variable A changes every 3 years and epoch variable B changes every 4 years, then an epoch change would, on average, occur every 2 years, as illustrated in Figure 5). An example of a component frequency in a context period is Moore's law, which states that the number of transistors that can be reasonably placed on an integrated circuit doubles every two years. For systems affected by this type of technology change, an epoch variable with a frequency of one change every two years would need to be considered. It is important to note that based on our definition of evolvability, evolutionary changes will occur in the redesign phase, between the fielding of successive systems. The goal of the evolvability-minded designer is to sync up the redesign frequency with the appropriate context frequency, when in their control. The reasoning behind this goal is that as the redesign frequency approaches the context frequency, successive generations can better take advantage of new technologies. In the case where context frequencies and redesign frequencies are out of the designer's control, Table 1 shows how important evolvability is as a design consideration. Systems with redesign frequencies on the same order as their context frequencies are the best candidates for having valuably evolvable designs. When the time comes to redesign the system, whether planned or as a reaction, having evolvability will enable a less resource-intensive redesign process. If the context frequency is much shorter than the redesign frequency, then evolvability is less important of a design consideration. It is likely that available technologies will be so different from what the system was originally designed with that clean sheet design is most likely the most viable option. On the other end of the spectrum, where the redesign frequency is much shorter than the context frequency, evolvability is a moderately important design consideration. In this case, the changes in available technologies and requirements will be relatively small for each generation, but the designer may still want their system to be able to take advantage of any new technology and meet any new requirements.

Table 1. When is Evolvability Important?

Relationship	*Relative Importance*
Redesign Frequency << Context Frequency	Low
Redesign Frequency ≈ Context Frequency	High
Redesign Frequency >> Context Frequency	Medium

New requirements often change the formulation of a utility function, which can be used to measure the perceived usefulness of a system, meaning that new requirements may result in a new definition of an "optimal" design point. This observation parallels Kevin Kelly's perspective that since "instability and imbalance are the norm in today's economy… systems optimized to a single design point will not last very long" (quoted in Fricke and Schulz 2005, p. 346). Therefore, designers desiring optimal solutions in the presence of changing requirements should strive to include evolvability in their designs if their plan is to field new designs (as opposed to modifying existing systems).

Candidate Evolvability Metrics

No perfect evolvability measure currently exists. Based partially on the metric qualities proposed by (Christian 2004), a "perfect metric" would be simple to state, state any environmental conditions, state time dependence, be quantitative, be relatively easy to measure, and help the user identify the system that best meets their needs. A perfect evolvability metric then has the specific requirements of characterizing ease and extent of changes that are possible for a design to incur between generations. An exploration of the current literature has exposed many attempts at evolvability metrics as well as candidate metrics from other categories, such as complexity and changeability. These metrics fall short in several ways: some actually measure a different ility (such as adaptability or complexity), some are time intensive and are potentially unreliable (such as relying on interviews), and still others require very highly developed models before they can be applied. These prior metrics will serve as starting points for creating a more comprehensive metric that aligns itself with the definition of evolvability used in this paper.

The interface complexity metric proposed by (Holtta-Otto 2005) was meant to be used to decide where modular boundaries (and therefore interfaces) should be defined in a given system. The metric expresses the relative difficulty of a given change to an interface, measured in percentage of original design effort, as a function of the percent change needed. An example of this relationship is shown in Figure 6. The three parts of the curve, as separated by the two discontinuities on the graph, represent three categories of redesign: (1) where no redesign is required, (2) where redesign is required and some relationship, not necessarily linear, relates the percent change to the redesign effort required, and (3) where the amount of change required is so great that a new module, component, or interface must be acquired or designed. The shortcoming of this metric is that it, as implemented by Holtta-Otto, relies on interviews to develop the relationships. To make this metric more viable, a more robust method of determining these relationships should be developed. Another way to adapt this metric to make it more useful for measuring evolvability would be to use it to measure cost and time required given a specific change in a parameter (as opposed to an interface).

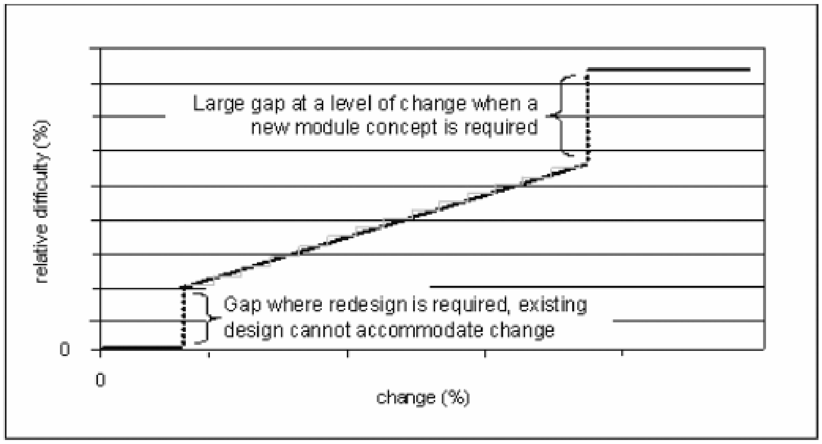

Figure 6 Example of Holta-Otto's Interface Complexity Metric (43)

Rowe and Leaney (1997) propose an ontological framework that fully defines the possible state space, the state functions (F_i), and reference frame of a system. They define a lawful state space as well, which is a subset of the possible state space determined by the laws of physics and other restrictions imposed on the system. Based on this framework, they define the relative change of a system in the i^{th} respect and with respect to α, shown in Equation 1. These authors propose interpreting the relative rate of change as the sensitivity of an architecture to changes in particular requirements that affect system properties. They also define the relative extent of change of a system in the i^{th} respect, with respect to α over the interval $[\alpha_1, \alpha_2]$, shown in Equation 2. They propose inter-

$$V_i = \frac{1}{F_i}\frac{\partial F_i}{\partial \alpha}$$

Equation 1. Relative Rate of Change (Rowe and Leaney 1997)

$$\Delta_i(\alpha_1,\alpha_2) = \frac{1}{\alpha_2-\alpha_1}\int_{\alpha_1}^{\alpha_2} V_i \cdot \partial\alpha = \frac{\ln F_i(\alpha_2) - \ln F_i(\alpha_1)}{\alpha_2-\alpha_1}$$

Equation 2. Relative Extent of Change (Rowe and Leaney 1997)

preting the relative extent of change as the normalized effort required to perform a given change. Both of these measures would contribute to measuring evolvability; the only drawback to this framework is that it requires the system designer to have a very well-defined and extensive model of the system, including all of its state functions in terms of all system parameters. An improvement on these metrics would be to add a cost function that could be combined, especially with the relative extent of change, to represent the cost of a given change. If the user defines a cost threshold, these metrics could then be used to explore the tradespace to see what subset is reachable with the given resources. This is very similar to the filtered outdegree metric for measuring changeability proposed by Ross (2006). The aforementioned metric measures the number of outgoing paths from an initial design to a transitioned design that fall below some cost threshold. The filtered outdegree metric also has the advantage of having been actually applied to case studies. Once a cost filter is set, the implementation only requires that change mechanisms, tradespace, and costs associated with given changes be defined. The algorithm can then search the outdegree from a certain design point and return how many paths are below the cost threshold.

Using design structure matrices (DSMs) is another potential method for measuring evolvability. DSMs are a matrix representation of the dependencies and connections in a system. Entries are traditionally binary and can represent a structural connection or a flow of information, mass, or energy. An often unseen drawback of DSMs is that they only capture first order relationships. Any systems engineer who has dealt with a complex system knows that a change to a parameter rarely affects just one component, and most likely is not even limited to affecting the components directly linked to the component being changed. This effect is characterized through change propagation analysis and the change propagation index in (Giffin et al. 2009), which can serve as guide for developing an evolvability metric based on similar concepts. Another method for seeing the full extent of these relationships, introduced by (MacCormack et al. 2007), involves raising the DSM to higher orders (until an empty matrix is reached) and summing the resulting matrices. Since not all matrices will reduce to zero at higher orders, a set number of iterations can be prescribed. The matrix that results from the sum is called the visibility matrix. The sum of an element's row is the visibility fan out (VFO) and represents the number of dependencies it has on other elements, both directly and indirectly. The sum of an element's column is the visibility fan in (VFI) and represents the number of elements that depend on it, both directly and indirectly. A modification to the visibility matrix and the VFO and VFI that might improve their ability to characterize evolvability is to use continuous scale rather than the binary scale for the DSM entries. Similar exploration is seen in Tyson Browning's work; he proposes adjusting the "coupling coefficient" in a DSM from linear to exponential to better capture the importance of a dependency (Browning 2001).

Current research efforts focus on building on the candidate metrics proposed above. By taking care to ensure all modifications to these metrics are based on sound reasoning, and are linked to key concepts in the definition of evolvability discussed in this paper, the end result should be a more applicable evolvability metric that can be applied to case studies.

Preliminary Evolvability Design Principles

The normative approach of this research involves deriving design principles from the metrics, assuming one or more metrics emerge that capture and reflect all essential aspects of evolvability. Techniques for designing a system that results in the highest evolvability score should constitute good evolvability design principles. Unfortunately, the metrics are not yet mature enough to be applied to case studies, so drawing design principles from theory and simulation is not yet possible. What follows are some evolvability design principles revealed in the literature.

Modularity is a design principle that is beneficial for many more reasons than just evolvability. The concept can be traced back all the way back to evolutionary biology, where data shows that the number of traits a variation affects is inversely related to its likelihood of being selected (Hansen 2003). The concept applies in the same sense to systems engineering; if a system is designed into modules such that a proposed change to a system only affects one module, potentially much less redesign will be required than the same change being implemented on a highly integral system. Modularity is not always good however; (Holtta-Otto 2005) points out that designing for modularity is accompanied by the potential for over-design and potentially inefficient performance. As mentioned earlier, being optimized to only one design point (efficient performance) does not allow for a system to thrive in uncertainty. Only a certain degree of modularity might be needed. For instance, if a system has components A, B, C, and D, but technology and requirements concerning A and B are constant and show no signs of changing, leaving A and B coupled will not hinder evolution.

In their paper on designing for changeability, (Fricke and Schulz 2005) list several extending principles for enabling changeability. Despite being applied to changeability, many of these principles appear to also be applicable to evolvability. The first principle is *integrability*, which is characterized by compatibility and common interfaces. This goes hand-in-hand with modularity; modules are only as good as the interfaces through which they interact. *Scalability* is another one of the extending principles that applies to evolvability, which can apply to either a single parameter or the entire system. The range of a parameter's scalability is determined by the capacity of the rest of the system to accommodate the change. The third applicable design principle proposed by Fricke and Schulz (2005) is *decentralization*. This principle calls for distributing resources to appropriate locations, rather than having them located at a single place. Decentralization, like modularity, aims to minimize change propagation. The final applicable design principle mentioned in their paper is *redundancy*. Redundancy allows for more constant performance and functionality in the face of potential faults or failures. Anticipating where these failures might occur "facilitates an identification of objects/units likely to be affected by architectural evolution" (Fricke and Schulz 2005). Redundancy and modularity can be very powerful together, because it potentially allows a module to be taken away without the system losing a critical function.

Reconfigurability is a design principle explored extensively by Siddiqi and de Weck, who claim reconfigurability aids evolvability through "[enabling the system to change] easily over time by removing, substituting, and adding new elements and functions" (Siddiqi and de Weck 2008). Siddiqi suggests two design principles that lend themselves well to designing for evolvability: using self-similar modules and maximizing information reconfiguration. *Self-similarity* can enable radical change and utilize the same components to achieve a very different function. *Maximizing information reconfiguration* is based on the fact that changing an informational structure is almost always less costly than physically reconfiguring a system or redesigning physical components.

Conclusions and Future Work

Evolvability is a powerful concept that describes the ability of a design to be modified across generations in the presence of changing contexts, allowing for the potential to deliver more value over the course of a family of systems' lifetime. The evolvability design principles described in this paper, including modularity, integrability, scalability, decentralization, redundancy, and re-configurability, (achieved through self-similarity and maximizing information reconfiguration) are as yet a preliminary list. Future research will expand on these and compare them to the principles that will be extracted from applying developed metrics to case studies and the biology-informed design principles. The descriptive approach will continue in finding applicable case studies to empirically test the initial set of principles and conduct informal and formal interviews with experts in different domains of engineering systems in order to elicit past applied evolvability approaches. The next goal for the normative approach is to develop the metrics to the point where they can be applied to case studies, which will eventually facilitate the extraction of further design principles. The eventual final product of this research will be a set of design principles and considerations that designers can use to increase the value robustness of their families of systems using evolvability.

References

Browning, T.R. "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions". *IEEE Transactions on Engineering Management*. 2001;48(3):292-306.

Fricke, E. and Schulz, AP. "Design for changeability (DfC): Principles to enable changes in systems throughout their entire lifecycle." *Systems Engineering*. 2005;8(4):342-359.

Giffin, M. et al. "Change Propagation Analysis in Complex Technical Systems". *Journal of Mechanical Design*. 2009;131.

Hansen, TF. "Is modularity necessary for evolvability? Remarks on the relationship between pleiotropy and evolvability." *Bio Systems*. 2003;69(2-3):83-94.

Holtta-Otto, K. *Modular product platform design*. Espoo: Helsinki University of Technology. 2005.

Kelly K. What Technology Wants. New York: Viking, pp. 406, 2010.

MacCormack, A. Rusnak, J. and Baldwin, C.Y. "The impact of component modularity on design evolution: Evidence from the software industry." papers.ssrn.com. (working paper) 2007.

Richards, M.G., Ross, A.M., Hastings, D.E., and Rhodes, D.H., "Two Empirical Tests of Design Principles for Survivable System Architecture," INCOSE International Symposium 2008, Utrecht, the Netherlands, June 2008.

Richards, M.G., Ross, A.M., Hastings, D.E., and Rhodes, D.H., "Empirical Validation of Design Principles for Survivable System Architecture," 2nd Annual IEEE Systems Conference, Montreal, Canada, April 2008.

Ross, A.M., *Managing Unarticulated Value: Changeability in Multi-Attribute Tradespace Exploration*, Doctor of Philosophy Dissertation, Engineering Systems Division, MIT, June 2006.

Ross, A. And Rhodes, D. "Using Natural Value-Centric Time Scales for Conceptualizing System Timelines through Epoch-Era Analysis." *INCOSE International Symposium 2008*, Utrecht, the Netherlands, June 2008.

Rowe, D. and Leaney, J. "Evaluating evolvability of computer based systems architectures-an ontological approach." *Proceedings International Conference and Workshop on Engineering of Computer-Based Systems*. 1997:360-367.

Siddiqi, A., and de Weck, O. "Modeling Methods and Conceptual Design principles for Reconfigurable Systems." *Journal of Mechanical Design*. 2008;130(10).

Suh NP. Axiomatic Design: The Principles of Design. Oxford University Press. 1990.

Ziman, .J ed. Technological Innovation as an Evolutionary Process. Cambridge University Press; 2000:398.

Biography

Clark Beesemyer is a masters student in the MIT Department of Aeronautics and Astronautics. Completing his undergraduate education at the United States Air Force Academy, he received a B.S. in Astronautical Engineering. There he served as the chief engineer for FalconSAT-5, a small satellite program exploring the effects of space weather. He now serves as a 2nd Lieutenant in the U.S. Air Force and will be attending pilot training at the conclusion of his masters program.

Dan Fulcoly is a masters student in the MIT Department of Aeronautics and Astronautics. Prior to attending MIT, he received a B.S. in Physics and a B.S. in Mathematics from the United States Air Force Academy. His prior research experience includes developing methods for non-resolvable space object identification for space situational awareness applications. Dan is a 2nd Lieutenant in the U.S. Air Force and will serve as a physicist after his masters program.

Dr. Donna H. Rhodes is a senior lecturer and principal research scientist at MIT, where she is Director of the Systems Engineering Advancement Research Initiative (SEAri). Prior to joining MIT, Dr. Rhodes had 20 years of experience in the aerospace, defense systems, systems integration, and commercial product industries.. She holds a Ph.D. in Systems Science from the T.J. Watson School of Engineering at SUNY Binghamton.

Dr. Adam M. Ross is a research scientist in the Engineering Systems Division at MIT. He has professional experience working with government, industry, and academia. He holds a bachelors degree in Physics and Astrophysics from Harvard University, two masters degrees in Aerospace Engineering and Technology & Policy, and a doctoral degree in Engineering Systems from MIT. He conducts and advises ongoing research in systems design and selection methods, tradespace exploration, managing unarticulated value, designing for changeability, value-based decision analysis, and systems-of-systems engineering. He has published over 50 papers in space systems design, systems engineering, and tradespace exploration.