Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 44 (2015) 443 – 453
Procedia Computer Science

2015 Conference on Systems Engineering Research

Towards a Prescriptive Semantic Basis for Change-type Ilities

Adam M. Ross[a]* and Donna H. Rhodes[a]

[a]*Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, USA*

Abstract

The concept of incorporating "ilities" into systems seems a self-evidently good idea, as expressed by both written and spoken positions by technical and political leaders. Indeed, incorporating lifecycle properties such as flexibility, adaptability, and recently affordability and resilience, into systems is touted as the solution to modern day's ever increasing complexity, schedule and budget pressures, and the need for finding sustainable solutions. While expressing desires for ilities seems straightforward, tracing these desires to verifiable system instantiations remains ambiguous at best. This paper describes the semantic challenge underlying the concept of a coherent set of system properties, a sampling of various efforts to ascribe meaning to particular ilities, and proposes a prescriptive 20 category semantic basis for specifying a set of ilities, while avoiding the assertion of new definitions. The intention for this first pass prescriptive semantic basis is to begin a structured approach for exploring the existence of one or more semantic fields, which together form a coherent semantic framework for tracing desired ilities into verifiable system requirements and specifications. Preliminary results indicate that at least three semantic fields exist within the larger set of system lifecycle properties including change-type, architecture-type, and new ability-type ilities.

Peer-review under responsibility of the Stevens Institute of Technology.

Keywords: ilities; semantic basis; system properties; changeability; flexibility; adaptability; robustness; scalability; modifiability; resilience; affordability

1. Motivation

These days, it is common to hear terms such as "flexible" or "adaptable" or "robust" in the parlance of politicians and technical leaders in the U.S. and other nations. In recent years, "affordability"[1] and "resilience"[2] have taken center stage. When terms such as these are used in speech or text, the precise meaning can sometimes be kept

* Corresponding author. Tel.: +1-617-253-7061; fax: +1-617-253-3641.
E-mail address: adamross@mit.edu

Peer-review under responsibility of the Stevens Institute of Technology.
doi:10.1016/j.procs.2015.03.040

intentionally vague in order to convey generalized impression of the intent of the speaker. In policy speeches, generalized impressions are more likely to garner support through various interpretations by the listeners. When it comes to implementation however, precise meaning becomes more important in order to validate that the mandates have been successfully carried out.

In order to better understand these "ilities" enough to ensure that systems predictably display these properties, this paper will propose a semantic-approach to frame the problem of disambiguating possible semantic field structure among the so-called "ilities." Semantics is the scientific study of "meaning" and is a promising area for clarifying the "ilities." In this context, it is important to recognize that meaning arises from the interplay of "use" (i.e. in speech) and "prescription" (i.e. definitions and dictionaries). Ideally meaning is universal and well defined, with direct correspondence between the prescriptive meaning and "use" meaning of terms. In practice, the correspondence changes over time with meaning in one sense impacting meaning in the other sense. A particularly relevant concept from semantics is that of the *semantic field*. A semantic field is a group of words with related meanings, for example, kinship terms or color terms[3].

1.1. Semantic challenges for ilities

One of the fundamental challenges for developing a clearer understanding of the semantics of "ilities" is the current ambiguity in these terms. Many of these terms are used colloquially and therefore inherit informal meaning. Their use in technical disciplines hinges on well-accepted prescription, which has not yet occurred within Systems Engineering, as evidenced by the abundance of definition offering papers with conflicting meanings. Additionally, the ilities terms, as currently used, display both polysemy and synonymy. Polysemy is "the property of [a term] having multiple meanings that are semantically related"[3]. An example of polysemy is two different, but related meanings for flexibility: "able to be changed" and "able to satisfy multiple needs." In contrast to polysemy, synonymy is "the property of multiple terms having similar meaning." An example of synonymy is the interchangeable use of flexibility (able to be changed) and changeability (able to be changed or change itself).

One of the reasons for this ambiguity in the technical usage of ilities is that typically ilities are mostly considered one at a time in the literature. As an example, consider treatments of engineering flexibility[4,5]. Some work has been done on sets of ilities, such as for system changeability/flexibility[7,8,9] and reconfigurability/flexibility[10].

2. Addressing sets of ilities

If the challenge of understanding the semantic relationships amongst the ilities can be addressed by looking at sets of ilities, how should the members of the sets be selected? Many approaches have been attempted in the past, some of which are briefly described below.

2.1. Academic expert opinion

In 2001, a committee of 11 MIT faculty developed a working paper called "ESD Terms and Definitions" in order to lay out the key concepts for the emerging field of Engineering Systems. These 14 ilities included flexibility, agility, robustness, fail-safe, adaptability, scalability, modularity, safety, durability, sustainability, quality, reliability, repairability, and maintainability. Ilities, as defined by this paper are: "requirements of systems, such as flexibility or maintainability, often ending in 'ility;' requirements of systems that are not necessarily part of the fundamental set of functions or constraints"[11]. On inspection of the list and the faculty listed as co-authors, the list appears to be reflective of the research interests of the committee members.

2.2. Practitioner interviews

An alternative approach is to ask practitioners about which ilities are well understood and which are important for their systems. An exploratory set of surveys and interviews was conducted with several Federally Funded Research and Development Centers (FFRDCs) and Prime contractors in the aerospace sector[12]. These two interviewee types were presented with the list of MIT ESD-proposed ilities, along with manufacturability, testability, and operability, which were ilities that arose as important during initial test interviews among practitioners. The results from these surveys showed little agreement between the FFRDCs and Primes, except that manufacturability, testability, and operability were actually quite important. According to the Primes (which are more like practitioners than the FFRDCs), the top six research priorities for (ESD-named) ilities in rank order are scalability, agility, adaptability, robustness, sustainability, and flexibility. Subsequent interviews revealed that in spite of the expressed desires of senior leaders and politicians for ilities, "'ilities' are not part of the lexicon of even the most successful project managers," so somewhere in the technical communication chain, the desire for ilities is not being translated into actionable concepts at the project level[12].

2.3. Crowd-sourced

Another alternative approach is to use a crowd-sourced method, with one notable example being Wikipedia. The Wikipedia page for "ilities" defines the concept as follows: "within systems engineering, quality attributes are non-functional requirements used to evaluate the performance of a system. These are sometimes named "ilities" after the suffix that many of the words share"[13]. The number of ilities reported on this webpage grew from 61 (on 24 April 2008) to 71 (on 3 March 2010) to 79 (on 8 Feb 2011) to 81 (on 13 Feb 2012). Changeability, manufacturability, quality, reconfigurability, and versatility are missing from the list.

2.4. Informed brainstorming

Illustrative of many uses of ilities in practice and literature is the selection of ilities based on brainstorming from experience and literature[14]. In one paper a large number of ilities were considered (more than 120) and down-selected through the efforts of the authors based on introspection and iteration in answering the following questions: "what are the overall objectives? What values are essential to ensuring effective [system] protection? What values are essential to architectures?" This study recognized that "no standard list of applicable "ilities" exists…" and "almost any attribute may be created by adding '-ility' to the end of the word…"[14]. The resulting sets of ilities were assigned to "architecture-focused properties" and "system-focused properties." Architecture-focused properties had accessibility, usability, modifiability, and accountability as top level ilities, while system-focused properties had capability, maintainability, and interoperability as top level ilities.

2.5. Standards for relationships

The ISO/IEC 9126-1 standard describes a set of software quality characteristics and sub-characteristics within a quality model. The model has six groups of high level ilities, with several ilities considered as lower level ilities within each group. The groups are: functionality, reliability, usability, efficiency, maintainability, and portability[15]. These ilities are described by as subcomponents of adaptability, which is asserted as the over-arching ility[16]. While the standard describes various ilities within the context of software engineering, the content of this standard mixes "ilities" and software-specific quality measures. Additional standards, as well as common groups of ilities exist, for example RAMS (reliability, availability, maintainability, and safety) and FURPS (functionality, usability, reliability, performance, and supportability)[13].

2.6. Descriptive surveys

A descriptive approach to determining ilities can also be used. In a recent study, the frequency of ilities mentioned in journal articles and Google hits are compared to show the relative descriptive occurrence of various ilities in these written media[17]. In an effort to understand the temporal relevancy within this data, an alternative representation shows the frequency of various ilities mentioned as a function of time. This shows a rise in mentioning of various ilities across distinct time periods (i.e. "epochs" of predominant system types), including the more recent rise of interoperability, sustainability, modularity, and testability.

While illustrative of the fact that not all ilities are mentioned with similar frequency; the work begs the question of whether frequency correlates with importance or relevance of the ilities relative to one another or in an absolute sense. The challenge is that these results do not capture relationships amongst ilities or the cause of the frequencies.

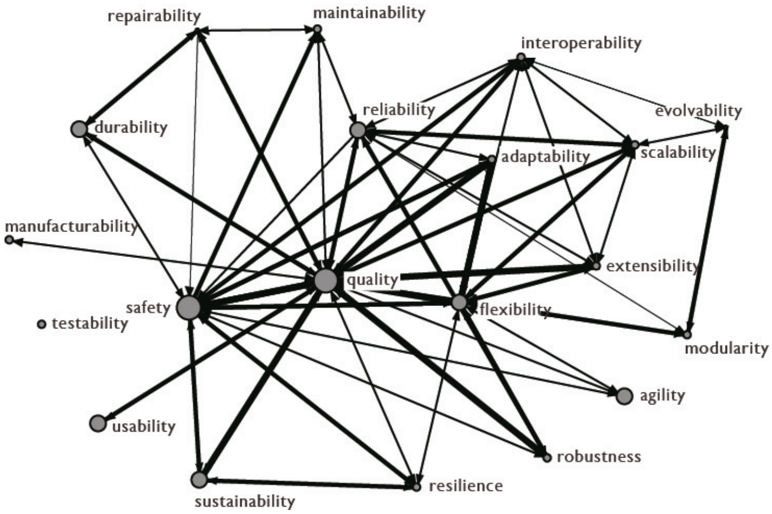

Fig. 1. Ility co-occurrence in the literature, with implied dependence.[17]

In a first step to answering this question, a related descriptive study looks at the co-occurrence of ility terms in the literature, with implied dependence amongst the terms[17]. A directed graph depiction (Fig. 1) tempts the reader to begin to read causal relationships into the links between ilities, but the existence of "co-occurrence" cannot describe the nature of the link, which could include complements, substitutions, or other tradeoffs between pairs of ilities[18]. The descriptive results are a good first step for proposing deeper inquiry into the nature of the relationship amongst ilities, but are sensitive to the particular list of ilities investigated, and does not indicate whether polysemy or synonymy confounds the results (i.e. the distinctiveness and consistency of the semantic content of a given term).

2.7. Prescriptive assertions

As a complementary approach to discovering relationships amongst ilities, prescriptive assertions can be based upon theory or experience, making conceptual leaps in proposing how ilities should relate to one another. Prescriptive semantic approaches, such as definitions in dictionaries, may not agree completely with one another and are dependent on particular assumptions. The strength of prescriptive approaches is the intended self-consistency within a particular approach, as well as transparency in terms of relationships derived from explicit assumptions. Weaknesses of prescriptive approaches include the appropriateness of assumptions and the universality of the results. As an example, one may look at several sources for definitions of terms, such as Oxford English Dictionary[19] or Merriam-Webster dictionary[20]. The definitions for particular terms will not be in exact agreement, but will tend to be similar. The similarity is based partly on the approach for generating these definitions, which is embedded in the linguistic research of the staff responsible for the definitions. At some point, casual dictionary users came to assume that the proposed definitions in the dictionaries are "correct" and therefore seek to adhere to these prescriptive meanings in their own usage of terms. More sophisticated users may be more familiar with the techniques used to generate the definitions within particular dictionaries (for example the Oxford English Dictionary explicitly traces the historical usage and evolution of meaning for terms, enabling users to trace assumptions regarding semantic stability).

In the domain of ility-related prescriptive approaches, at least two types of work have been proposed: experience-informed approaches and theory-informed approaches. Both of these approaches utilize induction and deduction, with the former rooted more in induction and the latter more in deduction.

2.7.1. Change-type ilities

One of the key papers asserting a relationship amongst a set of ilities uses the concept of "changeability" as a higher order ility that encompasses four key ilities: adaptability, robustness, flexibility, and agility[7]. In this work, a number of other ilities are mentioned in the context of "architecture principles" for achieving the changeability-related ilities. These include: simplicity, independence, modularity, integrability, autonomy, scalability, non-hierarchy, decentralization, and redundancy. Implicit in the paper is evolvability. The prescriptive nature of the framework for relating the ilities into change-type and architecture-type is based on the authors' research and experiences in German product development (e.g. BMW).

Another similar changeability-related work asserts "changeability" as an overarching ility, with five underlying related ilities along two relationship dimensions: adaptability, flexibility (change agent), and robustness, scalability, and modifiability (change effect)[8,12]. The concepts of change agent, change effect, and change mechanism are introduced as a means to generalize the concept of changeability and to provide a simple basis for deriving the other five ilities (i.e. adaptable scalability is an internal change agent instigating a change in the level of a system parameter). The "adaptable" vs. "flexible" ility label is dependent on whether the change agent is internal or external to the system boundary. This work introduces the concept that changeability is relative to one or more parameters of a system. In this way a system could display all five ilities. Additionally, this work introduced the concept of a verifiable changeability statement as a first step towards concretizing the concepts of these ilities into actionable project engineering and requirements (Fig. 2).

The system shall be	______________	______________	in	______________	for less than	________ .
	(change agent type)	(change effects)		(system parameter)		(resources)
	flexibly or adaptably	scalable, modifiable		with range		

Fig. 2. Template for verifiable changeability requirement.[8]

2.7.2. Means-ends hierarchy from prescriptive definitions

Recently, an initial exploratory study sought to uncover potential means-ends hierarchical relationships amongst a set of ilities[18]. In this study, four groups of graduate students constructed means-ends hierarchies from a given set of ilities. Each of the four groups independently derived distinct "hierarchies." In addition to connecting ilities with "means-ends" links, each of the four groups independently proposed a second grouping criterion called "level" or "depth" and structured their group hierarchies to display this quality. Fig. 3 below illustrates the aggregate of the four hierarchies, with solid lines indicating 3 or 4 groups in agreement of means-ends relationship between two ilities, and dashed lines indicating 2 groups in agreement. The vertical placement of each ility corresponds to the median level assigned to that ility across the four groups. The lack of consensus and emergent "depth" criterion suggests that more than "means-ends" relationships exist amongst the ilities.

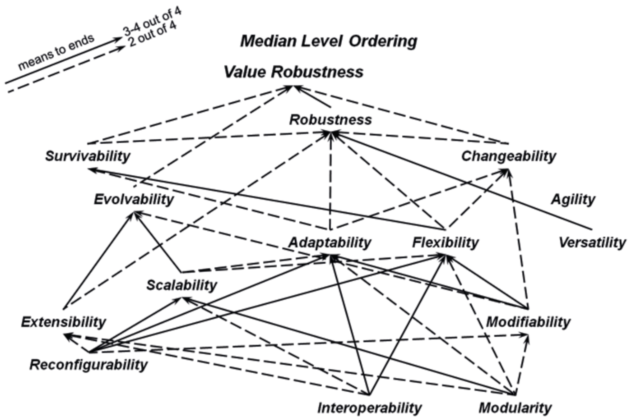

Fig. 3. Median level ordering means-ends hierarchy from study.[18]

Feedback from the study indicated that the “bottom” ilities reflected a different concept than the higher ilities. In particular, modularity and interoperability were viewed as a different perspective on a system lifecycle property than the others and implied particular architectural choices. This insight corresponds to the “architecture principle” concept described in Ref 7 and implies that these ilities may belong to a different semantic field than the others.

More recent research investigating ilities of interest for the DoD proposed a multi-level hierarchy, which emerged from investigating ilities as implied in use, and corroborated through literature and interviews[21]. This hierarchy is depicted in Fig. 4, illustrating some overlap between sets of ilities, such as robustness belonging to both dependability and resilience.

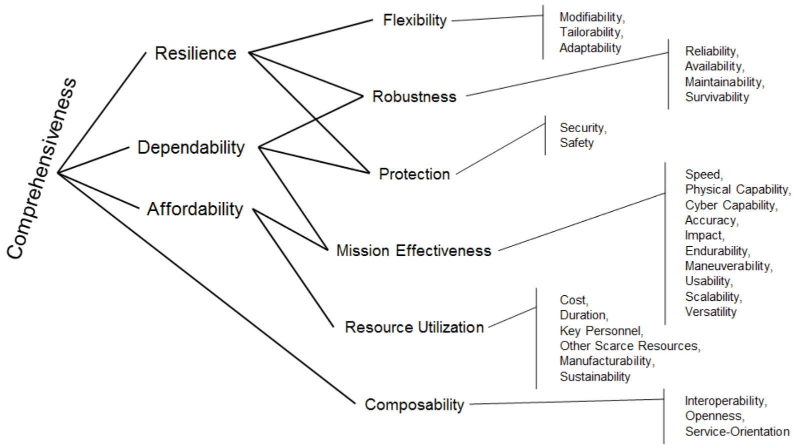

Fig. 4. Proposed DoD-centric ilities hierarchy.

3. Generalizing the Changeability Statement: A Semantic Basis

Building upon the insights from the various approaches for describing ilities above, and drawing inspiration from the linear algebra concept of a basis as a spanning set that defines a space, this paper now describes an initial approach for creating a prescriptive semantic basis for consistently representing ilities within a particular semantic field. At this time, the semantic basis, made up of twenty categories, is believed to span the change-type ility semantic field and excludes the architecture-related semantic field that includes “bottom” ilities[18] and “architecture principles”[7] described above. Through successive refinement, these categories were derived from an earlier effort to define a 10, and later 14, dimensional basis for describing change-type ilities[22].

Beginning with change agent and change effect as two categories for defining a change, a larger set of categories is proposed for defining a larger set of possible changes for a system. The twenty categories, which together form the semantic basis, are intended to collectively define a change in a system, thereby creating a consistent basis for specifying change-type ilities in formal statements. A system that can be verified to display the quality described in the statement can then have a traceable and consistent means for displaying a verifiable desired ility.

The twenty categories are: perturbation, context, phase, agent, impetus (nature, parameter, origin states, destination states, aspect), mechanism, outcome (effect, parameter, origin states, destination states, aspect), level of abstraction, and value qualities of the change (reaction, span, cost, benefit). Unique choices for each of these categories will formulate the change-type ility statement. The twenty categories along with their associated choices are illustrated in Fig. 5. The semantic basis aims to capture the essential differences among change-type ilities through specification of the following general change statement with regard to a particular system parameter:

In response to “perturbation” in “context” during “phase”, desire “agent” to make some “nature” impetus to the system “parameter” from “origin(s)” to “destinations” in the “aspect” using “mechanism” in order to have an “effect” to the outcome “parameter” from “origin(s)” to “destination(s)” in the “aspect” of the “abstraction” that are valuable with respect to thresholds in “reaction”, “span”, “cost” and “benefit.”

Prescriptive Semantic Basis for Change-type Ilities

In response to "perturbation" in "context", desire "agent" to make some "change" in "system" that is "valuable"

Perturbation	Context	Phase	Agent	Impetus Change					Mech	Outcome Change					System	Valuable* (this category is not complete)			

In response to "perturbation" in "context" during "phase" desire "agent" to make some "nature" impetus to the system "parameter" from "origin(s)" to "destination(s)" in the "aspect" using "mechanism" in order to have an "effect" to the outcome "parameter" from "origin(s)" to "destination(s)" in the "aspect" of the "abstraction" that are valuable with respect to thresholds in "reaction", "span", "cost" and "benefits"

Perturbation	Context	Phase	Agent	Impetus* (optional)					Mech	Outcome					Abstraction	Reaction	Span	Cost	Benefit
				Nature	Parameter	Origin	Destination	Aspect	Mechanism	Effect	Parameter	Origin	Destination	Aspect					
optional	circumstantial: required; general: optional	null	optional	null	required	optional	optional	null* (this is implied by "parameter")	Optional	null	required	optional	optional	null*	optional	required	required	required	required
"name"	"name(s)"		"name(s)"		"parameter"	"state(s)"	"state(s)"		"name"		"parameter"	"state(s)"	"state(s)"		"name"	"threshold w/units"	"threshold w/units"	"threshold w/units"	"threshold w/units"
none	circumstantial	pre-ops	none	decrease	level	one	one	form		decrease	level	one	one	form	architecture	sooner	shorter	less	more
disturbance	general	ops	internal	same	set	few	few	function		same	set	few	few	function	design	later	longer	same	same
shift	<empty>	inter-LC	external	increase	<empty>	many	many	operations		increase	<empty>	many	many	operations	system	always	same	more	less
<empty>		<empty>	either	not-same		<empty>	<empty>	<empty>		not-same		<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>	<empty>
			<empty>	<empty>						<empty>									

Fig. 5. Change-type prescriptive semantic basis in 20 categories.

Perturbation	Context	Phase	Agent	Impetus* (optional)					Mech	Outcome					Abstraction	Reaction	Span	Cost	Benefit	Ility Label
				Nature	Parameter	Origin	Destination	Aspect	Mechanism	Effect	Parameter	Origin	Destination	Aspect						
shift		ops								same	"Value"		few							Value Robustness
disturbance		ops								same	"Value"		few							Value Survivability
shift		ops								same			few							Robustness
shift		ops		not-same						same			few							Active Robustness
shift		ops		same		few	few			same			few							Passive Robustness
shift		ops	none	same		few	few			same	level		few	form	system					Classical Passive Robustness
disturbance		ops								same			few							Survivability
			either	not-same						not-same										Changeability
shift	general	inter-LC		not-same						not-same					architecture					Evolvability
			internal	not-same						not-same										Adaptability
			external	not-same						not-same										Flexibility
				not-same						not-same	level									Scalability
				not-same						not-same	set									Modifiability
		ops	either	not-same						increase	set									Extensibility
				not-same						not-same							shorter			Agility
				not-same						not-same						sooner				Reactivity
		ops		same	"Element set"	one	one	form		not-same	"Link set"			form						Form Reconfigurability
		ops		same	"Element set"	one	one	operations		not-same	"Order set"			operations						Operational Reconfigurability
		ops		same		one	one	form/ops		not-same	set		few/many	function						Functional Versatility
		ops		same		one	one	form/fnct		not-same	set		few/many	operations						Operational Versatility
		ops		same		one	one	fnct/ops		not-same	set		few/many	form						Substitutability

Fig. 6. Using the semantic basis to consistently identify ility term labels.

Application of the semantic basis begins with a user generating a change statement. The change statement is refined and assigned categorical choices within the basis, with the intention that the applicable ilities will emerge from the specified change statement. For this use to work, particular combinations of choices in the basis must be assigned an ility term label. Such an exercise is similar to assigning definitions for ilities, albeit with an imposed closed form "little language" supplied by the basis. For example, if "agent" is set to "external" then the "flexible" label will apply. In this way, a user does not need to use, or know, a particular ility label a priori, thereby avoiding semantic ambiguity in the terms. If the basis accurately and completely describes the underlying categories for change-type changes, then a user should be able to describe any change-type ility through the basis consistently.

In order to validate the proposed use of the basis, two activities need to be accomplished: (1) refinement of the basis itself, both in terms of complete categories and in terms of choices within those categories; and (2) generation of mapping(s) between patterns in the basis and ility terms. If the latter is accomplished, then any usage of the basis for specifying particular change statements will result in consistently derived ility term labels. It is hypothesized that particular change-type ilities will correspond to particular choice(s) in this basis. In this way, a consistent method for specifying ilities can be pursued. An example of using the semantic basis for mapping ility terms is illustrated in Fig. 6. The results in the figure are a work in progress and are meant for illustration purposes only.

3.1. Using the basis

The twenty categories can be a bit overwhelming at first; alternative versions where the "optional" categories are left out, can be used at different stages of the design lifecycle. For example, leaving out the "mechanism" column means that the statement leaves the mechanism ambiguous, allowing engineers to evaluate alternative mechanisms for meeting the statement. Likewise, leaving out the "impetus" categories results in an outcome-oriented change statement. These alternative use cases are currently being defined and will help to generate specific examples.

Beginning with a desired system parameter change statement, the user makes choices across the categories, signified by the double quotation marks (" ") in the general form of the statement. It is important that this statement be specific to a chosen system parameter, as currently the change-type ilities are defined only in relation to particular system parameters (i.e. there is no such thing as "generalized flexibility"). Choices in the categories for the considered change result in a categorized change description; optional categories do not have to be used.

As an example, a simple (optional categories excluded) statement could be: "in response to loud noises at night," desire "owner" to be able to "change the level of volume" of "his stereo" "in less than one second." This statement requests "flexible" (i.e. owner is external agent) "scalability" in "volume" and specifies the value proposition (i.e. "valuable" is when this change takes less than one second to execute). Alternative stereos can be evaluated on this basis, with those stereos able to change more quickly as being more valuably scalable in this regard. If ility labels have been previously mapped to the basis, then one can automatically derive the applicable ility terms implied by the categorized change statement (i.e. flexibly scalable in this case). The distinction is important since one could have generated an alternative statement that did not specify an external agent, rather leaving the agent open or even internal (e.g. the stereo changes its own volume with associated design implications, and is labeled "adaptable" instead of "flexible").

The categories of the basis are used to differentiate the change-type ilities. As mentioned above, in specifying a change statement, one must have in mind the "parameter" of the "system" that is being affected. The "perturbation" refers to whether the change is in response to a perturbation of finite duration (disturbance), or one likely to last (shift), or no perturbation at all (none), or if it doesn't matter (<empty>). The "context" refers to whether the response is desired in a particular case (circumstantial) or many cases (general), or it doesn't matter (<empty>). The "phase" of the lifecycle when the response occurs has choices of pre-ops (before operations), ops (during operations), inter-LC (between lifecycles), or doesn't matter (<empty>). The "agent" is the force that instigates the response through an impetus to the system; active response requires a change agent, which can be internal or external to the "system" boundary, while passive response doesn't require an agent (none), or it may not matter (<empty>).

Changes within the statement are framed as "impetus" and "outcome" to capture how one might want to specify a range of changes in inputs (impetus), and the resulting range of changes in outputs (outcome). Both the impetus and outcome sections of the statement are described by five categories: 1) the "nature" (impetus) or "effect" (outcome) of the change (decrease, remain same, increase, not-same, or doesn't matter=<empty>); 2) what type of "parameter" change (in level, set or <empty>); whether there is a target number of 3) "origin" states and 4) "destination" states (each ranging from one, to few, to many to <empty>); and 5) the "aspect" referring to whether the change is in the form, function, operations, or doesn't matter (<empty>). The "mechanism" specifies how the outcome is achieved.

The "abstraction" refers to the level of abstraction of the system, which can be at the architecture level (e.g. Boeing 737 aircraft), design level (e.g. 737-800), system level (e.g. 737-800, tail #: C-FTCZ), or doesn't matter (<empty>). Lastly, a grouped set of categories describe how the change can be evaluated as valuable. This set of "valuable" categories relates aspects of value to thresholds in "reaction" (timing) and "span" (duration), "cost" (resources), and "benefit" (utility). "Value" is separated from the rest since it represents a coupled set of tradeoffs that can be used to judge the goodness described in a change statement. For example, one might be willing to accept later, slower, and more expensive if it provides greater utility, but if it provides less utility, then maybe it should be sooner, faster, and cheaper. Many combinations of these can result in ility statements representing valuable change.

A more complete version of the earlier change statement example could then be:

> In response to a loud noise (*perturbation*) late at night (*context*), during operations (*phase*) of system, desire owner (*agent*) to be able to ***impetus***
>
> {increase (*nature*) the knob angle level (*parameter*) from one state (*origin*) to many states (*destination*) in the system form (*aspect*)}
>
> through turning the knob (*mechanism*) that results in the ***outcome***
>
> {increasing (*effect*) the volume level (*parameter*) from one state (*origin*) to many states (*destination*) in the system function (*aspect*)}
>
> in the owner's stereo system (*abstraction*) that takes less than 1 second (*valuable*).

3.2. Discussion

As research progresses on the prescriptive semantic basis, several open questions remain. For the particular basis described in this paper, what types of ilities can be represented in the basis? Can or should the basis be expanded or modified? What are the appropriate basis categories, and what are appropriate choices within each? With regard to mapping specifications within the basis to particular ility term labels, are there consensus patterns in matching ilities to the basis, given particular definitions for each ility? Are there consensus patterns for given ility terms without provided definitions (i.e. is there some inherent, general meaning within an ility term that can be more consistently expressed using the basis than through traditional English definitions)? Investigation into such matters would require engagement with the larger community to elicit feedback on both the basis and ility term label coding. Recent work has begun to implement a formal, web-based version of the basis to facilitate feedback from the broader community, promoting considerations to improve usability of the basis itself[23].

More generally, this research begins to structure the question regarding what semantic fields span the general set of ilities (and whether it is possible). Preliminary results indicate that at least three semantic fields may exist for the general set of "ilities" including change-type, architecture-type, and new ability-type (the last kind includes such ilities as "auditability," "learnability," and "drinkability"[24]). The list of ilities described in this paper is by no means complete and results from a union of terms used in the reviewed literature. Identifying and classifying in use ility terms into appropriate semantic fields will serve to eliminate ambiguity in meaning, usage, and application, as well as allow for the explicit consideration of trade-offs within the semantic field. A consistent basis within a field can allow for direct comparison of its members; for example kinship terms clearly distinguish between the meaning of uncle and cousin, even though a single person could serve in both roles. Using the proposed approach also allows one to consider whether each semantic field can be represented with an internally consistent basis.

Revisiting the concept of relationships amongst the ilities, the basis can provide a first order approximation to clarify semantic differences amongst ilities within a particular semantic field. For example the difference between "flexibility" and "adaptability" is whether the change agent is external or internal to the system's boundary, respectively. The basis can also show how a given change statement can display multiple ilities simultaneously. For example, agility relates to how quickly a change can be executed, so one could desire an agile, scalable change to describe a quick and level-increasing system parameter change. Additional research will be needed to clarify relationships between different semantic fields. For example, how are members of the "architecture-type" semantic field related to the "change-type" semantic field? Preliminary work has begun to describe relationships between particular ilities and design principles, but a unifying understanding of the relationship between semantic fields is not yet mature. Our working hypothesis is that "architecture-type" ilities are enablers for "change-type" ilities, as seen in the literature on design principles for changeability[7], evolvability[25], flexibility[26], reconfigurability[27], survivability[28], and SoS survivability/viability[29].

One of the possible emergent results of this work may be the discovery of "new" ilities that do not yet have ility term labels, and yet may represent important desired system properties, such as the distinction between functional versatility and operational versatility seen in Fig. 7. As an example, varying the versatility concept, by allowing the form to vary in order to achieve similar function using similar operations, results in a "new" ility we can label as "substitutability." On reflection, this ility is a property displayed in computers, where multiple different disk drives or monitors can be substituted for one another.

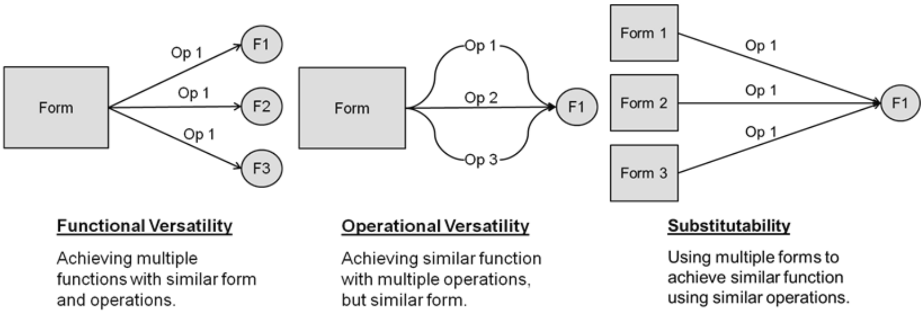

Fig. 7. (a) Functional versatility; (b) operational versatility; (c) substitutability as suggested by the semantic basis.

Given a stable, validated basis, can practitioners or academics use the basis to generate change statements, which will automatically label with the appropriate ilities? Do the ility term labels resonate with the users? The purpose of the research is not to generate more definitions, but rather, unambiguous, verifiable, standardized representations of desired system properties. The approach described in this paper serves partly as a strawman for fostering conversations in the research and practitioner community about a common means for describing relationships among ility terms, as well as to promote alternative means for addressing the semantics of ilities beyond asserted definitions. Computer science-based formal semantic methods, such as techniques used in the Semantic Web, are readily applicable to this problem and the authors hope this paper inspires knowledgeable researchers to address the posed problem with techniques other than those presented in this paper.

The ultimate goal of this research is to stimulate broad discussions and research around developing a basis, or bases, as prescriptive instrument(s) for spanning semantic fields whose union consistently encompasses sets of "ilities." With such an instrument, practitioners can have a consistent and (potentially) more complete list of possible ilities to consider for their systems, as well as a means to create verifiable requirements and system specifications. Academics can have a common basis for enhancing ility-related research and a means to advance the quantification and clarification of relationships amongst ilities in general, as well as a means to educate future engineering students so that one day ilities can become part of the lexicon of successful project managers. Ultimately, adoption and implementation of a unified theory of ilities would hinge on input from the broad community, as well as inclusion into community-based mechanisms, such as curricula, standards, and handbooks[30].

Acknowledgements

The authors would like to thank J. Clark Beesemyer for his SM thesis contributions to the early versions of the semantic basis, as well as Kevin Sullivan, Ke Dou, and Xi Wang of University of Virginia for their feedback on the representation and use cases of the semantic basis. This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract

HQ0034-13-D-0004. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

References

1. Department of Defense. *Better Buying Power*. [Available online: http://bbp.dau.mil].
2. Goerger SR, Madni AM, Eslinger OJ. Engineered Resilient Systems: A DoD Perspective. *12th Conf on Sys Eng Research (CSER).* Redondo Beach, CA. March 2014.
3. Akmajian A, Demers RA, Farmer AK, Harnish RM. *Linguistics*. Cambridge, MA: MIT Press; 2001.
4. Saleh JH., Mark GT, Jordan NC. Flexibility: a multi-disciplinary literature review and a research agenda for designing flexible engineering systems. *J of Eng Des (*2009); **20**(3):307-323.
5. Nilchiani R. *Measuring Space Systems Flexibility: A Comprehensive Six-element Framework*. PhD in Aeronautics and Astronautics. Cambridge MA: MIT; 2005.
6. de Neufville R, Scholtes S. *Flexibility in Engineering Design*. Cambridge MA: MIT Press; 2011.
7. Fricke E, Schulz AP. Design for changeability (DfC): principles to enable changes in systems throughout their entire lifecycle. *Syst Eng* (2005); **8**(4): 342-359.
8. Ross AM, Rhodes DH, Hastings DE. Defining changeability: reconciling flexibility, adaptability, scalability, nodifiability, and robustness for maintaining lifecycle value. *Syst Eng* (2008); **11**(3): 246-262.
9. Ryan ET, Jacques DR, Colombi JM. An ontological framework for clarifying flexibility-related terminology via literature survey. *Syst Eng* (2013); **16**(1): 99-110.
10. Ferguson S, Siddiqi A, Lewis K, de Weck OL. Flexible and reconfigurable systems: nomenclature and review. *Proceedings of the ASME 2007 IDETC & CIE 2007*. DETC2007-35745. Las Vegas, NV, September 2007.
11. ESD Symp Cmte. ESD terms and definitions. *MIT Engineering Systems Division Working Paper Series*. ESD-WP-2002-1 Ver 12; October 2001. p. 4. [Available online: http://esd.mit.edu/WPS/].
12. Ross AM. *Managing unarticulated value: changeability in multi-attribute tradespace exploration*. PhD in Engineering Systems. Cambridge, MA: MIT; 2006.
13. Wikipedia. *List of system quality attributes* (redirect from '*ilities'*), http://en.wikipedia.org/wiki/Ilities (last accessed 28 Feb 2012).
14. Cotton LD, Haase GA, Havlicek JD, Thal AE. Value-driven enterprise architecture score (VDEA-Score): a means of DoDAF architecture evaluation. *7th Conf on Sys Eng Research*. Loughborough, UK, April 2009.
15. ISO/IEC. *ISO/IEC 9126-1: Software engineering — product quality — part 1: quality model*; 2001.
16. Engel A, Browning T. Designing for adaptability by means of architecture options. *Syst Eng* (2008); **11**(2):.125-146.
17. de Weck OL, Roos D, Magee C. Life-cycle properties of engineering systems. In: *Engineering systems: meeting human needs in a complex technological world*. Cambridge, MA: MIT Press; 2011. p. 65-96.
18. de Weck OL, Ross AM, Rhodes DH. Investigating relationships and semantic sets amongst system lifecycle properties (ilities*). 3rd Int'l Eng Sys Symp: CESUN 2012*. TU Delft, 18-20 June 2012.
19. Oxford English Dictionary: the definitive record of the English language. Oxford University Press. [Available online: http://www.oed.com].
20. Miririam-Webster. An Encyclopedia Britannica Company [Available online at: http://www.merriam-webster.com].
21. Boehm B. (Ilities) Tradespace and Affordability – Phase 2. Final Technical Report SERC-20130TR-039-2. December 31, 2013. [Available online at http://www.dtic.mil].
22. Beesemyer JC. *Empirically Characterising Evolvability and Changeability in Engineering Systems*. SM in Aeronautics and Astronautics. Cambridge MA: MIT; 2012.
23. Dou K, Wang, X, Tang C, Ross A, Sullivan K. An evolutionary theory-systems approach to a science of the ilities. *13th Conf on Sys Eng Research*. Hoboken, NJ. March 2015.
24. Ross AM. Defining and using the new 'ilities' *MIT SEAri Working Paper Series.* WP-2008-4-1; June 2008. [Available online: http://seari.mit.edu].
25. Beesemyer JC, Fulcoly DO, Ross AM, Rhodes DH. Developing methods to design for evolvability: research approach and preliminary design principles. *9th Conf on Sys Eng Research*. Los Angeles, CA, April 2011.
26. Banerjee P, de Weck OL. Flexibility strategy— valuing flexible product options. *INCOSE/ICSE Conf on Synergy btwn Sys Eng and Proj Mgt*. Las Vegas, NV, 2004.
27. Siddiqi A, de Weck OL. Modeling methods and conceptual design principles for reconfigurable systems. *J of Mech Des*; 2008, **130**(10): p. 101102-1 – 15.
28. Richards MG, Ross AM, Hastings DE, Rhodes DH. Survivability design principles for enhanced concept generation and evaluation. *INCOSE Int'l Symp 2009*. Singapore, July 2009.
29. Mekdeci B, Ross, AM, Rhodes DH, Hastings DE. Examining survivability of systems of systems. *INCOSE Int'l Symp 2011*. Denver, CO, June 2011.
30. INCOSE SE Handbook Working Group. Haskins C, ed. *Systems Engineering Handbook v.3.2.2*. INCOSE-TP-2003-002-03.2.2. October 2011. [Available online at http://www.incose.org].