Basic Concepts and Distinctions for an Aerospace Ontology of Functions, Entities and Problems

Jane T. Malin
NASA Johnson Space Center
2101 NASA Road 1
Houston, TX 77058
281 483-2046
jane.t.malin@nasa.gov

David R. Throop
The Boeing Company
2100 Space Park Drive
Houston, TX 77058
281 460-8415
david.r.throop@boeing.com

Abstract—Aerospace problems and corrective actions involve hardware, software, processes, operations, humans and organizations. This paper describes a set of taxonomies for interpreting descriptions of aerospace entities, functions, properties and problems. Each category expresses a class concept. Categories have associated 'mapping' words and phrases (synonyms or names) that express these concepts. The taxonomies and mapping words are designed to support ontology development and aid text analysis. The text analysis has been used to semi-automatically generate system functional models from requirements. Text analysis can also be used to find trends and recurrences in problem reports and change requests. The paper describes an Upper Ontology that provides top-level distinctions for classifying objects, occurrences, properties and mathematical abstractions. Additional taxonomies partition Descriptions, Concepts, Entities, Functions/Actions, Problems, and Refining Properties. The paper describes a use case, semantic extraction and classification of key concepts in free text in the Space Shuttle problem reporting database. [1] [2]

TABLE OF CONTENTS

1. INTRODUCTION

Engineering complex aerospace systems involves many heterogeneous disciplines. Safety and mission assurance are broadly span heterogeneous viewpoints and nomenclatures from many specialties. Analysis of discrepancies, problems and mishaps typically involves considering a wide spectrum of potential causes, from mechanical to organizational domains. What is the hazard and what things or functions/actions produce it or are affected by it? Bridging across heterogeneous viewpoints is a significant challenge in development of aerospace standard nomenclatures, codes, taxonomies and ontologies [29].

Key information about a problem or discrepancy is commonly embedded in short titles or disposition summaries. Authors embed key information in free text fields because it can be hard to identify the right data codes. The codes can be confusing or out of date, and may not match the problem situation being reported. Individual keywords can be extracted from the text as tags, but simple keyword approaches are as brittle as codes. There are too many ways of conveying an idea in natural language.

The approach described in this paper uses ontologies to interpret and categorize key information in the text. Taxonomies define concepts and concept hierarchies. Ontologies include taxonomies and add value by using axioms, restrictions and properties to relate concepts. They not only accommodate many ways of expressing concepts, but the concepts in ontologies can also be combined (via relations between categorized terms) into phrases that can be extracted from text and analyzed. Text parsing that uses ontologies is thus "semantic". This semantic parsing can be used for tagging and processing knowledge from diverse dissimilar data sources for reuse.

Statistical text mining is another approach to categorizing documents or data records containing free text. Clusters and hierarchies of similar sources are extracted by finding text co-occurrences [18]. The strength of this approach is that very little needs to be known about the text to apply the approach. A serious weakness of the approach is that experts need to make sense of the clusters and hierarchies. Bridging this "semantic gap" [32] can be difficult and frustrating. Another weakness is use of unparsed words as features. Semantic parsing is needed to use syntactical text variants (word equivalents) as features. This point is illustrated in the parsing examples in Section 9.

1

[2] IEEEAC paper #1357, Version 3, Updated December 18, 2006

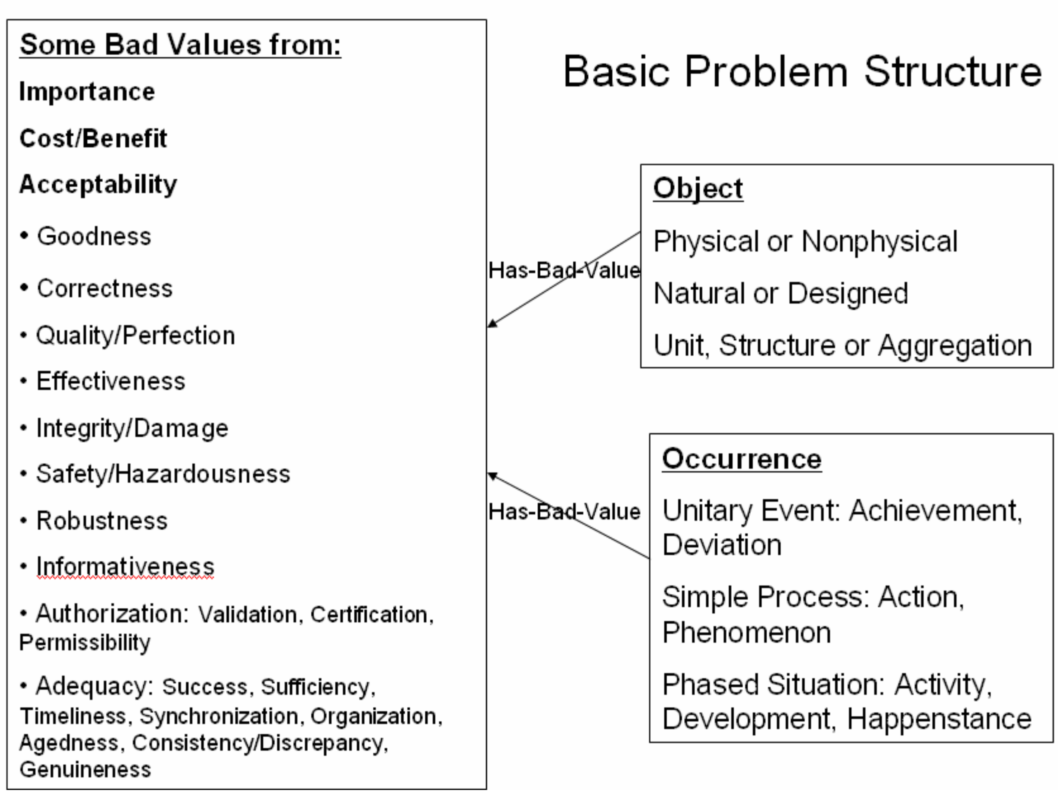


Figure 1. Basic relationships in problem situations. Many types of objects and occurrences can have many bad attributes

The weakness of the semantic parsing approach is that it requires development and maintenance of taxonomies and ontologies. This paper describes the development of a comprehensive aerospace ontology, the methods of development, and the hierarchies that have been developed. The aerospace ontology was first developed for extracting information from text for hazard identification and failure modes and effects analysis [16]. The semantic text parsing approach, in a tool called Reconciler, was applied to requirements text and risk data, to support semi-automated generation of system interaction models from text [17]. System structure (functional or physical), vulnerabilities and safeguards were captured for analysis. Equipment classes and Function/Action classes were also used to organize default information (standard hazards, vulnerabilities and safeguards) for system components in libraries. Defaults can guide users to identify potential system problems and mitigations.

Recently, the work has focused on identifying recurring problems in problem and discrepancy databases. The goal is to classify problems (hazard, damage, impairment or discrepancy) and the things that have the problems. Figure 1 illustrates the basic relationships in a problem situation. The goal is to find objects or occurrences that have problematic attributes, and use the classification of these things to browse and search the reports. The paper describes a pilot study where the ontology was used in a semantic parsing tool to extract equipment with problems from records in the Space Shuttle problem reporting and corrective action (PRACA) database.

2. METHODS

There are many criteria for a good ontology or taxonomy [3]. These include comprehensive coverage, flexibility for multiple uses and domains, usefulness for analysis and usability for consistent coding [21]. Rector emphasizes design for reuse, maintainability and evolution, and proposes methods for modularizing ontologies to make them clean and clear for maintenance [27].

There are various methods for developing taxonomies and ontologies, and they can be used in combination. Early work usually involves some hand building with domain experts, to ensure that critical domain distinctions are included. Two comprehensive hand-built ontologies are OpenCyc [26] and WordNet [43]. Protégé is an open-source tool for ontology development [25]. Statistical and empirical approaches can be used to enrich and update taxonomies [16, 41].

One approach to merging, extending and mapping nomenclatures is to adopt a high level set of basic abstract categories and major relations between them. Upper-level ontologies capture important basic distinctions about things at the highest level of abstraction, and helps organize terms and distinctions cleanly [27]. The aerospace ontology adopts high level distinctions that occur in several upper ontologies, principally DOLCE [6, 7, 42], the IEEE Suggested Upper Merged Ontology (SUMO) [22] and OpenCyc. The Description and Situation ontology [7] has also been adapted and extended in the aerospace ontology. The aerospace Upper Ontology will provide axioms, properties, restrictions, context and attachment points for the aerospace mid-level and domain modules.

A challenging fact about domain knowledge and is that it evolves. Just as important, domain uses for taxonomies and ontologies change. This means that ontologies will inevitably change and evolve. It is not unusual for perform significant re-engineering on an large multi-use ontology [upper bio]. Existing taxonomies and ontologies have been frequently reconciled and integrated [6, 10, 19, 22]. Noy [24] provides an overview of such approaches. Quality assurance and versioning tools and methods have been developed for evolving ontologies [28, 34]. Taxonomies and ontologies can be evaluated for quality, consistency, modularity and maintainability.

One approach to the problem of evolution is to use structures that facilitate evolution in both development and maintenance. Maintaining modularity in taxonomies and ontologies helps keep them clean, maintainable and reusable. Semantic relationships can be used to relate categories from the separated modular hierarchies. These category trees should use subsumption ("B is a kind of "A"). The semantic relationships can include Has-Bad-Value, Has-Part and Plays-Role, for example. See Rector for further details on approaches that can normalize and achieve modularity [27]. The aerospace ontology is currently being restructured to increase modularity and simplicity, using methods recommended by Rector.

Another approach to the problem of evolution is to use simpler and more changeable structures at first, delaying certain areas of formalism until the simpler structures have evolved to support multiple potential uses in the domain. This approach can support rapid evolution and enrichment. Almost all aerospace ontology development thus far has focused on modular taxonomic kind-of hierarchies with associated mapping words. This is possible because Reconciler can use nomenclature in this form, by embedding some abstract relationships and restrictions in the parsing tool. Formal RDF and OWL implementations in Protégé also could not keep up with the pace of change, and did not provide straightforward support for mapping words. As the aerospace ontology structures stabilize, formal languages can be used to represent concepts, properties, restrictions and axioms.

Aerospace ontology development began with a focus on hand-crafting taxonomies of objects, functions and problems for multiple subdomains. Each class in the taxonomy was populated with "mapping" words and phrases (synonyms or examples), which capture alternative ways that categories are expressed in free text. An example of mapping words is shown in Table 1. Many mapping words and phrases were also derived from semantic parsing of aerospace text, using the early versions of the ontology [16, 35]. Related mapping words have been identified using taxonomies, thesauruses and dictionaries available on the internet [1, 43]. There are currently about 10K mapping words in the aerospace hierarchies.

Table 1. Mapping Words for Categories of Uneven

Uneven [surface]: uneven, flawed, ridged, rough, roughened, gritty, non-uniform, bumpy, irregular, marred

- **Extrusions**: blistered, blemished, burred, with whiskers, with nodules, with extrusions, with protrusions, with asperities, with dendritic growth, built up, fuzzy, with fuzz, with flakes
- **Gaps**: pitted, dented, dinged, gouged, cratered, with voids, with cavities, with vesicles, with holes, with bubbles, with indentations, with gaps, with craters, permeable, porous
- **Eroded**: eroded, fretted, worn, spalled, galled, chafed, dulled, abraded, frayed, skinned, thinned, stripped
- **Marked**: scratched, flaking, scraped, marked, spotted, streaked, with drill marks

Three major modular, mid-level domain hierarchies have been developed for aerospace-relevant entities, functions/actions and refining qualities/attributes. These mid-level hierarchies provide a framework for identifying and classifying important elements of problem descriptions – participants, actions and negative effects. In semantic parsing, Entities classify nouns, Functions/Actions classify verbs and Refining Qualities can classify adjectives for entities or functions. The Problem Hierarchy combines negative Refining Qualities with Entities and Functions/Actions to describe types of damage, hazards, impairment and service failures and deficiencies.

A limited set of lower-level domain categories and mapping words have also been developed, to represent local conventions relating to engineering processes, software and electrical and power objects and problems.

Ontology-based tools can be evaluated for usefulness, integration into processes and systems of users, usability and improvement over alternatives (e.g., hits, misses and false alarms in classifying and extracting).

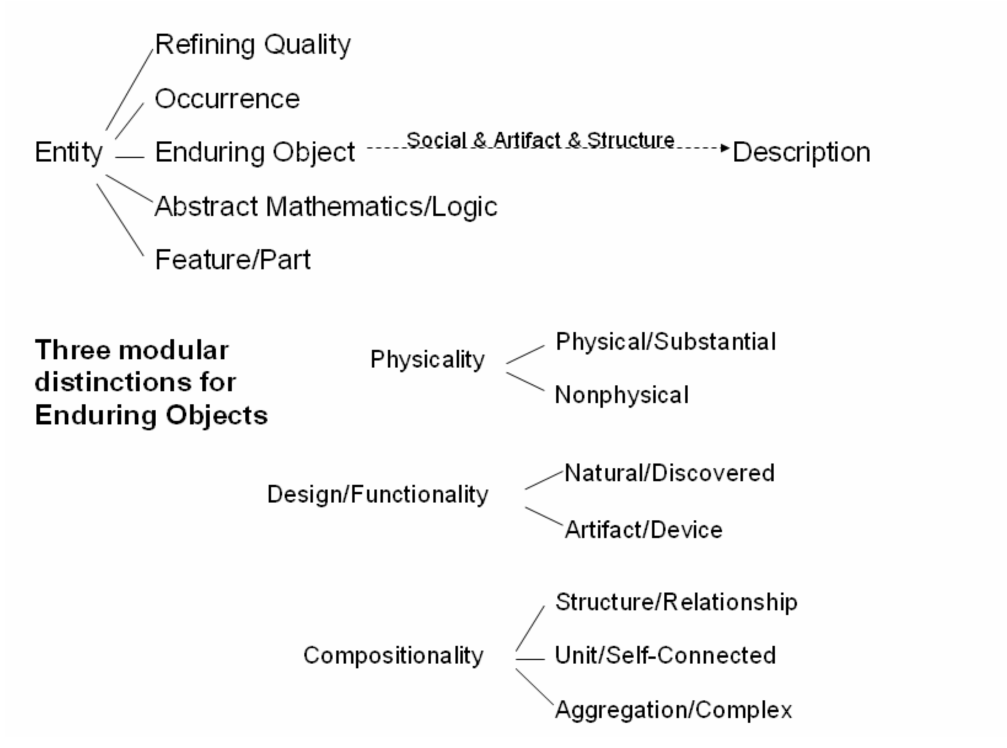


Figure 2. High-Level distinctions in Aerospace Upper Ontology

Developers of the KIM system describe a comprehensive evaluation of an ontology-based information retrieval tool [12]. Developers of the Corese Search Engine describe evaluation in real world information retrieval applications [4]. Thus far, the aerospace ontology and Reconciler have been combined for use only in pilot studies and demonstrations.

3. UPPER ONTOLOGY

An upper ontology provides the top-level distinctions that can be used to classify things, with associated assertions. These distinctions help define attributes and relationships that are appropriate for subcategories and instances.

The Aerospace Upper Ontology is designed to map to some widely available upper ontologies, including DOLCE, SUMO, OpenCyc and the Simple Bio Upper Ontology [30]. The comprehensive DOLCE upper ontology has been the most useful. The distinction between occurrences ("perdurants") and objects ("endurants") and the related distinction between Situations and Descriptions in [7] are useful in engineering. For aerospace uses, distinctions between specifications, capabilities and performances are important. SUMO is the main source of the Abstract Mathematics hierarchy. Distinctions from other upper ontologies have been incorporated and other distinctions have been added or changed, to accommodate safety concepts and enable links to the mid-level aerospace hierarchies. A sketch of the basic distinctions in the aerospace upper ontology is shown in Figure 2. The top half of the figure shows the basic distinctions. The bottom half shows three modular distinctions that can be combined in partitioning Enduring Objects. All three distinctions are used in defining an important class of Enduring Object called a Description (a Nonphysical (Social) Artifact Structure). This top-level structure for Objects is also shown in Figure 1. Figure 1 also shows the top level distinctions in Occurrences.

The three modular hierarchies for Enduring Objects were developed when problems were encountered with entangled distinctions in upper ontologies with strict hierarchy. The three distinctions can be applied to the same entity, leading to combinations of properties, relations and inferences that are associated with each viewpoint. For example, a person can be classified as a Social thing, a Natural thing and a Unit. Further Upper Ontology definitions and examples can be found in DOLCE category descriptions and at the WonderWeb project website [42]. A more detailed taxonomy for the Aerospace Upper Ontology is shown in Table 1 in the Appendix. To save space, some sets of subcategories are listed after the parent, separated by semi-colons.

Description and Concept object taxonomies shown in Table 2 in the Appendix. These taxonomies are based on the Descriptions and Situations Ontology [7] and Service Ontology [19]. Descriptions (or Specifications) and their underlying Concepts can be used for specifications, designs, explanations and agreements that are engineering products.

Concepts are parts of Descriptions that categorize their elements, including context, constraints, participants and actions. The Resource subcategories are influenced by Modarres [20]. The Function/Action class provides a hook for further elaboration in the Function/Action Hierarchy.

4. PROPERTIES

Refining Qualities and Features/Parts can be used to describe and select Enduring Objects and Occurrences. Abstract Mathematics/Logic provides the abstract foundations and scales for Refining Qualities and computational or logical occurrences or objects.

The Refining Qualities taxonomy contains types and mapping words that serve as adjectives for entities or functions. Refining Qualities describe states, statuses and evaluations. The Problem Hierarchy combines negative Refining Qualities with Entities and Functions/Actions to describe types of damage, hazards, impairment and service failures and deficiencies.

Types of Refining Qualities include physical aggregation and structural states, truth and time statuses and value statuses. A more detailed Refining Qualities taxonomy is provided in Table 3 in the Appendix, with some example mapping words.

The top level of the Value/Relation Status part of the taxonomy is shown in Figure 1 on the left hand side. The Assurance/Acceptability Value taxonomy is most elaborated, since evaluation is an important activity in certification and in describing problems and discrepancies. Many categories has a subcategory that is negative and a subcategory that is positive. Negatives and opposites are important for interpreting descriptions of problems and discrepancies. The Assurance and Acceptability hierarchies provide hooks for further elaboration in the Problem Hierarchy. They also relate to types of constraints in Descriptions and in the Entity Hierarchy.

6. FUNCTION HIERARCHY

The development of the Aerospace Ontology began with Function, Entity and Problem taxonomies. The Functional Hierarchy is one of the working hierarchies used by Reconciler. The remaining working hierarchies are the Entity and Problem hierarchies.

These working hierarchies are also used to organize browsing of tagged text descriptions to find recurring patterns. The hierarchies help users locate related sibling, parent and child classes of categorized "hits".

The Function/Action Hierarchy classifies functions and actions for processing, placing, serving, energizing and controlling/performing. The categories of Serving, Energizing and Controlling are meta-functions [13], since they can act on other functions. Sources for the distinctions in this hierarchy include Hirtz et al. [10], Kitamura and Mizoguchi [13], Modarres [20], WordNet [43], FrameNet [5] and VerbNet [40]. The organization and contents of the Control/Manage/Perform class are influenced by work on software goals [14] and on distinctions in organizational and cognitive psychology [11, 23]. These diverse sources provided a helpful challenge - to find common distinctions across viewpoints and uses.

The Function/Action Hierarchy is the first working taxonomy presented because the other two use functional distinctions. Some Entities can be categorized by their main function. Some types of problems are functional problems. Because of the importance of negatives and opposites in describing problems, the Function Hierarchy includes many such categories. There are currently about 250 classes in the Aerospace Function/Action Taxonomy, which is shown in Tables 4 and 5 in the Appendix.

The functions are expressed as verbs, as actions that can be viewed as part of specifications or as part of occurrences. In fact, initial work on function attributes has resulted in three versions of function attributes, corresponding to their use in a request (requirements), an offer (design) or an occurrence (implementation).

7. ENTITY HIERARCHY

The Entity Hierarchy provides more detailed classes and mapping words for objects, descriptions, occurrences and features/parts in the Aerospace Upper Ontology Taxonomy. Entities include types of equipment, substances, regions and interfaces. The development of the Entity Hierarchy began with engineering taxonomies inspired by Paredis [31]. The entities classified in the Entity Hierarchy play the roles of participants in descriptions: Performer/Agent/Actor, Instrument, Resource, Product or Patient/Operand. There are currently about 200 classes in the Entity Hierarchy, which is shown in Table 6 in the Appendix.

The categories in the Equipment/Implement/Tool category use the Function Hierarchy categories in Tables 4 and 5. Some of the Entity categories merely provide mapping words for branches of the previously described hierarchies. Table 6 highlights domain-specific categories, indicated by the "DOMAIN:" notation.

8. PROBLEM HIERARCHY

The Problem Hierarchy distinguishes types of damage, hazards, impairments, failures and deficiencies. Thus, it can be used to identify and categorize information about risks, symptoms and causes. These are the key concepts for finding and organizing clusters of problem types in the free text fields of aerospace problem reports.

The problem taxonomy was developed by starting with a theoretical framework influenced by the work of Stone and colleagues [36, 37] and by sources that described types of human [11] and material [9, 39] hazards and software problems [15, 23]. The mapping words and the hierarchy have been extended as the authors have used Reconciler to analyze text and databases with exemplars (descriptions, symptoms, causes) from aerospace problem reporting systems. Recently, an application to a Discrepancy Report database has resulted in new categories and mapping words for some process problems.

For easier maintenance and use, most classes in the Problem Hierarchy should be made up of combinations between negatively evaluated Refining Qualities and Objects or Occurrences. Work is under way to define these as combinations. There are currently about 250 classes in the Problem Hierarchy. The taxonomy of problems with Objects (as sources or sinks of damage or as impaired) is shown in Table 7 in the Appendix. The taxonomy of Performance Deviation problems is shown in Table 8 in the Appendix. Some examples of Problem mapping words and their use are given in Section 9.

9. FINDING KEY DATA IN PRACA TEXT

The Columbia Accident Investigation Board expressed concern about repeated operational anomalies [8]. The NASA Engineering and Safety Center has efforts to identify recurring anomalies from operational records such as the PRACA databases for the Space Shuttle. If each PRACA entry could be classified into one or more categories in the Problem Hierarchy, the results could be browsed for previously unrecognized clusters.

There are many powerful techniques and commercial tools for data browsing. These mostly depend on data values without typos – i.e. short entries in a rows-and-columns database. Such browsing is not possible in free text fields unless key information is extracted and tagged.

The titles of most Shuttle PRACA entries are English sentences stating the nature of a problem. There is typically a subject, which is some piece of equipment, followed by verb and object phrases expressing some problem condition. The challenge is to tag the phrases that express a problem condition, and also to tag the entity (usually a piece of equipment) having the problem. A complete solution is beyond the art of current text mining. However, this domain tolerates some false-positives and false-negatives from a tool that aids search and browsing. A cluster of recurring problems may still be recognized even if some occurrences are misclassified.

There are three top issues in PRACA tagging:.

(1) *Stemming and inflection*–a concept like *conform* can be expressed as a verb, noun (*conformance*) or adjective (*conformal.*) These are further modified by tense and number. The inflections' spelling patterns are only semi-regular.

(2) *Prefixes and auxiliary words*–negation is expressed with prefixes (*non-, anti-, im-)* and auxiliary words (*not, never, no, without)* which may not be adjacent to the concept word *('had never completely conformed', 'allowed the nonconformance.'*) Beyond negation, concepts about problems use several other style modifiers expressed either with prefixes, adverbs or auxiliary verbs – *too early, too late, too little, too much, irregular.*

(3) *Synonyms, related and variant terms*–concepts are expressed variously. True synonyms express near-identical concepts, but other pairs express important differences and yet are closely related (*pitted* and *corroded.)*

The Problem Hierarchy and its mapping words directly address (3). But issues (1) and (2) must be addressed also.

To match text phrases to the Problem Hierarchy, a method was developed to canonize words and phrases. Each word was analyzed and driven to a base-form and (optionally) a modifying style. The base-form drives nouns, adjectives and adverbs to their underlying verb-forms (or noun-forms when no verb-form was known.) I.e., (*conformal, conformation, conformally, conformed, conforming)* → *conform.* Phrases with style markers drive to a base word plus a style tag. For example, (*misalignment, non-aligning, never aligned*) → *not align.* Canonization was performed morphologically, supplemented with tables of irregular forms.

A synthetic example of program output is shown in Figure 3 – actual PRACA titles with certain controlled information removed, and with variant ways of expressing the *not align* concept. The program successfully clusters all of these together, while excluding instances of *align* that are not negated, like those shown in Figure 4.

Equipment	Failure	Examples	Sentence
TUBE	NOT ALIGN	YA324	TUBE BETWEEN FILTER AND TEE IS MISALIGNING VERTICALLY
		YB163	TUBE ASSY DISALIGNMENT APPROX .700 MAX ALLOWABLE PER SPEC .5625
		YC045	TUBE WAS NEVER ALIGNED TO ACCUMULATOR
		YC530	TUBE ASSY -149 NOT IN ALIGNMENT CAUSING
		YC715	TUBE SCRATCHED ON DYNATUBE SEALING SURFACE; IS MIS ALIGNED APPROX 1 INC
		YD116	DYNATUBE FITTING ON TUBE IS MISALIGNED TO SYS 1 TVC ISOL VALVE WITH 1/10 -TO-1/8 GA
		YD202	TUBE NON-ALIGNED
		YD654	SUPPLY INTERFACE CAUSED BY TUBE MIS-ALIGNMENTS.
		YD657	TUBE MISALIGNS WITH THE H2 #3 VENT QD THE MATING SURFACES.

Figure 3. Variants Example – Hits for "Not Align"

NON-Example Sentence
AIRFLOW TUBE WAS ALIGNED CORRECTLY BUT TOO SHORT.
TUBE ASSY BENT SLIGHTLY DURING REALIGNMENT (.05IN)
GUIDE TUBE PLATE REQUIRED EXTRA FORCE TO ALIGN FASTENER HOLES.

Figure 4. Variants Example – Rejects for "Not Align"

The following approach is used to tag problems in the PRACA titles:

(1) Read in the ontology, including the Problem Hierarchy. Tokenize the input, separating out punctuation, formatting marks, capitalization. Each mapping term is canonized and indexed.

(2) Read in and tokenize the text (PRACA titles) from Excel spreadsheets.

(3) Parse each title as a sentence or a noun-phrase. This step is not necessary; it often fails due to misspellings and garbled titles. But a successful parse limits the scope of style words – so that *'No bolts were used during alignment'* is not classified as *'not align.'*

(4) Pattern-match the text for common 'false-alarm' phrases which match Problem Hierarchy entries but do not signify a problem, ('*corrosion test', 'leakage inspection.*')

(5) Seek matches for each canonized title word in the Problem Hierarchy. Check to assure that the styles match (*incomplete halting* could match either *not halt* or *irregular halt,* but not *halt* or *premature halt.)*

(6) Tag occurrences of things-of-interest which can have problems – equipment, tools, agents, structure, test articles and some procedures.

(7) Display the output.

The output is structured hierarchically and can be presented in several ways in browsers and faceted search interfaces. These user interface approaches provide context for evaluating and exploring the results at various levels of abstraction.

Figure 5 shows eight different problem reports on bearings that express types of *unevenness*, matched to Unvenness and two of its subcategories in the Problem Hierarchy, by using the mapping words.

This work was done as a pilot study, to explore feasibility for PRACA. About 52,000 PRACA titles were parsed. Since there is a character limit on the length of the title, many of the titles are truncated. In about ¾ of the cases, despite this problem, both a problem and an object (piece of equipment) were found. In the pilot study, about 100 new mapping words and 20 new categories were discovered and used to expand the Problem Hierarchy. More recent expansion has been accomplished while processing Discrepancy Reports.

Equipment	Failure	Examples	Sentence
BEARING	1.1.1.1. Uneven [surface]: uneven, flawed, ridged, rough, gritty, non-uniform, bumpy, irregular, marred		
	ROUGHEN	CA453	NEEDLE BEARINGS AND BROKEN BEARING PARTS WERE ROUGHENED
	GRIT	CD205	FOUND BROWN GRIT COVERING BEARING TEMPERATURE SENSOR.
	NON-UNIFORM	CC919	THE BEARING OUTER EDGE DID NOT HAVE VERY UNIFORM APPEARANCE.
	RIDGE	CC894	BEARING DUKE EDGE BROKE; SHOWS RIDGES ON HOUSING.
	1.1.1.1.1. Extrusions: blistered, blemished, burred, with whiskers, with nodules, with extrusions, with protrusions, with asperities, with dendritic growth, built up, fuzzy, with flakes		
	BLISTER	CC919	BEARING ANTERIOR SUPPORT FAILED; SHOWING BLISTERED APPEARANCE.
	WITH NODULES	CC509	BROKEN WASHERS/SEALS IN FUEL PUMP SHAFT BEARING; WITH NODULES 7/8"
	1.1.1.1.2. Gaps: pitted, dented, dinged, gouged, cratered, with voids, with cavities, with vesicles, with holes, with gaps, with indentations, with craters, permeable, porous		
	PIT	CE423	WEAR UNDER BEARING ASSY ATTACH PLATE, PITTING ON L EDGE
	WITH CAVITIES	CE423	IRREGULARITIES WITH CAVITIES SHOWN BY INSPECTION OF SPHERICAL BEARING

Figure 5. Uneven Bearing Example - Cluster of Eight Hits

10. CONCLUSIONS

The Aerospace Ontology, taxonomies with mapping words are being developed for interpreting information about entities, functions and problems. Generality has been achieved by accommodating multiple diverse viewpoints and uses. Although the goal is an aerospace ontology, it has been practical to start with simpler taxonomies and mapping words. To prepare for a full ontology it has been important to adopt distinctions from upper-level ontologies and related ontologies. They will be sources for axioms, linking definitions and restrictions. Further work is needed to modularize, organize more specialized domain vocabularies and develop the full ontology formalisms. Nevertheless, the current taxonomies and mapping words are sufficient for use in the Reconciler tool for semantic text analysis. This paper provides a snapshot, while work continues toward an ontology.

Capability has already been demonstrated for extracting function-hazard models from specifications [17] and identifying clusters of related or recurring problems. Several browsing interfaces have been prototyped. These provide a variety of views and drilldown capabilities. Further work is needed for Reconciler to take advantage of combinations in the modular hierarchies and to move from operational prototypes to integration into a COTS-based text analytic implementation for robust use.

An advantage of semantic tagging is that it can be combined with other tagging and text mining approaches and tools, using an open-source environment such as UIMA [38]. Combining semantic and statistical approaches [33] with faceted browsing should lead to the powerful tool that NASA needs for extracting knowledge and models for text specifications and finding patterns in problem reports.

ACKNOWLEDGEMENTS

We are indebted to many nameless developers of taxonomies, lexicons, ontologies, OWL and RDF. Thanks to Dr. Luis Flores for Protégé implementations of earlier versions of the ontology. We thank Land Fleming, Martin Feather, Leila Meshkat and Steve Cornford, who advised us on important distinctions in risk analysis. Finally, we are indebted to Mark Shirley and Dan Duncavage, both for moral support and for arranging NASA R&D monetary support when possible.

REFERENCES

[1] Answers.com website http://www.answers.com

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," *Dependable ad Secure Computing*, 1: 11-33, 2004.

[3] Janez Brank, Marko Grobelnik and Dunja Mladenić, "A survey of ontology evaluation techniques," In: SIKDD 2005 at Multiconference IS 2005, Slovenia, October 2005.

[4] O. Corby, R. Dieng-Kuntz, and C. Faron-Zucker. Querying the semantic web with the corese search engine. In *Proc. 15th ECAI/PAIS*, Valencia (ES), August 2004.

[5] FrameNet website http://framenet.icsi.berkeley.edu

[6] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, "Sweetening WordNet with DOLCE," *AI Magazine*, 24 (3): 13-24, September 2003.

[7] Aldo Gangemi and Peter Mika, "Understanding the Semantic Web through Descriptions and Situations," *Lecture Notes in Computer Science Vol. 2888/2003: On the Move to Meaningful Internet Systems 2003: CoopIS, DOA and ODBASE*. pp. 689-706, 2003.

[8] H. Gehman, S. Turcotte, J. Barry, K. Hess, J. Hallock, S. Wallace, D. Deal, S. Hubbard, R. Tetrault, S. Widnall, D. Osheroff, S. Ride, and J. Logsdon. Columbia Accident Investigation Board (CAIB), Volume 1. NASA, Aug 2003.

[9] D. C. Hendershot, R. L. Post, P. F. Valerio, J. W. Vinson, D. K. Lorenzo and D. A. Walker, "Putting the 'OP' Back in 'HAZOP'," MAINTECH South '98 Conference and Exhibition, December 1998.

[10] Julie Hirtz, Robert B. Stone, Daniel A. McAdams, Simon Szykman, Kristin L. Wood, "A functional basis for engineering design: Reconciling and evolving previous efforts," *Research in Engineering Design*, 13(2): 65-82, March 2002.

[11] E. Hollnagel, "Accident Analysis and Barrier Functions," Halden, Norway: Institute for Energy Technology, 1999.

[12] A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, "Semantic Annotation, Indexing, and Retrieval." In Proc. ISWC 2003, pp. 484-499, Oct. 2003.

[13] Y. Kitamura and R. Mizoguchi, "Meta-functions of Artifacts," Papers of the 13th International Workshop on Qualitative Reasoning (QR-99), 136-145, 1999.

[14] E. Letier and A. van Lamsweerde, "Deriving Operational Software Specifications from System Goals", Proc. FSE'10: 10th ACM S1GSOFT Symp: Foundations of Software Engineering, November 2002.

[15] Nancy Leveson, *Safeware: System Safety and Computers*. Reading, MA: Addison-Wesley, 1995.

[16] J. T. Malin, D. R. Throop, L. Fleming and L. Flores, "Computer-Aided Identification of System Vulnerabilities and Safeguards during Conceptual Design," 2004 IEEE Aerospace Conference Proceedings, March 2004.

[17] J. T. Malin, D. R. Throop, L. Fleming and L. Flores, "Transforming Functional Requirements and Risk Information into Models for Analysis and Simulation," 2005 IEEE Aerospace Conference Proc., March 2005.

[18] M. W. McGreevy, "Reporter concerns in 300 mode-related incident reports from NASA's Aviation Safety Reporting System." NASA TM-110413, Ames Research Center, 1996.

[19] Peter Mika, Daniel Oberle, Aldo Gnagemi and Marta Sabou, "Foundations for service ontologies: aligning OWL-S to DOLCE," Proc. 13th international conference on World Wide Web, pp. 563 – 572, New York, 2004.

[20] M. Modarres, "Functional Modeling of Physical Systems Using the Goal Tree Framework," In AAAI-98 Workshop: Functional Modeling and Teleological Reasoning, July 1998.

[21] NASA Engineering & Safety Center Working Group Report #RP-06-11 ("Taxonomy Working Group Final Report"), 2006.

[22] I. Niles and A. Pease, A. , "Origins of the Standard Upper Merged Ontology: A Proposal for the IEEE Standard Upper Ontology." In Working Notes of IJCAI-2001 Workshop: IEEE Standard Upper Ontology, Seattle, Washington, August 2001. SUMO website http://suo.ieee.org/SUO/SUMO/index.html

[23] D. A. Norman, "Categorization of Action Slips," *Psychological Review*, 88(1): 1-15, 1981.

[24} N. F. Noy, Semantic integration: a survey of ontology-based approaches, *ACM SIGMOD Record*, 33 (4): 65-70, December 2004

[25] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. "Creating semantic web contents with Protege-2000," *IEEE Intelligent Systems*, 16(2): 60-71, 2001.

[26] OpenCyc website http://www.opencyc.org

[27] Alan R. Rector, "Modularization of domain ontologies implemented in description logics and related formalisms including OWL," Proc. 2nd Intl. Conf. on Knowledge Capture, pp. 121-128, Sanibel Island, FL, 2003.

[28] J. Rogers, A. Roberts and D. Solomon, "GALEN ten years on: tasks and supporting tools". In MedInfo 2001 Studies in Health Technology and Informatics, Haux R, Rogers R, Patel V (eds). IOS Press: Amsterdam, pp. 256-260, 2001.

[29] Nicolas F. Rouquette, Gary M. Wasserman, and Vanessa D. Carson. "OWL for Space Mission Systems development at JPL with semantic architecture styles," Proc. OWL: Experiences and Directions, Galway, Ireland, November 2005.

[30] Simple Bio Upper Ontology website http://www.cs.man.ac.uk/~rector/ontologies/simple-top-bio/

[31] R. Sinha, C. Paredis and P. K. Khosla. "Interaction Modeling in System Design," Proceedings of DETC '01, 2001 ASME Design Engineering Technical Conferences, September 2001, Pittsburgh, PA.

[32] A.W.M. Smeulders, Worring, M. , Santini, S. , Gupta, A. and Jain, R. "Content-based image retrieval at the end of the early years." *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12): 1349-1380, Dec 2000.

[33] Ashok N. Srivastava and Brett Zane-Ulman, "Discovering Recurring Anomalies in Text Reports Regarding Complex Space Systems," 2005 IEEE Aerospace Conference Proceedings, March 2005.

[34] Y. Sure, S. Staab and R. Studer, Methodology for development and employment of ontology based knowledge management applications. *ACM SIGMOD Record*, 31(4): 18 – 23, Dec 2002.

[35] D. Throop, "Reconciler: Matching Terse English Phrases," Proceedings of 2004 Virtual Iron Bird Workshop, NASA Ames Research Center, April, 2004.

[36] I. Tumer, Stone, R., and Bell, D., "Requirements for a Failure Mode Taxonomy for use in Conceptual Design", Proc. International Conf. on Engineering Design, ICED, Stockholm, # 1612, 2003.

[37] S. Uder, I. Y. Tumer and R. B. Stone, "Failure analysis in sub-system design for space missions," Proc. DETC '04, Salt Lake City, Utah, September 2004.

[38] UIMA website http://www.research.ibm.com/UIMA/

[39] V. Venkatasubramanian, J. Zhao and S. Viswanathan. "Intelligent Systems for HAZOP Analysis of Complex Process Plants," *Computers and Chemical Engineering*, 24: 2291-2302, 2000.

[40] VerbNet website http://seasrc.th.net/vnet/ (not the original site, but provides search access)

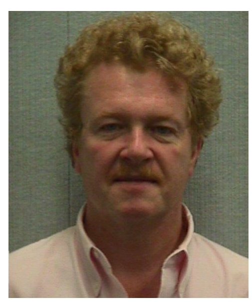
[41] Dominic Widdows, "Unsupervised methods for developing taxonomies by combining syntactic and statistical information," Proc. 2003 Conf. North American Chapter of Assoc. for Computational Linguistics on Human Language Technology, 1: 197-204. Edmonton, Canada, 2003.

[42] WonderWeb website http://wonderweb.semanticweb.org

[43] WordNet website http://wordnet.princeton.edu

BIOGRAPHY

***Jane T. Malin** is Senior Technical Assistant in the Intelligent Systems Branch, Automation, Robotics and Simulation Division, Engineering Directorate, NASA Johnson Space Center, where she has led intelligent systems research and development since 1984. She has led research on model-based hazard analysis and simulation for engineering intelligent systems. Her 1973 Ph.D. in Experimental Psychology is from the University of Michigan, with a dissertation on mathematical modeling of information processing in problem solving..*



***David R. Throop** has been an Artificial Intelligence Specialist with The Boeing Company since 1992. He provides engineering software support in the Automation, Robotics and Simulation Division in the Engineering Directorate at NASA Johnson Space Center. He oversaw development of FMEA modeling software and its use for the International Space Station. His 1979 Bachelors of Chemical Engineering is from Georgia Tech. His 1992 Ph.D. in Computer Science is from the University of Texas, with a dissertation on Model Based Diagnosis.*

APPENDIX

Table 1. Distinctions in the Upper Ontology of Entities (Things)

- Occurrence/Process (executed as a change in time-space)
  - Event/Achievement (unitary – e.g., arrival, failure)
  - Process/Performance (like stages – e.g., drilling, growing)
  - Situation/Event Sequence/Operation (distinct phases – e.g. accident)

- Enduring Object (present in time/space)
  - Three modular hierarchies; all can be applied to an object

  1. Physicality
     - 1.1. Physical
       - 1.1.1. Organic/Life Form
       - 1.1.2. Inanimate
       - 1.1.3. Energy
     - 1.2. Nonphysical
       - 1.2.1. Social (e.g., society, organization)
       - 1.2.2. Information/Pattern
  2. Design/Functionality
     - 2.1. Natural/Discovered
     - 2.2. Artifact/Device (designed to perform services or functions)
       - 2.2.1. Generic (unspecified function, e.g., hardware)
       - 2.2.2. System (has parts performing interacting functions)
       - 2.2.3. Implement/Equipment/Tool (specific types of function)
  3. Compositionality
     - 3.1. Unit/Self-Connected (but can have distinct parts and substances)
     - 3.2. Structure/Relationship (with defined interrelated parts)
     - 3.3. Aggregation/Complex
       - 3.3.1. Collection/Group (has members)
       - 3.3.2. Substance (homogeneous amount of matter)
         - 3.3.2.1. Pure substance: Element; Compound
         - 3.3.2.2. Mixture (by Main Phase, Design/Functionality, Physicality or Mixing Principle)

- Part/Feature (of object or occurrence)
  - Part
  - Region/Location/Place: Position; Boundary/Interface
  - Constituent/Member: Order; Description Variant

- Abstract Mathematics/Logic
  - Property Abstractions: Physical; Dimensional
  - Assertion/Fact: Theorem; Constraint
  - Set/class
  - Mathematical Structure: Relation; Field
  - Mathematical Operation

Table 2. Description and Concept Taxonomies

1. Description/Specification
    1.1. Scenario/Course (description of sequence or structure of events)
        1.1.1. Activity/Operation Specification (with goal/purpose, function, participants)
        1.1.2. Prediction (future scenario/course)
    1.2. Task Specification (element or step in Scenario/Course with goals, actions, participants)
    1.3. Activity Structure Specification
        1.3.1. Procedure (description of sequence or structure of tasks/activities)
        1.3.2. Plan (structure relating activities to goals and participants)
            1.3.2.1. Schedule (temporal plan)
            1.3.2.2. Policy (adopted social plan of action)
    1.4. Design Description
    1.5. Explanation
        1.5.1. Theory
        1.5.2. Rationale/Argument
    1.6. Assessment Description
    1.7. Certification
    1.8. Report
    1.9. Request
        1.9.1. Requirement
    1.10. Offering Description
    1.11. Agreement/Transaction: Promise/Obligation; Compromise, Settlement, Sale
2. Concept (element of description or specification that indicates the role of objects and occurrences)
    2.1. Circumstance (context, setting, assumptions)
        2.1.1. Fact
        2.1.2. Belief
        2.1.3. Source (reference, citation…)
        2.1.4. Decision
        2.1.5. Condition
    2.2. Need
        2.2.1. Goal/Purpose/Benefit
    2.3. Constraint/Restriction
        2.3.1. Acceptability Standard
        2.3.2. Synchronization Standard
        2.3.3. Cost/Exchange Limit
        2.3.4. Rule/Regulation (policy, rule, priority, guideline…)
    2.4. Function/Action
    2.5. Participant/Role
        2.5.1. Performer/Agent/Actor
        2.5.2. Instrument
        2.5.3. Resource
            2.5.3.1. Processing/Transporting: Power/Electrical; Thermal; Lubrication; Water/Hydraulic; Air/Pneumatic
            2.5.3.2. Control/Data
                2.5.3.2.1. Data/Signal
                2.5.3.2.2. Control/Monitor: Hardware; Software; Manual
            2.5.3.3. Store/Transport: Location/Place; Structure/Weight-bearing
            2.5.3.4. Skill Resource
        2.5.4. Patient/Operand
        2.5.5. Product: Effect/Product; SideEffect/ByProduct; Waste

Table 3. Refining Quality/Attribute Taxonomy

Property/Attribute

1. State
    - 1.1. Physical
        - 1.1.1. Material: Phase; Quality
        - 1.1.2. Organic (living, dead, embryonic, female…)
        - 1.1.3. Variety
        - 1.1.4. Concentration (damp, saturated…)
        - 1.1.5. Perceptual (red, smelly, bitter…)
        - 1.1.6. Behavioral/Operational (on, open, off..)
        - 1.1.7. Energy: Force; Pressure; Flow; Temperature; Radiation; Illumination; Electrical
        - 1.1.8. Mass
        - 1.1.9. Spatial: Extent, Shape, Coordinates
        - 1.1.10. Rate: Rate, Velocity, Acceleration
    - 1.2. Structural (complexity of n, degree of closure…)
    - 1.3. Aggregation
        - 1.3.11. Information (200 megabytes…)
        - 1.3.12. Frequency (n cycles per second…)
        - 1.3.13. Statistical: Descriptive (frequent, rare…); Predictive (likely, uncertain , certain…)
    - 1.4. Time Status
        - 1.4.14. Time Point of Occurrence
        - 1.4.15. Duration/Time span (of occurrence)
        - 1.4.16. Relative Time (of two occurrences, e.g., before, after, during, nonoverlapping…)
    - 1.5. Truth/Likelihood Status
        - 1.5.17. Truth: True; False
        - 1.5.18. Likelihood: Likely; Unlikely
2. Value/Relation Status (in relation to needs and expectations)
    - 2.1. Influence/Benefit: Contributing; Detracting; Controlling
    - 2.2. Expense/Cost: Expensive; Inexpensive
    - 2.3. Importance: Important, Unimportant
    - 2.4. Assurance/Acceptability value
        - 2.4.19. Goodness: Desirability; Pleasantness; Morality
        - 2.4.20. Correctness: Correct; Incorrect
        - 2.4.21. Quality/Perfection: Perfect; Imperfect
        - 2.4.22. Effectiveness: Capability; Dependability/Reliability; Variability; Efficiency; Difficulty
        - 2.4.23. Integrity: Whole; Damaged
        - 2.4.24. Safety: Safe; Unsafe
        - 2.4.25. Robustness: Robust; Not Robust
        - 2.4.26. Adequacy: Success; Sufficiency; Coordination; Agedness; Consistency/Discrepancy; Genuineness
    - 2.5. Informativeness: Knowledge; Objectivity; Openness/Availability; Clarity
    - 2.6. Authorization: Validation; Certification; Permissibility

Table 4. Function/Action Hierarchy: Process, Place, Serve, Energize

1. Process
    - 1.1. Convert: Change Phase; Change Hardness/Strength; Cook; Digest
    - 1.2. Change Amount
        - 1.2.1. Produce/Increase: Increase; Copy
        - 1.2.2. Reduce/Decrease: Decrease; Use Up
    - 1.3. Change Mixture:
        - 1.3.1. Interchange
        - 1.3.2. Combine/Contaminate: Combine; Contaminate
        - 1.3.3. Separate/Clean: Separate; Clean; Break Down
    - 1.4. Change Shape
        - 1.4.1. Shape: Compress; Expand; Crease
        - 1.4.2. Cut: Penetrate
        - 1.4.3. Change Surface: Smooth; Roughen
        - 1.4.4. Eat Away
        - 1.4.5. Break Open: Split; Rupture
    - 1.5. Destroy/Injure: Shatter; Obliterate; Burn; Irritate; Deprive
2. Place (Self or Other)
    - 2.1. Hold: Store; Carry/Channel; Support/Stabilize; GiveWay/Destabilize; Secure; Drop
    - 2.2. Shift/Distribute
        - 2.2.1. Send/Emit: Provision; Export; Remove; Excrete; Release
        - 2.2.2. Transfer
        - 2.2.3. Insert
        - 2.2.4. Receive/import/collect: Collect; Take; Accept; Absorb; Ingest
        - 2.2.5. Shift in Place
        - 2.2.6. Move (self): Come; Go; Move Smoothly; Change Altitude; Move in Place
    - 2.3. Arrange/Put
        - 2.3.1. Change Exposure
            - 2.3.1.1. Isolate/Close: Isolate; Close; Cover
            - 2.3.1.2. Expose/Open: Expose; Open; Uncover
        - 2.3.2. Change Position: Position, Displace; Deflect
        - 2.3.3. Change Assembly: Assemble; Disassemble; Disorder
        - 2.3.4. Change Connectedness
            - 2.3.4.1. Connect: Fasten, Bond
            - 2.3.4.2. Disconnect
            - 2.3.4.3. Collide
            - 2.3.4.4. Deflect
        - 2.3.5. Change Mobility: Immobilize; Free
3. Serve/Support
    - 3.1. Change Service Availability: Provide; Withhold
    - 3.2. Use: Demand; Forego; Waste
4. Change Energy
    - 4.1. Change Force/Pressure: Increase Force; Reduce Force
    - 4.2. Change Heat: Increase Heat; Reduce Heat
    - 4.3. Change Electrical Power: Increase Power; Reduce Power
    - 4.4. Change Illumination: Increase Radiation; Reduce Radiation

Table 5. Function/Action Hierarchy: Control/Perform/Manage

5. Control/Perform/Manage
   - 5.1. Control/Maintain (or not)
     - 5.1.1. Achieve: Maximize; Cause; Succeed
     - 5.1.2. Counteract/Safeguard/Mitigate
       - 5.1.2.1. Maintain: Preserve; Ensure
       - 5.1.2.2. Prevent: Avoid; Withstand; Guard
       - 5.1.2.3. Rectify: Minimize; Safe; Restore; Accommodate; Undo; Rework; Reset
     - 5.1.3. Fail: Fail; Aggravate; Allow
   - 5.2. Inform/Decide
     - 5.2.1. Communicate
       - 5.2.1.1. Indicate: Indicate; Record; Inform; Speak
       - 5.2.1.2. Author: Write; Revise
         - 5.2.1.2.1. Conceal
       - 5.2.1.3. Interact: Negotiate; Criticize
       - 5.2.1.4. Interfere with Communication: Conceal; Intercept
     - 5.2.2. Process
       - 5.2.2.1. Monitor
       - 5.2.2.2. Ignore/Mistake
       - 5.2.2.3. Classify: Classify; Measure; Convert
       - 5.2.2.4. Evaluate: Check; Test; Validate
       - 5.2.2.5. Determine/Construct Response: Investigate; Analyze; Decide; Construct
   - 5.3. Direct
     - 5.3.1. Manage
     - 5.3.2. Relinquish Control: Automate; Assign; Surrender Control
     - 5.3.3. Regulate/Guide/Modulate: Regulate; Limit; Guide; Adjust
     - 5.3.4. Command
       - 5.3.4.1. Select (mode)
       - 5.3.4.2. Control Enablement: Enable, Disable
       - 5.3.4.3. Control Start/Stop: Turn On; Turn Off
   - 5.4. Perform/Execute: Respond; Start; Stop; Continue; Repeat; Complete
   - 5.5. Coordinate
     - 5.5.1. Assist/Correlate: Assist; Correlate
     - 5.5.2. Interfere/Conflict: Interfere; Conflict
     - 5.5.3. Manage Certification: Certify; Change Certification
     - 5.5.4. Track Descriptions
     - 5.5.5. Respond to Regulations: Comply; Violate
     - 5.5.6. Administer Consequences: Reward; Punish; Forgive

Table 6. Entity Hierarchy Categories

1. Enduring Object
    1.1. Nonphysical Object
        1.1.1. Social Object
            1.1.1.1. Social Natural Unit (person, individual...)
            1.1.1.2. Social Artifact Structure (crew, team…)
                1.1.1.2.1. Description/Specification [see Table 2]: DOMAINS: Design Description; Engineering Procedure; Problem Report; Engineering Requirement; Engineering Agent; Production Agent; Operations Agent
            1.1.1.3. Social Artifact Aggregation-Collection (nation, audience…)
        1.1.2. Information/Signal Object
            1.1.2.1. Information Unit (signal, packet, pixel, measurement…)
                1.1.2.1.1. Identifier (name, part number…)
                1.1.2.1.2. Limit/Target (redline…)
                1.1.2.1.3. Message (communication, command, flag….)
                    1.1.2.1.3.1. DOMAIN: Problem Message (alarm, error message…)
            1.1.2.2. Information Aggregation-Collection (sample, folder…)
            1.1.2.3. Information Structure (list, pattern, form…)
                1.1.2.3.1. Description: DOMAINS: Problem Report; Invoice
                1.1.2.3.2. Structured Information Artifact: Document; Picture
                1.1.2.3.3. DOMAIN: Software structure (program, architecture…)
            1.1.2.4. Information Connection
    1.2. Physical Object
        1.2.1. Physical Unit (object, item…)
            1.2.1.1. Organic/Life-form (human, microbe…)
            1.2.1.2. Inanimate Natural (rock, debris…)
            1.2.1.3. Artifact/Device
                1.2.1.3.1. System Unit (assembly, kit, workstation…)
                1.2.1.3.2. Equipment/Implement/Tool [see Tables 4 and 5]
                    1.2.1.3.2.1. Processor: Converter/Amount Changer; Separator/Cleaner: Combiner/Shaper
                    1.2.1.3.2.2. Placer: Holder; Shifter; Arranger
                        1.2.1.3.2.2.1. DOMAINS: Information and Power Carriers; Protective Equipment
                    1.2.1.3.2.3. Entertainment Equipment
                    1.2.1.3.2.4. Energy Equipment
                        1.2.1.3.2.4.1. DOMAINS: Pyrotechnic; Lighting; Communication; Thermal; Electrical/Power; Pressure
                    1.2.1.3.2.5. Controller/Instrumentation
                        1.2.1.3.2.5.1. Instrumentation
                            1.2.1.3.2.5.1.1. Measuring/Monitoring Equipment: Physical Measure; Sensor
                                1.2.1.3.2.5.1.1.1. DOMAINS: Electrical; Pressure; Radiation
                            1.2.1.3.2.5.1.2. Recorder: DOMAIN: Information Storer
                            1.2.1.3.2.5.1.3. Controller: DOMAIN: Information Processing Equipment
                            1.2.1.3.2.5.1.4. Actuator
                            1.2.1.3.2.5.1.5. Indicator: DOMAIN: Error Equipment (error lamp…)
                        1.2.1.3.2.5.2. Test Equipment
                            1.2.1.3.2.5.2.1. DOMAINS: Electrical Test Equipment; Self-Test Equipment; Calibration Equipment
        1.2.2. Physical Aggregation [see Table 1] DOMAINS: Food Substance; Lubricant; Propellant
        1.2.3. Physical Structure (of parts with defined interrelationships – e.g., network)
        1.2.4. Physical Part/Feature: DOMAINS: Data Interface; Body Part; Electrical/Power Part
        1.2.5. Energy
2. Occurrence/Process [see Table 1]:
    2.1. Event/Achievement
    2.2. Process
    2.3. Situation/Event Sequence: Situation; Operation; DOMAINS: Engineering Activity; Launch Site Operations; Flight Operations; Landing Site Operations; Surface Operations

Table 7. Problem Hierarchy: Objects and Functional Impairment

1. Integrity Problem (faulty, defective, injured…)
    - 1.1. Damaged (change object integrity)
        - 1.1.1. Distorted
            - 1.1.1.1. Deformed
            - 1.1.1.2. Buckled: Bent; Creased
            - 1.1.1.3. Uneven: Extrusions; Gaps; Eroded; Marked
        - 1.1.2. Broken: Split; Burst; Perforated
        - 1.1.3. Disarranged
            - 1.1.3.1. Disconnected: Detached; Divided; Failed Join; DOMAIN: Electrically Disconnected
            - 1.1.3.2. Connected: Contacting; DOMAIN: Electrically Connected
            - 1.1.3.3. Displaced
            - 1.1.3.4. Disordered
        - 1.1.4. Destroyed: Shattered; Obliterated; Killed; Lost
    - 1.2. Converted (change substance phase, mixture, appearance)
        - 1.2.1. Separated; Broken Down; Combined; Eaten away; Burned;
            - 1.2.1.1. Contaminated: Fouled; Stained; With Debris
        - 1.2.2. Interchanged/Migrated
        - 1.2.3. Changed phase
    - 1.3. Uncomfortable
    - 1.4. Starved or Overwhelmed: Depleted; Overwhelmed
2. Acceptability Problem [see negative properties in Table 3]
3. Hazard/Safety Problem
    - 3.1. Burden/Shock (too much or too little): Thermal; Moisture; Pressure; Radiation
        - 3.1.1. DOMAIN: Electrical Burden: Voltage Excess; Electrified
        - 3.1.2. DOMAIN: Mechanical Burden: Stress/Load; Friction; Abrasion; Vibration; Sharpness; Acceleration
    - 3.2. Obscuring (perception or signal)
        - 3.2.1. DOMAINS: Electrical Noise; Obscuring Atmosphere
    - 3.3. Irritating (to human): Loud; Smelly
    - 3.4. Material Hazard
        - 3.4.1. Toxic/Infectious
        - 3.4.2. Caustic
        - 3.4.3. Fire Hazard
        - 3.4.4. Reactive
4. Functional Impairment
    - 4.1. Ineffective
        - 4.1.1. Incapable: Inoperative; Unprepared; Powerless
        - 4.1.2. Inefficient
        - 4.1.3. Undependable: Unreliable; Underreactive
    - 4.2. Mechanically Impaired: Weak; Fragile, Leaky; Blocked; Stuck; Loose
    - 4.3. Impaired Controllability
        - 4.3.1. Unperceptive: Disoriented; Not Processing; Misinterpreting
        - 4.3.2. Uncommunicative: Concealing; Intercepting
        - 4.3.3. Uncontrollable: Unchangeable; Unmodifiable; Disturbed; Overreactive
    - 4.4. Incompatible: Not Fitting; Out of Adjustment
    - 4.5. Incorrectly Supplied: Substitute; Excessive; Insufficient; Missing

Table 8. Problem Hierarchy: Performance Deviations

1. Performance Deviation/Error
    - 1.1. Process Deviation/Error
        - 1.1.1. Engineering Error: Requirements Error; Design Error; Development Error; Assembly Error
        - 1.1.2. Procedure Error
        - 1.1.3. Disorganized Situation
    - 1.2. Agent Deviation/Error
        - 1.2.1. Human Error (by role)
        - 1.2.2. Hardware Error (by equipment)
        - 1.2.3. DOMAIN: Software/Computer Error: Corruption; Overflow; Race; Faulty Component; Data Error
    - 1.3. Functional Deviation/Error
        - 1.3.1. Activation-Control Problem
            - 1.3.1.1. Commission: NotAllowed Function; Slip; Unsynchronized
            - 1.3.1.2. Omission: Missed; Did Not Control/Maintain [see Table 5]
            - 1.3.1.3. Not Responding
        - 1.3.2. Deviating Function
            - 1.3.2.1. Function Too Demanding: Difficult Performance; Wasteful Performance
            - 1.3.2.2. Function Performed Wrongly
                - 1.3.2.2.1. Coordination Deviation
                    - 1.3.2.2.1.1. Unsynchronized
                    - 1.3.2.2.1.2. Wrong Duration: Early Completion; Late Completion
                    - 1.3.2.2.1.3. Activation/Deactivation Delay
                        - 1.3.2.2.1.3.1. Early Reaction: Early Start; Early Stop
                        - 1.3.2.2.1.3.2. Late Reaction: Late Start; Late Stop
                        - 1.3.2.2.1.3.3. Reaction Failure: Failed Start; Failed Stop
                    - 1.3.2.2.1.4. Rate Deviation: Rate Variation; Slow Rate; Fast Rate; Too often; Not Often Enough
                - 1.3.2.2.2. Execution Quality Deviation
                    - 1.3.2.2.2.1. Unsuccessful
                    - 1.3.2.2.2.2. Incomplete
                    - 1.3.2.2.2.3. Overdone
                    - 1.3.2.2.2.4. Inconsistent
                    - 1.3.2.2.2.5. Reversed/Opposite
                    - 1.3.2.2.2.6. Incorrect Operand
                    - 1.3.2.2.2.7. Incorrect Direction/Destination
    - 1.4. Incorrect Input, Command, Output, Effect or Product
        - 1.4.1. Quantity Deviation
            - 1.4.1.1. Incorrect Quantity: Too Much; Too Little; Missing
        - 1.4.2. Quality Deviation
            - 1.4.2.1. NotAllowed Value/Relation
            - 1.4.2.2. Unclear: Ambiguous; Noisy
            - 1.4.2.3. Erratic
            - 1.4.2.4. Mismatched: MisAligned
            - 1.4.2.5. Out Of Limits: Reversed; Shifted; Value Above Limit; Value Below Limit