STPA (System-Theoretic Process Analysis) Compliance with MIL-STD-882E and other Army Safety Standards

Prof. Nancy Leveson
Aeronautics and Astronautics, MIT

STAMP and STPA were designed for software-intensive systems and the increasing complexity, coupling, and challenges that engineers must deal with today. This section of the report details how STAMP and STPA comply with or differ from the primary safety standards currently used by the Army.

7. MIL-STD-882E Compliance

STAMP is an embodiment of the principles and approach called System Safety that was created in the defense industry about 60 years ago to cope with the increasingly complex defense systems that were being developed. STAMP was purposely created to provide a means for implementing the principles underlying System Safety so it should not be a surprise that it and the tools built on STAMP comply with and support the tasks in MIL-STD-882. In this section, the compliance with and support for each of the tasks in MIL-STD-882E is discussed.

TASK SECTION 100 – PROGRAM MANAGEMENT AND CONTROL

STPA has a structured process for identifying hazards (as defined in MIL-STD-882E) that supports

TASK 101: Hazard Identification and Mitigation Effort using the System Safety Methodology
101.2.1 Establish and execute a hazard identification and mitigation effort within SE
101.2.8 As a minimum, report the following:
a. Hazards and associated risks.
b. Functions, items, and materials associated with hazards.
c. Recommended requirements for operation, maintenance, sustainment, and disposal.
d. Recommended mitigation measures.

Task 101. Note that the MIL-STD-882 definition (and the STPA hazards) include more than the "failures" that are the focus of some widely used system safety tools. STPA also identifies the functions associated with hazards and the system and component requirements for eliminating or mitigating hazards throughout the system life cycle. The causal scenarios that lead to hazards generated by STPA can be used to create requirements for safe and cyber secure design and operations.

TASK 102 - SYSTEM SAFETY PROGRAM PLAN
h. Describe the efforts to identify and control hazards associated with materials used during the system's life-cycle.
i. Describe a systematic software system safety approach to: *(1) Identify and describe the software contributions to system hazards.* *(2) Identify safety-significant (safety-critical and safety-related) software functions and software requirements.* *(3) Identify the safety requirements associated with safety-significant software components and safety-related items.* *(4) Identify and assign the Software Criticality Index (SwCI) for each safety-significant software function (SSSF) and its associated requirements.*

STAMP and STPA can support the hazard analysis and other goals in a System Safety Plan. With respect to software, STPA provides a systematic software safety approach to achieving the goals in Task 102 "i" above. An example is provided in Section 9.1.

Task 102 i.4 requires that software criticality be classified with respect to the specific function that it is implementing and how that function relates to the system hazards. That is exactly what STPA does. In systems today, software implements functions that previously were implemented by humans or hardware and can (and should) be treated in the same way. We do not assign a SwCI because in STAMP software can and should be treated in the same way as hardware, i.e., the hazards are identified along with causal scenarios leading to the hazards. Then engineers can eliminate or mitigate those causes according to standard system safety practice and design precedence, whether in software or hardware or both. STPA provides the information (not provided by most traditional hazard analysis methods) to do this. The criticality of software functions is related to the hazards that the software can cause or contribute to and the criticality (level) of those hazards. This is in turn related to the safety-related requirements. It is not related to the rigor of the software development process for non-safety-related requirements.

The limitations of the use of Design Assurance Level (DAL) as used in SAE ARP 4754/4761 or level of rigor for software is discussed in Sections 11.2 and 11.4. Briefly, the design assurance level does little to help with the overwhelmingly relevant problem in software safety, which is requirements flaws (e.g., missing cases, incorrect assumptions about the environment in which the software will be used, etc.). One can rigorously and perfectly implement unsafe requirements.

TASK 103 - HAZARD MANAGEMENT PLAN
TASK 104 - SUPPORT OF GOVERNMENT REVIEWS/AUDITS
104.2.2 Provide technical support to mishap investigations.

TASK 105 -INTEGRATED PRODUCT TEAM/WORKING GROUP SUPPORT
TASK 106 - HAZARD TRACKING SYSTEM
TASK 107 - HAZARD MANAGEMENT PROGRESS REPORT
TASK 108 - HAZARDOUS MATERIALS MANAGEMENT PLAN

The MIL-STD-882 tasks above are all related to program management and thus only indirectly related to STAMP and STPA. STPA is a hazard analysis technique, not a management process.

Task 104.2.2 mentions support of mishap investigation. One of the tools based on STAMP (which is not part of this demonstration) is a tool for performing mishap analysis called CAST (Causal Analysis using Systems Theory). CAST is being used on complex system mishaps in all fields, including U.S. military friendly fire mishaps [Johnson 2016, Leveson 2012].

TASK SECTION 200 – ANALYSIS

TASK 201 – PRELIMINARY HAZARD LIST
201.1 Purpose. Task 201 is to compile a list of potential hazards early in development. *201.2 Task description. The contractor shall:* *201.2.1 Examine the system shortly after the materiel solution analysis begins and compile a Preliminary Hazard List (PHL) identifying potential hazards inherent in the concept.* *a. A brief description of the hazard.* *b. The causal factor(s) for each identified hazard.*

As stated for Task 101, STPA has a structured process for identifying hazards (as defined in MIL-STD-882E) that supports the task. It identifies the causal factors for identified hazards, as required in the task (201.2.1.b). Note that the MIL-STD-882 definition and the STPA identified hazards include more than the "failures" that are the focus of some widely used risk analysis tools.

TASK 202 - PRELIMINARY HAZARD ANALYSIS
202.1 Purpose. Task 202 is to perform and document a Preliminary Hazard Analysis (PHA) to identify hazards, assess the initial risks, and identify potential mitigation measures.
202.2 Task description. The contractor shall perform and document a PHA to determine initial risk assessments of identified hazards. Hazards associated with the proposed design or function shall be evaluated for severity and probability based on the best available data, including mishap data (as accessible) from similar systems, legacy systems, and other lessons learned. Provisions, alternatives, and mitigation measures to eliminate hazards or reduce associated risk shall be included.

STPA can be (and is being) used for PHA. It includes all the factors mentioned in Task 202. Many of these factors are not handled well by traditional hazard analysis methods such as humans, software, interfaces and interactions among components, modes, operating environments and

constraints, etc. STPA provides the information needed to identify measures to eliminate or mitigate the hazards at a time when doing so is often very low cost.

One advantage of STPA is that there is a structured process (some of which can be automated) for identifying hazards based on a safety control structure model rather than depending on past experience and brain storming and history as suggested in the Task 202 description. Comparative evaluations on real systems have shown that STPA produces a much more comprehensive list of hazards.[1]

Beyond simply identifying hazards, STPA identifies their causal scenarios. Being able to start a causal analysis early in development before the design decisions are made allows a much better chance of eliminating hazards (according to the system safety design precedence) rather than having to deal with them the best that one can when potential causes are uncovered later after detailed design decisions have been made and are difficult or impossible to undo. The rework involved in undoing design decisions is, of course, very expensive. Engineers are using STPA as part of their early design efforts to help them make decisions that will increase safety and cyber-security before it becomes infeasible to reverse some of the basic design decisions.

STPA does not include specifying probability as that is unknown and unknowable until the detailed system design is created (and sometimes not even then). Many PHAs identify primarily hardware component failures as hazards, where the probability of failure can be adequately estimated. However, mishaps in complex systems today are not caused solely by hardware failures. For complex systems with the potential for design errors and for human and software errors, there is no basis for making probability estimates and, in retrospect, are often wrong and misleading. Humans and software do not behave or make mistakes randomly.

Software is just logic and therefore is deterministic. Attempts were made a long time ago to identify probabilities of “failure” for software, but those attempts have been discredited. SAE ARP 4761 concedes that such failure probabilities cannot be derived for software.

The argument against using probabilities for human behavior (called Human Reliability Analysis or HRA) is more complicated. Human errors can be divided into slips and mistakes. *Slips* involve unintended actions. For example, the pilot means to push button A but instead pushes button B. There is some randomness to this type of error, but it is also often related to the design of the controls, where, perhaps, buttons A and B are similar and located close together. Pilot errors related to accidentally shutting down the engines inadvertently have been reduced by making the controls very different and putting covers over ones that are hazardous (such as shutting down the engines in flight). In general, human behavior is impacted by the design of the system in which it occurs. We think we are randomly making errors and that we have complete control over our behavior, but we are often unknowingly affected by the design of the system we are operating. If one knew that there was a potential for a slip because of an interface design flaw, the proper response would be to change the interface design, not to assign a probability to the behavior without fixing the interface.

---

[1] The papers provided to support this report provide details for comparisons done previously for several defense systems.

Mistakes, in contrast, involve intentional human behavior. The person means to push button B and pushes it when the correct action was to push button A. This type of human error involves intentional (not unintentional) action and arises from misunderstandings about such things as how the system works, the current state of the system, or the effects of different types of actions. This type of error is not random. Human reliability analysis was first applied to workers doing repetitive actions to make widgets on assembly lines. The cognitively complex behavior required by operators on complex systems today cannot be treated in the same way.

STPA could incorporate probabilistic information for hardware and other simple failures, but there is no way to determine the probability of such hazards as "Software launches the missile prematurely." The same is true if the missile is being launched by a human operator. When cyber-security threats are considered as causes for such events, the problems are multiplied. Hazards of this type are sometimes omitted from the risk assessment or PHA because a probability cannot be derived.

Comparative studies for military systems [Abrecht 2016a and Abrecht et. al. 2016b] have been done between STPA and the results of PHA documented in the actual Safety Assessment Reports (SAR) produced for the system. These studies show many instances where hazards were identified and classified as marginal or unlikely in the PHA but the STPA analysis found perfectly reasonable scenarios leading to catastrophic results for the same hazards that had been dismissed as incredible in the PHA. In the analyses cited, these catastrophic scenarios could have been eliminated, usually with no or little cost. The same results have been found in comparative evaluations for nonmilitary systems. At the least, information about potential causal scenarios should be factored into such probability assessments.

Decades of psychological research have shown that humans are very bad at estimating probabilities and risk. The problem is called *heuristic bias*. Most often, risks are underestimated rather than overestimated.[2]

The bottom line is that STPA will comprehensively identify hazards and provide information about the causal scenarios leading to these hazards so that recommendations for design and operations to eliminate or mitigate them (according to the system safety precedence) can be generated. If users feel they are able to associate probabilities with the STPA-identified hazards and causal scenarios, they are free to do so.

TASK 203 - System Requirements Hazard Analysis
203.1 Purpose. Task 203 is to perform and document a System Requirements Hazard Analysis (SRHA) to determine the design requirements to eliminate hazards or reduce the associated risks for a system, to incorporate these requirements into the appropriate system documentation, and to assess compliance of the system with these requirements. The SRHA addresses all life-cycle phases and modes.

[2] Leveson has found after decades of reading mishap reports that nearly all had, during development, assigned a very low probability of likelihood to the factors that led to the actual loss. As a result, no action was taken to prevent those factors even in cases where the cost would have been minimal.

203.2 Task description. The contractor shall perform and document an SRHA to:

203.2.1 Determine system design requirements to eliminate hazards or reduce the associated risks by identifying applicable policies, regulations, standards, etc. and analyzing identified hazards.

a. *The contractor shall identify applicable requirements by reviewing military and industry standards and specifications; historical documentation on similar and legacy systems; Department of Defense (DoD) requirements (to include risk mitigation technology requirements); system performance specifications; other system design requirements and documents; applicable Federal, military, State, and local regulations; and applicable Executive Orders (EOs) and international agreements.*
b. *The contractor shall recommend appropriate system design requirements to eliminate hazards or reduce the associated risks identified in accordance with Section 4 of this Standard.*
c. *The contractor shall define verification and validation approaches for each design requirement to eliminate hazards or reduce associated risk.*

203.2.2 Incorporate approved design requirements into the engineering design documents, and hardware, software, and system test plans, as appropriate. As the design evolves, ensure applicable design requirements flow down into the system and subsystem specifications, preliminary hardware configuration item development specifications, software requirements specifications, interface requirements specifications, and equivalent documents. As appropriate, use engineering change proposals to incorporate applicable design requirements into these documents.

STPA supports this task by generating the system and component safety requirements, identifying their causes, and using the causal scenarios to produce detailed design and operational requirements.

Once STPA has identified the causal scenarios for the unsafe control actions (detailed hazards), determining how to verify and validate them is straightforward if the hazards cannot be completely eliminated. This verification includes generation of scenarios for human testing in simulators. In addition, the unsafe control actions (detailed hazards) can be used in a model-based development system (such as SpecTRM-RL) to generate models [Thomas, 2013] and from these formal models software test data can be generated to ensure testing coverage of hazardous software behavior.

Flow down should be enhanced by the connection of the requirements to the safety control structure.

TASK 204 - SUBSYSTEM HAZARD ANALYSIS

204.1 Purpose. Task 204 is to perform and document a Subsystem Hazard Analysis (SSHA) to verify subsystem compliance with requirements to eliminate hazards or reduce the associated

risks; to identify previously unidentified hazards associated with the design of subsystems; and, to recommend actions necessary to eliminate identified hazards or mitigate their associated risks.
204.2 Task description. The contractor shall perform and document an SSHA to identify hazards and mitigation measures in components and equipment. This analysis shall include Commercial-Off-the-Shelf (COTS), Government-Off-the-Shelf (GOTS), Government-Furnished Equipment (GFE), Non-Developmental Items (NDI), and software. Areas to consider include performance, performance degradation, functional failures, timing errors, design errors or defects, and inadvertent functioning. While conducting this analysis, the human shall be considered a component within a subsystem, receiving both inputs and initiating outputs.
204.2.1 At a minimum, the analysis shall: a. Verify subsystem compliance with requirements to eliminate hazards or reduce the associated risks. *b. Identify previously unidentified hazards associated with the design of subsystems.* *(1) Ensure implementation of subsystem design requirements and mitigation measures have not introduced any new hazards.* *(2) Determine modes of failure, including component failure modes and human errors, single point and common mode failures, the effects when failures occur in subsystem components, and from functional relationships between components and equipment comprising each subsystem. Consider the potential contribution of subsystem hardware and software events (including those developed by other contractors/sources, COTS, GOTS, NDIs, and GFE hardware or software), faults, and occurrences (such as improper timing).* *c. Recommend actions necessary to eliminate previously unidentified hazards or mitigate their associated risk. Ensure system-level hazards attributed to the subsystem are analyzed and adequate mitigations of the potential hazards are implemented in the design.*
204.2.4 The contractor shall update, as necessary, the SSHA following system design changes, including software design changes.

STPA is a top-down hazard analysis tool that can include all the subsystems. In STAMP and STPA, the subsystem hazard analysis is part of the general system analysis. STPA identifies the hazards and the causal scenarios, which may involve subsystem behavior, leading to the system hazards. The information in the causal scenarios (as with any hazard analysis technique) can be used to assist the designers in identifying design mitigations. Unlike many of the traditional hazard analysis techniques, STPA considers more than just failures but also design errors, timing errors, inadvertent functioning or functioning under the wrong conditions, etc. It also considers the human as a component of the system, as required in the 204.2 Task Description.

Updating the STPA analysis does not require extensive reanalysis. The parts of the safety control structure affected are easily identified and the reanalysis can be confined only to the parts

that were actually changed. An MIT master's thesis showed how this can be done and the associated reduction in effort for a real automotive system [Sgueglia 2015].[3]

TASK 205 - SYSTEM HAZARD ANALYSIS
205.1 Purpose. Task 205 is to perform and document a System Hazard Analysis (SHA) to verify system compliance with requirements to eliminate hazards or reduce the associated risks; to identify previously unidentified hazards associated with the subsystem interfaces and faults; identify hazards associated with the integrated system design, including software and subsystem interfaces; and to recommend actions necessary to eliminate identified hazards or mitigate their associated risks.
205.2 Task description. The contractor shall perform and document an SHA to identify hazards and mitigation measures in the integrated system design, including software and subsystem and human interfaces. This analysis shall include interfaces associated with Commercial-Off-the-Shelf (COTS), Government-Off-the-Shelf (GOTS), Government-Furnished Equipment (GFE), Non-Developmental Items (NDI), and software. Areas to consider include performance, performance degradation, functional failures, timing errors, design errors or defects, and inadvertent functioning. While conducting this analysis, the human shall be considered a component within the system, receiving both inputs and initiating outputs.
205.2.1 This analysis shall include a review of subsystems interrelationships for: ...[4]
205.2.4 The contractor shall evaluate system design changes, including software design changes, and update SHA as necessary.

Once again, this task is exactly what STPA was designed to support. It can be applied at the system or subsystem level. It has been successfully used for extremely complex systems (e.g., the U.S. BDMS) at the system level and even the "system of systems" level. The interfaces and their impact on hazards is a particular focus, as is software and humans (who are considered the same as any other system component). And, once again, changes are localized in the modeling and analysis so it is less difficult to update the analysis after changes are made.

TASK 206 - OPERATING AND SUPPORT HAZARD ANALYSIS
206.1 Purpose. Task 206 is to perform and document an Operating and Support Hazard Analysis (O&SHA) to identify and assess hazards introduced by operational and support activities and procedures; and to evaluate the adequacy of operational and support procedures, facilities, processes, and equipment used to mitigate risks associated with identified hazards.

[3] STPA is widely used in the automotive industry because of the quickly increasing autonomy in automobiles. Cars today have been estimated to have 100 million lines of software, which is many times more than the newest military aircraft.

[4] Details omitted for space reasons.

206.2 Task description. The contractor shall perform and document an O&SHA that typically begins during Engineering and Manufacturing Development (EMD) and builds on system design hazard analyses. The O&SHA shall identify the requirements (or alternatives) needed to eliminate hazards or mitigate the associated risks for hazards that could not be eliminated. The human shall be considered an element of the total system, receiving both inputs and initiating outputs within the analysis.

The STAMP safety control structure can include the operational control structure and an OSHA analysis is possible using STPA. In fact, it is currently being used to evaluate safety of operational flight test at Edwards Air Force Base [Montes 2016]. Planning for operational safety should start in the concept development stage so that operational safety can be built into the design of the system. STPA allows this early start.

TASK 207 – HEALTH HAZARD ASSESSMENT

207.1 Purpose. Task 207 is to perform and document a Health Hazard Analysis (HHA) to identify human health hazards, to evaluate proposed hazardous materials and processes using such materials, and to propose measures to eliminate the hazards or reduce the associated risks when the hazards cannot be eliminated.

STPA could potentially help with analyzing hazardous material processes**.**

TASK 208 - FUNCTIONAL HAZARD ANALYSIS

208.1 Purpose. Task 208 is to perform and document a Functional Hazard Analysis (FHA) of an individual system or subsystem(s). The FHA is primarily used to identify and classify the system functions and the safety consequences of functional failure or malfunction, i.e. hazards. These consequences will be classified in terms of severity for the purpose of identifying the safety-critical functions (SCFs), safety-critical item (SCIs), safety-related functions (SRFs), and safety-related items (SRIs) of the system. SCFs, SCIs, SRFs, and SRIs will be allocated or mapped to the system design architecture in terms of hardware, software, and human interfaces to the system. The FHA is also used to identify environmental and health related consequences of functional failure or malfunction. The initial FHA should be accomplished as early as possible in the Systems Engineering (SE) process to enable the engineer to quickly account for the physical and functional elements of the system for hazard analysis purposes; identify and document SCFs, SCIs, SRFs, and SRIs; allocate and partition SCFs and SRFs in the software design architecture; and identify requirements and constraints to the design team.

Once again, this task is exactly what STPA is meant to support and be used for. The STAMP safety control structure documents all this information. STPA analyzes this structure. The STPA results are broader, however, than specified in this task: Hazards are not just the result of functional failure or malfunction.

STPA identifies the system-level hazards associated with each function (and unsafe control action) so the classification as to severity comes from the classification of the system level hazards and their associated mishaps (losses).

TASK 209 - SYSTEM-OF-SYSTEMS HAZARD ANALYSIS
209.1 Purpose. Task 209 is to perform and document an analysis of the System-of-Systems (SoS) to identify unique SoS hazards. This task will produce special requirements to eliminate or mitigate identified unique SoS hazards which otherwise would not exist.
209.2 Task description. The contractor shall perform and document an analysis of the SoS to identify unique SoS hazards and mitigation requirements. The human shall be considered an element of the SoS, receiving both inputs and initiating outputs within the analysis.

STPA integrates system hazard analysis, subsystem hazard analysis, functional hazard analysis, and systems-of-systems hazard analysis into one process and produces the information needed for all of these tasks.[5] These tasks all ask for the same information but at different system levels or viewpoints. A top-down system hazard analysis technique like STPA, which is based on the STAMP causality model, includes all these views of the system by tracing down from the system to the subsystem levels at varying levels of abstraction.

TASK 210 - ENVIRONMENTAL HAZARD ANALYSIS
210.1 Purpose. Task 210 is to perform and document an Environmental Hazard Analysis (EHA) to support design development decisions. The EHA will identify hazards to the environment throughout all life-cycle phases and modes; document the hazards in the Hazard Tracking System (HTS); manage the hazards using the system safety process described in Section 4; and provide the system-specific data to support National Environmental Policy Act (NEPA) and Executive Order (EO) 12114 requirements.
210.2 Task description. The contractor shall perform and document an EHA in order to influence design decisions by integrating environmental considerations into the Systems Engineering (SE) process. The contractor should start the EHA process as early as possible consistent with initiation of the overall SE process. The contractor will continue to identify and manage environmental hazards using the system safety process described in Section 4 throughout the duration of the task.
210.2.1 Starting the EHA as part of the early SE processes is typically the most cost-effective means of minimizing environmental impacts from the operations and support of a new or modified system. Conversely, early design decisions made without consideration of environmental requirements may result in environmental impacts that cannot be easily

[3] Functional Hazard Analysis and Systems-of-Systems Hazard Analysis were added to MIL-STD-882 in 2012 in version E and were not included in any of the versions from 1969 to 2012. It is not clear why they were added because the objectives appear to be included in the previous tasks, which are still in the standard.

designed out and will require mitigation later in the acquisition process. These issues could potentially result in mission and operational constraints and compliance burdens for receiving installations, test, launch, and training ranges, depot maintenance installations, and operational training units.
a. The elimination of hazards or reduction of associated risks with an informed and structured risk assessment and acceptance process is essential. Early identification and introduction of environmental hazards into the SE process provides decision makers with a more complete and relevant picture of the potential risks during all life-cycle phases and modes, and will help mitigate the risk.

STAMP and STPA do not distinguish between different types of hazards—all are handled, including hazards to the environment. 210.2.1 states the argument for needing a hazard analysis technique that allows integrating hazard analysis into system engineering from the concept analysis stage. We did not include environmental hazards in this demonstration STPA analysis for FVL because we did not know what environmental concerns are important in the FVL development, but we easily could have done so. It requires no changes to our process or techniques and has been included in petrochemical system STAMP-based analyses where environmental hazards are paramount.

Task 210.2.1 contains exactly our argument for needing a hazard analysis process like STPA that can be started during concept analysis and used to guide system design.

TASK SECTION 300 – EVALUATION

TASK 301 - SAFETY ASSESSMENT REPORT
301.1 Purpose. Task 301 is to perform and document a Safety Assessment Report (SAR) to provide a comprehensive evaluation of the status of safety hazards and their associated risks prior to test or operation of a system, before the next contract phase, or at contract completion.
301.2 Task description. The contractor shall perform and document an assessment to identify the status, at the time of the report, of safety hazards, associated risks, mitigation measures, and formal risk acceptance decisions. This documentation shall include hazards that were identified and eliminated, and specific procedural controls and precautions to be followed to mitigate the risks of hazards that could not be eliminated.

STPA, like other hazard analysis techniques, provides input to the SAR in the form of the hazards, potential causes, and requirements for mitigating or controlling them. The final decision making about what was done about each of the hazards and the risk acceptance decisions are beyond the scope of STPA. As any hazard analysis technique, it can only provide input to this process.

In addition, STPA goes beyond the traditional hazard analysis methods in what it provides about hazardous human and software behavior. Specific design decisions related to how to prevent these hazards are provided by the analysis and used in the final SAR versus simply an argument about the level of rigor that was done on the software or a probability of a human error.

TASK 302 - HAZARD MANAGEMENT ASSESSMENT REPORT
302.1 Purpose. Task 302 is to perform and document a Hazard Management Assessment Report (HMAR) to provide a comprehensive evaluation of the status of hazards and their associated risks prior to test or operation of a system, before the next contract phase, or at contract completion.
302.2 Task description. The contractor shall perform and document an assessment to identify the status, at the time of the report, of hazards, associated risks, mitigation measures, and formal risk acceptance decisions. This documentation shall include hazards that were identified and eliminated and specific procedural controls and precautions to be followed to mitigate the risks of hazards that could not be eliminated. The contractor shall prepare a report that contains the following information:

Again, STPA provides input to this report in the form of the hazards identified (including the Unsafe Control Actions), potential causes, and recommendations for eliminating or mitigating them as well as any controls that are created to mitigate them. The actual risk assessment decisions go beyond the scope of a hazard analysis technique.

TASK 303 - TEST AND EVALUATION PARTICIPATION
303.1 Purpose. Task 303 is to participate in the Test and Evaluation (T&E) process to evaluate the system, verify and validate risk mitigation measures, and to manage risks for test events.
303.2 Task description. The contractor shall participate in T&E planning, support the preparation of test event Safety Releases, conduct post-test event actions, and maintain a repository of reports. The objective is to eliminate the hazards or reduce the associated risks when the hazards cannot be eliminated for both the system and the test events.

TASK 304 - REVIEW OF ENGINEERING CHANGE PROPOSALS, CHANGE NOTICES, DEFICIENCY REPORTS, MISHAPS, AND REQUESTS FOR DEVIATION/WAIVER
304.1 Purpose. Task 304 is to perform and document the application of the system safety process described in Section 4 of this Standard to Engineering Change Proposals (ECPs); change notices; deficiency reports; mishaps; and requests for deviations, waivers and related change documentation.

STAMP, STPA, and CAST can provide support for Tasks 303 and 304 by providing the causes of the hazards and using these to contribute to the development of test cases. A recent Ph.D. dissertation by Major Dan Montes (U.S. Air Force) showed how to apply STPA to Air Force flight test programs. Anomalies during test as well as mishaps, etc., can be analyzed using CAST (Causal Analysis based on System Theory).

STPA also assists in applying the system safety process to changes and change proposals. Using the safety control structure created while originally applying STPA, the changes can be located in the structure so that only those parts affected need to be examined to determine if the change

introduces or changes hazards. An MIT master's thesis demonstrated the practicality of this process at a major auto manufacturer [Sgueglia 2015].

TASK SECTION 400 – VERIFICATION

TASK 401 – SAFETY VERIFICATION
401.1 Purpose. Task 401 is to define and perform tests and demonstrations or use other verification methods on safety-significant hardware, software, and procedures to verify compliance with safety requirements.
401.2 Task description. The contractor shall define and perform analyses, tests, and demonstrations; develop models; and otherwise verify the compliance of the system with safety requirements on safety-significant hardware, software, and procedures (e.g., safety verification of iterative software builds, prototype systems, subsystems, and components). Induced or simulated failures shall be considered to demonstrate the acceptable safety performance of the equipment and software.

As stated for Task 303, the generation of tests and demonstrations from the identified causal scenarios is straightforward.

TASK 402 EXPLOSIVES HAZARD CLASSIFICATION DATA
TASK 403 EXPLOSIVE ORDNANCE DISPOSAL DATA

STPA has no relevance for the two tasks related to explosives.

Compliance Summary for MIL-STD-882E

STPA is totally compliant with MIL-STD 882 and, in fact, was created explicitly to support the tasks involving analysis in this standard. It is a top-down, system hazard analysis that can be used for the hazard analysis tasks (Tasks 201 to 209). It identifies hazards and generates both (1) system and component safety and cyber security requirements and (2) the information needed to eliminate or mitigate the identified hazards. The STAMP control structure provides a way to document much of the information about the system structure and flow down of requirements described in the tasks.

STPA modeling and analysis can begin in the concept development stage to assist engineers in decision making about basic architectural design and development issues in order to eliminate or mitigate hazards. Having specific causal scenarios can help to help to ensure that the PHA does not incorrectly categorize important hazards by a premature likelihood evaluation without causal information. STPA satisfies the need emphasized in Task 208 for obtaining safety-related information early in the program. As decisions are made, the analysis can be refined in an iterative fashion. The information generated can also be used in verification and validation activities.

8. Army Regulation (AR) 385-16 System Safety Engineering and Management

This regulation prescribes policies and procedures and identifies responsibilities to ensure hazards in Army systems and facilities are identified and the risks associated with these hazards are properly managed. It appears to take the tasks included in MIL-STD-882 and assign them to appropriate people to ensure the hazards are identified and eliminated or controlled. Because STPA is compliant with MIL-STD-882, it is also compliant with AR385-16. It can provide hazard-related information to the decision making responsibilities outlines in the standard. In addition, CAST (Causal Analysis based on Systems Theory), which is a tool for mishap analysis based on STAMP, could be used to assist with tasks associated with compiling lessons learned.

9. AMCOM Regulation 385-17 Software System Safety Policy

The primary purpose of this regulation is to provide a tailorable set of software safety requirements. One of the tasks specified is to "assist in generating the software safety requirements for the system components," which is what STPA does.

One of the primary reasons for creating the new, more powerful STPA hazard analysis technique was the increasing use of software in systems, which current analysis techniques do not handle very well. Therefore STPA's compliance with this regulation and its contributions to implementing the regulation is of great importance in any decision to use STPA on FVL. The bottom line is that STPA not only is compliant with and supports AMCOM Regulation 385-17, but STPA provides an effective and practical way to implement many of the requirements in the regulation that do not, on the surface, appear to be possible or practical to do *without* STPA.

In this section, the relevant requirements for software safety documented in AMCOM are presented along with a description of how STPA supports them. Those general software engineering requirements not directly related to STPA, such as configuration control, are omitted due simply to length reasons. The conformance of STPA with and assistance to AMCOM Regulation 385-17 can be divided into two parts: the activities required by the standard and the hazard analysis (assessment) approach. A final subsection is included to discuss the problem of safety and COTS.

9.1 AMCOM Regulation 385-17 Activities

The regulation includes the following assumption:

> "Software alone is not a safety issue; it is only an issue in the context of these larger systems and the hardware with which it interfaces. Hence, SwSS [Software System Safety] must begin with an understanding of the larger system and its mission/capabilities and be considered in the context of the system's associated hardware, environment, operators, and interfaces. The safety premise driving the analyses is that uncontrolled software hazard contributors can be propagated, via the interfaces, onto supported hardware and systems, and that those systems may be placed into a hazardous condition as a result."

This assumption is basically the same as that underlying STAMP, so the philosophical approaches to software safety are compatible.

The regulation goes on to state:

> "Software's contribution to system level hazards must be assessed within a structured and disciplined system safety program. … Software hazard controls are specified through requirements and those requirements are traceable to both the hazard analyses and the software design and implementation. The SwSS program must be able to provide evidence of end-to-end traceability (from system assessment and hazard identification through verification of safety controls) to satisfy software safety assurance requirements. The program's implementation of the requirements in this regulation, to include evidence of traceability, should be specified in the SSMP, the contractor's SSPP and software development documentation."

STPA supports such a structured and disciplined system safety program. It assists in generating the software safety requirements for the system components that include software. Traceability from the high-level system hazards to the software requirements is provided in the STPA process.

AMCOM 385-17 divides the software system safety program into five stages:

1. System Concept Refinement
2. Software Requirements and Architecture Development
3. Software Design, Coding, and Implementation
4. Verification and Validation
5. Support for Software Release

STPA can contribute either directly or indirectly to most of the tasks listed under these activities by assisting in doing the following:

- Identifying software-related hazards (software contributions to system-level hazards)
- Identifying the causal scenarios that can lead to the hazardous software behavior
- Providing information to both those who are designing the software to eliminate hazardous software behavior and those who are designing the system in which the software will operate to mitigate any potentially hazardous software behavior.
- Creating documentation and models (i.e., the safety control structure) for communication and common understanding about hazards, unsafe control actions, causes of unsafe control actions, safety-critical feedback to controllers, environmental threats (including cyber security threats), and coordination and communication within the system being considered and within a larger system.
- Performing software safety hazard analyses
- Tracking hazards, their consequences, and the control strategy selected in the hazard log
- Tracing system-level hazards into the software architecture

- Identifying safety-critical software functions and components as well as safety-critical feedback and communication requirements.
- Supporting software testing and verification activities
- Supporting the analysis of change requests with respect to introducing new hazards or potentially degrading previous hazard controls
- Supporting the generation of contractor's SOWs and scoping of safety efforts
- Generating leading indicators to identify when changes in the software or the system may be leading to a mishap.

In the demonstration of STPA in Section 6, the tracing from the ConOps and Objective MEP Definition to the STAMP safety control structure was shown for the Medium FVL FoS (Section 6.2). STPA can then be used to generate system and component functional requirements. For components that will be implemented by software, the specific hazards associated with that component are identified at the system level in the generation of the unsafe control actions for that component. The requirements that are generated from these hazards can be used to assist throughout the software system safety process defined in AMCOM 385-17.

For example, consider the Mission Controller and Weapons Processors in the partial example shown in Section 6. The high-level hazards directly associated with the weapon system function are uncommanded detonation, uncommanded launch, collateral damage or friendly fire, inadvertent launch, and non-deployment.

These system hazards can be refined further and associated with specific functions on the aircraft. For example, in the safety control structure in Section 6.2, the PIC oversees and has assigned responsibilities for weapons control by providing commands to and oversight (monitoring) of the Mission Processor. The Mission Processor provides the same role with respect to the Weapons Controller.

The UCA tables for the weapons control commands associated with these system functions provide the next level of refinement for the high-level hazards. For example, refined hazards (UCAs) for the Mission Processor include:

- The Mission Processor does not provide a weapon launch command upon receiving a command from the pilot or designated crew member.
- The Mission Processor issues a launch command when the PIC has not commanded a weapon to be launched.
- The Mission Processor issues a launch command for a target that has not been designated.
- The Mission Processor issues a launch command before an appropriate target has been designated.
- The Mission Processor issues a launch command when the target is no longer valid
- The Mission Processor issues a launch command before the launcher has been properly aimed, etc.

These hazards can be restated as requirements on the Mission Processor. Further analysis at the system level will identify interactions among the system functions, such as the interactions between detecting a threat and deploying a weapon.

The causal scenarios created by STPA will refine these hazards further. For example, UCA 1.1a: *The Mission Processor does not provide the weapons launch command when weapons are needed to complete the mission*. Some causes identified for this unsafe behavior by the Mission Processor are:

Causal Scenario 1.1a: The mission processor does not provide the weapon launch command when weapons are needed to complete the mission because it has a flawed process model of the mission environment. This could occur if:

1. The weapon launch commands from a higher level controller do not reach the mission processor.
2. The mission processor receives incorrect feedback indicating that the aircraft is in an unsafe state to fire the desired weapons.
3. The mission processor does not have sufficient targeting information to aim the weapon.
4. The mission processor believes it has already provided the weapon launch command, either because of an inadvertent error or enemy activity.
5. The mission processor believes that the ammunition has been depleted, again either through error or malicious activity.
6. The mission processor determines that the control commands are improper and decides to ignore them.

Requirements on the design to eliminate or mitigate these causes might include:

107) Backup control paths shall be provided that activate if the primary control path is not available.

108) When possible, multiple sources of feedback and comparisons shall be used to detect when a sensor providing hazard-related feedback is malfunctioning.

109) The PIC shall be alerted if there is not sufficient targeting information to deploy a weapon or the Mission Processor for any reason decides not to follow the control commands provided.

110) Sensors shall be used that check the actual hardware state to provide feedback when possible.

The information provided by STPA can be documented in hazard logs maintained throughout the lifecycle of the system. It can also be provided to contractors in the SOW to assist in their scoping of the safety effort.

The Software Safety Requirements (which are created by STPA on the system safety control structure) can be flowed down into the software architecture. They must be traceable to the lower level software requirements specifications for the software-enabled components. Basically, this process is enabled by STPA by refining each component's control structure, which reflects the functional architecture for the component. Other types of documentation and specification of the detailed software architecture can be generated from the STAMP control structure for the component. This process was not included as part

of this demonstration but could be a goal of a follow on research project. Any software architectures developed must satisfy the system-level software safety requirements generated by STPA.

We have found in extensive use of STPA on industrial projects that the safety control structure provides an extremely helpful tool for communication and common understanding of the safety critical functions. We believe that it will do the same in the Army process, and assist in the interaction between the government and the contractors.

AMCOM regulation 385-17 also requires that any software change requests are reviewed to determine if there is a safety impact. Identification of the impact of changes on the software-related hazards is a straightforward process using the STPA analysis results and the safety control structure. Because everything is done on a concrete model and traced to the components of that model, changes can also be traced to the model components and the analyses that are based on them.

The software safety analyses required by AMCOM 385-17 are supported by the use of STPA as the architectures and required software functions are refined. In fact, this later use of STPA has been its primary use on industrial projects although more companies are now trying to introduce the use of STPA in concept refinement. All of the activities listed for Software Requirements and Architecture Development are directly supported by the STPA analysis process.

9.2 AMCOM Regulation 385-17 Hazard Assessment Approach

The regulation states, as do the underlying assumptions of the STAMP model and of STPA, that software, as a standalone entity, is not hazardous. Software may, however, contribute to hazards. The regulation lists some generic software contributors, e.g., not performing a required function, performing a function not required, etc. All of these fall under and are included in the four types of unsafe control actions used in STPA to identify hazardous behavior.

The regulation goes on to say that "software contributors to hazards may arise through defects, errors, omissions, leading to a failure to operate *correctly*." "Correct," however, is not defined in the document. In software engineering (and usually in engineering in general), correctness is defined in terms of implementation of the specified requirements. Virtually all mishaps related to software behavior can be traced to flawed requirements [Leveson 1995, Leveson 2012]. Correctly implementing requirements that are unsafe will not prevent mishaps.

Similarly, complete correctness is not required for software to be safe as many requirements often have nothing to do with hazardous behavior or mishaps, although they may be important for other system goals such as reliability, maintainability, or efficiency. All incorrect software behavior does not lead to hazards.

AMCOM 385-17 tries to take both approaches, doing hazard analysis to identify and eliminate specific software behavior that can lead to system hazards and including general design guidance (not directly related to identified hazards) that attempts to eliminate all software errors.

STAMP takes the MIL-STD-882 approach to safety by defining a safe system as one that does not lead to mishaps. STPA identifies the functional requirements that must be enforced by the software. These must be traced to the components of the software architecture and verified to have been implemented correctly. STPA does not contribute anything to the parts of AMCOM-385-17 (mostly in the appendices) that list generic types of errors that should not be made or to design features that are deemed to be good practice. There is nothing wrong with following these good design practices, but they are at best only very indirectly related to system hazards and they need to be augmented with a software system safety hazard analysis driven process.

STPA provides a rigorous, defined step-by-step process for identifying the safety-critical (hazardous) software functional and design requirements specifically (not just generically) and their causal scenarios. It can assist, as stated above, with all the software development activities that stem from this initial identification.

Beyond simply assisting with identifying software safety requirements, STPA can assist with augmenting safety in the human–machine interface (HMI). Human operators are included as just another system component in STAMP and STPA. In the example in Section 6.4, UCI 1.1 is an unsafe behavior of the PIC: *The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight plan*. Causal scenarios identified by STPA include:

Example Causal Scenarios for UCA 1.1a: The PIC does not set aircraft attitude when the trimmed aircraft state is causing the aircraft to deviate from its flight path because the PIC believes that the aircraft is on the desired flight path, i.e., his/her process model is flawed. This could occur if:

1. The PIC does not receive detailed enough feedback through the PVI to detect small deviations from the flight plan. These small deviations become greater over time and the aircraft could violate separation with an obstacle. This problem can be compounded in tethered scenarios where the PIC is responsible for multiple vehicles and cannot easily check clearances for the tethered vehicles.
2. The PIC did not program a flight plan into the aircraft systems and is unable to project a flight path for the aircraft in the airspace because of degraded conditions or a high workload. This results in disorientation and a loss of aircraft control or violation of separation standards.
3. The aircraft is incorrectly indicating that it is following the desired flight path. This could occur if the navigation system or PVI supplies inaccurate or insufficient feedback to the PIC due to degraded operation, enemy kinetic/cyber attack, or poor design.
4. The PIC believes that autopilot is engaged and the aircraft computer systems are controlling pitch. This mode confusion could occur if there is not clear feedback to the PIC about the autopilot's state and/or the autopilot does not alert the pilot if it changes state.
5. Unauthorized communication from a second subsystem provides an incorrect state to the PIC. This could occur if subsystems are allowed to send arbitrary messages to displays leveraged by the PIC to inform his or her process model.
6. Malformed communication from sensors leads to incorrectly indicating the aircraft is following the desired path.

7. Adversaries purposely provide incorrect input to the pilot by altering the feedback to the displays.

These scenarios might identify HMI design requirements such as:

1) Navigation systems and interfaces shall allow for navigation with error less than TBD miles in manual flight modes.
2) The navigation systems and PVI shall be monitored for faults at TBD Hz to ensure that they are updating.
3) Navigation accuracy shall be confirmed through multiple independent sensors.
4) The PIC shall be alerted when the aircraft deviates from the flight plan by TBD feet.
5) The PIC shall be alerted if there is an object within TBD feet of separation minimums.
6) The aircraft shall provide the PIC with clear feedback to indicate what responsibilities the computer systems are currently taking.
7) The aircraft shall alert the pilot if there are uncommanded mode changes by the computer systems.
8) The PIC shall be provided with separation information about tethered vehicles and a visualization of the entire formation relative to the flight environment.
9) A minimal flight plan shall be provided for all missions so that the aircraft can be reoriented to the flight path if the PIC becomes disoriented.
10) Data sources for informational displays shall be specified. Integrity validation shall be performed on all data sources.
11) Explicit specification of data formats shall be provided for informational displays so that the displays can be verified to correctly parse data and correctly respond to malformed data.

References

Abrecht, Blake (2016a) *Systems Theoretic Process Analysis Applied to an Offshore Supply Vessel Dynamic Positioning System*, Master's Thesis, Massachusetts Institute of Technology.

Abrecht, B., Arterburn, D., Horney, D., Schneider, J., Abel, B., and Leveson, N. (2016b), A New Approach to Hazard Analysis for Rotorcraft*, American Helicopter Society Specialists Meeting on Development, Affordability, and Qualification*, Huntsville AL, Feb. 9-10.

Johnson, Kip (2016), Extending Systems-Theoretic Safety Analyses for Coordination, Ph.D. Dissertation, MIT Aeronautics and Astronautics Dept., September.

Leveson, Nancy (1995) *Safeware: System Safety and Computers*, Waltham, MA: Addison-Wesley.

Leveson, Nancy (2012). *Engineering a Safer World*, MIT Press, 2012.

Montes, Daniel (2016) *Using STPA to Inform Developmental Product Testing*, Ph. Dissertation, Aeronautics and Astronautics Dept., Massachusetts Institute of Technology.

Sgueglia, John (2015), *Managing Design Changes using Safety-Guided Design for a Safety Critical Automotive System*, Master's Thesis, Massachusetts Institute of Technology.

Thomas, John (2013) *Extending and Automating a Systems-Theoretic Hazard Analysis for Requirements Generation and Analysis*, Ph.D, Dissertation, Massachusetts Institute of Technology.