Early Concept Development and Safety Analysis of Future Transportation Systems

Citation: Fleming, Cody H. and Nancy G. Leveson. "Early Concept Development and Safety Analysis of Future Transportation Systems." IEEE Transactions on Intelligent Transportation Systems 17, 12 (December 2016): 3512–3523 © 2016 IEEE

As Published: http://dx.doi.org/10.1109/TITS.2016.2561409

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Persistent URL: http://hdl.handle.net/1721.1/115299

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Early Concept Development and Safety Analysis of Future Transportation Systems

Cody H. Fleming and Nancy G. Leveson, *Member, IEEE*

***Abstract*—As transportation systems become increasingly complex and the roles of human operators and autonomous software continue to evolve, traditional safety-related analytical methods are becoming inadequate. Traditional hazard analysis tools are based on an accident causality model that does not capture many of the complex behaviors found in modern engineered systems. Additionally, these traditional approaches are most effective during late stages of system development, when detailed design information is available. However, system safety cannot cost-effectively be assured by discovering problems at these late stages and adding expensive updates to the design.**

Rather, safety should be designed into complex, intelligent transportation systems from their very conception, which can be achieved by integrating powerful hazard analysis techniques into the general systems engineering process. The primary barrier to achieving this objective is the lack of effectiveness of the existing analytical tools during early concept development. This paper introduces a new technique, which is based on a systems- and control-theoretic model of accident causality that can capture behaviors that are prevalent in these complex, software-intensive systems. The goals are to (1) develop rigorous, systematic tools for the analysis of future concepts in order to identify potentially hazardous scenarios and undocumented assumptions, and (2) extend these tools to assist stakeholders in the development of concepts using a safety-driven approach.

***Index Terms*—safety analysis, automation, human-automation interaction, systems engineering**

I. Motivation

Emerging technologies show a tremendous potential for transforming and improving all modes of transportation in the future. Driver assist features in today's automobiles promise to improve the driver experience in many ways including convenience, efficiency, and safety [1]; intelligent rail systems will increase the efficient use of existing and future infrastructure [2], [3]; and next generation air traffic management technologies promise to increase traffic throughput while minimizing environmental impact [4].

However, implementation and use of new transportation technologies is often inhibited by stakeholders' inability to ensure that the systems are safe. That is, a lack of safety often inhibits, and potentially prohibits, the implementation and use of these new transportation technologies.

With the continued introduction of software into transportation systems, new kinds accidents are occuring, such as the Wenzhen train accident [5]. In addition to the potential for tragic consequences, a lack of safety has other detrimental effects. Software-related recalls in the automotive industry damage automaker reputations and consumer trust [6] in addition to causing accidents, while entire air transport fleets have been grounded due to issues with application software on tablets [7] or large-scale computer systems [8], [9].

Often there is perception among stakeholders involved in developing a complex system that safety is expensive. Safety-related features are also seen as intrusive because they seem to result in reduced performance, increased weight, or unnecessary complexity. In fact safety often *is* costly, both in terms of economics and technical performance, but this is not due to any intrinsic property of safety itself. Rather, one reason safety costs so much is that it is often considered only after the major architectural tradeoffs and design decisions have been made. Once the basic design is finalized, the only choice is to add expensive redundancy or excessive design margins [10].

It has been estimated in the defense community that up to 80% of the decisions affecting safety are made in the early concept development stages of a project [11]. Compensating later for making poor choices at the beginning can be very costly, as illustrated in Fig. 1. Safety must be designed and built into systems from the very beginning of concept development.

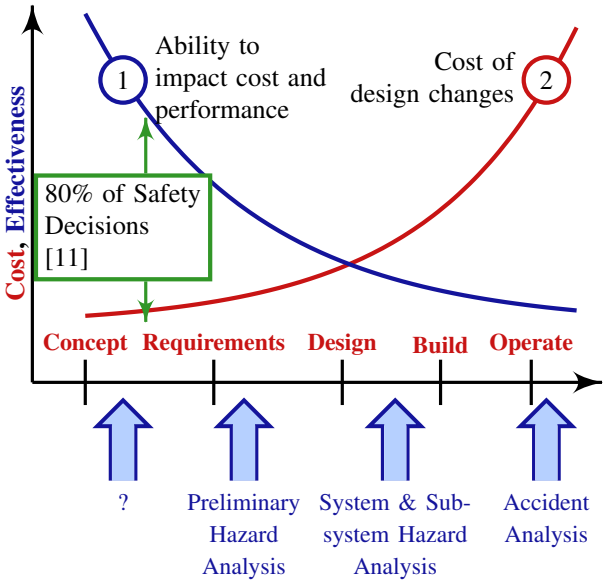

Figure 1. Decision Effectiveness during Life Cycle (adapted from [12])

This paper first presents a brief background on concept development, traditional approaches to safety engineering,

Manuscript received Month Day Year. This research was supported by NASA LEARN grant NNX14AC71A.

C.H. Fleming is an Assistant Professor of Systems & Information Engineering at the University of Virginia.

N. Leveson is Professor of Aeronautics and Astronautics at the Massacusetts Institute of Technology.

0000–0000/00$00.00

and an outline of the characteristics of future transportation systems, particularly air transport. A new approach to rigorously and systematically including safety during early concept formation follows this review, and the new approach is then demonstrated on a future air transportation system. The paper then concludes with a a summary of contributions and proposes potential extensions to the work.

This research has two main objectives. The first objective is to present a rigorous, systematic framework for the analysis of future transportation systems in order to identify hazardous scenarios and undocumented assumptions during concept formation. Related to this theme is the problem of assessing safety-related risk when little design detail is available, with the goal of assisting concept development and design when modifications are most effective. The second objective is to demonstrate how these tools assist stakeholders in the development of concepts using a safety-driven approach. Ideally, this safety-guided concept development would supplement existing activities used for developing transportation systems, including architectural and design studies that occur during tradespace exploration. Both objectives especially apply to systems where the tradespace includes human operation, automation or decision support tools, and the coordination of decision making agents—all characteristics of many of tomorrow's transportation systems.

II. Background

While safety is almost always a priority for stakeholders involved in developing a system with any societal impact, particularly for transportation systems, safety has not explicitly been considered in many of these early phase system engineering activities [13].

Most of the effort in proceeding from a set of stakeholder needs to a design that can be implemented involves the selection of *how* to meet those requirements. Stakeholders trade off different properties or design decisions in order to obtain a system that meets requirements and satisfies certain performance criteria. Common system properties used in tradespace exploration of transportation systems include travel time, travel range, fuel usage and efficiency, system lifecycle costs, and others [14], [15].

The following review describes the current state of the practice as it relates to safety engineering during early system development activities. This review is followed by a brief description of an alternative approach to safety analysis, and then finally a description of the challenges facing those stakeholders charged with developing tomorrow's transportation systems.

A. Traditional Approach to Safety Engineering

This section is not intended to review hazard analysis techniques in general. Rather, this review pertains to how safety is integrated—or not integrated—into general engineering activities, particularly concept development of complex transportation systems. Traditionally, safety-related activities conducted during the preliminary phases of an engineering program include developing Preliminary Hazard Lists (PHL), performing Preliminary Hazard Analysis (PHA), and informing decision-makers by using risk assessment techniques, such as a risk matrix. This traditional approach assesses risk by combining the likelihood and consequence (and sometimes mitigation measures) of a particular hazard [16], [17], [18], [19].

Table I
Sample PHA Worksheet, adapted from [20]

PRELIMINARY HAZARD ANALYSIS PROGRAM: _____ DATE: _____ ENGINEER: _____ PAGE: _____					
HAZARDOUS CONDITION	CAUSE	EFFECTS	RAC	ASSESS-MENTS	RECOMMEN-DATIONS
List the nature of the cond-ition	Describe what is causing the stated condi-tion to exist	If allowed to go uncorr-ected, what will be the effect or effects of hazardous condition	Hazard Level assign-ment	Probability, possibility of occurr-ence: -Likelihood -Exposure -Magnitude	Recomm-ended actions to eliminate or control the hazard

These efforts, however, are not explicitly intended to enable the development of requirements or the comparison of different system architectures or design alternatives. Rather, comparison of system architectures and design alternatives are based on trade studies that incorporate performance objectives such as travel time, travel range, and efficiency, as well as cost and schedule estimates. Safety is rarely included in these trade studies [13], and the preliminary hazard analysis is conducted separately from architecture generation.

Preliminary hazard analysis (PHA) is a guided analysis effort that occurs early in the engineering process, when detailed design information is not available. Standard preliminary hazard analyses include a list of hazards to be avoided, potential causes of those hazards, effects on the system, severity level of the hazards, and supporting comments or recommendations [20]. Table I shows a generic PHA table and expected contents.

Currently and in the past, PHA has focused on failure modes of sub-systems or components, but such a focus is limited for several reasons. The types of causes identified by PHA techniques are limited to electro-mechanical faults or very generic causes related to human or software behavior. For example, a recent PHA in the aerospace domain listed "Design flaw, coding error, software OS problem" and "Human error" as potential hazard causes [21]. These generic types of causes are not particularly useful for guiding the design. That is, hardware, software, or humans cause all hazards. Simply listing a generic set of factors is not very useful for the overall engineering effort, and PHA techniques suffer from a lack of guidance in identifying causal factors that lead to specific

hazardous states that stakeholders wish to avoid.

B. Systems Approach to Safety

Many existing hazard analysis tools, such as Fault Tree Analysis (FTA) and Failure Modes and Effect Analysis (FMEA), are more appropriate in the later stages of system development when detailed design information is available [22]. Perhaps more importantly, these tools are also limited in the types of scenarios they identify. These tools were developed long ago—Bell Laboratories developed FTA in the 1960s [23] and the U.S. Department of Defense developed FMEA in 1949 for its weapon systems program [24]—when the primary cause of accidents was due to mechanical failure [25]. Modern systems exhibit hazardous behavior due to factors that extend well beyond hardware failure. The introduction of new technology, such as computers and software, is changing the types of accidents we see today [26], [27].

Hazardous behavior arises in systems due to unsafe interactions between components, even when the components have not necessarily failed. Given the complexity of today's systems, these interactions are increasingly difficult to understand and predict. The underlying assumptions of traditional hazard analysis tools also oversimplify the role of human operators [28], [29], [30] and software requirements errors [10], [31]. Not only are traditional hazard analysis techniques incapable of analyzing systems that are immature in terms of design detail, they are also very limited with respect to these new accident causation factors, which will become increasingly prevalent in tomorrow's systems.

This paper, therefore, introduces a new technique based on an different view of accident causality.

System-Theoretic Accident Model and Processes (STAMP) was created to capture more types of accident causal factors including social and organizational structures, new kinds of human error, design and requirements flaws, and dysfunctional interactions among failed and non-failed components [32], [27]. Rather than treating safety as a failure problem or simplifying accidents to a linear chain of events, STAMP treats safety as a control problem in which accidents arise from complex dynamic processes that may operate concurrently and interact to create unsafe situations.

Accidents can then be prevented by identifying and enforcing constraints on component interactions. This model captures accidents due to component failure, but also explains increasingly common component interaction accidents that occur in complex systems without any component failures. For example, software can create unsafe situations by behaving exactly as instructed or operators and automated controllers can individually perform as intended but together they may create unexpected or dangerous conditions.

STAMP is based on systems theory and control theory. In systems theory, emergent properties are those system properties that arise from the interactions among components. Safety is a type of emergent property. The emergent properties associated with a set of components are related to constraints upon the degrees of freedom of those components' behavior [33]. There are always constraints or controls that exist on the interactions among components in any complex system. These behavioral controls may include physical laws, designed fail-safe mechanisms to handle component failures, policies, and procedures. Such controls must be designed such that the safety constraints are enforced on the potential interactions between the system components. In air traffic control, for example, the system is designed to prevent loss of separation among aircraft.

System safety can then be reformulated as a system control problem rather than a component reliability problem—accidents occur when component failures, external disturbances, and/or potentially unsafe interactions among system components are not handled adequately or controlled, leading to the violation of required safety constraints on component behavior. System controls may be managerial, organizational, physical, operational, or in manufacturing. In STAMP, the safety controls in a system are embodied in the *hierarchical* safety control structure. The next section describes hierarchy theory in greater detail.

Accidents arise from inadequate enforcement of safety constraints, for example due to missing or incorrect feedback, inadequate control actions, component failure, uncontrolled disturbances, or other flaws. STAMP defines four types of unsafe control actions that must be eliminated or controlled to prevent accidents:

1) A control action required for safety is not provided or is not followed
2) An unsafe control action is provided that leads to a hazard
3) A potentially safe control action is provided too late, too early, or out of sequence
4) A safe control action is stopped too soon or applied too long

One potential cause of a hazardous control action in STAMP is an inadequate process model used by human or automated controllers. A process model contains the controller's understanding of 1) the current state of the controlled process, 2) the desired state of the controlled process, and 3) the ways the process can change state. This model is used by the controller to determine what control actions are needed. In software, the process model is usually implemented in variables and embedded in the program algorithms. For humans, the process model is often called the "mental model" [32]. Software and human errors frequently result from incorrect process models. While process model flaws are not the only cause of accidents in STAMP, it is a major contributor.

The generic control loop in Figure 2 shows other factors that may cause unsafe control actions. Consider an unsafe control action for a train control system: a train conductor is instructed to proceed to the next station while a maintenance crew works on the track. The control loop in Figure 2 would show that one potential cause of that action is an incorrect belief that the track ahead of the train is clear (an incorrect process model). The incorrect process model, in turn, may be the result of inadequate feedback provided by a failed sensor, the feedback may be delayed, the data may have been corrupted, etc. Alternatively, the system may have operated

exactly as designed but the designers may have omitted a feedback signal.

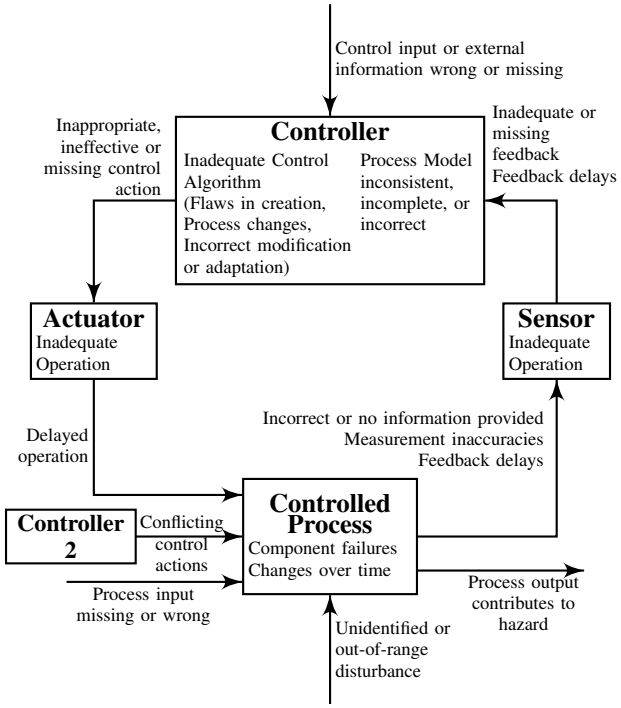

Figure 2. STPA Control Loop with Causal Factors

C. Future Transportation Systems

Intelligent transportation systems call for increased focus on more efficient travel, a shift in responsibility from human-in-the-loop control to decision support tools and automated safety features, and many other changes. In short, upgrades in future transportation systems will result in increased reliance on automation, greater coupling between infrastructure and vehicle technology, a major shift in the way roadway, railway, airspace, and other information is gathered and disseminated, and a total revamping of how vehicle paths are managed. All of these changes will come as the result of incremental upgrades that span years and even decades. These developments also put ever increasing pressure on early systems engineering activities.

1) Engineering of Long-term Transportation Concepts: Early engineering activities such as concept generation, architecting, and early requirements generation do not explicitly include safety even though it would be highly beneficial during this part of system development. This exclusion of safety-related activities is perhaps appropriate given the current state of the art. Though some of the concept generation frameworks have become increasingly formalized, the techniques still do not yield the level of detail necessary to perform most traditional types of hazard analysis [22]. Often, one of the only prominent artifacts of early engineering concepts is a Concept of Operations document.

A Concept of Operations (ConOps) can be developed in many different ways, but usually share the same properties. In general, a ConOps will include a statement of the goals and objectives of the system; strategies, tactics, policies, and constraints affecting the system; organizations, activities, and interactions among participants and operators; and operational processes for fielding the system [34]. A ConOps

> describes how the system will be operated during the life-cycle phases to meet stakeholder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals [17].

As is the case with ConOps, one of the challenges is that much of the documentation in these early phases is informal, and specifications do not yet exist.

2) Air Transportation Issues: Like many transportation systems, the United States air transportation system is under stress and demand in aircraft operations is expected to increase significantly in the next decade and beyond [35]. There are also growing concerns about air transportation's effect on the environment and national security. Current technologies and procedures in the national airspace cannot meet these increasing demands; therefore, the United States is creating the Next Generation Air Transportation System (NextGen) air traffic management modernization program. The goals of NextGen are to expand capacity, ensure safety, protect the environment, and grant flexibility and equity to airspace users.

Past attempts at modernizing the national airspace have failed or fallen exceedingly short of expectations, in part due to the ineffectiveness or lack of tools necessary to develop and integrate new technologies and procedures [36], [37]. The methods described in this paper represent an attempt to overcome some of the difficulties that have plagued past attempts at developing intelligent air transport systems, in particular that safety-related flaws have not been discovered until too late in the development process.. The following methodology is applied to one aspect of NextGen called Trajectory-based Operations, which will be described further in Section IV.

III. Methodology

A key contribution of this new approach to the engineering of complex transportation systems is that it explicitly models the interactions among various components and decision-making agents in an organized, top-down fashion. Two fundamental concepts of systems theory—hierarchy and emergence, and communication and control—are fundamental to STECA. Control-theoretic concepts are used first to construct a model of the system, and theories of hierarchy and emergence (in addition to control and communication) are then used to analyze the model itself. The process is conducted according to Figure 5. The following sub-sections describe the theoretical development as well as provide a brief example for illustrative purposes.

A. Systematic Control Model Development

Modeling a concept of operations consists of two basic steps. The first step parses the informal, natural language text and/or graphics of a ConOps into control theoretic elements.

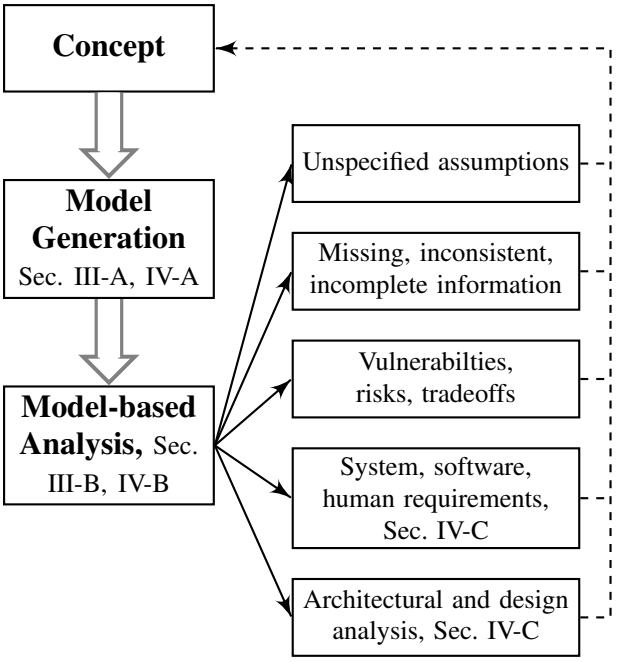

Figure 3. STECA Methodology

The next step synthesizes these control theoretic elements into a hierarchical control structure.

1) Identifying Control Concepts : The first modeling step consists of examining the text (or graphics) of a ConOps and considering the basic functions of each entity in the control loop. That is, what is required of each entity in the control loop for effective, safe system behavior? What are the responsibilities of the controller, actuator, controlled process, and sensor? How do these entities interact with each other, with the environment, and with other control loops?

The Controller:

- creates, generates, or modifies control actions based on algorithm or procedure and perceived model of system
- processes inputs from sensors to form and update process model

The Actuator:

- Translates controller-generated action into process-specific instruction, force, heat, torque, or other mechanism

The Controlled Process:

- Interacts with environment via forces, heat transfer, chemical reactions, or other input
- Translates higher level control actions into control actions directed at lower level processes (if it is not at the bottom of a control hierarchy)

The Sensor:

- Transmits continuous dynamic state measurements to controller
- Transmits binary or discretized state data to controller
- Synthesizes and integrates measurement data

The roles of the controller, actuator, controlled process, and sensor, and their interactions with the environment and other control loops can be summarized with 15 generic keywords or guide words. Figure 4 depicts these guide words in the familiar control loop format. With a proper accounting of these 15 items, the control loop can achieve the necessary conditions of process control [38] and adequately interact with its environment, other processes, and other controllers. In other words, these guide words are necessary to ensure that a controlled process is controllable and that the control loop is coordinable with other controlled processes.

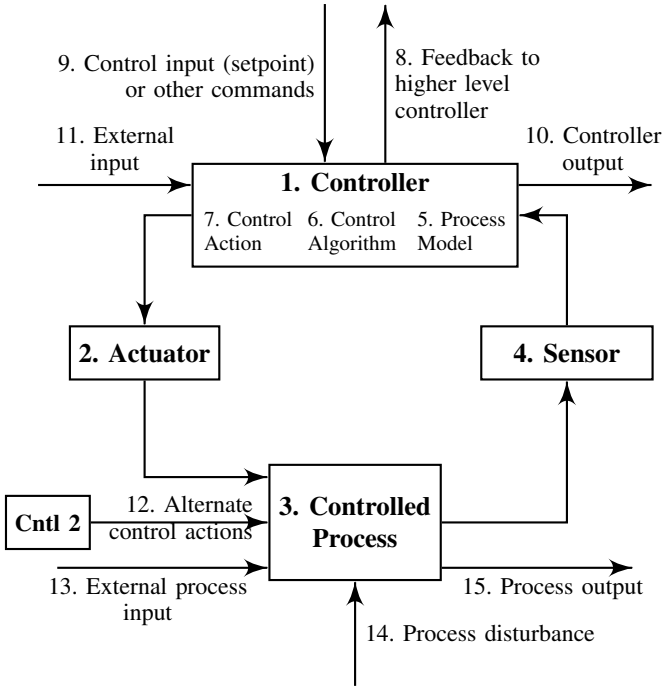

Figure 4. Control Loop with generic entities

The information in Figure 4 and the above lists (Controller, Actuator, Controlled Process, Sensor) can then be used to systematically parse and query the natural language description or graphical depiction in a concept of operations. The resulting model and subsequent database are easy to interrogate and visualize. These qualities help the analyst to check for internal inconsistencies and/or missing information that may result in unsatisfied control conditions, and also to check for inconsistencies across the system hierarchy.

Table II provides a series of prompts that an analyst can use when reading a text or graphic in a ConOps.

A formal, mathematical description of the control models is developed elsewhere by the author [39] and will be developed in the following section.

2) Synthesizing Control Information into System Hierarchy: The previous development involves parsing the ConOps by mapping informal textual or graphical information into control-theoretic constituents defined by Figure 4. The resulting control loops do not, however, by themselves represent a model of the entire concept. The above analysis should result in a set of controllers, each with its own actuators, processes, and sensors. The approach described in this paper relies not only on basic control theory but also on general

Table II
CONTROL-THEORETIC ANALYSIS OF TEXTUAL OR GRAPHICAL INFORMATION

Source / Subject	What is the primary subject of the text? What is the primary source of action that the text (or graphic) is describing?
Role	Is the Source or Subject a Controller, Actuator, Controlled Process, or Sensor?
Behavior Type	For the given role, which type(s) of behavior does it exhibit? See the lists in the body text above for each control role
Justification / Context	Provide a justification for categorizing the text (or graphic) in the chosen manner.

systems theory. In systems theory it is inappropriate to analyze individual control loops and then make a determination about the overall behavior of the system. Furthermore, it is inappropriate to analyze individual components like sensors, actuators, or controllers.

Rather, the behavior of the system can only be determined in the context of all the components and their *interactions*. Instead of focusing solely on understanding the behavior of each component, the relevant issue here relates to how the control elements relate to each other. This section develops an approach, based on hierarchy theory, to determine the relationships between control components.

This sub-section presents heuristics for *identifying* or constructing the hierarchy based on information contained in the ConOps, while the following sub-section presents a method for checking the consistency across the hierarchy. This paper proposes the use of several abstractions that can be used to determine the "vertical" and "horizontal" relationships between control components. Section IV also introduces the formal mathematical notation

Several different (but related) notions of hierarchy, or abstraction, may be used to generate a system-level control model from the individual control loops generated in the analysis using Table II and Figure 4. For example, [40] describe three types of hierarchies: strata or levels of description, layers or levels of decision complexity, and echelons or organizational decomposition. In the controls literature, the hierarchy is typically layered in terms of time scale; for example, scheduling (weeks), system-wide optimization (days); local optimization (hours); supervisory, predictive, or advanced control (minutes); and regulatory control (seconds) [41]. The echelon hierarchy—described by Mesarovic and used in the controls literature [42]—is often used to decompose a system using the notion of decision-making authority. That is, some decision-making units are influenced or controlled by others.

A specific characteristic of the echelon hierarchy is that there are many elements within a given level, which implies another dimension of organization. Intent Specifications [43] organize system information according to three types of hierarchy: level of intent, part-whole abstractions, and refinement. Part-whole abstraction provide another horizontal decomposition of the system. That is, while decision complexity, time scale, or authority defines a hierarchy *vertically*, part-whole abstractions describe the organization and relationships *horizontally* within a given level.

To summarize, higher levels of a hierarchical control system typically:

1) have decision making authority over lower levels. That is, lower level control agents are required to respond to higher level control commands due to formal engineering design, procedures, and/or law;
2) are concerned with larger portions of the system;
3) have to make increasingly complex decisions. Increasingly complex decisions tend to lack well-defined and complete specification of uncertainties, input conditions, problem constraints, and processes involved in transforming input conditions into desired output states [44];
4) exchange with the environment takes place at a lower frequency, the dynamics of concern is slower, and the period between decision time is longer;
5) deal with more abstract descriptions of the system. A control agent may be concerned with–from increasing to decreasing level abstraction–functional purpose, abstract function, generalized function, physical function, or physical form. *Abstraction hierarchies* [45] decompose the system in terms of level of description.

Using the above concepts as a guide, the analyst synthesizes the control theoretic elements (i.e. the individual control loops) into a hierarchical control struture. This hierarchical model provides the basis for analysis.

B. Systems-Theoretic Analysis of Model

Much of the control- and systems-theoretical foundation used to develop the model of the concept is also used to guide the analysis of the model. However, the focus shifts from modeling to identifying potential causal factors and invalid assumptions. There are several general vulnerabilities in a hierarchical system. "At each level of the hierarchical control structure, inadequate control may result from missing constraints (unassigned responsibility for safety), inadequate safety control commands, commands that were not executed correctly at a lower level, or inadequately communicated or processed feedback about constraint enforcement" [27, p.81].

The control-theoretic approach emphasizes the importance of process models in enforcing adequate control: a process model must contain "the required relationship among the system variables (the control laws), the current state (the current values of the system variables), and the ways the process can change state" [27, p.87], or the dynamics of the process. The four fundamental requirements of process control (see 1.a-d below) described in the previous sub-section must also be satisfied.

Once the control model of the ConOps has been built (previous sub-section), the hazardous scenarios and causal factors can be identified using these systems-theoretic views of accident causality. Specifically, the analysts, engineers, and stakeholders should ask:

1) Are the control loops complete? That is, does each control loop satisfy a Goal Condition, Action Condition, Model Condition, and Observability Condition?

a) Goal Condition—what are the goal conditions? How can the goals violate safety constraints and safety responsibilities?
b) Action Condition—how does the controller affect the state of the system? Are the actuators adequate or appropriate given the process dynamics?
c) Model Condition—what states of the process must the controller ascertain? How are those states related or coupled dynamically? How does the process evolve?
d) Observability Condition—how does the controller ascertain the state of the system? Are the sensors adequate or appropriate given the process dynamics?

2) Are the system-level safety responsibilities accounted for, or are there gaps?
3) Do control agent responsibilities conflict with safety responsibilities?
4) Do multiple control agents have the same safety responsibility(ies)?
5) Do multiple control agents have or require process model(s) of the same process(es)?
6) Is a control agent responsible for multiple processes? If so, how are the process dynamics (de)coupled?

As in the previous section, these questions have been formalized elsewhere [39] and will be developed in Section IV. Question 1 relates to completeness of the individual control loops, questions 2-3 relate to assigning safety-related responsibilities to various control agents, and questions 4-6 relate to coordination of multiple control agents. The analysis therefore proceeds through three basic areas, which are explored in the following subsections and depicted in the bottom left of Figure 3.

The following section builds on this description of STECA and demonstrates its application on a complex transportation system. In addition to demonstrating the approach, the underlying formalism of STECA is developed and it is shown how the above questions provide insight into potentially hazardous behavior, particularly when a transportation system involves real-time sensing, computation, and human-computer interaction.

IV. Analysis of Intelligent Transport System

Trajectory-Based Operations (TBO) is a shift from the current Air Traffic Management (ATM) and control strategy of clearance-based operations, which rely on discrete clearances from air traffic controllers that modify the heading, airspeed, or altitude of individual aircraft. Today's operations rely on relatively little automation, in comparison to the TBO framework where aircraft will follow four dimensional paths, called trajectories, which are computed by autonomous systems and decision support tools (DST) [46] and will be continuously monitored and updated. When fully realized, these trajectories will represent an aircraft's gate-to-gate movement and will be the basis for Air Traffic Control (ATC) and Air Traffic Management (ATM) that focuses on traffic flow and airspace use and autonomy of individual aircraft.

The primary themes of TBO are: moving from clearance-based to trajectory-based airspace management, increasing reliance on automation and decision support tools, and distributing traffic management responsibilities throughout the system. The shift from clearance-based to trajectory-based traffic management is intended primarily to increase capacity and improve efficiency.

A. Model Development

In the TBO ConOps [47], there is a chapter dedicated to conformance monitoring, which is a function that measures the degree to which an aircraft follows–or *conforms* to–its agreed-upon trajectory. This example is intended to show how these control-theoretic concepts can be used to (1) query a certain aspect of a concept and then (2) to use the resulting information to build a system model. Querying a ConOps is done in a recursive fashion, looking at individual sentences or paragraphs and attempting to parse control-theoretic information.

The example quote for this analysis is shown at the top of Figure 5. To begin, the analyst must ask: What is the primary source, subject, or actor in the text, and in what way does this source relate to control theory? The quoted text describes conformance, or conformance monitoring. Next, what is the source's role in control theory? Conformance monitoring acts as a sensor, and in this text there appear to be two versions of the sensor: one in the aircraft and another on the ground. Of the three generic roles that a sensor can take in the proposed framework, the conformance monitoring sensor provides two. Figure 5 includes a graphical depiction of how this information is mapped into a control model, where numbers in the text correspond to the numbered boxes in the control model. A separate model should also be developed for the ground conformance monitor.

This process—identifying the behavior associated with a specific source of information in a ConOps, and then inserting it into the appropriate place in a control model—is repeated recursively over the entire ConOps document. For example, Figure 6 on page 14 shows the model development for a different aspect of conformance monitoring. Though the behavior described in this new piece of text is similar to that shown in Figure 5, the new text shows slightly different detail and conformance monitoring is done in a different component of the system.

The process of parsing informal, textual information will result in a set of individual control loops, as in Figure 5 and 6. These individual control loops are then synthesized into a hierarchical control structure according to the guidelines described in Section III-A. A very generic, functional control hierarchy may consist of a route or trajectory management function for all of the airspace, a navigation function that navigates individual aircraft along their prescribed paths, and lower level functions that manipulate the various control surfaces of the aircraft. According to the guidelines from Section III-A, the route management function would be at the highest level because it involves the greatest level of complexity and uncertainty with respect to decision making.

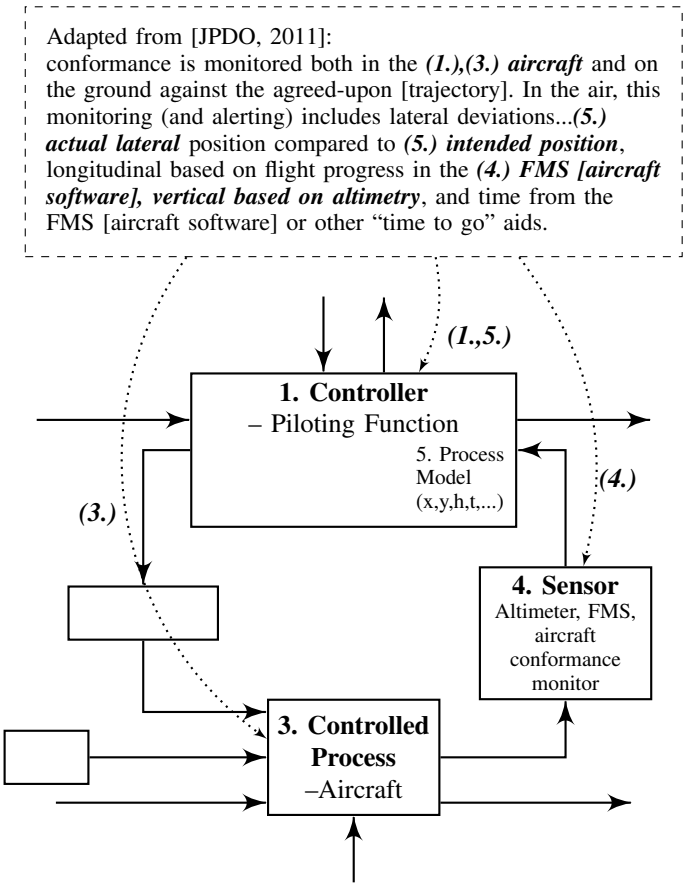

Figure 5. Graphical Control Model of Airborne Conformance Monitor

In addition, the TBO ConOps states elsewhere that the "ANSP's authority over the airspace and the flight crew's authority over the aircraft's trajectory do not change" [47]. It is therefore appropriate and intuitive to map the ground-based conformance monitoring loop to the Trajectory Management Function and the airborne loop to the Piloting (navigation) Function of Figure 7.

Figure 7 depicts the control structure that results from analyzing the full text dedicated to conformance monitoring, as well as other chapters, in the TBO ConOps. Each of the blocks and lines in Figure 7 has an underlying formalism, which results from mapping the information in the ConOps into basic control theory. For example, sensor behavior is a mapping from measured variables, $\mathcal{V}$, to controller inputs, $\mathcal{I}$ (see, e.g. [48] for an overview of the notation). For the ground-based conformance monitor, labeled "Conformance Monitor [Gnd]" in the figure, this mapping is

$$\mathcal{B}_{\mathcal{L}G} := \mathcal{V}_{mG} \rightarrow \mathcal{I}_{Gc}. \tag{1}$$

where $\mathcal{I}_{Gc}$ is the signal going to the ground control agent regarding conformance, and the functional behavior of the sensor is

$$\mathcal{V}_{mG} = \mathcal{L}_G \times D_{c,i} \tag{2}$$

where $\mathcal{L}_G$ is a measurement or model of the airspace state and $D_{c,i}$ is the decision criteria regarding conformance of aircraft i. The "ground", (or ANSP) conformance model is defined as the set of dynamic variables,

$$\mathcal{L}_{\text{G}} := \{z_{\text{int},i}, z_{\text{act},i}, \rho, \tau, P_r, W, E_{cm}, F_D;\ i \in \mathbb{G}\} \tag{3}$$

where

$$\begin{aligned} z_{\text{int},i} &:= \{G, C, t\}_{\text{int},i} \\ z_{\text{act},i} &:= \{G, C, t\}_{\text{act},i} \\ \rho &:= \text{Traffic density} \\ \tau &:= \text{Operation type} \\ P_r &:= \{\text{RNP}, \text{RTP}\} \\ W &:= \text{Wake turbulence model} \\ E_{cm} &:= \text{Elliptical conformance model} \\ F_D &:= \{F,\ z_{\text{int},i}\} \end{aligned}$$

and G is the ground track, C is the climb performance, and t is time of arrival, constituting the aircraft state, z. The $_{\text{int}}$ and $_{\text{act}}$ subscripts represent intended and actual performance, respectively, for the i^{th} aircraft within ANSP jurisdiction, which is the set $\mathbb{G}$. RNP and RTP represent standard definitions of required navigation and time performance, and F_D is a downstream flow model consisting of a general flow model (F) of a particular airspace and a prediction of aircraft arrivals into that space.

Criteria for determining whether an aircraft conforms with its assigned trajectory are,

$$D_{c,i} = \{\, z_{\text{act},i} |\, z_{\text{act},i} \notin \bar{z}_i\,(z_{\text{int},i}, E_{cm}, a_{\text{G}})\,, \quad \forall i \in \mathbb{G}\} \tag{4}$$

where $\bar{z}_i$ is an allowed volume for aircraft i, as a function of the intended aircraft state in time, the elliptical conformance model at a given time (E_{cm}), an alert parameter set by the operator (a_{G}), and $\mathbb{G}$ is the set of aircraft under ANSP jurisdiction. Evaluation of Equation (4) to True indicates that aircraft i does not conform to its assigned trajectory. This is a relatively primitive form of conformance monitoring that simply alerts the user—either a ground controller, flight crew, and/or other software functions that require information about conformance—whether the aircraft is following the assigned trajectory. More advanced forms of conformance monitoring could be developed, but the definition in equation (4) is consistent with the TBO ConOps.

B. Systems-Theoretic Analysis of Model

Due to space constraints, the following development focuses on issues related to coordination and consistency, but completeness and gaps in safety-related responsibilities[1] may also be addressed systematically using this approach [39].

Any two control agents with safety-responsibilities related to the same process states must ensure certain consistency characteristics. These control agents must have a consistent understanding or model of the current state, actions that can affect that state, and how the state will evolve from those actions. Recall from Section II-B the importance of controller process models in ensuring safe control of a system.

[1]Completeness and safety-related responsibilities are addressed in questions 1 and 2-3, respectively, of Section III-B.

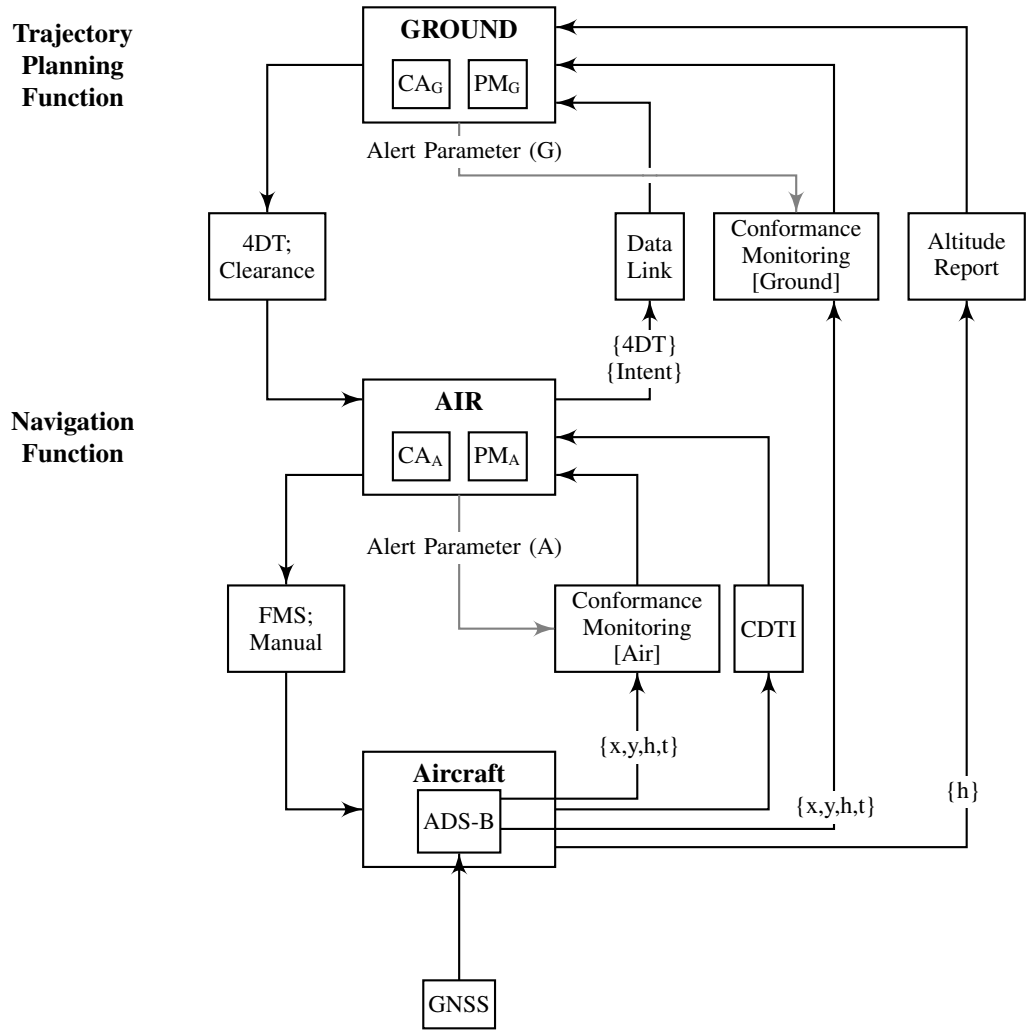

Figure 7. TBO Conformance Monitoring Control Structure

The consistency principle for two controllers is developed as follows. Define the action process predicate, $\boldsymbol{A}(\mathtt{c}, \mathcal{V})$ iff $\mathtt{c} \times \mathcal{V} \rightarrow \mathcal{V}'$. That is, $\boldsymbol{A}$ is true whenever the action $\mathtt{c}$ can cause a change in state(s) from $\mathcal{V}$ to $\mathcal{V}'$. For any two control agents, *i* and *j*, with available control actions $\mathscr{C}_i$ and $\mathscr{C}_j$, respectively, a lack of coordination may arise whenever the following condition evaluates to false:

$$(\forall v \in \mathcal{V},\, \forall \mathtt{c} \in \mathscr{C}_i,\, \forall \mathtt{d} \in \mathscr{C}_j \mid \mathtt{A}(\mathtt{c}, v) \wedge \mathtt{A}(\mathtt{d}, v)) \\ [\rho_i(a, v) \equiv \rho_j(a, v) \wedge G_i \equiv G_j] \quad (5)$$

The intuition behind equation 5 is as follows: whenever two control agents can apply an action to the same process state, then they must have a consistent model of that process state as well as consistent goals about how that state should evolve. $\rho_k(a, v)$ is the k^{th} control agent's model of system's state dynamics, v, subject to inputs, $a = \{\mathtt{c}, \mathtt{d}, \epsilon\}$. The latter aspect of this set, ϵ, represents "other" inputs to the process, such as environmental disturbances. G_k defines the goal condition of the k^{th} controller. The equivalence properties in equation (5) can be described using a property of aggregation called *dynamic exactness* [49]. The preceding formalism paves the way for automatic model checking and validation, which will be explored in future work..

Conformance monitoring is intended to assure consistency among the various actors in TBO, including the ANSP, flight crews, and operating centers. Despite its intent in the TBO ConOps, the architecture of conformance monitoring is actually a source of potential inconsistencies and lack of coordination.

Consider again a conformance model, generalized from the development on the preceding page. Conformance *monitoring* is a mapping from surveillance and other data to a binary (or discrete set of) signal(s) that indicates whether an aircraft conforms to the desired trajectory. Conformance *alerting* is a function of current surveillance data in four dimensions and desired aircraft state in four dimensions, some allowed tolerance volume, and an "Alert Parameter".

This mathematical formulation of conformance monitoring and alerting (equations 1–4) helps identify issues with coordination and consistency, in at least three ways. First, the mapping, $\mathcal{L}_{cm} \times D_{cm}$, is a function of an "Alert Parameter", a_{cm}. This parameter is available to any agent with a conformance monitor, ground or airborne, and is thus independent and

potentially inconsistent. For example, the ground controller sets an alert parameter for his or her own monitor, while each flight crew in the airspace independently does so for their monitor.

The TBO ConOps does not describe or specify the rationale for including this function, but it may be assumed that it is to counteract alarm overload or over- and under-sensitivity. Furthermore, the TBO ConOps does not specify what the alert parameter entails with respect to human-computer interface design. The ConOps does refer to existing aircraft functions such as altitude alerts for the airborne monitor, but it is unclear how either the ground or airborne agents will "set" these parameters.

Several natural design questions arise with respect to the conformance monitoring automation itself, and these questions would typically be addressed during later detailed design phases. Perhaps the most fundamental design question is this: what actually constitutes "non-conformance"? That is, how should the conformance monitoring algorithm(s) be specified? Based on these questions and the information in equations (1)-(4), a few follow-on questions must be addressed. For example,

- Does non-conformance imply that an alert is flagged at any instant the aircraft leaves a (continuously updated) elliptical shape, E_{cm}? Alternatively, does the monitor take an average over a specified time interval and compare that trajectory to the allowable shape?
- Control issues often arise due to improper accounting of timing (e.g. feedback delays in Figure 2 on page 4). How often is the elliptical shape updated? How do the relevant agents receive these updates?
- Is a deviation in one direction given greater importance than a deviation in another direction? For example, a deviation in the direction of other traffic should perhaps be given greater priority relative to a deviation in the opposite direction, given that no traffic exists in the opposite direction.

While these questions could be addressed in a straightforward fashion with the appropriate level of technical expertise, the approach described in this paper provides some important insights about the overall system behavior. In particular, the modeling approach identifies several subtle interactions of components that were otherwise considered to be independent in the original concept of operations.

With the level of abstraction in the model of Figure 7, the conformance monitor occurs in two separate (independent) blocks: "Conformance Monitor [Gnd]" and "Conformance Monitor [Air]". A more detailed model would indicate that there are multiple ground controllers and as many as thousands of aircraft, each with its own conformance monitor. These monitors will likely be developed by independent technical teams and may contain different algorithms for determining conformance, different levels of sensitivity to nonconformance, different approaches to timing and updating of the software functions, and many other factors. Indeed, the conformance monitors *are* independent, as the TBO ConOps asserts. In addition to these technical factors, the individual Alert Parameters represent another source of independent behavior, as described above.

The independence assumptions of the individual conformance monitors is valid. However, the issues arise because the behavior of those agents who use these monitors is, in fact, quite *dependent*.

By explicitly modeling the interactions between control agents, as shown in Figure 7 and in equations (1)-(4), it can be shown by inspection that the consistency principle does not hold. Let $\rho_G(a, v)$ be the ground control agent's process model, which is informed and updated, in part, by the information delivered from the conformance monitor, $\mathcal{V}_{mG}$. Likewise, the air control agent's model, $\rho_A(a, v)$, contains the information $\mathcal{V}_{mA}$. As explained above, $\mathcal{V}_{mG}$ and $\mathcal{V}_{mA}$ have several sources of inconsistency, which are described qualitatively above, and therefore equation (5) does not hold.

The preceding analysis provides valuable information for the early development of complex transportation systems. Some of those implications are now explored. Analysis of completeness and safety-related responsibilities (questions 1-3 in Section III-B), which are not treated further in this paper, provides additional detail that should be used to support early engineering activities.

C. Supporting Early Engineering Activities

Hazard analyses or safety assessments should not be used to merely state whether the systems or components are "Safe" or "Unsafe". The results should drive the design of the system. Once the hazardous scenarios and causal factors have been identified, the key to safety-driven design is reasoning about (a) how to prevent the scenarios and (b) how to mitigate the scenarios if they occur.

1) Generating Safety Requirements: The following hazardous scenarios drive the identification of functional and safety-related requirements. The example scenarios contain a reference to the related hazard(s) and are then followed with example requirements and safety constraints. Requirements are based on both the included scenario as well as the analysis presented in the previous section. The requirements are represented as a nested, hierarchical list.

Hazardous Scenario 1: Ground does not issue a DE-conflicting command because it is unaware that the aircraft is not conforming or unaware that the flight crew begins taking action to "close" the trajectory. [→Hazard: Loss of separaton beween two or more aircraft]

R.1) Air component must notify ground of any changes to velocity (change in heading, airspeed, or altitude)
Rationale: Flight deck typically must request ANSP for a deviation from the filed flight plan, or from the current trajectory. This type of change to the velocity is actually due to the intent of *staying on* the trajectory, but it changes the aircraft's inertial state, or velocity in all three directions; see associated causal factors
Associated Causal Factors: due to the differences in conformance alerting models, the ANSP may be unaware of the need for change. The ANSP may have a different "Alert Parameter" setting in general, or may

have adjusted the setting due to other circumstances (e.g. alarm fatigue, managing other conflicts, etc)
Note: This requirement may be levied to either the flight crew or avionics; this design choice would require further analysis of potentially dysfunctional interactions

R.1.a) Flight deck must notify ANSP that changes to velocity are due to non-conformance
Rationale: this is not a "nominal" change, e.g. a change in direction that was part of the flight plan.

R.1.b) ...additional requirements related to R.1...

Hazardous Scenario 2: Flight crew does not conform to trajectory, pursuant to the ANSP conformance model. This non-conformance could arise even if the ANSP has instructed the aircraft to do so, and the flight deck has confirmed compliance. [→Hazard: Loss of separaton beween two or more aircraft]

R.2) ANSP must issue commands that result in the aircraft closing on the ANSP's own conformance model. That is, the command should directly result in velocity changes that cause the aircraft to enter into desired, protected volume. This clearance is heretofore called a "Close Conformance".
Rationale: ANSP must do more than notify the flight deck that it is not conforming and instruct it to close. This requirement also assumes that the ANSP model of the overall airspace (and thus the conformance model) takes precedence over the flight deck model
Associated Causal Factors: Flight deck may try to close the trajectory to its own model, or already believe that it is conforming (and thus believe it is complying with the instruction). See also causal factors in ←R.1

R.2.a) ANSP must be able to generate aircraft velocity changes that close the trajectory within TBD minutes (or TBD nmi).
Rationale: TBO ConOps is unclear about how ANSP will help the aircraft work to close trajectory. Refined requirements will deal with providing the ANSP feedback about the extent to which the aircraft does not conform, the direction and time, which can be used to calculate necessary changes.

R.2.b) ANSP-generated clearances used to close trajectories must not exceed aircraft flight envelope [→Hazard: aircraft enters uncontrolled state]

R.2.c) ANSP must be provided information to monitor the aircraft progress relative to its "Close Conformance" change request
Rationale: See "Associated Causal Factors" listed in ←R.2
Associated Causal Factors: e.g. ANSP "turns off" or changes Alert Parameter once flight deck has confirmed that it will comply

R.2.d) ...additional requirements related to R.2...

In addition to specifying requirements and constraints on behavior, STECA can be used to inform design trade studies or to revise and modify the concept itself via system architecture analysis.

2) Engineering Trade Studies: As Figure 1 illustrates, many important commitments are made early in the concept development process. STECA helps stakeholders to reason, in an increasingly formal way, about alternative approaches to achieving a system's objectives in terms of technical design and architecture.

For example, the previous section described potential inconsistencies that arise due to independent conformance alert parameters in the TBO ConOps. Several design alternatives exist. One decision might be to eliminate the "Alert Parameter" entirely[2] and require that the black box models of all conformance monitors—every aircraft and on the ground—are identical. That is, the sensitivity and underlying mathematical models of conformance would be required to be identical in such a design.

Alternatively, the control structure itself, which is one representation of the system architecture, could be modified. One control structure modification that could mitigate against inconsistency might involve eliminating the airborne conformance monitors. In Figure 7, conformance monitoring could be done centrally by the ground. Conformance would then be ensured by the instructions going from the ground to air. Perhaps less desirably with respect to consistency, conformance monitoring could be distributed to the air and eliminated from the ground. This latter solution may eliminate inconsistencies between the ground and air, but other issues arise because the ground control agent has a primary responsibility to assure separation between aircraft. Questions 2) and 3) on page 3 relate to analysis of safety responsibilites, and safety responsibilities of the TBO example have been covered in greater detail elsewhere [39].

V. Conclusions

This paper has presented a new approach, called systems-theoretic early concept analysis (STECA), for performing hazard analysis on a concept of operations and a safety-driven approach to concept development. STECA is based on systems- and control theory and its usefulness and practicality is demonstrated on an important upgrade to the air transport system, called Trajectory-Based Operations.

STECA is based on two basic steps. The first step involves recursively applying control-theoretic concepts using guide words, heuristics, and feedback control criteria to parse an existing ConOps document, resulting in the development of a hierarchical control model of the concept. The second step—analysis—consists of examining the resulting model with the explicit goals of identifying hazardous scenarios, information gaps, inconsistencies, and potential trade offs and alternatives. That is, the analysis identifies incompleteness or gaps in the control structure, assures that all safety-related responsibilities are accounted for, and identifies sources of uncoordinated or inconsistent control.

A. Contributions

The primary research contributions are in integrating safety earlier when developing complex transportation systems via rigorous, systematic methods. Specifically, STECA:

[2]This is not to suggest that there will be no alerts, only that the respective users cannot tune the sensitivity. See section IV-B.

- applies more rigor to the concept development process. The process described above is repeatable and can be applied by individuals to understand a transportation system in systems-theoretic terms. Alternatively, the process can be used by teams and working groups to build consensus on how the system should (and should not) behave, to allocate responsibilities to various actors, and to define the interactions between those actors;
- identifies a class of hazardous scenarios that have been difficult to find during concept development. Most accidents and incidents in modern, intelligent transportation systems arise due to factors that extend beyond component reliability, which is the focus of most efforts based on PHA techniques (see Table I). Accidents arise due to unsafe interactions among components, which include software and human operator behavior, and STECA provides a powerful way to identify these types of interactions;
- makes explicit the assumptions that are often undocumented or implicit during early concept generation. Because ConOps documents are typically developed by subject matter experts, many details that are obvious to a particular expert may seem obvious and thus go undocumented. The model development approach in this thesis forces many of these assumptions to be made explicit, and often the various subject matter experts who generate the ConOps actually make competing or inconsistent assumptions about various aspects of the system.

B. Future Work

There are many potential paths of future research that build upon this work. While this paper briefly described how the results of the analyses can be used to generate alternative designs and architectures, future work should demonstrate how these alternatives can be compared. Stakeholders identify potential tradeoffs or synergies, and tradeoffs could be made with respect to safety and/or extended to other system properties.

Tools should be generated to assist in the STECA process. Existing model-based systems engineering frameworks could be adapted or integrated into the systems- and control-theoretic processes described in Sections III, and IV.

Although this research focuses on the early phases of systems engineering (far left side of Figure 1), the framework has potential to applied to the very last phases of systems engineering (far right side of the figure). That is, when systems become operational, it is unfortunately necessary in some cases to perform accident and incident investigations. Accident reports typically share similar characteristics to Concept of Operations documents—they contain informal natural language text, use informal graphical depictions of events, and are often developed by committees comprised of potentially disparate views of the system. There exists an accident analysis process, called CAST (Casual Analysis using STAMP), based on the same systems- and control theory that has been successfully applied to accidents in a variety of domains [50], [51], [52], [53]. However, there is not yet a rigorous way to generate the necessary models from all the different sources of data associated with any major accident, and the methods presented in Section III represent a potential way to accomplish this goal.

Perhaps most importantly, it is important to investigate the extent to which STECA can be applied to other future transportation systems, including rail, road, water, and other air transportation concepts.

REFERENCES

[1] R. Bishop, *Intelligent vehicle technology and trends*. 2005.
[2] K. Dierkx, "The smarter railrail: An opportunity for the railroad industry," *IBM Global Business Services: Travel and Transportation*, 2009.
[3] J. Josey, "Intelligent infrastructure for next-generation rail systems," *Cognizant 20-20 Insights*, 2013.
[4] K. Toner, "Opportunities and challenges for technology research and development," *ATCA Technical Symposium, Atlantic City NJ.*, May 2011.
[5] J. Areddy and Y. Jie, "How chinas train tragedy unfolded," *China Realtime Report, The Wall Street Journal, China, available at: http://blogs.wsj.com/chinarealtime/2011/12/29/wenzhou*, vol. 80, 2011.
[6] C. Jensen, "G.M Steering issue pushes automaker's limits."
[7] B. Carey, "Application software bug delays american airlines flights," *Air Transport, AIN Online*, 2015.
[8] A. Scott and J. Menn, "Exclusive: Air traffic system failure caused by computer memory shortage," *Reuters*, 2014.
[9] B. Farmer, "Flights in chaos across uk as air traffic control computers fail: latest," *The Telegraph*, 2014.
[10] N. G. Leveson, "Software challenges in achieving space safety," 2009.
[11] F. Frola and C. Miller, "System safety in aircraft management," *Logistics Management Institute, Washington DC*, 1984.
[12] A. Strafaci, "What does BIM mean for civil engineers?," *CE News, Tranportation*, 2008.
[13] E. Crawley, O. de Weck, S. Eppinger, C. Magee, J. Moses, W. Seering, J. Schindall, D. Wallace, and D. Whitney, "The influence of architecture in engineering systems," in *1st Engineering Systems Symposium*, MIT Engineering Systems Division, Cambridge, Massachusetts, 2004.
[14] A. M. Ross and D. E. Hastings, "Assessing changeability in aerospace systems architecting and design using dynamic multi-attribute tradespace exploration," in *AIAA Space*, 2006.
[15] M. O'Neill, J. Dumont, T. Reynolds, and J. Hansman, "Methods for evaluating environmental and performance tradeoffs for air transportation systems," in *11th AIAA Aviation Technology, Integration and Operations Conference, Virginia Beach, VA, AIAA Paper*, vol. 6816, 2011.
[16] US DoD, *MIL-STD-882E, Department of Defense Standard Practice System Safety*. U.S. Department of Defense, 2012.
[17] S. J. Kapurch, *NASA Systems Engineering Handbook*. DIANE Publishing, 2010.
[18] FAA, *Safety Management System Manual*. Federal Aviation Administration Air Traffic Organization, 2008.
[19] SAE, *ARP-4754, Guidelines For Development Of Civil Aircraft and Systems, Revision A*. Society of Automotive Engineers, 2010. S-18.
[20] J. W. Vincoli, *Basic Guide to System Safety, Second Edition*. John Wiley & Sons, Inc., Hoboken, NJ, USA., 2005.
[21] JPDO, "Capability safety assessment of trajectory based operations v1.1," tech. rep., Joint Planning and Development Office Capability Safety Assessment Team, 2012.
[22] E. Harkleroad, A. Vela, J. Kuchar, B. Barnett, and R. Merchant-Bennett, "Atc-045 risk-based modeling to support nextgen concept assessment and validation," tech. rep., MIT Lincoln Laboratory & Federal Aviation Administration, 2013.
[23] H. Watson, "Bell telephone laboratories. launch control safety study," *Bell Telephone Laboratories, Murray Hill, NJ USA*, 1961.
[24] USDoD, *MIL-P-1629 - Procedures for performing a failure mode effect and critical analysis.* United States Department of Defense, 1949.
[25] W. E. Vesely, F. F. Goldberg, N. H. Roberts, and D. F. Haasl, "Fault tree handbook," tech. rep., DTIC Document, 1981.
[26] N. G. Leveson, *Safeware: System Safety and Computers*. Addison Wesley, 1995.
[27] N. G. Leveson, *Engineering a Safer World*. MIT Press, 2012.
[28] S. Dekker, *Ten questions about human error: A new view of human factors and system safety*. CRC Press, 2005.
[29] J. Rasmussen, "Risk management in a dynamic society: a modelling problem," *Safety science*, vol. 27, no. 2, pp. 183–213, 1997.

[30] D. D. Woods, L. J. Johannesen, R. I. Cook, and N. B. Sarter, *Behind human error*. Ashgate Publishing Company, 2010.
[31] R. R. Lutz and I. Carmen Mikulski, "Operational anomalies as a cause of safety-critical requirements evolution," *Journal of Systems and Software*, vol. 65, no. 2, pp. 155–161, 2003.
[32] N. Leveson, "A new accident model for engineering safer systems," *Safety Science*, vol. 42, no. 4, pp. 237–270, 2004.
[33] P. Checkland, *Systems thinking, systems practice: includes a 30-year retrospective*. John Wiley & Sons, Inc., 1999.
[34] J. McGregor, "Arcade game maker pedagogical product line: Concept of operations," *Version*, vol. 2, p. 2005, 2005.
[35] FAA, "Nextgen implementation plan," *Washington, DC, Available: http://www.faa.gov/nextgen/implementation/media/NextGen_Implementation_Plan_2013.pdf (accessed March 18, 2014)*, 2013.
[36] E. Cone, "The ugly history of tool development at the faa," *Baseline Magazine*, vol. 4, no. 9, 2002.
[37] U. S. G. A. Office, *Air traffic control: FAA's advanced automation system acquisition strategy is risky: report to the Secretary of Transportation*. The Office, 1986.
[38] W. R. Ashby, *An Introduction to Cybernetics*. Chapman & Hall Ltd., 1957.
[39] C. H. Fleming, *Safety-driven early concept analysis and development*. PhD thesis, Massachusetts Institute of Technology. Department of Aeronautics and Astronautics., 2015.
[40] M. D. Mesarovic and Y. Takahara, *General systems theory: mathematical foundations*, vol. 113. Academic Press New York, 1975.
[41] S. Skogestad, "Control structure design for complete chemical plants," *Computers & Chemical Engineering*, vol. 28, no. 1, pp. 219–234, 2004.
[42] M. Morari, Y. Arkun, and G. Stephanopoulos, "Studies in the synthesis of control structures for chemical processes: Part i: Formulation of the problem. process decomposition and the classification of the control tasks. analysis of the optimizing control structures," *AIChE Journal*, vol. 26, no. 2, pp. 220–232, 1980.
[43] N. G. Leveson, "Intent specifications: An approach to building human-centered specifications," *Software Engineering, IEEE Transactions on*, vol. 26, no. 1, pp. 15–35, 2000.
[44] H. A. Simon, "The structure of ill-structured problems," in *Models of discovery*, pp. 304–325, Springer, 1977.
[45] J. Rasmussen, "Information processing and human machine interaction: An approach to cognitive engineering. 1986," *New York. North-Holland*, vol. 1, no. 2, p. 3, 1986.
[46] JPDO, "Joint planning and development office concept of operations for the next generation air transportation system," *Version 3.2 (September 30, 2010)*, 2010.
[47] JPDO, "JPDO Trajectory-Based Operations (TBO) study team report," tech. rep., Joint Planning and Development Office, 2011.
[48] N. Leveson, "Completeness in formal specification language design for process-control systems," in *Proceedings of the third workshop on Formal methods in software practice*, pp. 75–87, ACM, 2000.
[49] N. Kheir, *Systems modeling and computer simulation*, vol. 94. CRC Press, 1995.
[50] A. Dong, *Application of CAST and STPA to railroad safety in China*. PhD thesis, Massachusetts Institute of Technology, 2012.
[51] J. J. P. Hickey, *A system theoretic safety analysis of US Coast Guard aviation mishap involving CG-6505*. PhD thesis, Massachusetts Institute of Technology, 2012.
[52] M. B. Spencer, *Engineering financial safety: a system-theoretic case study from the financial crisis*. PhD thesis, Massachusetts Institute of Technology, 2012.
[53] R. Hosse, D. Spiegel, J. Welte, E. Schnieder, *et al.*, "Integration of petri nets into stamp/cast on the example of wenzhou 7.23 accident," in *Control and Automation Theory for Transportation Applications*, vol. 1, pp. 65–70, 2013.

Cody H. Fleming Cody Fleming started as Assistant Professor of Systems and Information Engineering at the University of Virginia in August 2015. He obtained a doctoral degree in Aeronautics and Astronautics at the Massachusetts Institute of Technology in January 2015. He holds a BS degree in Mechanical Engineering from Hope College and masters in Civil Engineering from MIT. Prior to returning to MIT, Cody spent 5 years working in space system development for various government projects.

Nancy Leveson Dr. Nancy Leveson is Professor of Aeronautics and Astronautics at MIT and a member of the National Academy of Engineering. She has authored over 200 published papers and two books, Safeware: System Safety and Computers (1995) and Engineering a Safer World (2012). She conducts research on all aspects of system safety and system engineering, including requirements, design, operations, management and social aspects of safety-critical systems.

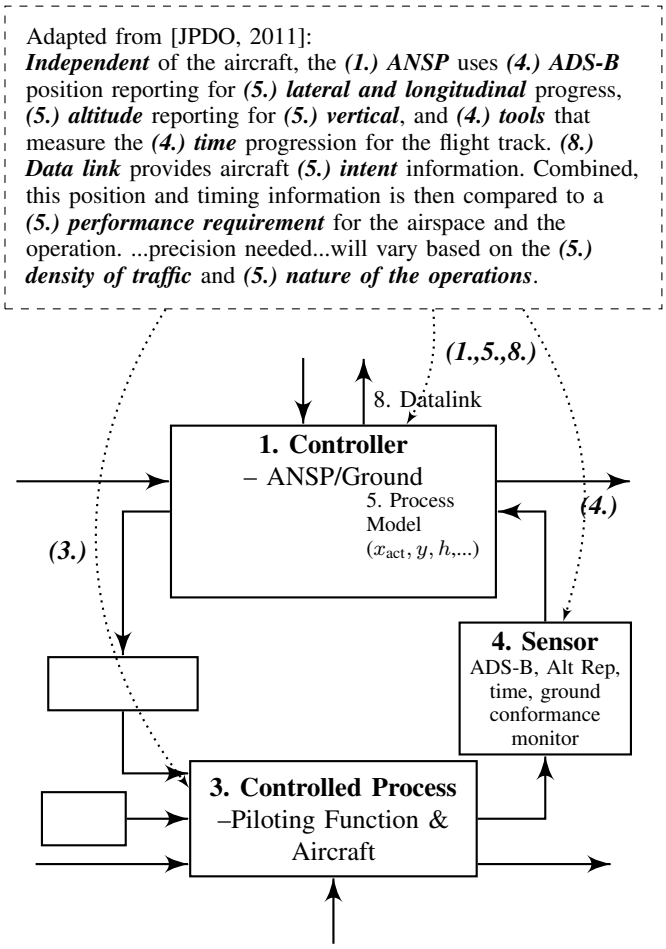

Figure 6. Graphical Control Model of Ground Conformance Monitor