Development of a Safety Knowledge Base

Philip R. Lewis
Chief Engineer – J&P Technologies

Abstract

The salient features of the proposed Safety Knowledge Base, an integrated toolset for tying together Hazard causes and controls to development requirements and implementation, are presented. The need for such a tool, its logical basis of its construction, and scope of applicability are discussed.

INTRODUCTION

The Safety Knowledge Base, upon implementation, will provide the ability to serve in Project Concept Development by tying major capabilities to generic hazards, in Requirements Definition by tying the lower tiered requirements to specific hazards and their causes, in Development to verify design safety compliance with the implemented hazard controls, and support anomaly resolution and monitoring of safety controls in Operations. Prior JSC programs have used distinct tools, typically each with its own database, for each program phase and application. These separate tools and datasets do not have common data formats and limited scope of applicability.

Current space flight projects rely heavily on the complex interactions between software and hardware to accomplish the intended mission. In the past, hardware and software were handled as separate entities such that separate safety requirements could be levied on each. One result of this approach is that it allows the pursuit of mutually exclusive design solutions that are only revealed in the last phases of the project's development life cycle or in operations. Another result is that specific instances of software performing hazard control functions are not adequately captured in the lower level software requirements specification, and therefore may not be adequately verified. The Safety Knowledge Base process brings software and hardware interaction to the forefront allowing hazard controls to be analyzed as complete control loops. Lacking this veiw, safety assessments and anomaly resolution conducted during operations require system expertise and data searches to obtain the needed "as built" data. This process is not only labor intensive, but requires an in-depth knowledge of the system and subsystem interactions in order to be performed effectively.

The Safety Knowledge Base captures this data and its configuration/functional relationships and provides different "views" of system configuration and safety design data depending on the user's need. This knowledge will provide greater utility to both the verification and operation of the system, providing a user-friendly capability to a more diverse set of users. In this way greater utility is provided and greater efficiency is realized.

The Safety Knowledge Base will also serve as source data for knowledge mining in support of the "Design For Safety" initiatives. The lessons of the past designs are captured and robustly documented for future generations of engineers to mine and learn from. The tool will be available to serve other programs such as CTV, Shuttle and 2nd Generation Reusable Launch Vehicle. In its final form the tool will provide knowledge in a variety of formats. It will interface with COTS products to generate Fault Trees, RBD's, and serve as source data for other tools in performing system and mission simulations.

An integrated tool providing different "views" of system configuration and safety design data will significantly enhance the efficiency of resources devoted to safety and mission assurance activities as well as provide some immunity to loss of corporate knowledge.

ORIGINS

The International Space Station is growing more complex with each mission and the number of software Problem Reports (PRs) grows with the addition of each new module and software release. Determining the severity, or "Safety" implications, of a software PR is, or should be, determined from the applicable Hazard Report, but attempting to do this revealed some shortcomings in the ISS Hazard Reports.

The present ISS Safety Review Process captures Hazards, Hazard Causes, and their Controls and documents them in Hazard Reports. In their present form they lack implementation detail, most often they simply call out an end effector, e.g. an isolation valve, as a control, ignoring the software that controls and powers the effector, and, as a result, are ineffectively used in the performance of safety assessments and anomaly resolution conducted during operations. Additionally, this lack of information about the software controlling the end effector in the Hazard Report did not allow the software safety engineers to present an informed defense of high PR severity ratings before the Control Boards. In order to provide the software engineers with a data set that could be readily used in evaluating PR's, a Software Hazard Mapping Task was undertaken. This task was to map the various CSC/s/CSCI's to the hazard causes and controls as documented in the hazard reports. It was during this task that the true complexity and extent of the hardware/software interaction was made apparent. While mapping the hardware/software required to implement the hazard controls, it was realized the usefulness of the data extended far beyond its original software PR evaluation function, and was actually applicable to the entire project life cycle. The technique used to gather and store the data was in essence building an electronic schematic of the implemented hazard controls with all the complex subsystem interactions in place.

IMPLEMENTATION BASIS

The following describes the logical relationship between project capabilities and hazards. Specifically, it shows that generic and specific hazards are a subset of both the selected project capabilities and the requirements allocated from those capabilities. The Safety Knowledge Base will capture this interrelationship as well as the specific details of the hazard controls. This aspect of the Safety Knowledge Base not only provides valuable information to the current project, but also provides a complete knowledge capture for use by future projects.

A project starts with a defined set of capabilities, C_i ;

The capabilities are allocated to Level A requirements: $R_{i,j}$, *i.e.*

$$C_i = \underset{j=1}{\overset{x}{\text{AND}}} R_{i,j}$$

The Level A requirements are allocated to Level B requirements: $R_{i,j,k}$, *i.e.*

$$R_{i,j} = \underset{k=1}{\overset{y}{\text{AND}}} R_{i,j,k}$$

The Level B requirements are allocated to Level C requirements: $R_{i,j,k,l}$

$$R_{1,j,k} = \underset{l=1}{\overset{z}{\text{AND}}} R_{1,j,k,l}$$

The Level C requirements provide the basis of implementation.

If the capabilities set is complete and the lower level requirements properly allocated, then

Generic Hazards arise from the negated capabilities: $\{\sim C_i\}$, i.e., failure to provide a specific capability is hazardous.
Specific Hazards and hazardous requirements (harbors a hazard) are then developed from the allocation of the capabilities as a subset of the negated allocated requirements *e.g.*

If $\sim C_3$ is identified as a Generic Hazard then the Specific Hazards is given by

$\overset{x}{\underset{j=1}{OR}} \sim R_{3,j}$, or lower level decomposition, and the causes will be a subset of $\left\{\sim R_{3,j}\right\}$, or lower level decomposition.

From this the hazardous requirements and implementation details are a subset of the further allocation of the identified hazardous subsets.

The safety engineer's task becomes one of examining the negated capabilities to determine hazards or determining that a capability is missing because a known hazard does not derive from the negated capabilities. As the requirements allocation develops the safety engineer need only examine those requirements whose parent has been determined to harbor a hazard. Ultimately the trace allows one to know where all aspects of a hazard control are implemented.

SAFETY KNOWLEDGE BASE TOOL CONCEPT OF OPERATION

The Safety Knowledge Base tool and supporting products are intended to enhance the overall safety of complex projects. This tool and supporting implementation process are intended to not only capture data for Design for Safety Initiatives, but also supplement and enhance the current safety processes used throughout NASA. The heart of this approach is the centralized body of knowledge related to space flight safety. It will capture the knowledge of a given project from concept definition through decommissioning. This knowledge will assist in current development projects as well as provide an invaluable resource to future complex undertakings.

The Safety Knowledge Base will capture the initial intent of a project developed during the project concept phase and trace it throughout the project lifecycle. One of the primary focuses of this approach is to bring the software used to control hazards up to the same level of robustness as the hardware used to control hazards. This is achieved by identifying the software safety aspects of the hazard control early in the development and requirements definition processes. All too often we spend more time on hardware details of a hazard control (i.e. what seal material and how many seals are used) that we spend in evaluating the actual control loop that implements the hazard control. The Safety Knowledge Base process proposes to close this gap by requiring a detail definition of the as built control loop along with all the supporting utility functions. Figure 1 presents a general concept of the Safety Knowledge Base.

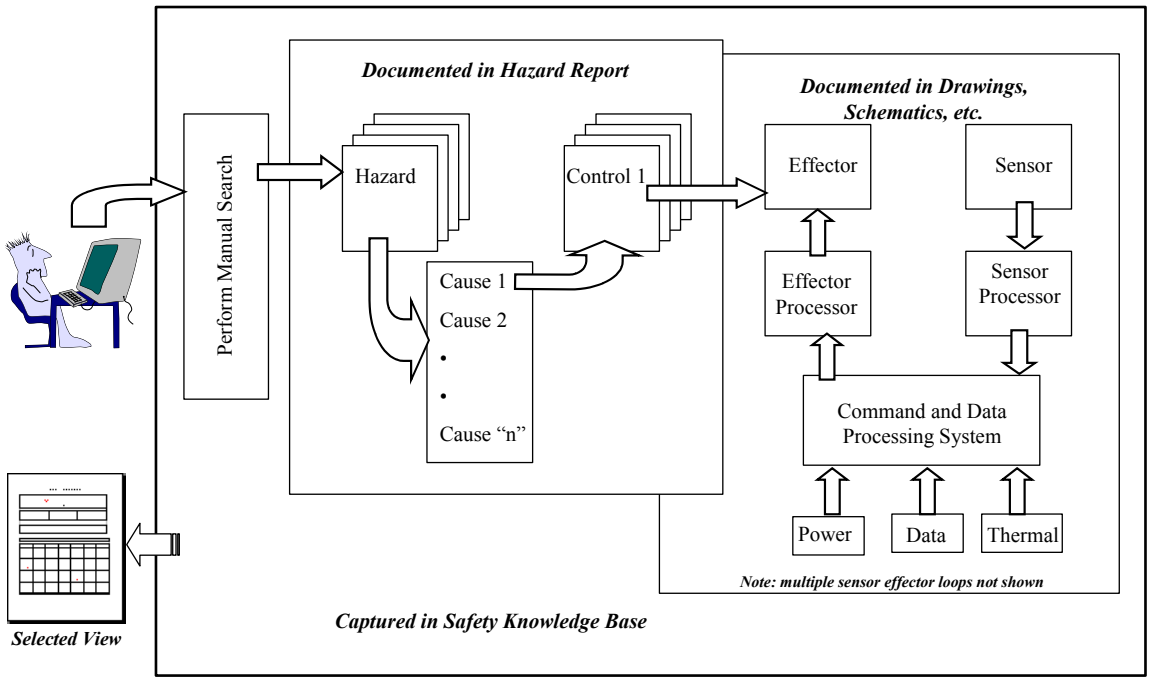

Figure 1: Safety Knowledge Data Location and Hazard Control Process

PROPOSED IMPLEMENTATION GUIDELINES AND PROCEDURES

The following is a set of guidelines that can be applied during the projects life cycle. As the Safety Knowledge Base is populated, its usability in the earliest stages of project formulation also grows. Once sufficiently populated, users during the project concept phase will have access to potential hazards and their controls simply by searching the Safety Knowledge Base for hazards linked to the major capabilities identified in projects of the past. Potential project risks are identified during the earliest phase of project definition; as well as risk mitigations techniques used in the past that maybe applicable to the current project.

CONCEPT DEVELOPMENT PHASE

A Preliminary Hazard Analysis (PHA) should be performed during the conceptual stage. Scheduled updates may be performed as design development progresses to provide a basis for establishment of design safety requirements early in the program. Ideally, this also would eliminate the possibility of costly design changes later in the program.

The generic hazards are developed from the major capabilities defined during the concept development phase of a projects life cycle. The Safety Knowledge Base ties these major capabilities to the generic hazards allowing full requirement traceability to be established. Future projects will be able to extract these hazards of the past from the Safety Knowledge Base based upon selecting proposed major capabilities.

REQUIREMENTS DEFINITION PHASE

In this phase of a projects lifecycle, the major capabilities (or mission objectives and goals) are decomposed (or allocated) to Level A requirements. These requirements are then further decomposed, as required (Level B, C, etc.), until all the requirements are verifiable via test, analysis, or other accepted method.

As the requirements are allocated, the specific hazards can be identified along with their causes and potential controls. The Safety Knowledge Base can assist in safety requirement traceability by tracking the allocated requirements from major capability development, through requirement allocation, and finally capturing the design details of the implemented hazard control.

The PHA must be updated during this phase to identify the specific hazards, their causes, and potential controls. Each specific hazard must be linked to its' respective generic hazard and lowest tiered requirement thus providing traceability back to the major capabilities. This aspect of the Safety Knowledge Base provides future programs with insight into the hazards and/or applicable safety requirements associated with major project capabilities during the earliest phases of the design process. In addition, full safety requirement traceability is obtained.

Once the potential hazard controls are identified, they should be sorted into one of the following groups:

1) achieved by design alone, e.g. material selection,
2) purely manual control, e.g., close a valve manually,
3) automatic notification to perform a manual action, e.g. alarms identify necessary actions,
4) automated execution of manual input, e.g. user issues command to close valve,
5) purely automated system response, e.g. smoke detector senses fire and terminates power to the area.

Categories 3, 4, and 5 involve software in the control loops, which should be identified in the Hazard Report. The software requirements associated with each "non-manual" hazard control can be identified.

DEVELOPMENT PHASE

This is the most crucial phase of the Safety Knowledge Base implementation process. It is during this phase of the project that the details of the implemented hazard controls mature and must be added to the Safety Knowledge Base. Implementation of this technique post development can be an almost insurmountable task due to the sheer volume of data and its complex interactions.

The Safety Knowledge Base process will supplement the normal safety review processes currently in use throughout NASA. It does not require a significant increase in effort since the same data is essentially required to approve a hazard report. It does on the other hand, require a systematic approach to the data gathering and reporting process be implemented such that both the top level hazard control sensor effector loops and their required supporting utilities (i.e. power, data, thermal) are sufficiently defined such that all interfaces and must work functions are identified.

For example, the hazard report may define the control for a depress hazard to be closing of a specific valve as well as identifying that the close valve command is issued by a given processor. Figure 2 shows an excerpt from an ISS Hazard Report for the depress hazard.

Module	Control 3.4	Control 3.5
2. USOS	Verification: An analysis is performed to show egress can be accomplished within 3 minutes. The assessment includes planned drag through items… Closed to VTL.	Intermodule Ventilation (IMV) valves, on the USOS will be closed automatically upon detection of a rapid depressurization. IMV valves can also be closed by a manual override. C&C MDM sends isolate commands to Lab, Node 1, & airlock; cabin fan/CCAA off command for Node, Lab, and Airlock; and automatic command of MCA to standby, and/or SDS isolation…….. …..Fans off (<30 sec), valve closed (<1min), between all modules (except A/L in Campout, Node 1 stbd. Valves open) All external valves commanded closed <30s.

Figure 2: Hazard Report Excerpt for ISS Depress

What is missing from this Hazard Report are the actual details of the sensor effector loop required to perform this task. This would not be adequate information to truly verify the control nor is it adequate for implementation into the Safety Knowledge Base. What would be required is identification of what sensor is used to detect the event, the processor that controls/reads that sensor along with its supporting utility needs (i.e. power, data thermal), what data path is taken to the next higher processor which may actually issue the close valve command (and its supporting utilities), and so forth all the way to the actual processor, or firmware controller, that actually closes the valve. The software, which actually executes on each processor, must also be identified and should be traceable to a software requirement that actually states that this requirement is associated with a given hazard control. Figure 3 presents a different view of the same hazard as it relates to closing the Vacuum Resource System (VRS) Vent Valve, which is an external valve located in the Lab and requires automatic closure upon detection of a depress event.

As can bee seen from Figure 3, this process is in essence building an electronic schematic of the as built hazard control along with the complex interactions of the supporting utility functions. It is in this way that a complete verification of the hazard control is achieved and documented for future use.

The change and problem resolution processes associated with both the development and operational phases of a project are greatly enhanced by implementing the Safety Knowledge Base process. When a software problem report is reviewed, the reviewer can quickly identify if the problem affects software safety simply by requesting a search of the Safety Knowledge Base, determining what hazards are actually controlled by that piece of software. The same approach applies to hardware, but it is the software side of the hazard control that is usually harder to trace and interpret.

ISS-ECL-0210-8A
Depressurization of ISS

Cause 1: Pressure loss (general)
Control 1: Response to a rapid dp/dt results in automatic isolation
a) Close VRS Vent Valve

Sensor	Processor	Effector
Lab PCA	CCS	VRS Vent Valve

Data Flow

Lab PCA → INTSYS → CCS → INTSYS → LSYS3 →VS-LCA → VRS Vent Valve

Electrical Power Chain

Lab PCA	INTSYS (INT-1)	CCS (CCS-2)	INTSYS	LSYS3	VS-LCA	VRS Vent Valve
	RPCM: LAF1-1B-A On Bus: LB SYS-LAB-1 Controlled by: INTSYS-Pri	RPCM: LAF5-2B-A On Bus: LB SYS-LAB-2 Controlled by: INTSYS-Pri	RPCM: LAF1-1B-A On Bus: LB SYS-LAB-1 Controlled by: INTSYS-Pri			
RPCM: LAFWD-1B-E On Bus: LB SEPS-HAB-14 Controlled By: INTSYS-Pri	RPCM: LAFWD-1B-D On Bus: LB SYS-LAB-1 Controlled by: INTSYS-Pri	RPCM: LAAFT-2B-C On Bus: LB SYS-LAB-2 Controlled by: INTSYS-Pri	RPCM: LAFWD-1B-D On Bus: LB SYS-LAB-1 Controlled by: INTSYS-Pri	RPCM: LAAFT-2B-A23 On Bus: LB SEPS-N2-23 Controlled by: INTSYS-Pri	RPCM - LAFWD-1B-F on Bus LB SEPS-N2-14 Controlled by INTSYS-Pri	RPCM - LAFWD-1B-F on Bus LB SEPS-N2-14 Controlled by INTSYS-Pri
DDCU: LAFWD- 1B On Bus CB GNC-1 (RS BUS 7) Controlled by CCS	DDCU: LAFWD- 1B On Bus CB GNC-1 (RS BUS 7) Controlled by CCS	DDCU: LAAFT-2B On Bus: CB GNC-2 (RS BUS 8) Controlled by: CCS Primary	DDCU: LAFWD- 1B On Bus CB GNC-1 (RS BUS 7) Controlled by CCS	DDCU: LAAFT-2B On bus: CB GNC-2 (RS BUS 8) Controlled by: C&C Pri	DDCU: LAFWD- 1B On Bus CB GNC-1 (RS BUS 7) Controlled by CCS	DDCU: LAFWD- 1B On Bus CB GNC-1 (RS BUS 7) Controlled by CCS
DCSU: P6LWR-4B On Bus: UB PVB-24-1 Controlled By PVCU-2B/4B	DCSU: P6LWR-4B On Bus: UB PVB-24-1 Controlled By PVCU-2B/4B	DCSU: P6UPR-2B On Bus: UB PVB-24-2 Controlled by: PVCU-2B/4B	DCSU: P6LWR-4B On Bus: UB PVB-24-1 Controlled By PVCU-2B/4B	DCSU: P6UPR-2B On bus: UB PVB-24-2 Controlled by: PVCU-2B/4B	DCSU: P6LWR-4B On Bus: UB PVB-24-1 Controlled By PVCU-2B/4B	DCSU: P6LWR-4B On Bus: UB PVB-24-1 Controlled By PVCU-2B/4B
Power Channel: 4B	Power Channel: 4B	Power Channel: 2B	Power Channel: 4B	Power Channel: 2B	Power Channel: 4B	Power Channel: 4B

Figure 3: Safety Knowledge Base "Hazard Control View"

OPERATIONS PHASE

It is in the operational phase of a major project or program that this process really begins to pay off. Not only have the hazard controls been verified and tested to the greatest extent possible, the resources required to trouble shoot on-orbit anomalies are greatly reduced. The effects of on-orbit failures are quickly identified as they relate to hazards. The console operator can quickly determine the state of any hazard control by entering into the system the ID of the failed hardware and requesting a hazard summary report. In addition, system effects related to planned maintenance

activities are quickly and accurately identified. For example, if a maintenance activity requires shutting down a power source, the effects as they relate to hazard controls are readily available, since all hazard controls that rely on that power source in their sensor effector loops are already identified in the Safety Knowledge Base. This same scenario also applies to software anomalies.

SAFETY KNOWLEDGE BASE CHECKLIST

Figure 4 presents a preliminary checklist that can be used to ensure a new project or program is compatible to the fullest extent possible with the Safety Knowledge Base process. The "Procedures and Guidelines" section of this report along with the Safety Knowledge Base Schema (not supplied) should be used as additional support data to assure the compatibility of a new development project with the Safety Knowledge Base Process.

Phase	Task	Data Requirements	Product/Benefit
Concept Development	Define project's major capabilities	Lessons learned from previous projects	
	Perform Preliminary Hazard Analysis (PHA) on Defined Capabilities to identify generic hazards	Define relationship between selected capabilities and generic hazards	Traceability between capabilities and generic hazards
Requirement Definition	Allocate capabilities to Level A requirements	Relationship between capabilities, requirements, and generic hazards. Hazards are also used to establish top safety requirements.	Traceability between requirements and hazards.
	Further allocate requirements until verifiable.	Relationship between each requirement level.	Maintain requirement traceability
	Update PHA	Specific hazards, causes and controls linked to lowest level traceable requirement	Provides link to verifiable requirements maintaining traceability
	Sort potential hazard controls identifying those with potential S/W or firmware interface (non-manual)	List of all potential hazard controls.	Hazard controls that require software interface are identified, thus software requirements with safety impact can now be allocated.
Development	Perform formal hazard analysis	PHA, design details	Specific hazards, causes and controls are identified and linked to generic hazards.
	Perform hazard control verification	Data supporting the definition of the complete sensor effector loops including utility function architecture and data flow requirements	Identified further detail and location of actual software interfaces. .
	Identify all instances	Data flow and supporting	Each instance of software

Phase	Task	Data Requirements	Product/Benefit
	of software interaction with the control loop	utilities processor requirements	should be linked to a specific software development requirement paragraph number.
Operations	Functional state of the hazard control should be verified	Downlink data and other on-orbit anomaly reports	Active hazard control can be identified, thus the state of the vehicle maintained.

Figure 4: Safety Knowledge Base Checklist

DEFINITIONS

Fault Tree Analysis: A graphic representation of a logical thought process used to analyze an undesired event. Using inductive logic, all causes that can lead to the undesired, or top, event are listed on an inverted tree. These causes then become events for which causes are listed. This analysis is continued to determine all of the events and combinations of events that can lead to the top event.

Preliminary Hazard Analysis: The identification of generic hazards, basic technical safety requirements, hazard controls, evaluation of alternative concepts, and documentation of initial safety risk associated with the proposed concept or system.

Hazard Analysis: The determination of potential sources of danger and recommended resolutions in a timely manner for those conditions found in either the hardware/software systems, the person-machine relationship, or both, which cause loss of personnel capability, loss of system, or loss of life or injury to the public.

Safety: Freedom from chance of injury or loss of personnel, equipment or property.

Safety Analysis: A systematic and orderly process for the acquisition and evaluation of specific information pertaining to the safety of a system.

Safety Assurance: The review and assessment of equipment provider safety analysis and hazard reports for completeness and compliance to safety requirements.

Safety Data Package (SDP): The compilation of all related technical data and safety analysis in support of a specific flight or group of flights.

Safety Review: An incremental process that focuses on: assuring that all hazards and hazard causes inherent in the design and operations have been identified; evaluating the means to control the risk; and assessing the methods identified to verify all hazard controls.

System Safety: The optimum degree of risk management within the constraints of operational effectiveness, time and cost attained through the application of management and engineering principles throughout all phases of a program.

Software Safety: The application of the disciplines of system safety engineering techniques throughout the software life cycle to ensure that the software takes positive measures to enhance system safety and that errors that could reduce system safety have been eliminated or controlled to an acceptable level of risk.

REFERENCES AND APPLICABLE DOCUMENTS

SSP 30309, *International Space Station Program Safety Analysis and Risk Assessment Requirements Document*

SSP 30599, *International Space Station Program Safety Review Process*

SSP 50021, *International Space Station Program Safety Requirements Document*

SSP 50038, *International Space Station Program Computer Based Control System Safety Requirements*

SSP 41000, *System Specification for the International Space Station*

NHB 1700.1(V1), *NASA Safety Policy and Requirements Document*

NASA-SDT-8719.13A, *Software Safety NASA Technical Standard*

NPG-7120.5A, *NASA Program and Project Management Processes and Requirements*