The Orion GN&C Data-Driven Flight Software Architecture for Automated Sequencing and Fault Recovery

Ellis King
17629 El Camino Real, Houston TX 77058, 281-212-1107, eking@draper.com

Jeremy Hart
Houston, TX 77058, 281-483-0001, jeremy.j.hart@nasa.gov

Ryan Odegard
17629 El Camino Real, Houston TX 77058, 281-212-1140, rodegard@draper.com

The Orion Crew Exploration Vehicle (CEV) is being designed to include significantly more automation capability than either the Space Shuttle or the International Space Station (ISS). In particular, the vehicle flight software has requirements to accommodate increasingly automated missions throughout all phases of flight. A data-driven flight software architecture will provide an evolvable automation capability to sequence through Guidance, Navigation & Control (GN&C) flight software modes and configurations while maintaining the required flexibility and human control over the automation. This flexibility is a key aspect needed to address the maturation of operational concepts, to permit ground and crew operators to gain trust in the system and mitigate unpredictability in human spaceflight. To allow for mission flexibility and reconfigurability, a data driven approach is being taken to load the mission event plan as well as the flight software artifacts associated with the GN&C subsystem. A database of GN&C level sequencing data is presented which manages and tracks the mission specific and algorithm parameters to provide a capability to schedule GN&C events within mission segments. The flight software data schema for performing automated mission sequencing is presented with a concept of operations for interactions with ground and onboard crew members. A prototype architecture for fault identification, isolation and recovery interactions with the automation software is presented and discussed as a forward work item.

TABLE OF CONTENTS

1. INTRODUCTION

As compared to NASA's previous human spaceflight programs, the design of the Orion vehicle (Figure 1) allows for the astronaut crewmembers to take on a more supervisory role in the configuration of the flight software. This has been accomplished by designing the Orion vehicle to include capabilities to automatically make hardware and software reconfigurations according to predefined sequences resident onboard the spacecraft. The design requirements for automated functionality and represent a departure from the limited automation in the software on-board the Space Shuttle with the intent of creating increased automation capabilities and reducing the burden of manual configuration on the crew [2]. Astronauts aboard the Space Shuttle are required to manually configure spacecraft hardware and software via command for even the most routine and pre-defined events. The Orion spacecraft is designed to include automated functionality to perform this type of configuration allowing the role of the crew to shift from primarily manual configuration to monitoring and situational awareness of pre-defined events. These events are implemented in the Orion design as sequences of parameters that specify the configuration of hardware and software components for a given mission plan.

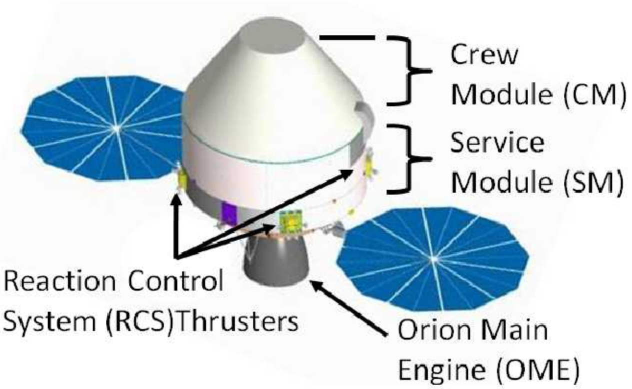


Figure 1: The Orion Crew Exploration Vehicle (CEV)

The choice to implement the sequences of automated configurations via parameters is to allow the software to be versatile enough to accommodate changes to the mission plan needed to safely adapt to the often dynamic nature of human spaceflight. When unexpected contingencies occur the design must allow changes to the pre-defined configurations that are no longer valid. The alternative of hard-coding automated capabilities would also be too brittle to allow for changes in crew and operational preferences that change and mature over time. This flexibility is critical since Orion is being designed to operate for the next 30 years. Over the lifetime of the Orion project the operational preferences and even missions themselves will continue to mature and evolve. A flexible and 'data-driven' architecture specified via parameters allows for many different types of missions and operator interaction, but also presents unique design challenges.

The following sections describe how the data-driven configuration items are created and managed, in particular for the GN&C subsystem. This paper represents aspects of the Orion design as baselined at the Preliminary Design Review (PDR) in the summer of 2009. The design will continue to mature and undergo development as the Orion program proceeds towards the Critical Design Review (CDR).

The first section describes the Orion vehicle-level mission sequencing handled by the Timeline Management (TM) software, which is responsible for coordinating the Orion subsystems. The Orion mission sequencing is important background to understand how GN&C is coordinated with the other subsystems for both nominal operations and for fault recovery. The next section illustrates how the GN&C software architecture relates to the configuration and automated sequences. This section includes an example of the typical sequence of events involved in the execution of an on-orbit burn maneuver. The example includes an in-depth description of how the individual software algorithms are grouped, activated for execution, and configured. After these sections the reader will have the background necessary to understand the data-driven parameters described in the following sections. In particular, the hierarchy of data decomposed into Mission Phases, Mission Segments, GN&C Activities and GN&C Modes, referred to as PSAM.

Following the description of the GN&C software architecture, the PSAM schema section presents the data schema for the data-driven artifacts used to configure the GN&C software. This schema follows the hierarchy mentioned above and also specifies parameters that each of the algorithms will use during execution. The overview of the PSAM schema includes details of key design features that are important to the overall goal of data-driven mission sequencing. After the description of the schema, the next section describes the PSAM database tool that can be used to create and manage the GN&C configuration sequences. This tool serves the critical role of helping to manage, visualize, and understand the data-driven parameters both during development and throughout the life of the program.

Forward work is then discussed, including the preliminary designs for the interactions between GN&C and external Vehicle Systems Management (VSM) software. This includes Fault Detection, Isolation, and Recovery (FDIR) for both Orion-level and GN&C-level faults. In addition forward work design decisions are presented which are being considered as display and control software requirements are flowed into the GN&C sequencer designs. Finally, conclusions summarize the important and novel aspects of the Orion GN&C design that have been addressed in this paper.

2. Orion Mission Sequencing

This section describes the vehicle-level mission sequencing hierarchy of the Orion TM Flight Software as it relates to the Orion Subsystems, including the GN&C subsystem. Each of level of the mission sequencing hierarchy is implemented via data-driven parameters.

Orion Vehicle-Level Sequencing

The responsibility for sequencing and coordinating Orion mission events is distributed among the TM software and the vehicle subsystems. The TM software is responsible for the overall mission timeline and coordination of the Orion subsystems, including GN&C. This knowledge of the overall timeline is based on a sequence of Mission Phases and Mission Segments, which is designed and tested on the ground prior to flight and loaded onto the vehicle as data-driven parameters.

Mission Phases and Mission Segments represent the high-level and intermediate-level decompositions of the overall mission timeline, respectively. The Mission Phases represent the major operational portions of the timeline, such as Ascent, Low-Earth Orbit (LEO) Configuration, Entry, etc. Each Mission Phase contains multiple Mission Segments, which correspond to the major events that will occur during that Mission Phase. The Mission Phases and Mission Segments are used to coordinate the configuration of the Orion vehicle subsystems. The current Mission Phase and Mission Segment are communicated to and used by the Orion Subsystems to provide knowledge of the point in the mission plan. For example, during the Ascent Mission Phase the subsystem configuration will change in response to the transition from the First Stage Mission Segment to the Second Stage Mission Segment. In some cases the subsystem response to the current Mission Phase and Mission Segment will be internal sequencing, as is often the case for GN&C as detail in the following section. Figure 2 shows an example decomposition of an Orion mission to the International Space Station (ISS). In this example the current Mission Segment is designed to perform an orbital plane change burn, called 'NPC Burn'. The Mission Phase and Mission Segment communicated to the Orion subsystem software will result in all of the necessary configurations and subsystem sequencing to execute that burn event. Each subsystem will provide a 'Segment Transition Indication' when it has completed the necessary mission objectives for that Mission Segment. The Mission Phases and Mission Segments are then sequenced by TM based on mission elapsed time and the Segment Transition Indications provided by the Orion subsystems.

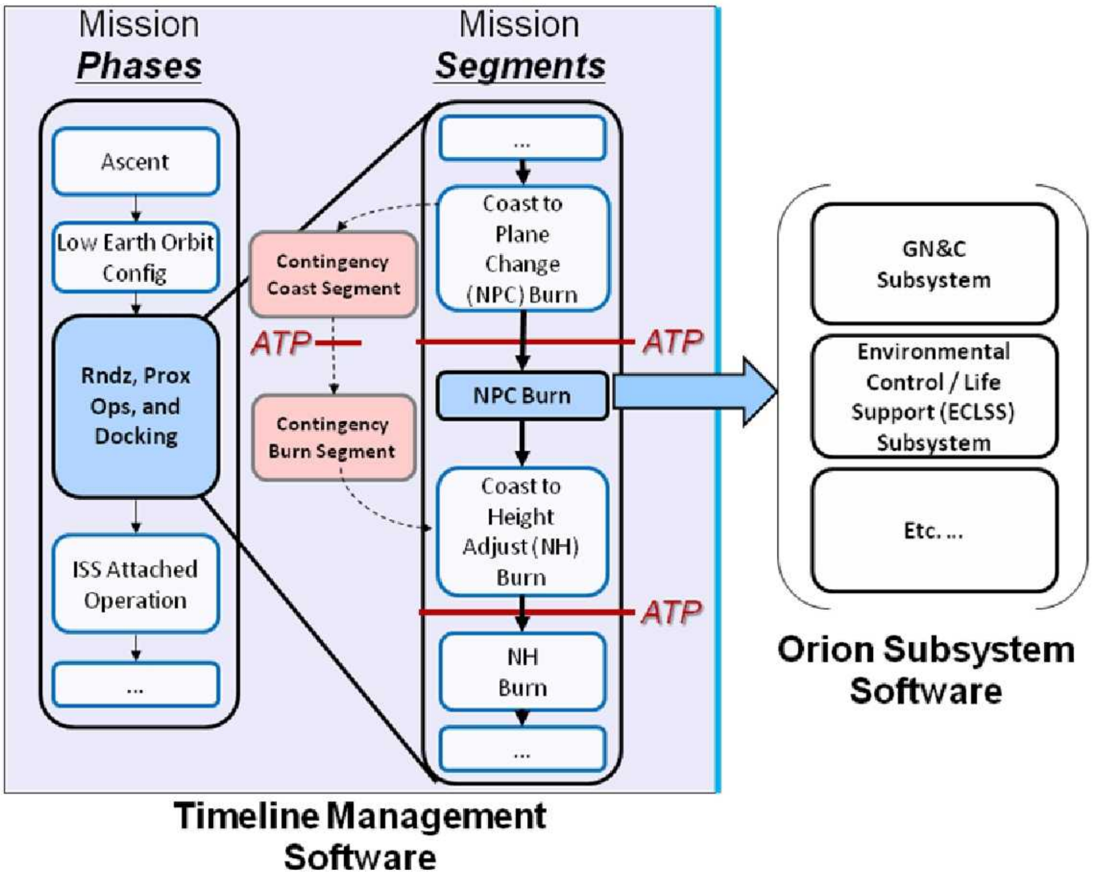

Mission Phases
Mission Segments
Ascent
Low Earth Orbit Config
Rndz, Prox Ops, and Docking
ISS Attached Operation
...
Contingency Coast Segment
Contingency Burn Segment
ATP
...
Coast to Plane Change (NPC) Burn
ATP
NPC Burn
Coast to Height Adjust (NH) Burn
ATP
NH Burn
...
GN&C Subsystem
Environmental Control / Life Support (ECLSS) Subsystem
Etc. ...
Orion Subsystem Software
Timeline Management Software

Figure 2: This figure shows the Orion mission sequencing hierarchy as it is decomposed between the Timeline Management software and GN&C Subsystem software.

Another aspect of the Orion mission sequencing design is that the Mission Segments can be configured to require a crew- or ground operator-issued Authority-To-Proceed (ATP). If an ATP is required, a ground or crew operator must issue a specific command before a transition to the next Mission Segment will be permitted in the flight software. This mechanism is used as a gate to provide control over automated functionality of the vehicle. Therefore, an ATP is often desired prior to the start of a certain critical mission event. [2] This type of operator interaction is also implemented using a data-driven approach to allow the required mission flexibility.

An example path for fault recovery is also shown in Figure 2. The two segments denoted as 'Contingency' and shown in red can be predefined as the response to a vehicle fault. This path would allow the completion of the NPC Burn with an alternate set of vehicle configurations corresponding to the fault conditions. As with the nominal sequence the contingency paths, of which there could be many, are specified via data-driven parameters. Specifying the contingency sequences as data-driven allows the fault recovery plans can also mature over time as the common vehicle faults and their desired responses mature over time.

3. GN&C FSW ARCHITECTURE

As described in the previous section, the TM software is responsible for broadcasting the current Mission Phase and Mission Segment to each of the Orion vehicle subsystems. The subsystems are each responsible for helping to achieve the objective of the Mission Phases and Segments by performing the desired subsystem functionality and making the necessary configuration updates. Because Mission Segments are defined at a course level of granularity, in many cases the GN&C subsystem must be further decompose the Segments into lower level sequences of commands to accomplish the objectives of a given Mission Segment. As a result, automated sequencing capability is distributed to the GN&C subsystem as well. This section presents the GN&C architecture in place to permit automated sequencing at the subsystem level and fulfill the vehicle level requirements for Orion automation [1]. The following subsections describe the GN&C Flight Software architecture and provide the necessary context to understand how the GN&C subsystem uses data-driven parameters to perform the required automation, which is detailed in later sections. An example of these concepts that summarizes all of the functionality and integrated capability required within the GN&C automation software is included in the final subsection.

Automated GN&C Sequencing Software

The GN&C subsystem has a unique role on the vehicle in that it directly affects the trajectory and attitude throughout the mission. Therefore many of the parameters sequenced internally to GN&C dictate mission profile, attitude timelines and other vehicle operating constraints and thus impact the entire vehicle. From the perspective of mission sequencing, GN&C parameters effectively define the mission being flown. To keep these mission level parameters and constraints flexible to accommodate sequences for as yet undefined future mission sequences, a data-driven design approach is taken at the GN&C subsystem level to permit adaptable mission sequences to be developed over the life of the program. As with vehicle level automation software, the onboard GN&C sequencing software must work together with -- or as part of -- the integrated fault recovery logic on the vehicle. It must be capable of responding to inhibits, overrides and other parameter updates from the ground or onboard crew to provide control over the automation at all times [1].

Together the guidance, navigation and control algorithms within the GN&C subsystem require a complex interaction of these tightly integrated software components. A centralized onboard sequencing engine provides the primary control over these interactions as well as the capability to update static parameter data required to operate each of these algorithms. This functionality is collectively referred to as the GN&C Executive software. The executive software is architected as a data-driven state machine engine which processes parameterized sequences of ***GN&C Activities.*** A GN&C Activity refers to the commands issued by the GN&C Executive used to configure the GN&C Subsystem.

Within the GN&C executive software, ***GN&C Activities*** provide the capability to 1) specify the active GN&C algorithms (by issuing flight software modes), 2) configuring the GN&C algorithm behavior via static parameters, and 3) initiate algorithm re-initialization. GN&C Activities provide a useful and convenient means to collect groupings of functionality and parameters together for one or more GN&C software components, for example one GN&C Activity can specify and configure the active algorithms for the entire subsystem. GN&C Activities may be strung together into a sequence to form a list of updates to the configuration GN&C subsystem. Each Activity may perform one or many actions thereby providing the capability to define and execute coordinated actions for the entire subsystem. More detail will be provided in the example to follow. The following section features key derived design requirements for internal GN&C Activity list sequencing and data-driven logic specification.

GN&C Sequencing logic

To satisfy the requirement for truly reconfigurable mission sequences, it is necessary to provide a means to parameterize the logical conditions between GN&C Activities as well as the Activities themselves. This capability permits mission designers to recombine or add new logic to existing sequences or even create entirely new sequences without rebuilding the flight software, saving considerable rework and testing expense over the life of the program [**REF?**]. On Orion the use of logical normal forms will be employed to permit mission designers to define logical expressions of multiple variables, static parameters, and operations (AND/OR/NOT) [2]. These logical parameterization schemes are straightforward, efficient and easily extended to provide for multiple logical instantiations, wherever necessary.

Within the GN&C subsystem, this data-driven design permits the ***transition criteria*** between each GN&C Activity to be parameterized along with the other details of each Activity. In this design each Activity has an association to a single logical object whose purpose is to define that Activity's transition conditions. Satisfying the transition conditions in an Activity enables sequencing to occur to subsequent Activities in the list. As shown in Ref. [2], transition conditions may be composed of complex logical expressions of multiple variables.

In addition to nominal sequencing GN&C provides the capability to define one or more logical paths through each Activity list. These logical paths may be utilized to invoke simple nominal logic, or provide responses for detected faults. Within the GN&C subsystem, ***activation criteria*** define the conditions which permit one or more Activities to be dynamically skipped, as needed. An example Activity list making use of both transition and activation criteria is presented at the end of this section.

Figure 3 depicts the relationship of GN&C Activity activation and transition criteria schematically in a flow chart. It depicts several key aspects of the sequencing logic design. The "GN&C Reconfiguration" block represents the automated commanded updates to the GN&C software at the start of each Activity. These commanded updates define boundaries of algorithm execution or changes to parameters within the software. When the GN&C subsystem is "in Activity A" it implies that Activity A's software reconfiguration commands have been issued and the executive software is monitoring for the successful completion of the Activity transition criteria. When the transition criteria are satisfied, the software enables sequencing to subsequent Activities in the list. As depicted in Figure 4, the GN&C Activity activation criteria are not associated with the GN&C Activities directly, rather they are parameterized within a parent "Activity List" object. This relationship permits activation criteria to be dynamically evaluated to skip over several Activities in a list, without knowledge of any of the particular Activity details. (This particular implementation detail will be revisited again in Section 4). The logical parameterization of activation criteria is the same as GN&C transition criteria; as such it is possible to define complex logical conditions if required. In the event that none of the Activity List activation criteria are valid, the executive software simply remains in the currently active Activity and does not make any transition.

In Section 2 the concept of Segment transition indication from the vehicle subsystems was mentioned. This indication signals to the TM software that each subsystem, including GN&C, has completed the objectives of the Segment and is ready to be commanded to a new Mission Segment. Although it is not depicted in Figure 4, this logical condition is parameterized in exactly the same manner as the Activity transition criteria, using a data driven logical association. This condition may be specified independently of any of the Activity transition criteria, or it may be setup to use exactly the same logical conditions. This is an important association as it completes the data driven logical parameterization for the Mission Segment and makes the mission sequencing information a completely parameterized aspect of the mission plan.

In summary, this subsection has provided an overview of the executive software in place to parameterize sequences within the GN&C subsystem. It has been designed to provide the necessary level of flexibility for both nominal and contingency sequencing capability. The following subsections will focus on the GN&C architecture design in place to support the automated updating of GN&C flight software modes as well as providing the capability to make the parameter updates to individual GN&C algorithms.

GN&C Executive and FSW Domain Interaction

The previous subsection described the GN&C executive software Activity sequencing capabilities in response to Mission Segment commands received from the TM software. This subsection deals with the GN&C architecture used to distribute Activity commands within the GN&C subsystem. In addition, a description of the infrastructure for updating static GN&C algorithm parameters is detailed.

The GN&C software is divided into 16 separate ***GN&C Flight Software Domains***, which each represent a grouping of common GN&C functionality. The GN&C FSW Domains are divided across functional categories (Guidance, Navigation and Control) as well as by major operational flight phase (Ascent, On-Orbit and Entry). This breakdown facilitates distributed model-based development [3] and also has CPU throughput advantages to effectively 'Idle' large portions of the code when they are not in use. Table 1 presents a complete listing of the GN&C FSW Domains by GN&C function. Note that the GN&C Health Manager Domain interaction with the rest of the GN&C architecture is described in Section 6, after concepts related to the nominal sequencing capability are presented. The GCI Domain contains the GN&C Executive software as well as other software components not related to the discussion at hand.

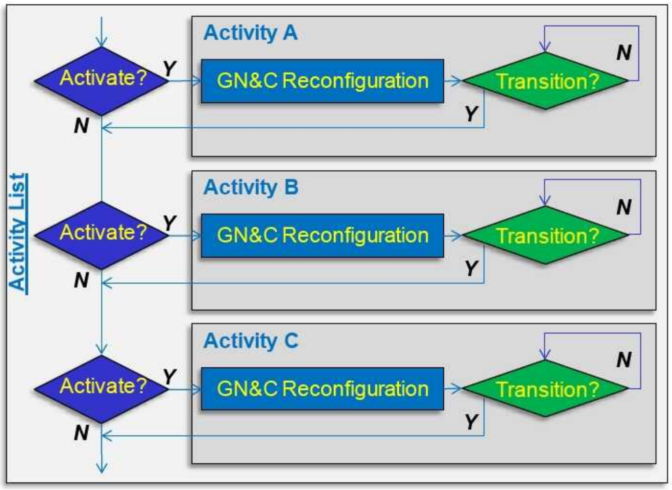


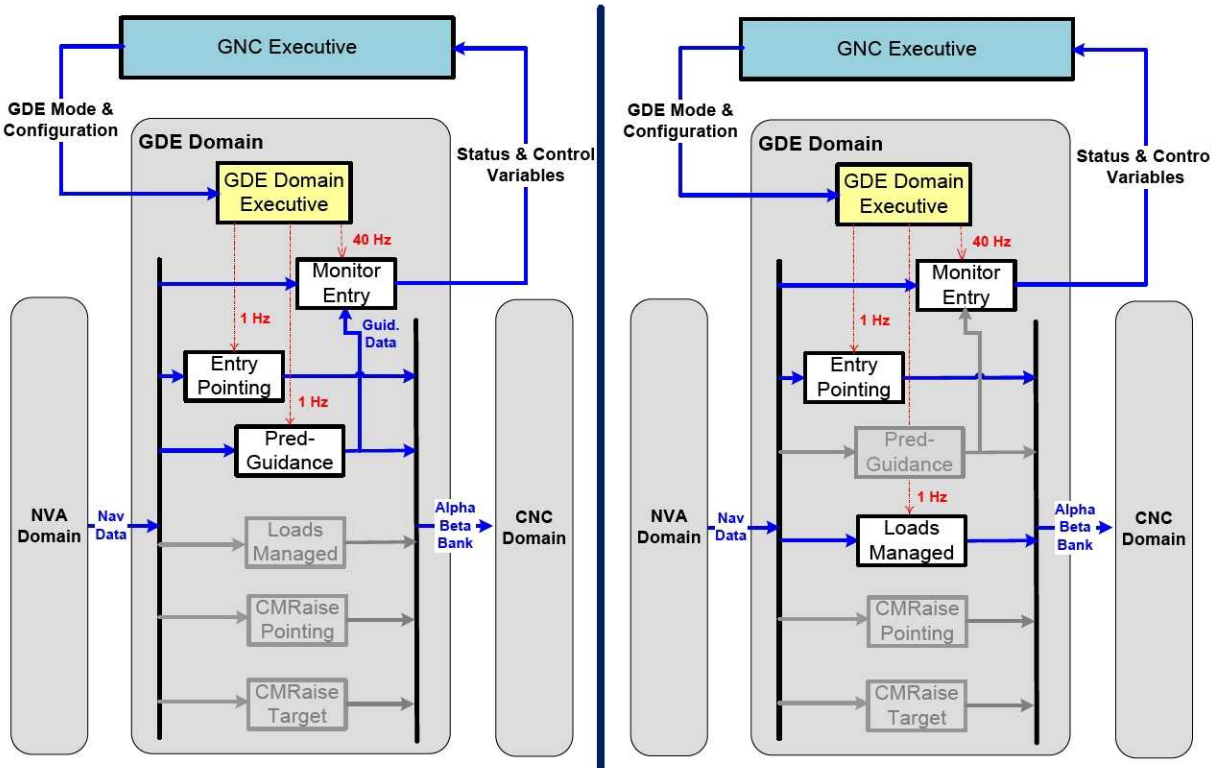
Figure 4: Flowchart of activation and transition criteria for GN&C Activity sequencing.

Table 1. The GN&C FSW Domain Definitions, sorted by Navigation, Guidance and Control function respectively.

GN&C Domain	GN&C Domain Functionality
GCI	GN&C Command Interface
NVA	Absolute Navigation
NVR	Relative Navigation (Rendezvous)
NVE	Ephemeris Processing
NHM	Navigation Health Manager
GMP	Vehicle Mass Properties
GDA	Ascent Guidance
GDE	Entry Guidance
GDO	On-Orbit Guidance
GHM	Guidance Health Manager
CNC	Command-Module (CM) Control (Entry)
CNS	Service-Module (SM) Control (Ascent Aborts, On-Orbit)
CNL	Launch Abort System (LAS) Control (Ascent Aborts)
CNE	Propulsion Engine Controls
CNP	Propulsion Systems Control
CHM	Control Health Manager

The GN&C subsystem behavior is defined by the currently executing ***GN&C Domain Modes.*** Each GN&C Domain has a list of possible modes of execution that define the active GN&C algorithms, thereby defining the behavior of that domain. The Domain Mode for each Domain is issued by the GN&C Executive software and may be changed via GN&C Activity updates. In many cases the Navigation, Guidance and Control Mode updates must be coordinated to occur at the same time to ensure algorithm validity. Hence, the centralized GN&C executive software is responsible for coordinating these actions across Domains via GN&C Activities.

In addition to the Domain Mode, the GN&C executive is responsible for providing configuration information to specify both Domain and GN&C algorithm-related static parameters. These parameters may be specified via GN&C



(a) Entry Guidance Domain (GDE), Entry Mode

(b) GDE, Loads Managed Mode

Figure 5: Example GN&C Executive and Domain Mode Functionality for the Entry Guidance Domain (GDE).
(a) The active CSUs corresponding to the (Guided) Entry Mode
(b) The active CSUs corresponding to the (Unguided) Loads Managed Mode

Activity independently of the Domain Mode, or as part of a coordinated update.

Figure 4 illustrates the GN&C Flight software architecture including a representative GN&C Flight Software Domain. The GN&C executive broadcasts the Domain Mode and configuration updates to each Domain in the GN&C subsystem as specified in the GN&C Activity sequence. Each Domain receives a unique Mode and configuration command specific to that Domain (not depicted in the generic example of Figure 4). The GN&C Domains each contain a "Domain Executive" which is responsible for receiving the Mode and configuration update commands and ensuring they get distributed to the appropriate algorithms in the Domain as described below.

Domain Executive and GN&C Algorithm Execution

The GN&C algorithms themselves are partitioned into specific Computer Software Units (CSUs) that are activated as a function of the Domain Mode command received from the GN&C executive. The CSUs are each defined at manageable levels of complexity to facilitate distributed model based development within the Domain [3]. As shown in Figure 4, each CSU is defined with interfaces for dynamic I/O data (in blue) from other GN&C CSUs or external subsystems. Generally there are very complex I/O relationships between Domains and CSUs within the GN&C subsystem. Static input parameter structures are also defined (in pink) for each CSU. The static parameter values may be comprised of different types of quantities used for different purposes. For example, guidance parameters might include values of gains and control parameters might include deadband values. These parameter values are controlled via the GN&C executive software as configuration updates as specified by GN&C Activities. More detail and examples of the Domain and CSU parameters are provided in the subsequent sections.

Domain Modes are defined such that each Mode activates a unique combination of CSUs within the Domain. Activation of a particular Mode causes the Domain executive to trigger the fixed-rate execution of CSUs for the commanded Domain Mode. The Domain executive also relays to each CSU which parameter set to use for the current GN&C Activity, as well as when to perform re-initialization functions.

The concept of CSU execution via Domain Mode command is illustrated through a specific example in the Entry Guidance Domain (GDE) shown in Figure 5. The CSUs in this Domain are responsible for collectively generating the commanded bank angle of the Crew Module (CM) (shown in Figure 1) during the Entry phase of flight, as well as other functions not pertinent to this example. Figure 5a and 5b depict the difference between two distinct entry guidance schemes and the distinct set of active CSUs for two example GDE Domain Modes (Entry and Loads Managed Mode, respectively). In this example, the Entry Monitor CSU is responsible for monitoring for time-critical events, such as triggering the parachute deployment. The outputs of the Entry Monitor CSU are required by the GN&C executive for triggering the pertinent GN&C Activity and Mission Segment transitions. Although static parameter structures are not depicted here, each of the active CSUs shown would require commanded updates from the GN&C executive software to function properly in each of these modes. The following subsection provides more detail and examples on the mechanisms for making parameter updates within GN&C CSUs for this purpose. .

CSU Parameter Structures

The previous subsection provided an overview of the Domain Modes and how they are used to control the GN&C subsystem behavior through CSU activation. CSU static parameters are also important in controlling the GN&C subsystem behavior and performance as they are the primary means by which mission level sequencing information is communicated to the GN&C algorithms. In this context CSU parameters refer to mission specific target values, performance thresholds, physical constants or other non-dynamic data. In many cases CSU parameters are also used to internally change the algorithm from one execution state to another, through flags, enumerated types or other means. As a result, the mechanisms by which these parameters are updated in the software are a very important aspect of the automated sequencing capability of the Orion spacecraft. The following is a discussion of the GN&C architecture infrastructure required to support static CSU parameter updates on Activity boundaries.

In the existing GN&C architecture, the algorithm parameters fed into a CSU may come from one of three separate sources:

❍ GN&C – Level Structure
❍ Domain – Level Structures
❍ CSU – Level Structures

The implication is that a particular CSU parameter may have scope at different levels of the software depending on the definition of the parameters in other parts of the GN&C software. The highest level is the GN&C parameter structure which is defined and updated by the GN&C executive software directly and contains parameters destined for all of the GN&C Domains. The intermediate level is a Domain parameter structure that contains parameters common to multiple CSUs within a Domain. The Domain parameter bus is updated by the Domain executive software since these parameters are needed by multiple CSUs within the domain and not destined for an individual CSU. The lowest level source of parameters is the CSU parameter structure which contains parameters that are only specific to a particular CSU. (The three parameter update sources are not depicted in Figure 5 to maintain simplicity of the Figure)

In some cases it is appropriate to elevate parameters out of a CSU-level parameter bus to either the Domain or the GN&C parameter. One reason to separate these parameters to different levels is to ensure parameter consistency across multiple CSUs. When a specific parameter is required in multiple CSUs within the same domain it may be elevated to the Domain parameter bus as long as its scope is limited to one domain. Similarly if a parameter has scope in multiple CSUs across multiple FSW Domains, it becomes a candidate for being moved to the GN&C parameter bus. Elevating parameters in this fashion ensures that the affected CSUs are obtaining the exact same instance of a particular parameter whenever it is updated. Examples for parameters of this type include physical constants (g, μ, c, etc.) as well as specific vehicle constants (engine thrust, I_{SP} etc.).

Another reason to elevate parameters above the CSU bus is based on the frequency of change of the parameter. When parameters are frequently modified in a mission sequence or via command from the crew or ground, they could be candidates for assignment from the GN&C parameter structure. These types of parameters are referred to as ***GN&C command parameters*** because they often represent the items in software which are specific to a particular mission objective that crew or ground operators require access to command updates, such as the target landing site, spacecraft attitude and rendezvous burn targets. Such values change on a mission to mission basis or even more frequently. These parameters may be elevated up to the GN&C level parameter structure even if only a single CSU requires them. Assigning a parameter directly from this level provides a large benefit in that the quantities which frequently change can be segregated from the parameters which are more fixed and tightly controlled. The advantage of having direct access to the command parameters will be demonstrated through example in the following section. However before moving on to this section it will be helpful to present an example summarizing the concepts presented in this section.

GN&C Activity Sequencing Example

This section has summarized the GN&C architecture to support automated sequencing and data-driven mission planning for the Orion spacecraft. This final subsection presents a high-level sequencing example utilizing the concepts and architecture presented thus far.

Figure 6 depicts an example for how a list of GN&C Activities may be used to sequence through a typical automated "Burn" Mission Segment using the Orion Main Engine (OME). In this example, the burn Segment is generically defined such that targeting for the burn has previously been achieved in the prior Segment. Upon the start of the burn Segment, the vehicle first performs an attitude maneuver to the desired attitude obtained from the targeting solution. Once the attitude maneuver has been completed, the OME (shown in Figure 1) is used to execute the rendezvous burn to within some nonzero threshold of the targeted burn velocity (called Velocity-to-Go (VGO)). If necessary, a second "trim burn" is performed immediately following using the Reaction Control System (RCS) to complete the burn execution within a higher accuracy tolerance since the RCS system is capable of achieving much lower residual Velocity-to-Go (VGO) threshold. Finally, the vehicle is placed into a safe Post-Burn Configuration to await the next Mission Segment.

Figure 6 shows the current GN&C Activity as the RCS Trim Burn. The GN&C Executive sends active GN&C Mode Commands to the Absolute Navigation (NVA), Orbit Guidance (GDO), and SM Control (CNS) Domains as well as updated configuration parameters to the GDO and CNS Domains. The GN&C Executive also sends 'Idle' commands to the remainder of the Domains.

In the Burn Segment example, the RCS Trim Burn Activity will become active only if the residual VGO is above the threshold for RCS trim burns. If that activation criterion is true, the RCS Trim Burn Activity will execute. When the transition conditions are true (residual VGO below the threshold), the GN&C Executive will transition to the next Activity in the list, Post-Burn Configuration. If the residual VGO happens to be less than the RCS threshold following the OME burn, the RCS Trim Burn Activity is skipped and GN&C sequences to the Post-Burn Configuration Activity immediately. In this case there is no need to start the RCS burn and the Activity sequence logic defined to accommodate that.

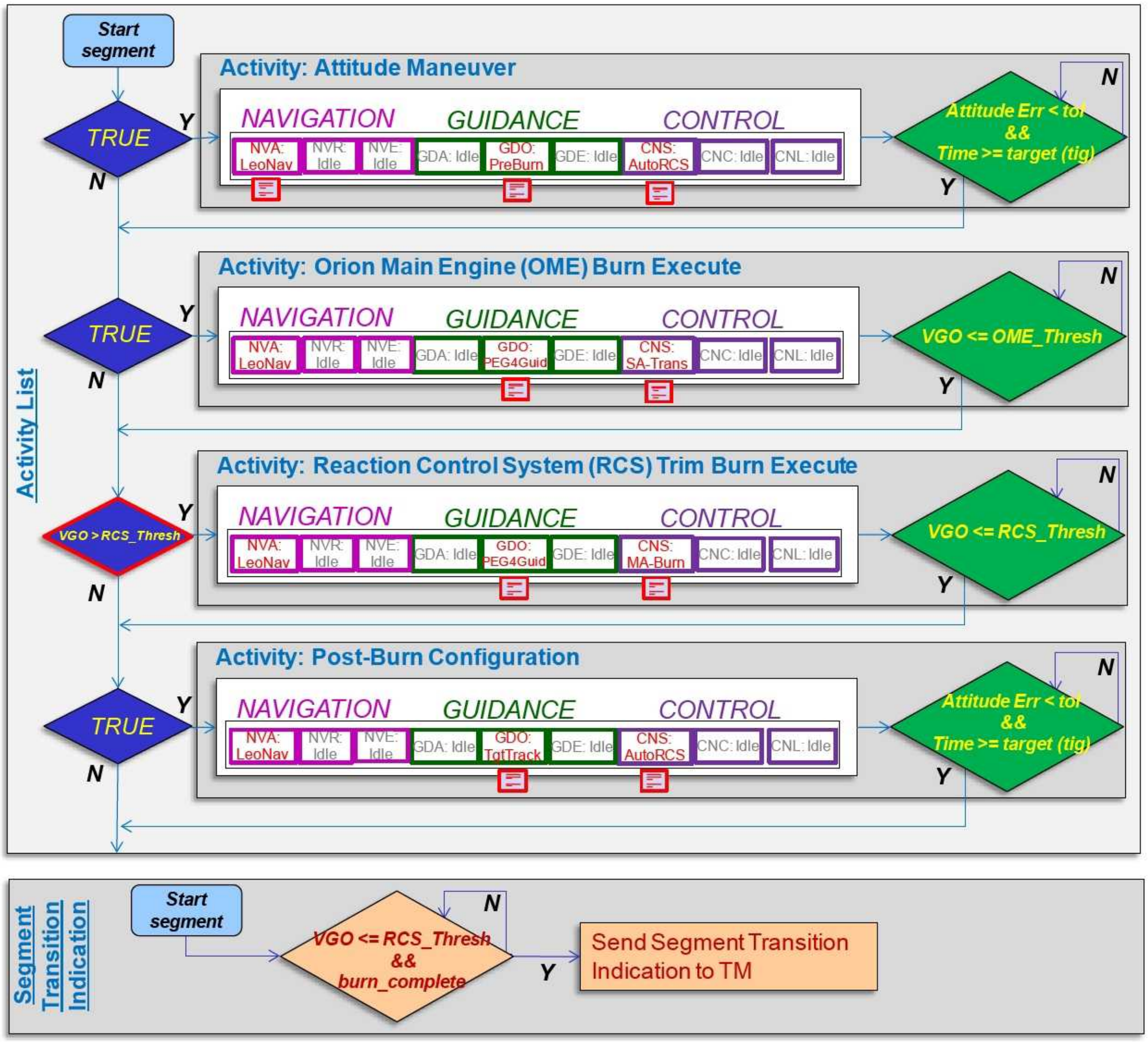


Figure 6: Example of GN&C Activity sequencing for a "Burn" segment utilizing the Orion Main Engine (OME) followed by a conditional clean up "Trim Burn" Activity using the Reaction Control System (RCS). The Changes to GN&C domain modes are shown highlighted with the domain configuration parameter updates for each Activity. The Segment transition indication criteria is shown as a separate logical condition required to complete the segment.

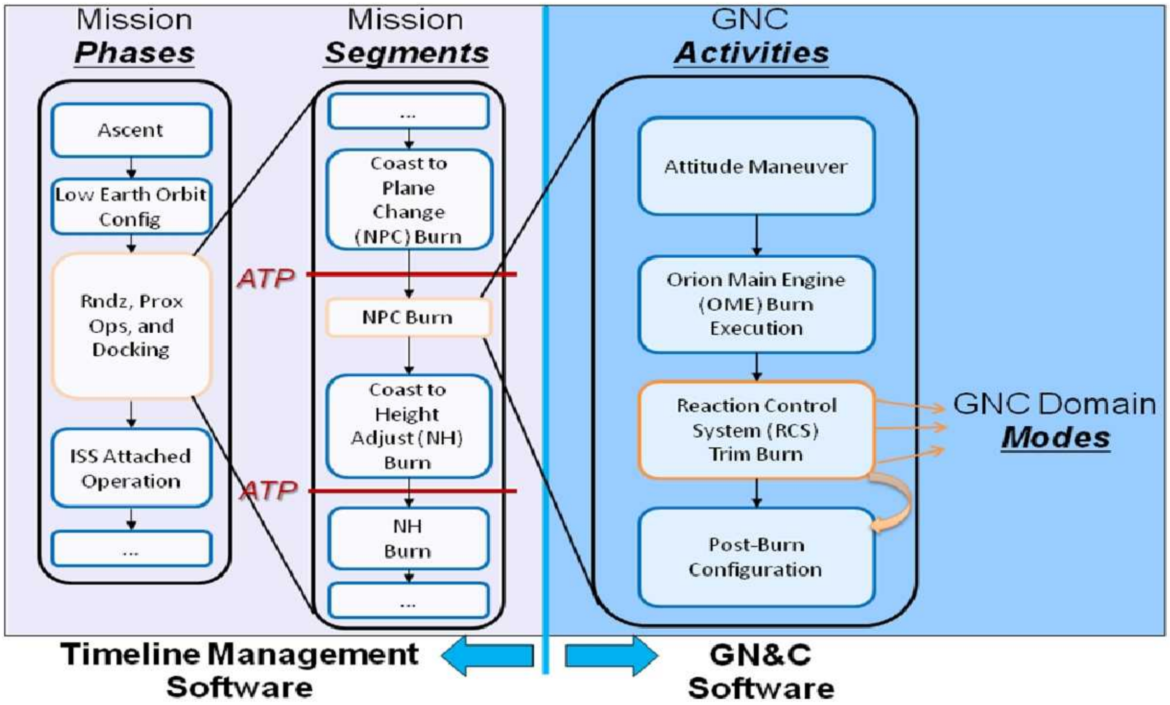

Mission
Phases
Mission
Segments
GNC
Activities
Ascent
Low Earth Orbit Config
Rndz, Prox Ops, and Docking
ISS Attached Operation
...
...
Coast to Plane Change (NPC) Burn
ATP
NPC Burn
Coast to Height Adjust (NH) Burn
ATP
NH Burn
...
Attitude Maneuver
Orion Main Engine (OME) Burn Execution
Reaction Control System (RCS) Trim Burn
Post-Burn Configuration
GNC Domain
Modes
Timeline Management Software
GN&C Software

Figure 7: PLACEHOLDER FIGURE ILLUSTRATING THE PSAM. shows the Orion mission sequencing hierarchy as it is decomposed between the Timeline Management software and GN&C Subsystem software.

When the objectives of the GN&C subsystem for the Burn Segment are completed (the burn has been completed), the 'Segment transition condition' is set true resulting in the Segment Transition Indication being sent to the TM software. In this case the segment transition criteria is the same as the completion of the final Activity in the Activity List, but there is enough flexibility built in to the executive software to allow for different conditions to be specified. Upon receiving indication that the Burn Segment is complete, the TM software is responsible for commanding a new Mission Segment to all of the vehicle subsystems.

A potential extension of this automated burn example incorporates additional logic within the Activity list to accommodate the case that the OME fails during the burn and an immediate fault response to switch engine effectors is required. In this case the Orion auxiliary engines (not represented in Figure 1) are used as a backup to complete time critical burns. To encode this fault response within the Burn sequencing example shown in Figure 4, a separate "Aux Burn" Activity with associated activation and transition criteria would be included between the OME and Trim Burn Activities. The Aux Burn Activity would be nominally skipped, but in the event an OME failure occurred the logic would be in place to respond to the failure by immediately activating it. This type of functionality is not fully defined at this stage of the program, but there are many instances where it may be applied.

This example has provided details of how several aspects of the data-driven GN&C automated sequencing capabilities can be used. The following section will describe the schema used to capture each portion of the Orion Mission Sequencing hierarchy, including Mission Phase, Mission Segment, GN&C Activity, and GN&C Domain Mode (also referred to as the PSAM schema)

4. PSAM Schema

The previous sections described the mission sequencing capability within the GN&C software and mechanisms through which this sequencing may be controlled. As described in these sections, the hierarchy of automation software collectively forms a mission plan which is distributed between the Timeline Manager and GN&C software. This hierarchy of mission plan information is composed of Mission Phases, Segments, GN&C Activities and Domain Modes (PSAM). This hierarchy of sequencing information is depicted in Figure 7, showing the allocation between both TM & GN&C. For each mission segment, a sequence of Domain modes and reconfiguration commands are issued to the Guidance, Navigation and Control Domains via the list of configured GN&C Activities. In this section, the software artifacts facilitating data-driven mission design are presented. In addition, the design philosophy and concept of operations for building a data driven sequencer onboard a manned system is discussed.

Because Orion is required to perform automated sequencing, mission data for the nominal and contingency Mission Phases and Segments will be stored onboard the vehicle[1]. The current FSW design provides for an onboard memory store of GN&C configuration data from which sequencing can be performed. This data is loaded from configuration files at the start of the mission and may be refreshed from these files at discrete points during the mission, or reloaded if changes need to be made to the existing configuration data onboard the vehicle. Sequencing from configuration data in memory requires that a data schema is established whereby the configurations for all of the CSU parameters may be assigned to specific storage locations in memory, queried and reinitialized if necessary. This data schema permits configurations from multiple levels of the PSAM hierarchy to be defined, linked and sequenced as needed during the particular mission being performed.

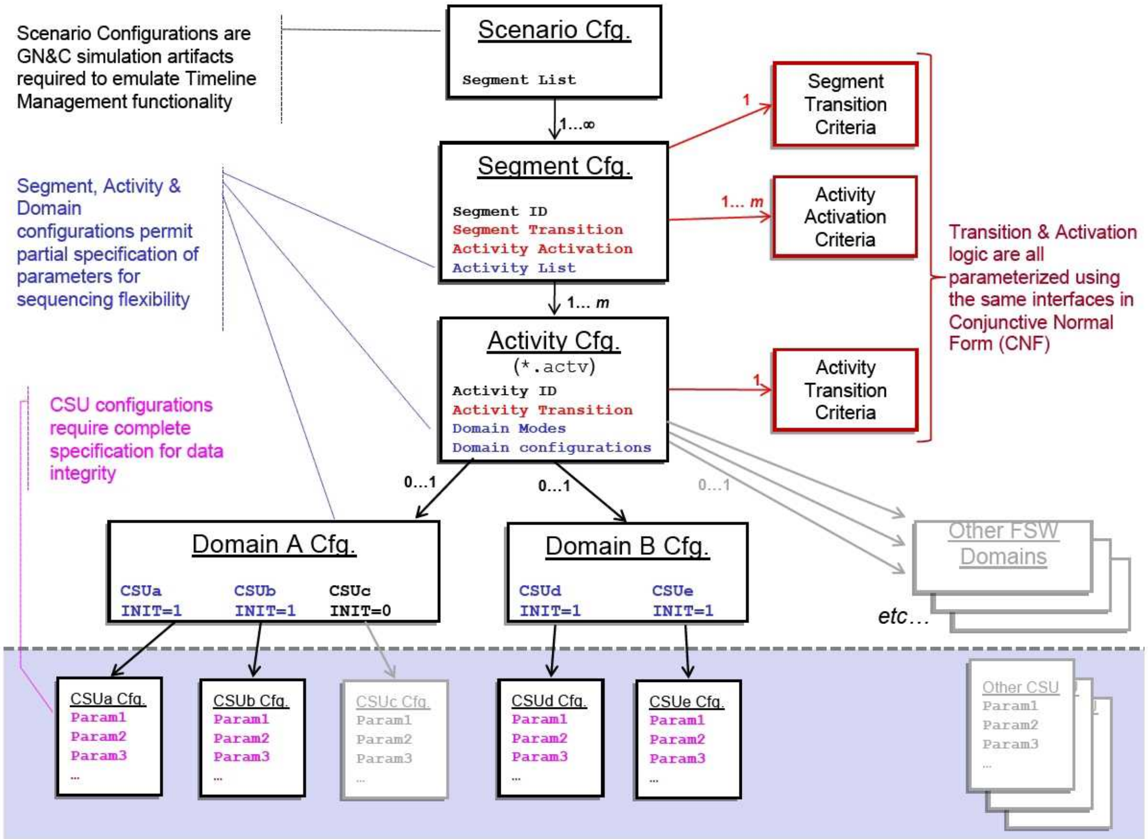
The design provides for configuration elements at each tier of the FSW hierarchy to make it possible to configure the vehicle at each level of detail. Each lower level in the hierarchy provides access to specific parameters within the GN&C FSW. In this sense commanding can be performed at a very high level (Segment), or very detailed, parameter level (CSU). The intermediate Activity and Domain levels provide corresponding intermediate levels of access to parameters at those levels of detail.

GN&C Automation Parameter Hierarchy

An overview of the GN&C PSAM Data Schema is depicted in Figure 8. This schema depicts the structure of the data elements required by GN&C to sequence through Mission Segments and GN&C Activities, with configurations specified for the GN&C Domains and CSUs. Each of these elements represents a specific instance of configuration data which is linked to others through memory addresses and may be accessed via the onboard software as needed to execute the mission plan..

Segment transition criteria, Activity activation criteria and Activity transition criteria are all parameterized through the PSAM data schema. All of these logical conditions may be parameterized through the same data specification and are linked with the appropriate parameter field as shown in Figure 8. The Segment configuration contains an association to the Segment transition criteria, associations to each of the Activities in the sequence, and an association to an Activity activation criteria for each of the Activities in the list. A GN&C Activity specifies how each of the GN&C Domains is configured by specifying the GN&C Domain Mode and configurations for each Domain. The Activity configuration contains the Domain Modes, the Domain configurations to be used for that Activity, as well as the transition criteria associated with that Activity.

In addition to Domain Modes and configurations, GN&C-level parameters are specified within each GN&C Activity. Recall from Section 3 that these parameters may have been elevated because they are parameters which must be


Scenario Configurations are GN&C simulation artifacts required to emulate Timeline Management functionality

Segment, Activity & Domain configurations permit partial specification of parameters for sequencing flexibility

CSU configurations require complete specification for data integrity

Transition & Activation logic are all parameterized using the same interfaces in Conjunctive Normal Form (CNF)

Figure 8: The GN&C PSAM Data Schema depicting the primary configuration elements required to perform mission sequencing.

coordinated among CSUs in multiple Domains, or because they contain very mission specific parameter information. An example of how this parameter hierarchy might be used in practice to segregate parameters is presented in the following subsection on the *"CSU Parameter Structure"*

Figure 8 shows that GN&C Activities may be instantiated with 0…1 (zero or one) associations to each domain configuration structure. This is meant to imply that an Activity is not required to completely populate the parameter list of domain configurations. If a Domain configuration update is not present in an Activity it continues processing without changes. Although it is not depicted in the figure, this is also true for the Domain Mode parameters specified in the Activity. This is an important aspect of the Orion GN&C automation design which permits the automated sequence to be specifically targeted to only the areas of the software which need to be updated. Without this capability, the ground and the crew may have difficulty making any changes to the vehicle software without first disabling the automation onboard the vehicle.

The GN&C data schema is architected to permit the specification of a new set of Domain configurations for each Activity. The purpose of each Domain configuration element is to define changes to Domain and CSU level. As with the higher levels of the PSAM schema, only those CSU configurations which need to be updated are specified at this level; when a CSU configuration set is left unspecified it continues to run with the previously specified values. This is indicated in Figure 8 schematically using 0…1 (zero or one) associations to each CSU parameter element.

CSU Parameter Structure

The CSU configuration element controls all of the CSU-level parameters for a CSU and specifies the values for those parameters for the current Activity. CSU parameters are subdivided into groups of parameters that are specific to a particular function within the CSU, or parameters that are frequently updated. For example, it is common for a CSU to split parameters into groups which are related to

(1) Hardware Interfaces including IMU mounting angles, lever arm corrections, or other similar quantities.

(2) Mathematical / Physical Constants specific to a particular algorithm.

(3) Algorithm Performance such as control gains, filter coefficients, etc.

These groups are largely defined by the CSU form and function, and each CSU may be defined with a number of different configuration elements within the overall schema. As shown in Figure 8, these configuration sets may be updated independently from one another or in a coordinated fashion. CSU Command parameters such as attitude targets, landing site targets, etc. are not handled within these configuration sets to provide more direct access to them through GN&C Activities.

A basic example illustrating this CSU parameter specification hierarchy comes from the SM Control Domain parameterization for attitude pointing utilized while on orbit. To operate the attitude pointing algorithm, a detailed list of algorithm-specific constants related to performance and computation method are required. In addition, a number of parameters define the specific pointing targets, as well as the pointing method desired. In this case all of the attitude target parameters as well as the pointing method are elevated into the GN&C parameter bus to make it part of the Activity specification, while algorithmic detail parameters are separated out into separate configuration elements for selection as part of the Domain configuration. The detailed parameter sets include generic options for high accuracy, medium accuracy and low accuracy for general use, as well as configuration sets which are specifically tailored for particular mission events. In this software architecture mission planners and analysts are able to specify the mission specific parameters at the Activity level as well as define which configuration set is required for each particular Activity. In this sense the GN&C Activity parameters provide a high level "user interface" to the GN&C Subsystem and insulate the details of the GN&C algorithms into lower level configurations.

GN&C Schema Specification Standards & Evolveability in Orion Automation

The Orion GN&C automation software is being designed with the flexibility to make large configuration changes as well as individual parameter updates on Activity boundaries. This type of commanding refers to the ability to specify only those values which have to be updated on a particular Activity boundary. However, this does not preclude a mission planner from fully specifying a large subset (or complete set) of GN&C parameters within an Activity. Parameters which are not updated remain static across each Activity boundary. Likewise, GN&C Domain Modes may be specified in the same manner. Only those Modes which need to be updated have to be set within the GN&C Activity.

The capability to build up lists of Activities which are not completely specified is important to ensure that the Activity lists are robust to external changes by the ground or crew, FDIR responses and contingency cases. Providing this feature permits Activity lists to be built up which incrementally affect the configuration of the GN&C software. In this paradigm ground or crew operators can make updates to the GN&C configuration or potentially update Domain modes without necessarily being forced to inhibit the automation software or make large updates to subsequent Activities configurations. The key to this is designing mission sequences which only affect the minimal set of GN&C parameters required as they are executed.

Operationally partial Activity specification is also a benefit because the automation software behaves similarly to the manner in which the onboard crew would if they were issuing the same commands. If the crew wished to manually issue a sequence of commands, the exact same mechanisms to update the GN&C software could be used. Activity sequences which completely specify the full set of GN&C parameters must be updated if crew or ground commands are required to persist across Activity boundaries. Partial Activity specification is thus necessary to mitigate complex interactions between the sequencing and FDIR software, as well as the crew.

This concept of operations requires a very well understood decision tree to progress between each Mission Segment to ensure that the software is not put into an unsafe or undesirable configuration. It is likely that each Mission Segment will include an initial Activity that is completely specified to setup the necessary preconditions in entering the Mission Segment. This procedure ensures that the GN&C software is properly configured regardless from which Mission Segment it is transitioning.

The flexibility to partially specify GN&C parameters is a powerful capability which permits mission designers the ability to make changes to specific areas of the GN&C flight software and leave others unaffected as the mission progresses. However this capability requires mission planning tools and good visibility into the configuration of the vehicle as a mission is executed. The following section addresses how the PSAM mission sequence might be visualized, manipulated and updated using prototype database tools.

5. PSAM Database

The PSAM data schema is used to parameterize automated sequences for the GN&C subsystem, and it enables coordination among all of the elements of the GN&C flight software. For a particular mission, there will be parameter data associated with the Mission Segment, Activity, GN&C Domain Mode, as well as parameter data for individual CSUs. Prior to and during each flight, algorithm developers, analysts, and mission planners will need to manage and manipulate this data for many nominal and contingency scenarios. Because of the potentially imposing task of managing this data manually through data configuration files, a database was developed to manage the PSAM data content associated with a variety of mission scenarios.

Motivation

The Orion GN&C development is currently at a stage where the vehicle flight software for guidance, navigation, and control is nearing complete integration. Subsequent to that effort will be extensive analysis and testing, which requires the management of data as described in the previous section. It is therefore desirable to have a means of storing and manipulating this data in a flexible and reliable way. This is part of the motivation for building a database for the PSAM data content. Furthermore, useful features of databases include:

- Queries to extract the relationships between datasets
- Flexibility in making changes
- Consistency in data records
- Ability to build user interfaces to easily manipulate the underlying data and generate summary reports.

These database features will be useful for assessing timeline sequences, GN&C configurations, and the effects changes have on both. For example, the modification of CSU parameters that are also used in another portion of the mission may have effects that need to be understood, and the database could readily provide information on these impacts. The database will evolve into a critical aspect of using the GN&C sequencing software.

Microsoft Access was chosen as the initial prototype application for its ease of use and availability. Once stability is achieved and other database tools are developed across the Orion project, this work will be transferred to a more permanent platform. Future versions of this tool will be essential for managing and tracking the hundreds, if not thousands, of configuration items required to operate and configure the GN&C subsystem.

PSAM Database Features

The "PSAM Database" consists of a series of tables and relationships that correspond to the data schema of the GN&C subsystem. The full hierarchy of configuration data, from the scenario level down to the algorithm/CSU level is captured in the schema of the database, and mirrors the PSAM data schema.

There are several beneficial features of the database. One useful capability is the quick and easy replication of data. This allows PSAM scenario configuration data to be quickly modified. For example, if one scenario is set up to test a nominal entry, another scenario testing an alternate guidance scheme can be built by copying the nominal scenario hierarchy and altering which guidance mode runs. Both scenarios can be saved in the database for future analysis.

Another benefit of the database is that it facilitates access to and visualization of the data. A series of user interface forms have been built to enable easy manipulation of PSAM data. The primary form in the database, the scenario management form shown in Figure 9, enables user input for information related to the Mission Segments and the GN&C Activities, including Domain Mode and configuration specification. There are also regions on the form to specify the criteria for Segment and Activity transitions, as well as Activity activation criteria. When a user needs to specify a new transition condition, a button brings up a separate transition form for building the transition and activation criteria. Once a condition is created using the transition form, it can be applied to Segments and Activities on the scenario management form. When a user needs to specify a new domain configuration, a button brings up a separate domain configuration form. With this interface the user can specify which CSU parameter sets to use for a particular Activity. The specification at this level is what dictates the data values that will be used by the GN&C algorithms during different portions of the mission.

There are certain aspects of the GN&C architecture design that will change on occasion and affect the PSAM data schema. One example is the set of variables that is used to compose transition conditions. To readily adapt to these periodic changes, the the database incorporates utilities to automatically read in external data. This allows for updates to be readily merged with the data in the database.

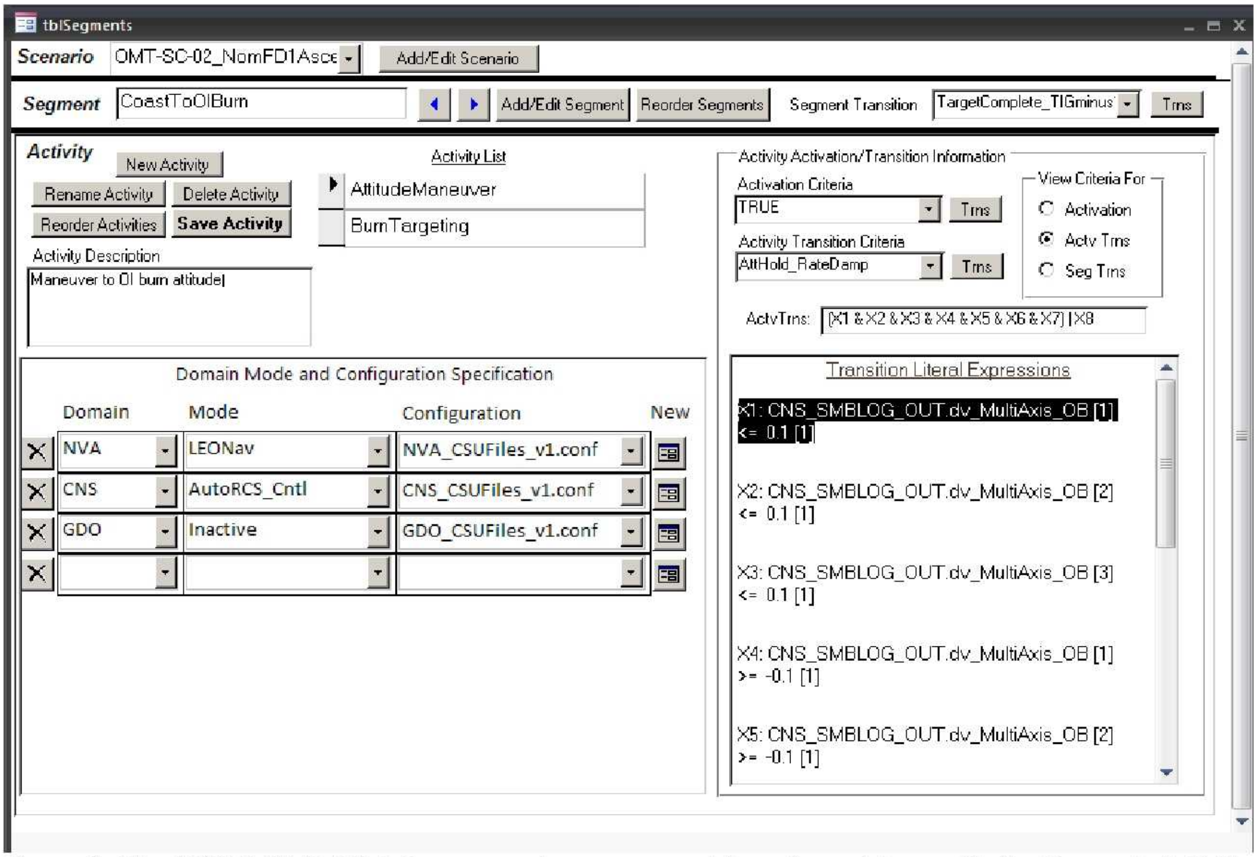
Another feature of using a database is the automatic generation of configuration artifacts and summary reports. Once the mission data is input for a particular scenario, the generation of configuration files can be performed automatically. These instances of configuration data are what will be loaded into the flight vehicle for use by the flight software during a mission. Also, this automatic file generation allows analysts working on testing algorithms and software to quickly manipulate data sets, run simulation and analysis tests, verify vehicle performance, and test software functionality. Alternatively, this would have to be done by manually editing and managing numerous configuration sets that contain the PSAM data. Furthermore, this capability also insulates users from changes to the format of the configuration artifacts since the files can be regenerated to match a new standard.

Generation of summary reports can be automated for a number of purposes. These include, but are not limited to:

- Assessment of Segment and Activity content
- CSU parameters being used
- Attitude timelines
- Trajectory timelines

User Communities

Use of the PSAM database in the near-term will be by GN&C analysts who are developing scenarios that include specific Segment and Activity sequences they wish to analyze. The database allows users to enter, store, and output the data associated with the tests they need to carry out. Therefore, the primary function currently is for GN&C analysis and test case development.



Scenario: OMT-SC-02_NomFD1Asce

Segment: CoastToOIBurn

Segment Transition: TargetComplete_TIGminus

Activity List:
AttitudeManeuver
BurnTargeting

Activity Description: Maneuver to OI burn attitude

Activation Criteria: TRUE

Activity Transition Criteria: AttHold_RateDamp

View Criteria For: ○ Activation ● Actv Trns ○ Seg Trns

ActvTrns: (X1 & X2 & X3 & X4 & X5 & X6 & X7) | X8

Domain Mode and Configuration Specification

Domain	Mode	Configuration	New
NVA	LEONav	NVA_CSUFiles_v1.conf	
CNS	AutoRCS_Cntl	CNS_CSUFiles_v1.conf	
GDO	Inactive	GDO_CSUFiles_v1.conf	

Transition Literal Expressions

X1: CNS_SMBLOG_OUT.dv_MultiAxis_OB [1] <= 0.1 [1]

X2: CNS_SMBLOG_OUT.dv_MultiAxis_OB [2] <= 0.1 [1]

X3: CNS_SMBLOG_OUT.dv_MultiAxis_OB [3] <= 0.1 [1]

X4: CNS_SMBLOG_OUT.dv_MultiAxis_OB [1] >= -0.1 [1]

X5: CNS_SMBLOG_OUT.dv_MultiAxis_OB [2] >= -0.1 [1]

Figure 9: The GN&C PSAM Database scenario management form is used to specify the Segment, Activity, and domain data for each scenario.

A future step will be to integrate the GN&C data and schema into the databases being developed for the entire Orion vehicle. The consistent use of data variables as well as vehicle sequencing that includes interactions between all the subsystems are important steps in the overall design of Orion.

Eventually the process of controlling the GN&C domains through Activity sequences will also be merged with mission planning and ground operations during flights. The testing of mission plans is an important effort, and the data-driven nature of the Orion flight software architecture will be simultaneously powerful and adaptable . Because of this flexible approach, ground tools that enable operations personnel to understand and manage data not only for GN&C but also for the entire vehicle will be needed. A tool with user interfaces will aid in this greatly because it can provide indications of valid configurations for the vehicle and quickly generate flight software products for testing. The PSAM database currently under development will provide guidance on how to establish those planning and operational capabilities for the future.

6. Off-Nominal GN&C Automation

The previous sections described the nominal sequencing capability and touched on certain preliminary aspects of the design related to contingency sequencing and fault recovery. This final section presents issues related to off-nominal interactions with the architecture for issues related to fault detection support as well as manual interactions with the automation software. These issues represent some of the complication in architecting automation software on a fault tolerant, man-rated vehicle and intersect directly with the PSAM data-driven schema in the specific ways presented here. At the time of writing of this abstract, these topics are less mature than the material presented in earlier sections and should be taken as forward work items. These areas are of critical interest to the GN&C community and will be addressed on the forward path to CDR.

GN&C Fault Recovery Software

The Orion Fault Detection Isolation and Recovery (FDIR) software is distributed between VSM and GN&C, much like the mission sequencing software described in the previous sections. All failures that affect multiple subsystems or require changes to the vehicle power or resource states are addressed by VSM software, specifically Systems Management (SM) and Timeline Management (TM) software. The SM software handles vehicle level FDIR logic by collecting health and status from all the vehicle subsystems, resolving anomalies using root-cause determination, and commanding responses via the contingency commanding mechanism introduced in Section 2. This contingency Mission Segment results in the GN&C subsystem transitioning to a new set of GN&C Activities for that Segment, however it should be noted that this behavior is identical to actions taken during any nominal Segment transition. In this case, the GN&C Activities are instead designed to respond to the specific contingency situation at hand.

The GN&C FDIR logic is responsible for responding to failures that must be resolved quickly within the subsystem, as well as reporting all internal faults to the VSM and display software such that the appropriate actions can be taken at that level. In a most cases, the appropriate GN&C action is to place the software into a "safe" state until the TM/SM software can respond with a new Segment, however there may a select few cases in which GN&C must take action.

Figure 10 represents a block diagram of the key interactions between GN&C and VSM related to FDIR, as well as several options for how the GN&C Health Manager domains may interact within the existing GN&C architecture. A similar version of the GN&C architecture (Figure 3) is depicted with an example "FDIR CSU" included. From an architecture perspective, a FDIR CSU simply has embedded logic which provides capability to detect or isolate faults in hardware, sensors, or the algorithms themselves. They may be distributed throughout the flight software and respond to parameter updates and Mode changes like any other CSU. In addition FDIR CSUs are capable of providing dynamic input to any other CSU as a dynamic switching mechanism to affect direct changes within the FSW domain as shown in Figure 10.

As referenced in Table 1, the GN&C Subsystem includes Navigation, Guidance and Control health manager domains for the purpose of consolidating status signals and providing common interfaces to both internal and external sources. It is likely that the health managers will require sequencing information as well as parameter updates for specific Segments or Activities. For example, there may be a set of algorithms that perform FDIR during an Orion Main Engine burn. These monitoring algorithms will not be required during other portions of the mission, and the health management parameters may be specific to that particular burn. Since the ability to mode algorithms into different states is accomplished in the other GN&C domains through enacting GN&C Domain Modes, a similar approach may be used for the health management domains in this respect. To implement this functionality in the existing architecture additional associations in the PSAM schema would be made to include details for each of the Health managers as similarly shown for the existing GN&C Domains.

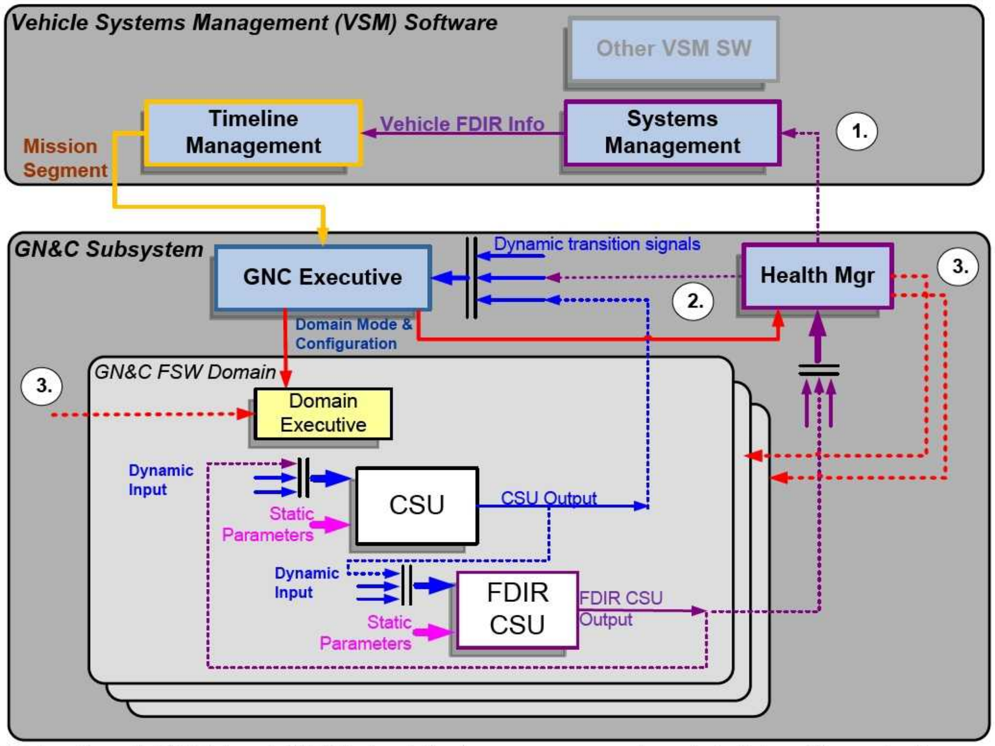


Figure 10: GN&C-level and VSM-level FDIR Architecture components related to off-nominal sequencing and fault response. At least three potential options are described for providing fault responses within the GN&C subsystem via the Health Manager Domains.

Figure 10 describes three non-mutually exclusive options for fault response logic which may be architected utilizing the Health Manager domains for each of the GN&C domains grouped in Table 1. Option 1) relies on the Systems Management software to correctly update the Mission Segment to a valid contingency response based on the indicated fault(s) from GN&C as well crew/ground decisions. Option 1) is the preferred method for most faults, since issuing new Segments via TM is also the nominal commanding mechanism. However there will also be latency associated with Segment changes, and in some cases this latency may be greater than safely allowable, thereby making one of the other options more suitable.

Option 2) describes the use of the GN&C Executive to directly update the modes and/or configurations of the affected GN&C Domains utilizing the centralized sequencing logic described in Section 3. In this case the Health Managers (or CSUs themselves) report faults to the GN&C Executive software as well as the vehicle level software. One benefit of this method is that it is possible to achieve coordinated GN&C responses to particular faults and this scheme will be at least as fast as the proposed commanding path described in option 1). Depending on the complexity of the fault responses required, option 2 would potentially require updates to the PSAM data schema to accommodate complex interactions of Domains.

Option 3) provides the least reconfiguration latency at the expense of additional distributed complexity within the Domain Executive and Health Manager software. This option provides some limited capability within the Health Manager software to update modes and configurations in other GN&C Domains immediately after faults are detected. In this scheme Domain Executives would be required to accept reconfiguration commands from *either* the GCI Domain, or one or more of the Health Manager Domains as shown in Figure 10. For example, assume that Table 1 represents the GN&C Domain execution order (from top to bottom), and the health manager Domains have the capability to change Modes or update the Domain configuration to a predefined state in the event a particular fault occurs. In this example, it is plausible to trace a fault recorded in Navigation through to the NHM Domain and immediately trigger an 'Idle' Mode change in the currently active control Domain. This action would effectively "safe" the vehicle by preventing thruster firings until such time that a contingency Segment were provided by the TM software. There are many variations on this particular example, and it may be extended these other cases as well.

While option 3) provides almost no response time to protect the GN&C software from faults, it requires the most complex interaction of GN&C recovery software. An open question left to be resolved is how many faults of this type will need to be covered with this sort of protection. In addition, since the Health Managers may have reloadable parameters like any other GN&C Domain, it is conceivable to use a data-driven specification to define faults as well as their responses onboard the vehicle. In this case, the PSAM schema would be impacted and modified to include the necessary artifacts to parameterize these logical conditions and their responses in a very similar way to how the existing architecture is parameterized. If implemented this architecture would enable the GN&C community to continue to redefine and create new fault response functionality within GN&C through the use of configuration data and extensions to the PSAM database over the life of the program.

Displays & Control Interactions, Manual Commanding

The Orion Display & Control requirements are currently under development and these requirements will impose additional complexity within the GN&C subsystem. One area of complexity may be in crew initiated Activity commanding. It is currently uncertain as to the level of Activity commanding that will be required by the onboard crew. At a minimum the crew will have the capability to inhibit and enable Activity sequencing, as well as perform a series of manually initiated Activities. The modification of specific parameters within the Activity list will also be available for edit by the crew via display interfaces. However, the capability to insert, delete and reorder GN&C Activities within the current or an upcoming Activity list would not be available given the current automation architecture. In this commanding scheme, ground operators will have the primary responsibility to reload any Activity list or CSU configuration(s) if the situation requires such actions.

A more sophisticated commanding capability might include features such as insertion, deletion and possibly reordering of Activities within a current or future Activity list when none of the pre-planned contingency Segments[1] suit the situation at hand. This functionality would permit onboard or ground operators to modify Activity lists when unforeseen events occur, however this capability would come with an increased risk that an Activity list could mistakenly put the vehicle into an unsafe configuration or cause the Mission Segment objectives to become unachievable. Insertion, reordering and capabilities of this nature inherently make validation much more difficult to perform. This type of commanding capability has to be weighed against the cost of the extra software validation it would require.

To support such features not only would the display software have to support it, but GN&C Activity sequencing would likely need to become more sophisticated as well. Figure 11 depicts a modified Activity sequencing flow chart with both entrance criteria as well as exit criteria associated with every Activity. The extra entrance criteria would be needed in this case to collect all of the preconditions which must be satisfied before the Activity can be issued to the GN&C Domains. Contrasted with Figure 4 shown earlier, the primary benefit is that specific conditions can be more readily associated with the Activity with which they belong, rather than being collected together in one larger and more complex logical expression called the 'transition criteria.' Note that permitting this extra logical association would add one additional logical association to the Activity parameter specification depicted in Figure 8. Making this extra logical association makes it explicitly clear what preconditions must be achieved to perform each GN&C Activity as well as what confirmation / indication conditions are required to proceed onward in the list. These associations make it more likely that an operator would be able to confidently insert, remove or reorder Activities without having to make changes to the remainder of the data in the Activity list. It provides a means by which each Activity may be logically isolated from the others in the list and thereby reordered, or removed without being required to make complex changes to multiple elements of the schema.

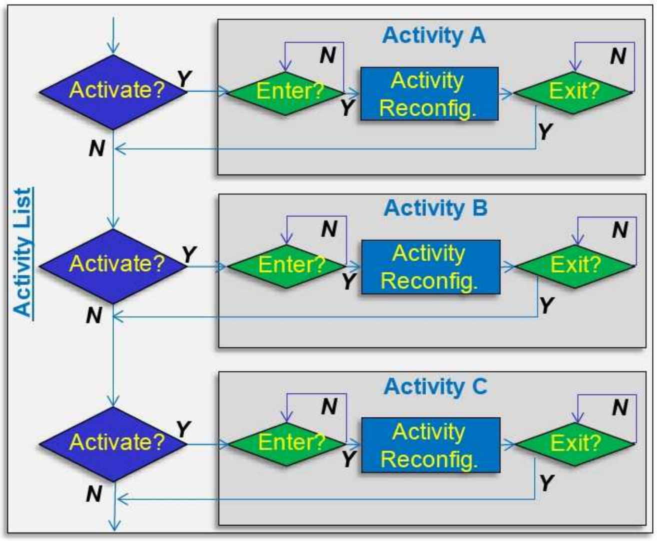


Figure 11: Activity sequencing logic with entrance criteria, exit criteria and activation (skipping) logic. Providing the capability for sequences to specify entrance as well as exit criteria makes it operationally feasible to consider Activity insertion and reordering features.

As an example of how this new association schema aids in Activity insertion, recall the Burn Segment Activity list shown in Figure 6. In this example the first transition criteria is listed as "attitude maneuver complete" && "mission elapsed time" >= "time of ignition." In this case, the first condition clearly belongs as the exit condition to the first Activity (the attitude maneuver), while the second condition belongs as the entrance condition to the "Burn" Activity. Separating these conditions permits an operator to conceivably insert an Activity between them without having to make changes to the preceding or subsequent Activity in the list. If these conditions were to be kept within one transition criteria logic statement, it would be necessary to modify the attitude maneuver Activity for the list to function as intended and make such a function much more difficult to perform operationally. The capability to add GN&C Activities will be evaluated in more detail as the design matures and human-in-the-loop evaluations are conducted going towards CDR.

7. Conclusions

This paper has presented an overview of the flight software architecture as it relates to the latest design for automation implementation within the GN&C subsystem for Orion. The hierarchy of configuration elements was presented to clarify a basic concept of operations for the automation software as well as how the automation capabilities will be parameterized and maintained by GN&C analysts and mission planners.

Several forward work items were detailed on the path to the project CDR. First, the interactions between the vehicle-level FDIR software as well as the distributed GN&C FDIR software must be further understood and integrated. Secondly, the interactions between GN&C and the vehicle-level configuration management software must be further understood and documented. Finally, the command and control interfaces with displays and GN&C automation must be designed. The GN&C community recently established a simulation environment for testing these capabilities in a closed-loop engineering simulation and more fidelity will continuously be added to these simulation environments to aid in driving out the correct functional allocation for all of these software functions.

Acknowledgements

There are several people and organizations we would like to thank… Funding provided by...

References

[1] Hart, J., King, E., Miotto, P., Lim, S., "Orion GN&C Architecture for Increased Spacecraft Automation and Autonomy Capabilities" AIAA Aug 2008

[2] King, E., Hart, J., Miotto, P. "Orion GN&C Executive Architecture for Increased Automated and Autonomous Spacecraft Operations" AAS Feb 2009

[3] Tamblyn., S., Henry, J., King, E. "A Model-Based Design and Testing Approach for Orion GN&C Flight Software Development" IEEE Mar 2