The Space Congress® Proceedings

2004 (41st) Space Congress Proceedings

Apr 30th, 8:00 AM

Paper Session II-B - A Conceptual Model for the Program Driver Environment for Future Spaceflight

Roefof L. Schuiling
NASA KSC, YA-E6

A Conceptual Model for the Program Driver Environment for Future Spaceflight

Roelof L. Schuiling
NASA KSC
YA-E6

Abstract

This paper proposes a conceptual model for the initiation, planning, and execution environment of future human exploratory programs or projects. The conceptual model assumes that program drivers exist and that these extend beyond the technical. Such drivers are often not within the control of program personnel, as requirements definition may be. However, they drive the program by defining the possible. By considering these drivers as elements of sets we may identify them and examine their interactions. The model is presented in Venn Diagrammatic form and the components of the model are identified and described.

The paper addresses the relationships of major parameters which join to form the environment in which major space-related initiatives are conceived, planned, and operated. The paper also provides functional examples of this conceptual model by examining its application in terms of several historical human spaceflight initiatives to show the interrelationship of the model's components. This characterizes spaceflight initiatives in terms of the model's components.

Concept

We would like to examine the factor of major program drivers. These, ultimately, form the environment that drives the requirement determination process. Too often, however, the major program drivers are not fully identified. This leads to a restricted approach to the requirements determination process. In addition we would like to consider the varying program drivers as elements of sets – the sets to be determined by similar characteristics of the program driver elements. The contention is that by such a process the program drivers may be identified; (1) prior to the requirements process, and (2) a more extensive and accurate population of program drivers may be identified.

We may begin by considering the various characteristics of the program development environment as discrete elements. In general these elements having similar characteristics may be collated into a series of sets. As an example; the set of policy elements bearing on a development program may include many aspects of both national and international factors.

A traditional method of illustrating sets of elements so as to provide a conceptual illustration is the use of Venn diagrams. A Venn diagram is a pictorial representation of sets where sets are represented by enclosed areas in a plane. In a Venn Diagram the sets are show in a relation to one another. Usually the convention is to show sets as circular diagrams. In Figure 1 we illustrate two sets: Set A and Set B. No elements of Set A are also in set B and no elements of B are in A.

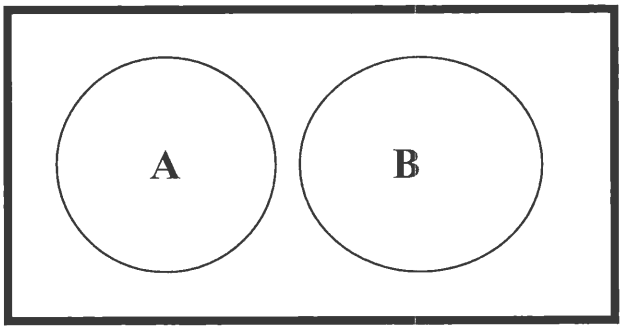

Figure 1. Two Sets, A and B with No Common Elements

Similarly Venn diagrams may be used to show sets with some elements in both sets. Figure 2 shows two sets with overlapping elements which together form a set of those elements in either of two sets and is termed a *union* of two sets. Figure 3 shows two sets with overlapping elements and also represents a sub set made of only elements in both sets at the same time. This is termed an *intersection*.

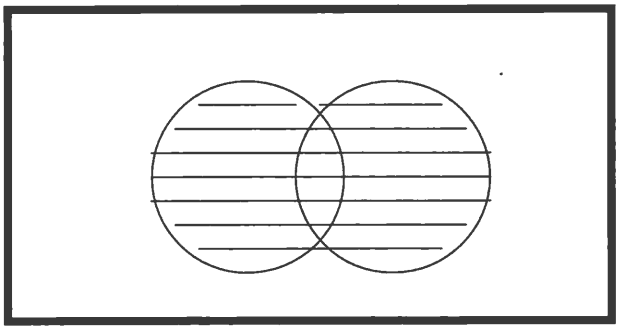

Figure 2. A Union of two Sets

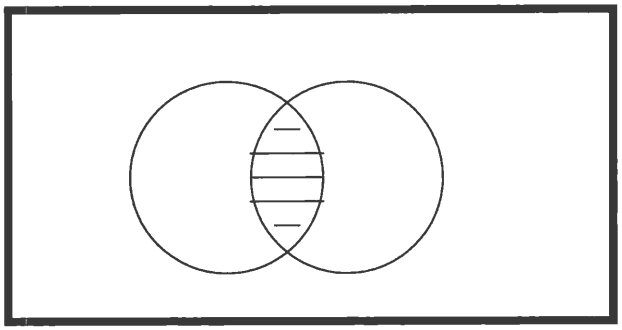

Figure 3. An Intersection of Two Sets

The above illustrations show how the elements of two sets may be depicted; however the Venn diagram presentation concept is not limited to two sets and may depict the relation between the elements of more than two sets. They may also be used to describe sub sets.

Program Examples

Let us examine this approach in terms of some historical examples. We may begin by looking at three sets of major programmatic drivers in our example. In this example we will use sets of technical, policy, and operations programmatic drivers.

Technology program drivers may come in several forms. They may be limiting factors; however, they may also be the reason driving the program in the sense of a program having technology development as its goal.

The set of policy program drivers, ultimately, may have a very large set of elements. Such elements may include Congress committees, subcommittees, appropriations subcommittees, authorization subcommittees, multiple executive branch agencies, civil and defense agencies, commercial, and international policy issues.

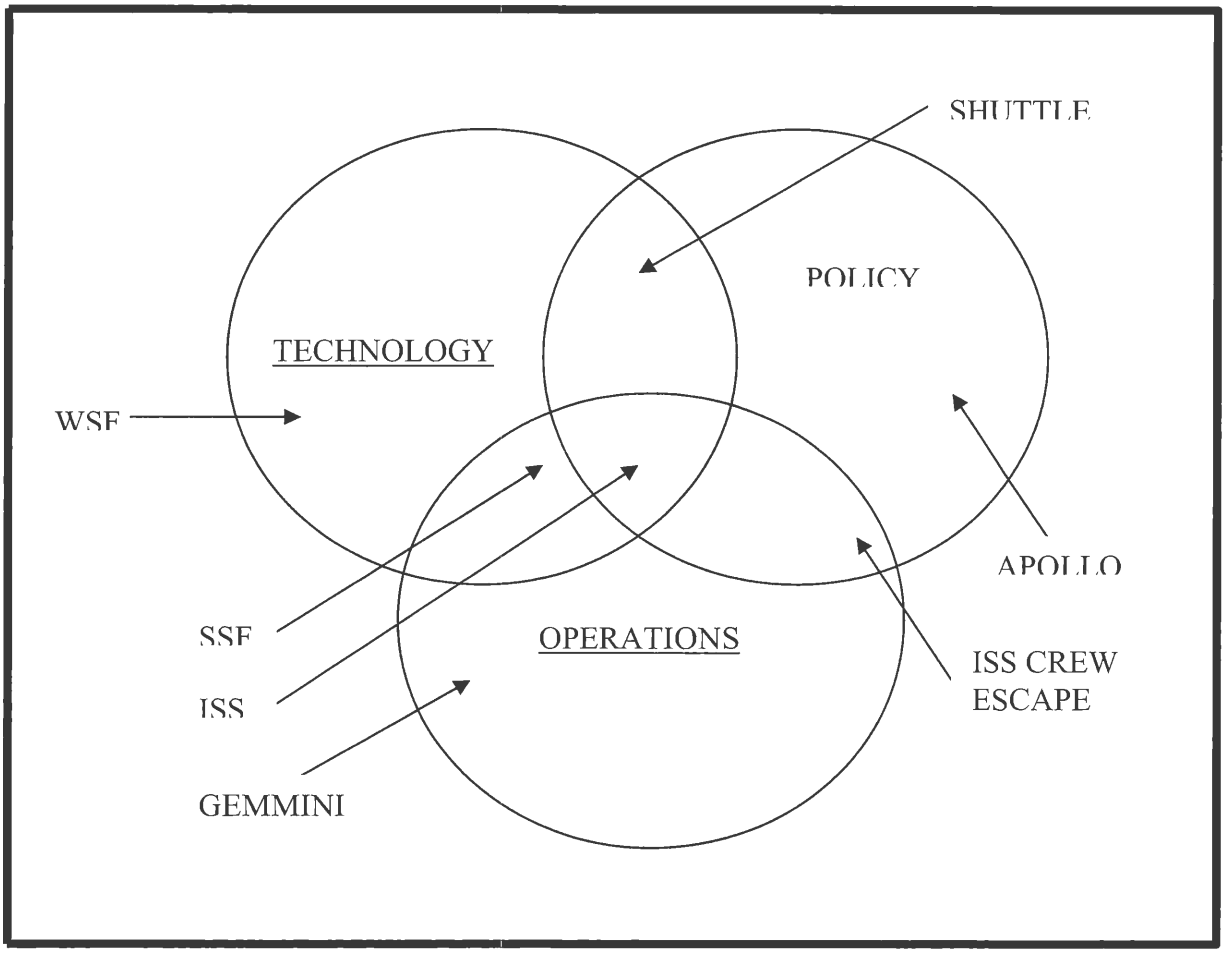

Figure 4. Examples of Program as a Function of Major Program Drivers

Operations programmatic driver set elements may also include a variety of elements such as payload recovery or non-recovery, single or multiple missions, mission goals, A three-set Venn Diagram of program driver elements might look like Figure 4. This shows how a number of example programs Venn Diagrams might appear.

Although there are certainly technology, operational, and policy aspects of these example programs in Figure 4, we are concerned here with the programmatic drivers – not merely aspects of the programs. For instance:

WSF: The Wake Shield facility was an example of a program with a set of program drivers that were basically technology in that WSF hoped to develop a technology for producing extremely pure materials on orbit. This involved extending a circular shield up from the Shuttle's payload bay and producing the material in the vacuum behind that shield so as to avoid any residual molecules at orbital altitude.

GEMINI: Gemini is an example of a program with basically operations drivers. Its goal was to demonstrate operations concepts that would later be used in the Apollo program.

APOLLO: The set of program drivers for Apollo were primarily policy-oriented. Although there were great technology and operations challenges, the program driver was the demonstration of superior American capabilities in space in a Cold War political environment.

SHUTTLE: The Space Shuttle program drivers for a set that is the intersection of technology and policy sets. Development of reusable launch vehicle technology meshed with the policy decision for the government to abandon commercial ELV launchers as well as to offer delivery of payloads to LEO with a government-developed launch vehicle.

ISS CREW ESCAPE: This program effort was characterized by sets of drivers involving operations, in that a safe crew return from an existing operational environment, as well as policy drivers in the wish to enhance utilization of the ISS.

SSF: The Space Station Freedom program, despite a rather political name, was driven by the intersection set of technology and operations driver elements. Operational drivers involved the use of orbital flight to produce reduced gravity as well as variable centrifuge gravity environments together with developing the capability to operate with permanent human crews in LEO. Elements of the set of technology drivers included the development of technologies from microgravity research as well as the development of procedures and capabilities for orbital assembly and construction.

ISS: The intersection sets of drivers characterizing a program are not temporally fixed and may change over the life of the program. Perhaps no better illustration of this could be found than the change of the space station program from the Space Station Freedom to the International Space Station. Although the program driver elements of the Space Station freedom did not disappear when it became the International Space Station, the major change was the incorporation of an additional subset of policy-related drivers. The result was to produce an intersection set of ISS program drivers made up of elements of the technology, operations, and policy sets. This was characterized by the policy drivers of bringing additional international partner – basically Russian – involvement. Although

this policy addition forced additional operational aspects – the need to vary the orbital inclination of the ISS from that of SSF – the added program drivers were political.

Obviously we are interested in the intersections of the sets of program drivers as the unions would include all possible driver elements within the sets.

Thematic Variations

In that a set may have a number of elements, these may also be graphically described and may have relevant unions. An example of several sub sets that make up the set of policy-oriented elements is shown in Figure 5.

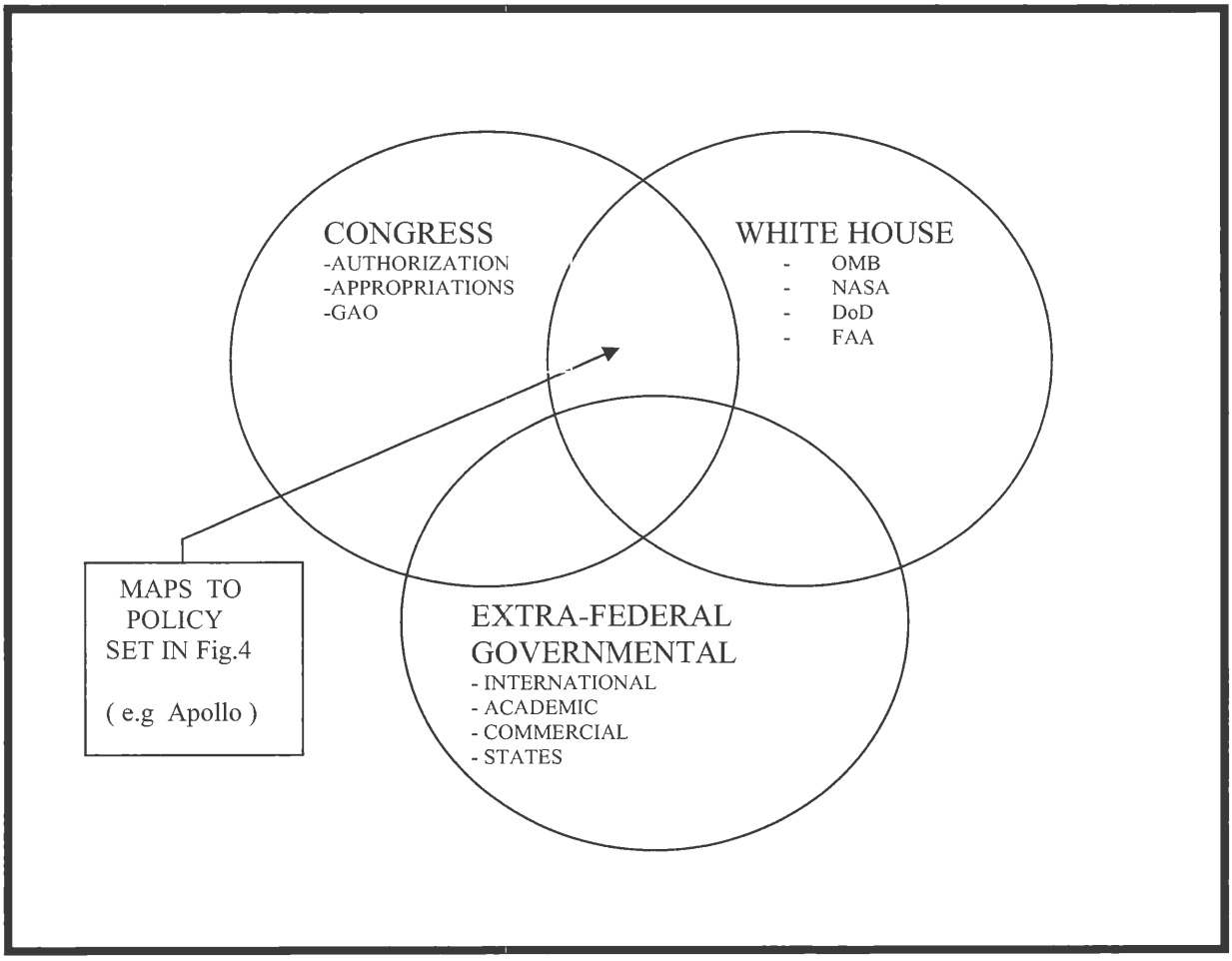

Fig. 5. Example of Policy Set with Subsets Designated

Are sets other than those three above applicable? Certainly. A typology of sets can be developed for any specific program that may vary from those above. These might be sets

with elements made up of program divers characterized by resources, timeline, or other characterizations as well as those above.

Applicability

Looking at the above example might be interesting from the standpoint of the past. It may be of interest to space historians and space policy wonks. It may help such individuals develop a typology of drivers for past programs and form a basis that would allow them to make comparative analyses of past programs. However, what applicability is it to planning, defining, and characterizing future programs?

Well, the above example has actually been backing into an analytical approach rather than going in from the front. The historical nature of the above examples are responsible for this. In relating to proposed future programs, however, we would be entering the analysis from the front. We would no longer be defining how past programs fit into their universe of program drivers as an illustration, but we would be defining what program drivers will affect the program over its life cycle. As we saw above with the Space Station Freedom's evolution into the International Space Station, programmatic drivers may vary extensively over the life cycle

Identification and characterization of the driver elements will not be an easy task. Potentially large numbers of such may have to be considered. However a rigorous analytical effort to identify them and to develop the sets of program drivers beforehand can enhance the probability of success of the program. This effort would define the environment within which program requirements must be defined. It would also serve as a forcing function such that requirements in areas that would normally be overlooked would now be identified and considered. The program drivers making up the various sets would also now be identified and considered by an algorithmic process rather than a less stringent approach.

With the identification of the programmatic drivers and their characterization as elements within sets we might have the opportunity to influence the program requirements process. If we were to identify the definition of program requirements with their characteristic numerical levels, this analytical exercise could almost be characterized at the Level "Minus One" level in comparison to traditional requirement definition.

In addition to defining a more extensive environment of program drivers such an analysis may also identify potential programmatic stumbling blocks and dead ends. Also, the analysis may surface conditions characterized by program drivers with mutually exclusive or contradictory natures. This may provide an opportunity such that the program might be modified at an early stage rather than having to do so after the program is in development.

Putting this into Practice

The next step in developing this analytical approach would be to propose a mythical spaceflight program as a model. Elements of real programs may be incorporated in this as a check, if desired. The exercise of identifying the detailed program drivers making up the relevant sets would then have a simulated specimen. The key to this effort would be to develop the capability to do so in detail, with rigor, and for the elements to be comprised of realistic program drivers. Following this development, a requirements definition effort could be simulated to assess the effectiveness of the approach.

The next step would be the development of a software-based system that could emulate the universe of program drivers and separate them into sets applicable to a potential program throughout its life cycle. This would also allow the early identification of mutually conflicting program drivers at a very early stage. The output of this effort could serve as an entrance to the program requirements definition for various real programs. As we saw above with the Space Station Freedom evolving into the International Space Station, such a process would have to have iterative capabilities such that it could be examined with varying time-related outputs.

Summary

Spaceflight programs are subject to a program driver environment that affects the conceptualization and development process. The early identification of these drivers acts to ensure realistic program requirements development. By characterizing the potential program drivers they may be conceptualized as elements of specific sets. These may then be represented so as to result in a Venn diagrammatic representation of those sets that accurately defines the program driver environment and leads to a more accurate requirements development process.

By providing this process in a software-resident status it may be modified in accordance with potential changes in any of the program drivers so as to indicate how the change might affect the program's requirements. In addition, the exercise may be run at differing time-related planned points in the program's development so as to identify how changes might arise.