

ARCHITECTURE FOR PAYLOAD PLANNING SYSTEM (PPS) SOFTWARE DISTRIBUTION

11-61-CR

Jeff Hagopian
Advanced Planning Branch (EO45)
Marshall Space Flight Center, AL 35812

Eric Howell
Boeing Defense and Space Group
Huntsville, AL 35824

ABSTRACT

The complex and diverse nature of the payload operations to be performed on the Space Station requires a robust and flexible planning approach, and the proper software tools to support that approach. To date, the planning software for most manned operations in space has been utilized in a centralized planning environment. Centralized planning is characterized by the following: performed by a small team of people, performed at a single location, and performed using single-user planning systems. This approach, while valid for short duration flights, is not conducive to the long duration and highly distributed payload operations environment of the Space Station. The Payload Planning System (PPS) is being designed specifically to support the planning needs of the large number of geographically distributed users of the Space Station. This paper provides a general description of the distributed planning architecture that PPS must support and describes the concepts proposed for making PPS available to the Space Station payload user community.

1. INTRODUCTION

The key to the success of any tool is not how well it performs its functions, but whether it performs the right functions well. There are a number of existing planning tools which have been used to support centralized planning; however, these tools were not designed with distributed planning in mind. The tools needed for Space Station distributed planning must have the following characteristics:

- Graphical interface which is easy to learn and use
- Ability to run on commonly available hardware/software platforms
- Powerful data managing and reporting mechanisms
- Functions which allow the scheduling problem to be subdivided and distributed among the users of the Space Station
- Scheduling and modeling algorithms which conceal the complexity of the underlying Space Station resources.

These tools will allow each user group to participate in the planning for their experiments without requiring detailed knowledge of the Space Station systems required to support those operations. This paper focuses on the software distribution aspects necessary to provide this capability.

2. PLANNING CONCEPT

Because of the diverse and dynamic payload complement, no one organization will have the knowledge and expertise required to perform detailed planning for all payloads on the Space Station. Since the knowledge and expertise is spread across a number of control centers and payload user facilities throughout the world, it makes sense to distribute the planning as well. While there are many possible ways of supporting distributed planning, the hierarchical distribution of resources appears to be the approach which is best suited for Space Station payload operations planning.

Figure 2-1 provides an overview of the architecture which supports this approach. This architecture consists of various levels of planning, where the functions of a particular level are performed by one or more organizations. Rather than expressing this concept using Space Station specific terminology, the concept is described in general terms which can be applied to other distributed planning problems.

In general, there are three basic levels of planning: 1) Upper Level Planning Function (ULPF), 2) Intermediate Level Planning Function (ILPF), and 3) Lower Level Planning Function (LLPF). The ULPF represents the controlling authority and is ultimately responsible for the integrated plan of payload operations. There is only one ULPF, although there may be many organizations which support its functions. The LLPF rep-

resents the individual users of the Space Station. These individuals have specific payload operations which need to be scheduled, and are in competition with one another for the limited resources available to support those operations. The ILPF represents the organization or organizations which serve as the interface between the ULPF and the LLPF. In most cases, the ILPF represents the sponsoring organization or country of the users. In cases where there is no ILPF organization, the LLPF interfaces directly with the ULPF. There may be multiple ILPF levels, where one ILPF organization exists to serve the ILPF organizations which fall under its authority. Refer to Figure 1 for a pictorial representation of this architecture and the relationships between the ULPF, ILPF, and LLPF organizations.

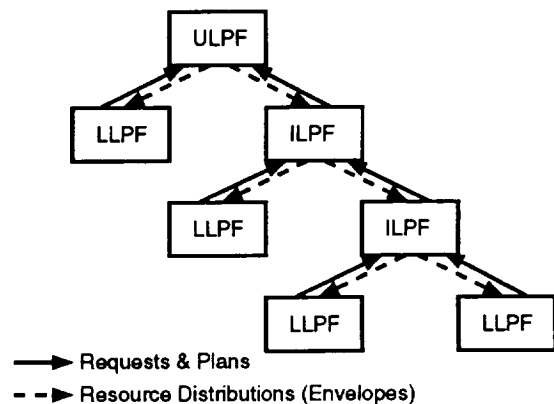


Figure 2-1 Architecture

The basic premise of this concept is that resources are distributed in a manner which allows for *concurrent and independent* planning at each level in the architecture. Requests for resources are passed from the LLPF upwards through the ILPF level(s) to the ULPF. The ULPF, taking into account all of the requests for resources, distributes the available resources to the ILPF. Each ILPF then distributes its resources to the level below it, either another ILPF level or

the LLPF. At the LLPF level, the users develop plans within their resource distributions and pass those plans back up through the path to the ULPF. Each level, by having visibility into all of the requests for resources from the levels below, can ensure a distribution of resources which best satisfies the needs of its users. Conversely, since each level of the planning hierarchy is isolated from all other peer levels, planning can be accomplished without regard for work being done at other planning centers. This independence allows the total mission planning problem to be distributed across many planners, which decreases the amount of wall-clock time required to develop schedules. The flow of this information from one level to the next is depicted in Figure 2-1.

3. SYSTEM DISTRIBUTION

There are two principal options available for distributing a computing problem among several processing sites. The first is to pass the partial and evolving solution between sites until everyone is satisfied with the results. In general this tends to serialize the process and can require a large amount of information to be explicitly transferred between sites. The second option is to centralize the storage of the information and arrange it in such a way that many users can work on their data simultaneously without interfering with other users' work. The key problem here is in making all the information readily available to the appropriate users. The architecture that implements this solution is called "*client-server*". The following sections provide an overview of client-server systems, a description of the PPS client-server design, and the different ways this design can be implemented by the various users of the payload planning system.

3.1 CLIENT-SERVER OVERVIEW

A client-server architecture provides for re-

source sharing between systems. The server in such an architecture provides services to other systems. The client is the system which requests the services. There are two immediate advantages of this architecture. Since there is one server for all the data, it can apply centralized data managing techniques such as a relational database management system or a file management system to the data under its control. These solutions are readily available and provide substantial control over and protection for shared data. The second advantage is the independence of the client from the server. Any hardware/software combination which can effectively interface with the server can help process data in the client-server system. As shown in Figure 3.1-1, three different hardware/software platforms are connected as clients to the server, which provides them each with a view of the database at the top of the figure. This allows substantial flexibility to the users of this system, since they can each use hardware they own and are familiar with and which has been sized properly for their portion of the problem.

In developing a client-server system the following items should be kept in mind:

- The system should be sized to support the anticipated number of users
- The system should be sized to manage the appropriate volume of data
- The system should not be sensitive to the addition or removal of clients.

The operation of a client-server system is also heavily dependent on the network which transfers information between the clients and the server. A server can generally support several methods for network communication with clients, helping to further isolate the hardware dependencies from the information processing.

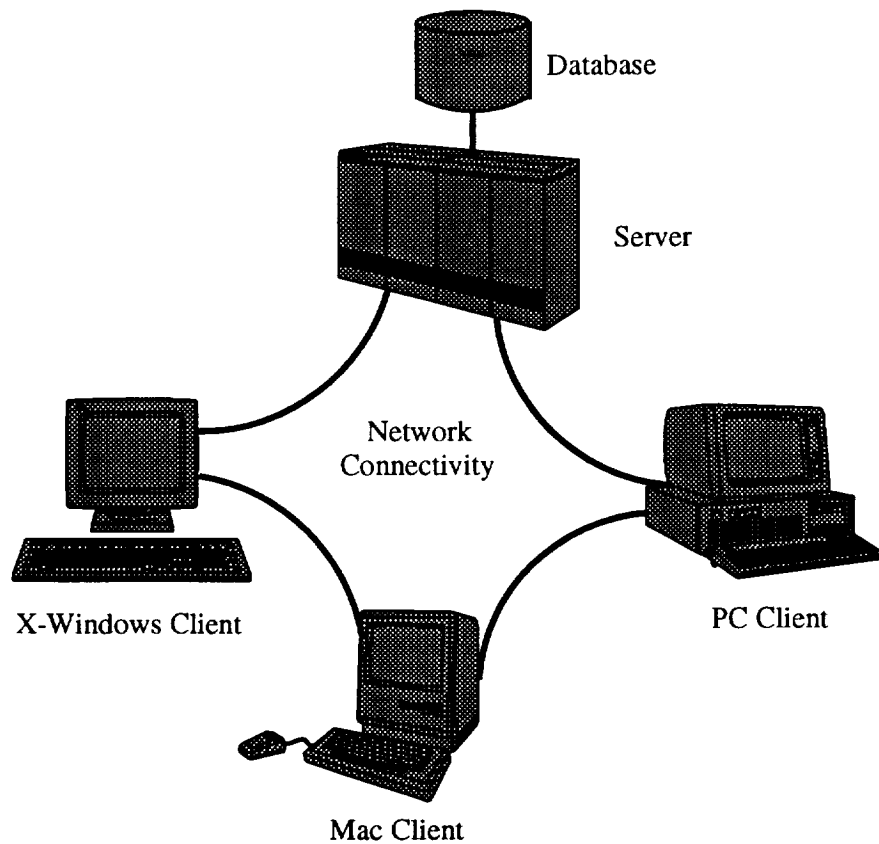


Figure 3.1-1: Client-Server Overview

3.2 PPS CLIENT-SERVER DESIGN

PPS is not a single application, but a collection of applications which together comprise the entire system. These applications work together to solve the planning problem through the sharing of data. Figure 3.2-1 provides an overview of the PPS architecture. The external systems shown on the figure represent those systems with which PPS must exchange data. These external system interfaces are subject to change as PPS matures and as new systems come into existence.

Within the PPS network, the PPS application may be found on any number of hosts. Each of these applications interacts with the database through a common data server. In this manner, the majority of the processing workload is shifted from the server to the in-

dividual hosts on which the PPS application is being executed. Each host on which the PPS application resides can support a number of simultaneous users. The size and number of hosts required to support the planning process will be determined by the number and types of users the host must support.

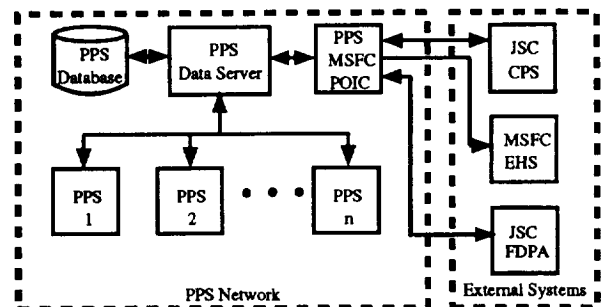


Figure 3.2-1: PPS Architecture

3.3 PPS CLIENT-SERVER IMPLEMENTATION

The PPS client-server design is very extensible, and provides individual users as well as groups of users with several options for utilizing the system. The following section provides a sampling of the ways in which members of the user's community can fit into the client-server environment supported by PPS.

3.3.1 Remote User Access

The remote user access implementation provides a low cost method through which individual users can be provided with access to PPS. In this implementation, the user provides an x-windows terminal. The PPS application resides and executes on a Space Station program provided host. Refer to Figure 3.3.1-1 for a graphical representation of this option. The x-windows terminal merely

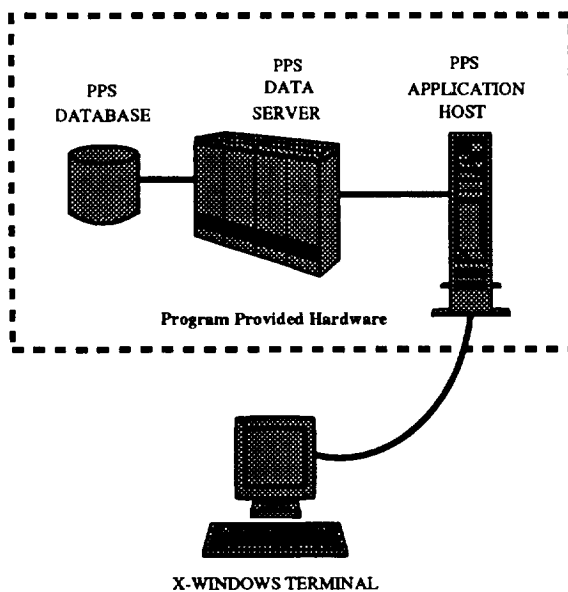


Figure 3.3.1-1: Remote User Access

serves as the display device for the PPS application. While this is a fairly low cost option for individual users, it is not without drawbacks. The program provided host may be required to serve large numbers of users which ultimately will affect response time.

The number of users that can be supported by this implementation will be defined to a large extent by the size of the host provided by the program. The response time will also be affected by the type and throughput of the networks being utilized by the user.

3.3.2 User Provided Host

The user provided host implementation is more costly to the user; however, this option can provide significantly better response time. The user in this case not only provides the display device, but also provides a host for the PPS application. This eliminates competition for processing time with other PPS users. Communications is still a concern, as the PPS application must request data over the network from the PPS data server. A drawback to this implementation is that the user is responsible for the systems management and maintenance of the local host. Refer to Figure 3.3.2-1 for a graphical representation of this option.

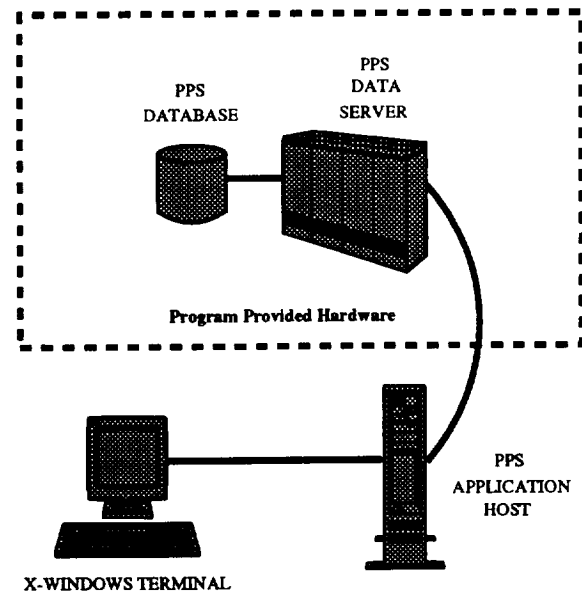


Figure 3.3.2-1: User Provided Host

3.3.3 Facility Provided Host

The facility provided host implementation

may provide the most cost effective and efficient option for user access to PPS. A facility can be any organization which sponsors or supports a number of users. In this implementation, the program provides the PPS database and the database server, the facility provides the PPS application host, and the user provides the x-windows terminal. In this manner, the costs associated with purchasing and maintaining the PPS application host can be spread across a number of users.

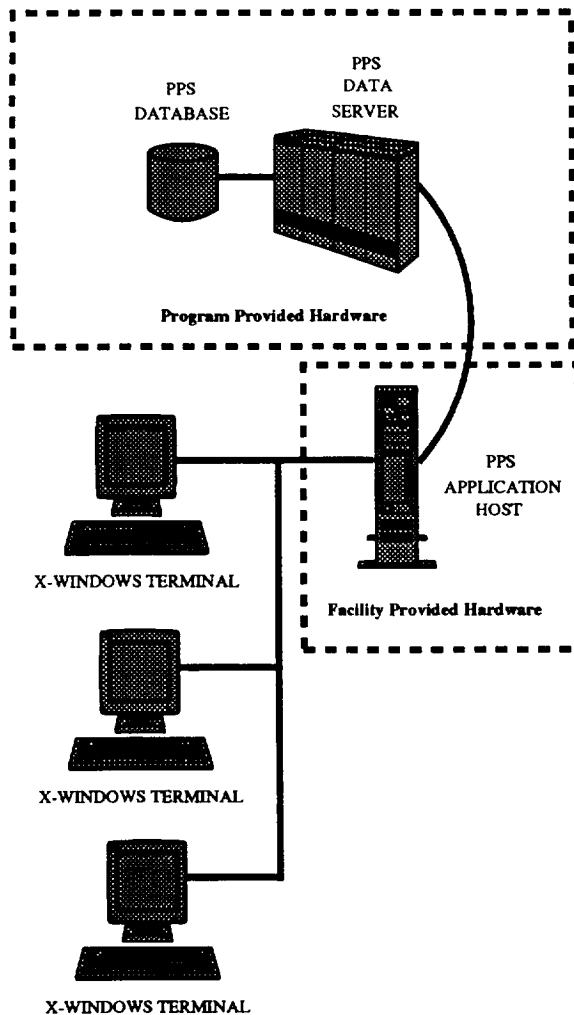


Figure 3.3.3-1: Facility Provided Host

The host can be sized specifically for its intended users. Response time for each user should be adequate if the ratio of users to host size is appropriate. Refer to Figure 3.3.3-1 for a graphical representation of this

option.

4. SUMMARY

[to be rewritten] The general approach to Space Station distributed mission planning was discussed, along with generic client-server architecture and specific configuration options for PPS. Every user of the Payload Planning System will have the option to configure his planning system assets to best suit his requirements for cost, availability, redundancy, and performance. Users who form cooperative planning facilities will realize the most effective combination of these attributes.

5. REFERENCES

1. Payload Planning System (PPS) System Specification (SS); SW683-70256-1.
2. Segment Specification for the United States Ground Segment (GSS); SSP 41161.
3. Mission Planning Operations Concept Definition (MPOCD) Update: Final Report, Informal; D683-10686-1.
4. Concept of Operation and Utilization (COU) Volume III; SSP 50011.
5. Hagopian, J.; Maxwell, T.; Reed, T.: "A Distributed Planning Concept for Space Station Payload Operations", Third Symposium on Space Mission Operations and Ground Data Systems, Greenbelt, MD., Nov. 15-18, 1994.