COST REDUCTION VIA COMMON ONBOARD SOFTWARE

John A. Rohr
Jet Propulsion Laboratory
Pasadena, California USA
John.A.Rohr@Jpl. Nasa.Gov
(818) 354-4096

March 26, 1995

The Jet Propulsion Laboratory has been building, launching, and operating deep-space probes for more than 30 years. Initial JPL probes were sent to the moon in the 1960's. Subsequent probes flew past nearby planets Mercury, Venus, and Mars. In the 1970's, two Viking probes landed on Mars and two Voyager spacecraft began their journey to the edge of the Solar System. In the 1980's and 1990's, other probes have been sent toward the Sun, planets, and outer space to increase our scientific knowledge of the Solar System.

Each space mission requires an onboard commanding and sequencing capability to direct the spacecraft through various actions to accomplish the scientific and engineering tasks required by the mission objectives. Early JPL spacecraft used programmable sequencers. These devices were hard-wired logic machines which could accept specific parameters. The first special-purpose computing capability consisted of a computer with a limited memory of 128 words and a set of 16 special-purpose, two-address instructions. The same computer was later enhanced by expanding the memory to 512 words. All onboard commanding and sequencing was accomplished within these constraints.

The first JPL spacecraft to have a general-purpose computer were the Viking Orbiter spacecraft. These spacecraft carried two identical computers which implemented the Computer Command Subsystem. Each computer had a 4,096-word memory and a set of 64 instructions, a few of which were dedicated to operating the custom input-output units which were included. Also onboard were dual telemetry computers and dual attitude control computers. Computers almost identical to the Viking (1-biter computers are used on the Voyager spacecraft.

The software used for commanding and sequencing onboard the Viking and Voyager spacecraft utilizes an onboard interpreter driven by a set of tables generated by ground software. The tables implement a language called Virtual Machine Language (VML). This language is used to program the spacecraft to accomplish the scientific and engineering tasks required by the mission objectives. VML is a very efficient language. All of the tables required for commanding and sequencing the spacecraft fit in about 2000 words in each computer memory. These words are reloaded as required from the ground.

The Galileo spacecraft utilizes a Command and Data System which is based on a distributed set of six RCA 1802 computers. This is a significantly different architecture from that of Viking and Voyager. The total amount of memory is approximately 500,000 words. The commanding and sequencing language for Galileo is very different from that of Viking and Voyager, but it is also called VML (for Virtual Machine Language). The Galileo VML implements all the capabilities required for commanding and sequencing onboard the Galileo spacecraft.

The Cassini spacecraft is using an **IBM 1750A** (GVSC) for its onboard computing capability. This is yet another architecture and yet another, different VML has been invented for the mission. other missions have also used VML's which were different from those of previous missions.

The design and use of a different VML for each JPL mission has been expensive both in terms of the design and implementation effort for the language and in terms of the effort required to develop a ground system to support the language. Even though different hardware has been used, many of the functions required onboard are the same. in the future, multiple missions should be able to share one common language for onboard commanding and sequencing.

A study has been recently completed at JPL which investigated the use of a commercial language to provide a common language which could be used on multiple JPL missions. Related work at JPL facilitates building a unique database of spacecraft commands for each mission. A prototype effort is currently underway to demonstrate the feasibility of using a commercially available language for commanding JPL spacecraft. The prototype will implement command generation and execution from the beginning of command sequence generation in ground software through execution onboard a simulated spacecraft. Two different languages and a single database of spacecraft commands will be demonstrated to show that the concept is feasible independent of implementation in any specific language.

The use of a common language for onboard commanding and sequencing should reduce the cost of developing ground software for new missions. JPL has invested a considerable effort over recent years to develop a multimission ground system. The use of a common language onboard for multiple missions would further help to reduce costs by eliminating the need to develop software to support new onboard languages. Software developed to support a common onboard language would require only minimal modification, if any, from mission to mission. Also, all personnel would know the common onboard language and would be able to work on multiple missions or transition from one mission to another without needing to learn another onboard language.

The use of a commercial language as the common onboard language would relieve JPL of the necessity of developing and supporting a language, but some adaptation of both the JPL multimission software and the commercial language itself may be required for all components to work together effectively in the JPL operations environment. By using a commercial language, JPL could take advantage of capabilities in the language not available in the various VML's now being used and JPL could share experience gained during over 30 years of space exploration.

The current schedule for the prototype development calls for a demonstration in August, 1995. By that time, some of the issues involved in using a commercial language as a common onboard language and issues involved in its integration into the JPL multimission operations environment should have been addressed. An up-to-date report will be made at the workshop. Experience with the development of the prototype and the implementation of the two commercial languages as well as the common database of commands will be discussed. Lessons learned and ideas for possible future work will be presented.