TECHNICAL REPORT AE-84-1

SPaCE I

SPACECRAFT PRELIMINARY AND CONCEPTUAL ENGINEERING I

DEFINITION DOCUMENT

By Gary Pearce Barnhard
NASA Graduate Student Researcher
Code 502
Goddard Space Flight Center
Greenbelt, Maryland 20742
(301) 344-7992

January 1984

This work is sponsored by the National Aeronautics and Space Administration (NASA) Graduate Student Researchers Program in conjunction with the NASA Goddard Space Flight Center (GSFC) and the University of Maryland Aerospace Engineering Department. This work is intended to support the on-going Advanced Earth-Orbital Spacecraft Systems Technology RTOP (Research and Technology Objectives and Plans) study at GSFC and to serve as an additional resource for the Mission and Data Operations Directorate activities in the area of advanced automation systems.

TABLE OF CONTENTS

I. INTRODUCTION

This paper is intended to serve as the initial definition document for the construction of a knowledge-based (“expert”) system to aid in the preliminary and conceptual engineering of Earth-orbital free flyer spacecraft. The system, known as Spacecraft Preliminary and Conceptual Engineering I (SPaCE I), is under development at GSFC by the author of this paper in consultation with a wide range of NASA personnel.

SPaCE I, is being designed to perform three major functions in conjunction with a spacecraft systems engineer. The first function is the production of specification sets for spacecraft designs that effectively fulfill a spacecraft’s purpose. An effective spacecraft design is one of the set of spacecraft designs that meet all the definite requirements and the external constraints that relate to the spacecraft system. The appropriate spacecraft design is the design which can be judged to meet the requirements and constraints in the most effective manner. The second function is the generation of comparison reports between spacecraft design specification sets. This function allows “What if?” questions to be answered as well as aiding the selection of the appropriate spacecraft design by providing an evaluation tool. The third function is the generation of review reports which show how each of the definite requirements identified contribute to set of specifications which define the appropriate spacecraft.

SPaCE I, is a prototype system being developed to demonstrate that knowledge based systems can serve as valuable tools to facilitate the spacecraft systems engineering process. A knowledge based system has two components. One is the knowledge base which contains the expert knowledge of the problem domain. The other is the inference mechanism which accepts the user input, interacts with the knowledge base to make inferences, and then produces the output. A schematic of a knowledge based system is shown in Figure 1.

Conventional programming systems that have been used for spacecraft preliminary and conceptual engineering have a number of inherent limitations. The most critical limitation is that they require complete and specific information for the parameters that they consider. They cannot deal with missing parameters or variable parameters. They are totally dependent on the ability of the user to provide a suitable set of parameters for each case to be considered. Another severe limitation is that no explanation is available on how answers are arrived at short of tracing through the program code line by line. Lastly, they cannot generate comparison reports between potential designs. A knowledge based system need not be subject to these limitations.

The motivation for SPaCE I comes from an identified requirement in the advanced Earth-orbital Spacecraft Systems Technology RTOP underway at the GSFC (managed by Paul Studer, Engineering Directorate), of which the author is a participant.[1] The RTOP participants include experts on all sub-systems and functional technologies that are commonly involved in Earth-orbital free flyer spacecraft systems. The sub-systems represented include: attitude control, power, thermal, propulsion, communication, command & data handling, structures and navigation. The functional technologies include: lasers, cryogenics, optics and electro-mechanical. These experts are committed to providing the RTOP with a state- of-the-art assessment of sub-system technologies in their area of expertise, as well as 5 and 10 year technology projections. In addition, they will provide qualitative relationship rules and quantitative relationship functions, where possible, that define the potential interactions their sub-system has with other sub-systems and external constraints. For the purposes of the RTOP, the external constraints are considered to be the environment (all those that the spacecraft system will be exposed to), the payload/instrument package and the requirements dictated by retro-fit and refurbishment needs. This information is then to be assembled into an overall interaction matrix.

In the initial RTOP proposal, the prospect of incorporating the knowledge base that the RTOP would generate into some form of spacecraft modeling system was mentioned. This was later identified as a so called “math model” system which still remained an undefined prospect. This prospect has evolved over the past year into the SPaCE I system now under development. The text that follows uses the development of a plausible model of spacecraft systems engineering as a starting point for defining the SPaCE I system.

[1] Advanced Earth-Orbital Spacecraft Systems Technology RTOP, Goddard Space Flight Center, 506-62-26 (1983).

II. THE SPACECRAFT SYSTEMS ENGINEERING PROCESS

Spacecraft systems engineering is the discipline concerned with assuring the ability of all the sub-systems and payload/instrument components to work together to achieve the spacecraft's purpose in the most effective manner. Spacecraft systems engineering is a piecewise iterative process. It is accomplished by the dedicated efforts of highly experienced individuals and by multiple, and often redundant, review procedures. NASA characterizes the spacecraft systems engineering process as a set of fairly discrete phases which can be associated with a number of factors. These factors include the degree of confidence in the design, the status of the funding, the level of resources committed and the level of detail of the current work (see Figure 2). Spacecraft systems engineering is playing an increasingly major role throughout the lifecycle of spacecraft systems. This trend is being driven by a number of factors including: the increased use of spacecraft optimization constraints (design to cost, design to space available, etc.); efforts to expand the degree of autonomy/automation; provisions for servicing and/or repairing; allowing for a high degree of modularity and the use of standard interfaces; providing for the ability for the spacecraft to be evolved to suite future mission needs; and, to control overall system lifecycle costs.

A high level diagrammatic view of spacecraft systems engineering is shown in Figure 3. The design of a spacecraft system starts with a "purpose". The purpose provides the basis for understanding and evaluating the efficacy of the overall spacecraft system. The purpose serves to constrain the domain of possible spacecraft from a functionally infinite set to a finite set that have definable characteristics. In addition there exists a set of critical parameters (needs) that have been identified for the proposed instruments/payload. This information tends to be an ill-defined mixture, varying from extreme specificity to rather oblique generalizations. Quite often there is competing if not contradictory information supplied. Yet, in spite of the tenuous nature of this information, it provides a necessary starting point for systems engineering process. Some of the critical parameters (also referred to as mission requirements) that often come into consideration are listed in Figure 4. It is by the consideration of the implications of the critical parameters, in light of the spacecraft's purpose, that the definite spacecraft requirements and their relationships emerge. These requirements and relationships provide the basis for the generation of potential spacecraft designs/configurations. In the process of evaluating the potential designs/configurations, revised and/or additional spacecraft requirements emerge. The modifications mandate another iteration of the systems engineering process. When the project management has sufficient confidence in a given spacecraft system design/configuration, such that no further changes are foreseen, the design can be converted into the appropriate specifications necessary for detail design and construction. Any changes in the specifications after construction has begun become increasingly more costly in terms of funds, time and political will.

As the complexity of the spacecraft system being designed increases, the systems engineering process becomes ever more difficult and tedious. This trend of increasing difficulty, inherent in the systems engineering process, has been dealt with by adding additional review levels and analytical depth to the analysis process. However, the difficulties with the current generation of spacecraft and associated hardware (e.g. SMM, LANDSAT D, TDRSS, INDSAT and IUS system) lend credence to the idea that new tools are needed to deal with the growing complexities of spacecraft systems both now and especially in the years to come. In a more general sense, NASA's goal of creating Space Station System may require a whole new approach to spacecraft systems engineering in order to be successful.[2]

[2] NASA LaRC/OAST Space Station Technology Workshop Summary Presentation Document, March 1983.

III. REQUIREMENTS FOR AN EXPERT SYSTEM

NASA and the aerospace industry have built, launched and operated successfully a myriad of spacecraft systems over the past 25 years and continue to do so. Therefore, experts in spacecraft systems engineering do exist. Accordingly, the most fundamental requirement for the construction of an “expert” system is satisfied (i.e. experts have to exist in the domain under consideration).

A second requirement is that is must be possible to determine how the expert works in the domain. In the previous section, a plausible model of the overall spacecraft systems engineering process was presented. This model applies from the genesis of the spacecraft purpose through the actual construction process, and by analogy, through to the end of the lifecycle of the spacecraft system. This is because, after a spacecraft system is launched, decisions are continually made that have ramifications which often cascade throughout the entire system in a non-trivial manner; hence, the continuing role of spacecraft systems engineering.

A third requirement for the construction of an expert system is a suitable knowledge base. In the case of spacecraft systems engineering, the knowledge base exists as many scattered elements in a variety of forms including rules, qualitative description, and quantitative functions. Unfortunately a significant amount of the knowledge has never been codified; it exists only in the minds of the experts involved. Furthermore, the knowledge also exists at many different levels of refinement, particularly with respect to quantitative functions/tools. For the purposes of SPaCE I, the knowledge base will be supplied by the Advanced Earth-orbital Spacecraft Systems Technology RTOP participants (see list in Figure 5). The knowledge base supplied will be supplemented to the extent necessary to provide for satisfactory prototype development. The critical consideration is that a knowledge base does exist and will be put in the form required to allow it to be incorporated into SPaCE I.

A further requirement that must be satisfied is that a functional inference mechanism must be available. In order to deal with the incomplete information that will be presented to the system, and which must be manipulated in a variety of forms, a number of inference capabilities must be present. A number of domain independent expert system construction “toolkits” exist that may be applied directly or used in conjunction with other research programs as models for the construction of a suitable overall inference mechanism (or set of cooperating inference mechanisms).[3] [4] [5] The inference mechanism required for SPaCE I is considered to be well within the state-of-the-art, and by most assessments, a straight forward piece of AI applications work.

A fifth requirement that is often difficult to come to terms with, is deciding what the program is really expected to do. In artificial intelligence applications work, the capabilities that an expert system must have are clearly defined. The system either performs successfully or it does not. The success criteria are fundamentally different for artificial intelligence research work. For research work the capabilities of an expert system must have are fluid goals subject to revision as the researcher feels appropriate. Accordingly, both a conceptual understanding of the system’s purpose and a definite outline of the expected output is required. The SPaCE I system meets this requirement.

A sixth requirement is that a workable architecture/process can be defined. This involves defining what must be in the knowledge base, what capabilities the inference mechanism must have and how both must interrelate to function properly. The structure of the knowledge base, and the processes needed to produce a viable expert system, represent the most fundamental AI problem addressed by the SPaCE I system. Creating a system that can cope with incomplete and multiple levels of knowledge in a real time environment, and function as a meaningful aid in the preliminary and conceptual engineering of spacecraft systems, is a non-trivial problem. Until now knowledge based technologies have not been applied in this domain.

The last requirement considered here is that a tractable implementation scheme exists. The hardware must support the knowledge based system within an operational environment that allows for the system to be developed, tested and evaluated without unreasonable constraints of memory space, CPU time or access. Furthermore, an adequate resource of qualified personnel and support funds is required to make the system operational. Due to the nature of the applications environment, SPaCE I must

---

[3] Reggia, J.A.; Perricone, B.T.; KMS (Knowledge Management System) Manual, University of Maryland Department of Computer Science TR# 1136.

[4] Waterman, D.A.; Hayes-Roth; An Investigation of Tools for Building Expert Systems, Rand Report R-2818-NSF, 1982.

[5] Gavarter, W.B.; An Overview of Expert Systems, NBSIR 82-2505, 1982.

have early demonstrable capabilities and show noticeable growth, as additional resources are committed to it.

IV. SPaCE I SYSTEM OVERVIEW

For the purposes of SPaCE I, a spacecraft is defined in one of several ways. A spacecraft exists as both an abstract concept which has a qualitatively defined purpose and as a collection of specified components which have known quantitative characteristics. The SPaCE I system seeks and appropriate synthesis between these two descriptive extremes. The SPaCE I system presumes that the spacecraft designer knows the purpose of the spacecraft and the critical parameters of the payload/instruments. SPaCE I is intended to better enable the designer to define an appropriate support system. Such a system maximizes the ability of the payload/instruments to achieve the mission purpose within the constraints imposed on the system. This paper will henceforth refer to the support system mentioned above as the "spacecraft". The overall system, including the payload/instruments, shall be referred to as the "spacecraft system". An example of this hardware dichotomy is Landsat D which uses the Multimission Modular Spacecraft and a separate instrument module (see Figure 6). SPaCE I does not address the design of the payload/instruments directly; it must however, meet their specified needs and document how those needs relate to the design of the appropriate spacecraft.

The spacecraft consists of a related set of sub-systems which together perform the functions critical to the operation of the spacecraft system. One example is the Multimission Modular Spacecraft which consists of a set of modules with known interfaces (see Figure 7).[6] The system has modules for: attitude control, power, command and data handling, structural support, and two propulsion modules. The Multimission Modular Spacecraft has been used for both the Solar Maximum Mission and the Landsat D spacecraft systems. As well, the modules are being considered for use in a number of future spacecraft systems (Upper Atmosphere Research Satellite (UARS) is one example). The trend toward modularity in the spacecraft sub-systems is firmly established and is expected to become the rule, rather than the exception, in the years to come. The importance of this example is that a spacecraft has been built which can be used in multiple configurations.

[6] Multimission Modular Spacecraft External Specifications Document, GSFC 1978.

Hence, it is possible for individual sub-system interfaces to be clearly defined independent of position. Furthermore, the interfaces can be defined for arbitrary sub-system juxtapositions in consideration of external constraints.[7] This is the key to the modeling system used in SPaCE I.

The sub-systems considered by SPaCE I are treated as “black boxes” which have defined surfaces (idealized geometric or actual) and interactions that are not limited by spatial adjacency. The sub-systems can therefore be considered as discrete components of the model structure. The sub-system breakdown used by SPaCE I is: attitude control, power, thermal, propulsion, communication, command and data handling, structure and navigation.

In addition to the discrete components of the model structure, two additional components exist. The first consists of distributed components which comprise a set of functional technologies that present specialized problems to the designer. These include: laser, cryogenic, optic and electro-mechanical technologies. The second consists of external constraints. These constraints stem from user supplied information that cannot be inferred and/or data this is best accepted by fiat. The external constraints include: critical parameters of the payload/instruments, retro-fit and refurbishment requirements, and the environment. The environment is considered to include information on the prelaunch, launch interface, post-launch, orbit transfer and on-orbit environment as appropriate.

The discrete, distributed and external constraint model structures represent a first level of a definition taxonomy for spacecraft (see outline in Figure 8). These structures define the problem boundary of the SPaCE I system (see Figure 15). The elements of these structures will contain the majority of the SPaCE I spacecraft systems engineering knowledge base.

The elements correspond to the areas that the Advanced Earth Orbital Spacecraft Systems Technology RTOP group has designated to be included in the overall spacecraft interaction matrix being produced. Accordingly, SPaCE I is designed to interact with a knowledge base of that format.

---

[7] Barnhard, G.P.; Plans for a Spacecraft “Math Model”, unpublished presentation first given 12/9/82.

An example of an incomplete version of the spacecraft interaction matrix including only some qualitative relationships is shown in Figure 9.[8] Once the matrix reached a considerable degree of completeness on a qualitative level, coping with the complexity of the interactions appeared to be a most formidable problem. A solution to the problem was found by considering the nature of possible interactions between the elements that make up the interaction matrix. It was postulated that there should be some reference frame by which the interactions could be viewed that would allow them to be reduced to a conceptually and computationally practical set. A suitable reference frame was found by considering the interactions shown on the matrix to be "flows". These flows all belong to a taxonomy (shown in Figure 10) which has been determined to have twelve constituents under three basic genres.[9] Each constituent, or flow type, can be defined by filling in a suitable "frame"[10]. The information necessary to understand the nature of each type of flow is independent of where the flow is from and where the flow goes to. Suitable frames can be constructed for each of the possible flow types. A partial example of a possible electrical flow frame is shown in Figure 11.

Each of the flow types has a flow matrix whose elements are separate instantiations of the flow type frame (see outline in Figure 12) and are labeled as individual flow specifications. The collection of flow specifications comprises a data base which is used to construct flow models using the functions/rules resident in the knowledge base. A complete flow model consists of a flow specification, the function/rule invocations (list of activated rules), the entry points into the documentation that give the rational for the invocations and the citations on additional information sources (see outline in Figure 13).

The completed flow models, merged with the other knowledge obtained or inferred by SPaCE I, allow for the synthesis of an overall model of the spacecraft system (see outline in Figure 14).

---

[8] Studer, P; Earther-Orbital Spacecraft Systems Technology Preliminary Plan, GSFC, Nov. 1982.

[9] Barnhard, G.P.; Plans for a Spacecraft "Math Model", unpublished presentation first given 12/9/82.

[10] Minsky, M; A Framework for Representing Knowledge, MIT AI Lab Memo No. 306.

The SPaCE I “expert” system schematic is shown in Figure 16. Top level descriptions of the schematic components (Input, Knowledge Base, Inference Mechanism, and Output) follow.

The regular user input to SPaCE I will include a variety of forms. These forms include both qualitative descriptions (using a restricted sub-set of natural English relevant to spacecraft systems engineering) and quantitative descriptions. A number of different quantitative descriptions are possible. They include:

1) Fixed Values – invariant quantity
2) Target Values – preferred quantity
3) Range Values – upper and lower bounded quantity
4) Minimum Values – minimum quantity
5) Maximum Values – maximum quantity
6) Function Defined Values – quantities given by a function
7) Curve/Tabular Values – quantities given in tables/curves

The user must designate the appropriate description for their input. The input required includes: the instrument/payload description and the operating parameters; the retro-fit and refurbishment goals; and, the time horizon (initial operational capability).

The Knowledge Base will consist of a number of elements that will be considered to be discrete for the purposes of system design. The elements must provide the knowledge necessary to accomplish the following:

1) Interpret the user input
2) Delineate the definite spacecraft requirements
3) Define the appropriate spacecraft specifications for the set of spacecraft designs that can meet the definite requirements of the spacecraft system.
4) Generate comparison reports and any other possible outputs

Descriptions of the Knowledge Base elements presumed to be required for SPaCE I may be found in Section XI. The specific analytical tools that the knowledge base elements need to allow SPaCE I to function successfully are only alluded to in this paper. Considerable detail on the tools available can be found in the works cited in the bibliography (Section IX) both from a computer science and from systems engineering perspectives. The determination of which qualitative and quantitative tools should be included is a continuing problem with "expert" system design. For a spacecraft preliminary and conceptual engineering "expert" system to be viable, the user must be allowed some choice over the tools applied. This choice must also allow for the ability to add new tools and to redefine how existing tools may be applied. SPaCE I addresses these problems by creating an environment where the user is given such choices, provided that they explain why. This provides a basis for continual revision of the system as appropriate. Accordingly, the question of which specific tools will be used becomes strictly an "expert" domain question rather than a program structure question.

The inference mechanism required for SPaCE I to function needs to be able to handle the following types of activities (grouped from highest level to lowest level):

1) Hypotheses & Test – required for coping with missing and/or incomplete information
2) Frame Based Deduction – required for making indirect inferences based the data supplied an open slots which have some data that can be related to them (i.e. by analogy)
3) Rule Based Deduction – required for making direct inferences from the rule base
4) Direct Calculation – required for executing sub-routine functions
5) LISP Function Execution – required for knowledge base manipulation

These activities are pursued as dictated by the PROGRAM PROTOCOL and the META-LEVEL SOLUTION STRATEGIES knowledge base elements (see Section XI for descriptions). Should an activity prove to be at too low a level to achieve results on a given sub-problem, the next higher activity is invoked until some result can be produced. The activity requirements specified can be met by a number of the domain independent expert system construction toolkits.

One possible toolkit is the Knowledge Management System (KMS) that was developed at the University of Maryland. [11] However, this system was designed for diagnostic problem solving, a guise which may prove intractable for implementing

[11] Reggia, J.A.; Perricone, B.T.; KMS (Knowledge Management System) Manual, University of Maryland Department of Computer Science TR# 1136.

The SPaCE I “expert” system schematic is shown in Figure 16. Top level descriptions of the schematic components (Input, Knowledge Base, Inference Mechanism, and Output) follow.

The regular user input to SPaCE I will include a variety of forms. These forms include both qualitative descriptions (using a restricted sub-set of natural English relevant to spacecraft systems engineering) and quantitative descriptions. A number of different quantitative descriptions are possible. They include:

1) Fixed Values – invariant quantity
2) Target Values – preferred quantity
3) Range Values – upper and lower bounded quantity
4) Minimum Values – minimum quantity
5) Maximum Values – maximum quantity
6) Function Defined Values – quantities given by a function
7) Curve/Tabular Values – quantities given in tables/curves

The user must designate the appropriate description for their input. The input required includes: the instrument/payload description and the operating parameters; the retro-fit and refurbishment goals; and, the time horizon (initial operational capability).

The Knowledge Base will consist of a number of elements that will be considered to be discrete for the purposes of system design. The elements must provide the knowledge necessary to accomplish the following:

1) Interpret the user input
2) Delineate the definite spacecraft requirements
3) Define the appropriate spacecraft specifications for the set of spacecraft designs that can meet the definite requirements of the spacecraft system.
4) Generate comparison reports and any other possible outputs

Descriptions of the Knowledge Base elements presumed to be required for SPaCE I may be found in Section XI. The specific analytical tools that the knowledge base elements need to allow SPaCE I to function successfully are only alluded to in this paper. Considerable detail on the tools available can be found in the works cited in the bibliography (Section IX) both from a computer science and from systems engineering perspectives. The determination of which qualitative and quantitative tools should be included is a continuing problem with "expert" system design. For a spacecraft preliminary and conceptual engineering "expert" system to be viable, the user must be allowed some choice over the tools applied. This choice must also allow for the ability to add new tools and to redefine how existing tools may be applied. SPaCE I addresses these problems by creating an environment where the user is given such choices, provided that they explain why. This provides a basis for continual revision of the system as appropriate. Accordingly, the question of which specific tools will be used becomes strictly an "expert" domain question rather than a program structure question.

The inference mechanism required for SPaCE I to function needs to be able to handle the following types of activities (grouped from highest level to lowest level):

1) Hypotheses & Test – required for coping with missing and/or incomplete information
2) Frame Based Deduction – required for making indirect inferences based the data supplied an open slots which have some data that can be related to them (i.e. by analogy)
3) Rule Based Deduction – required for making direct inferences from the rule base
4) Direct Calculation – required for executing sub-routine functions
5) LISP Function Execution – required for knowledge base manipulation

These activities are pursued as dictated by the PROGRAM PROTOCOL and the META-LEVEL SOLUTION STRATEGIES knowledge base elements (see Section XI for descriptions). Should an activity prove to be at too low a level to achieve results on a given sub-problem, the next higher activity is invoked until some result can be produced. The activity requirements specified can be met by a number of the domain independent expert system construction toolkits.

One possible toolkit is the Knowledge Management System (KMS) that was developed at the University of Maryland. [11] However, this system was designed for diagnostic problem solving, a guise which may prove intractable for implementing

[11] Reggia, J.A.; Perricone, B.T.; KMS (Knowledge Management System) Manual, University of Maryland Department of Computer Science TR# 1136.

SPaCE I. Other toolkits, such as Attempt to Generalize (AGE), may be available from other NASA centers.

The choice of adopting a ready-made inference mechanism or trying to custom craft one is an implementation decision that is dependent on the machine made available for the work and the available resources for supporting code acquisition and maintenance. At this point it is unclear which is the best route to obtain satisfactory results from a prototype SPaCE I within time constraints of the project. Regardless, both routes will rely heavily on existing software, either directly or indirectly.

The inference mechanism chosen does not need to justify the absolute correctness of the paths of action it undertakes. However, it must generate a reconstructable path for explanation capabilities, and allow for the deactivation (either directly or indirectly) of paths which lead to identified errors. A spacecraft designer is concerned with producing useful results that are logically defensible. Only results that have been arrived at by a clearly understandable set of procedures will be used.

The typical user of the SPaCE I system is envisioned to be an engineer involved in the conceptual and/or preliminary design of a spacecraft system.

The SPaCE I system is intended to perform three main functions. They are as follows:

1. The generation of the appropriate specifications for a spacecraft, SPaCE I is not intended as a replacement for systems engineers, rather it is intended to extend their capability by acting as a decision support system. Such a decision support system could reduce errors, streamline the process (i.e. reduce the amount of resources that have to be committed to obtain the necessary analysis) and allow for greater flexibility in considering alternate paths that may result in a more optimal spacecraft design/configuration.

2. The generation of comparisons between complete versions of a spacecraft system data sets. This is useful for showing the utility of choosing one type of sub-system over another. It provides a resident “what if?” answering capability. This could be very important to justifying the use of a more advanced technology sub-system when no requirement forces its inclusion.

3. The determination of which requirements are driving the design of the spacecraft

and the sensitivity of the spacecraft system design to variations in how those requirements are met. This function is useful for resolving which of the spacecraft specifications generated are really “hard” (i.e. they can be traded off for better performance in other sub-systems). The information provided can be evaluated to determine how the effective of the spacecraft system may be increased.

The three functions can provide output to video terminals, printers, and plotters. A number of outputs are possible. Data can be displayed in tabular, chart or 3-D drawing form. Explanations are in text form only; with or without citations. These functions were identified by RTOP team members as functions that they would use if a system such as SPaCE I were available.

A typical Input/Output (I/O) Schematic for SPaCE I is shown in Figure 17. The schematic illustrates the varying levels of interaction between the user and the SPaCE I system. From sign-on until the initial data set is finalized SPaCE I is highly interactive with the user. After the initial data set is finalized, until a complete data set is generated, SPaCE I interacts with successively more rigorous qualitative and quantitative tools. These tools may reside with the Knowledge Base or be analytical packages which can be accessed by protocols contained in the Knowledge Base. Once a complete data set has been generated, SPaCE I returns to being highly interactive with the user in order to produce the appropriate output. This I/O schema allows users to minimize their involvement in the mechanical aspects of the spacecraft systems engineering process. This should facilitate the spacecraft conceptual and preliminary engineering activities. The SPaCE I system allows the spacecraft systems engineer to concentrate on the most ill-defined aspects of spacecraft design problems. Once a technique or fact becomes associated with an appropriate application it will not be forgotten or overlooked. Over time, a suitably structured system could develop into a very powerful tool for spacecraft conceptual and preliminary engineering. Using a system like SPaCE I could yield a variety of benefits, including:

1) More design iterations per unit of resources committed
2) More detailed design iterations
3) Lessen the chance of omissions or errors

4) Lessen the possibility of viable alternatives being ruled out by omission or personal bias
5) Allow more direct comparison of alternatives
6) Reduce the lead time required to produce an operating system

The primary intention of the first generation prototype of SPaCE I is to demonstrate that a system can be constructed to meet the defined needs of the RTOP given a sufficient commitment of resources. Artificial Intelligence technologies will not be the limiting factor for systems along the lines of SPaCE I. Rather, the capabilities will be limited by the willingness and the ability of experts to articulate their knowledge of the domain of spacecraft systems engineering.

The next section outlines a process description for SPaCE I.

V. SPaCE PROCESS DESCRIPTION

The SPaCE I system architecture/process flow diagram is shown in Figures 18-24. After the various sign-on related procedures there are six major modes that the SPaCE I system is intended to function in (see Figure 18). They are as follows:

1. HELP! – This mode is shown in Figure 19. It is intended to help the user. It has two sub-modes. The first is the SPaCE I Tutorial which is intended to provide lessons on how to use various aspects of the system. The user is given a choice of the lessons available. The tutorial exits to the major mode menu. The second sub-mode is the query answering capability. This is currently envisioned to be a “Key Word Out of Context” (KWOC) parser which is capable of accepting near natural language queries from the user (within the domain of spacecraft systems engineering) at any time, regardless of the current state of the job. The parser constructs an entry table into the SYSTEM VOCABULARY and SYSTEM DOCUMENTATION elements that corresponds to the key words found in the query. The entry table is displayed to the user, who is allowed to request the display of the documentation accessible by the entry table. Once the user is through with the query, control is returned to the point in the program where the job was interrupted.

2. UPDATE EXISTING KNOWLEDGE BASE – This major mode is shown in Figure 20. It is intended to allow authorized users to update the existing knowledge base. This major mode has two sub-modes which represent the two manners in which knowledge base elements may be updated. The first sub-mode allows the user to edit a copy of a knowledge base element. Once the editing has been completed the element is put through the appropriate element test. The element tests are used to verify that the element format is correct and where possible to verify that the element will perform in the expected manner. If the element fails the test, the errors found are displayed and the user is returned to the editor to revise the element. If the element passes the test, the user is requested to determine the location that the old element should be retired to. The second sub-mode allows the swapping of two versions of the same knowledge base element. The user is queried for the location of the new element and then the

element is tested by the appropriate element test as described in the preceding sub-mode. The sub-mode paths are identical from that point on.

3. CONTINUE EXISTING JOB – Is the third major mode. It is intended to allow continued analysis of an existing Interim or Complete spacecraft system data sets. This mode is shown in Figure 21. This mode has no sub-modes, but it does have an iteration option. This mode is responsible for the spacecraft system model component generation and the synthesis of those components both on qualitative and quantitative levels. This mode allows access to all other analytical packages known by the system. This mode also evaluates the model status and provides the requested output. If the user does not choose to iterate the analysis the program returns to the major mode menu.

4. EDIT/APPEND EXISTING JOB – This major mode is illustrated in Figure 22. It is intended to allow a user to update or append an existing spacecraft system data set regardless of the status of the data set. The system determines if the user desires a duplicate of the existing version of the data set for his/her own purposes. If a duplicate is not requested, the system creates its own back-up data set just in case there is an editing problem. Editing of a version of the data set is then allowed. After editing, the system recourses along the EXPLANATION TREE element until the changes made in the data set can be accommodated with no secondary effects. This level becomes the new design state level assigned to the data set. The user is returned to the major mode menu.

5. END THIS SESSION – Is the major mode shown is Figure 23. This mode allows the user to end the session as well as save or delete data sets. Before sign-off, the status of the job, files and other relevant information is displayed. The user is then asked to confirm their sign-off command. If the command is not confirmed the user is returned to the major mode menu. If the command is confirmed then SPaCE I proceeds to sign-off and shut down.

6. GENERATE NEW DATA INITIAL DATA SET – This major mode allows the generation of new initial spacecraft data sets. The mode is illustrated in Figure 24. The mode has no sub-modes and does not iterate. It solicits the spacecraft purpose and other additional available data from the user. It makes direct and indirect inferences

VII. SUMMARY

This paper has described the current state of the SPaCE I system under development at GSFC. It is this author's contention that a viable expert system can be constructed in the domain of spacecraft systems using the architecture defined in this document as the initial basis for the system construction. Furthermore, such a system could serve as a decision support system to aid in the preliminary and conceptual engineering of spacecraft systems to whatever extent it is implemented.

SPaCE I will meet the needs of the RTOP, but it could have further value. It could serve as a precursor to a more general computer aided systems engineering (CASE) capability. A CASE system could conceivably work with users from the conceptual phase until the end of the lifecycle of the spacecraft system. Several diagrams which illustrate this are included as Figures 26, 27 and 28.[12]

Advanced automation systems which utilize AI techniques can play a vital role in extending human capabilities to handle very complex problems and cope with harsh environments. They can help provide the critical support needed to allow space development to begin in earnest, or put in other terms enable the development to begin in earnest, or put in other terms enable the development of a viable Space Station System. Building a Space Station System will require finding new approaches that allow the seven year lead times for hardware development and delivery to be reduced. As well, NASA cannot afford to fund a new marching army on the ground to support a Space Station system. Lastly, the astronauts cannot spend 80-90% of their time keeping themselves alive. When you also consider that the Space Station System is going to have to evolve over time, the fact that some new tools are necessary becomes very clear.

Advanced automation systems can provide some new tools both now and in the future to help address the problems associated with the systems engineering of complex space systems.

[12] NASA LaRC/OAST Space Station Technology Workshop Summary Presentation Document, March 1983.

VIII. REFERENCES

Advanced Earth-Orbital Spacecraft Systems Technology RTOP, Goddard Space Flight Center, 506-62-26 (1983).

NASA LaRC/OAST Space Station Technology Workshop Summary Presentation Document, March 1983.

Reggia, J.A.; Perricone, B.T.; “KMS (Knowledge Management System) Manual” University of Maryland Department of Computer Science TR# 1136.

Waterman, D.A.; Hayes-Roth; “An Investigation of Tools for Building Expert Systems”, Rand Report R-2818-NSF, 1982.

Gavarter, W.B.; “An Overview of Expert Systems”, NBSIR 82-2505, 1982.

“Multimission Modular Spacecraft External Specifications Document”, GSFC 1978.

Barnhard, G.P.; “Plans for a Spacecraft ‘Math Model’”, unpublished presentation first given 12/9/82.

Studer, P; “Earth-Orbital Spacecraft Systems Technology Preliminary Plan”, GSFC, Nov. 1982.

Minsky, M; “A Framework for Representing Knowledge”, MIT AI Lab Memo No. 306.

Speray, D.E; “Raster Graphics Language Extensions and Tools for Increased Productivity”, LaRC RTOP proposal document.

Figure 1.
"Expert" System Schematic

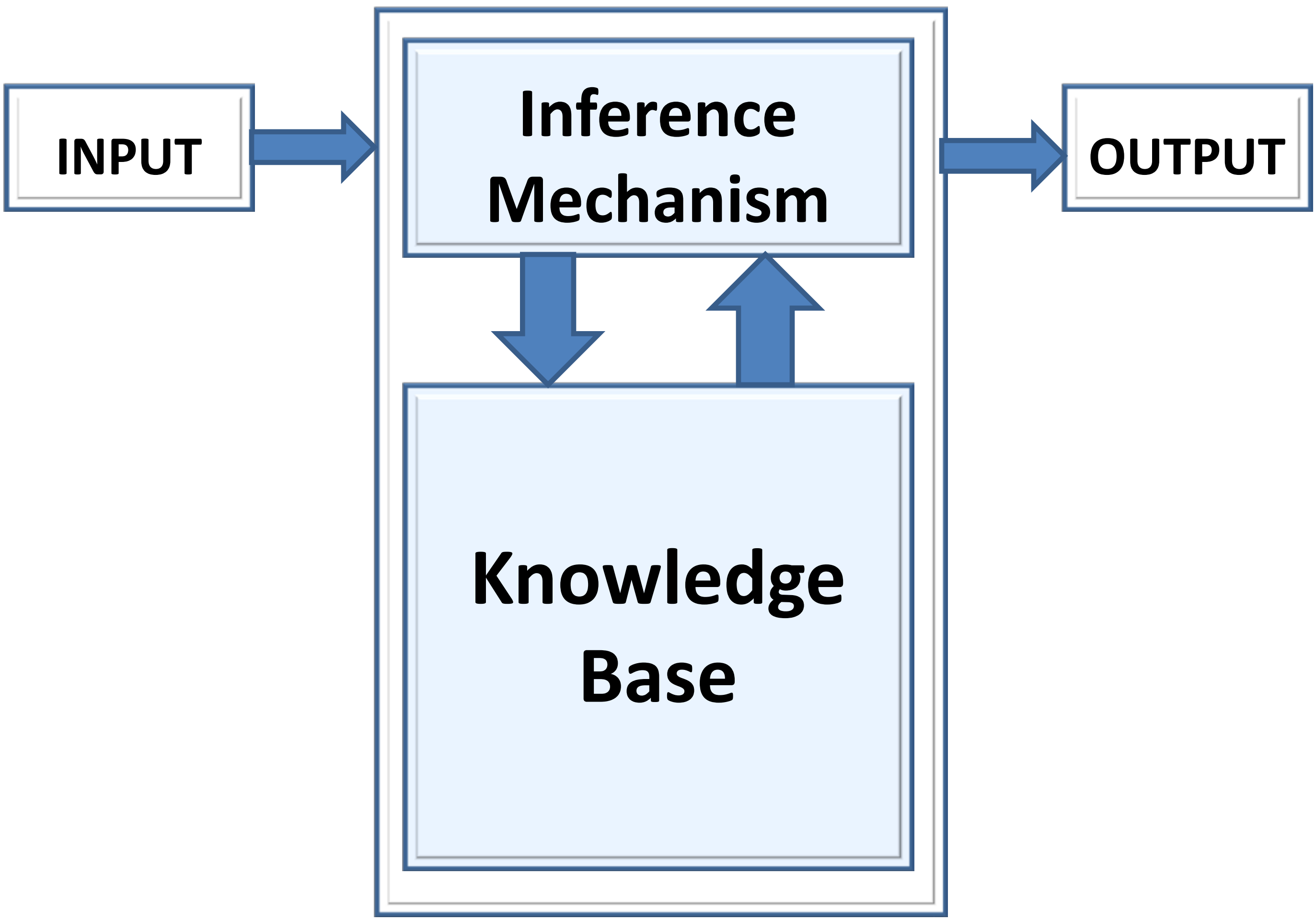


Figure 2.
The NASA Systems Engineering Process Phase Breakdown

Designation	Phases	Some Characteristics
Conceptual Engineering	Prephase A (Conceptual Study) Phase A (Mission Analysis)	= Rough Calculations 1x Resource Commitment 3-D Model Length, Width, Mass Functionality Resources Required
Preliminary Engineering	Prephase B (Advanced Definition) Phase B (Definition)	= Finite Element Work 10x Resource Commitment Buy Off Point Assessed Multi-Discipline Work
Detail Design	Phase C/D (Execution)	= Every Bolt and Rivet
Construction	Phase C/D (Execution)	
Operations	Phase E (Operations)	

Figure 3.
The Spacecraft Systems Engineering Process Circa 1983

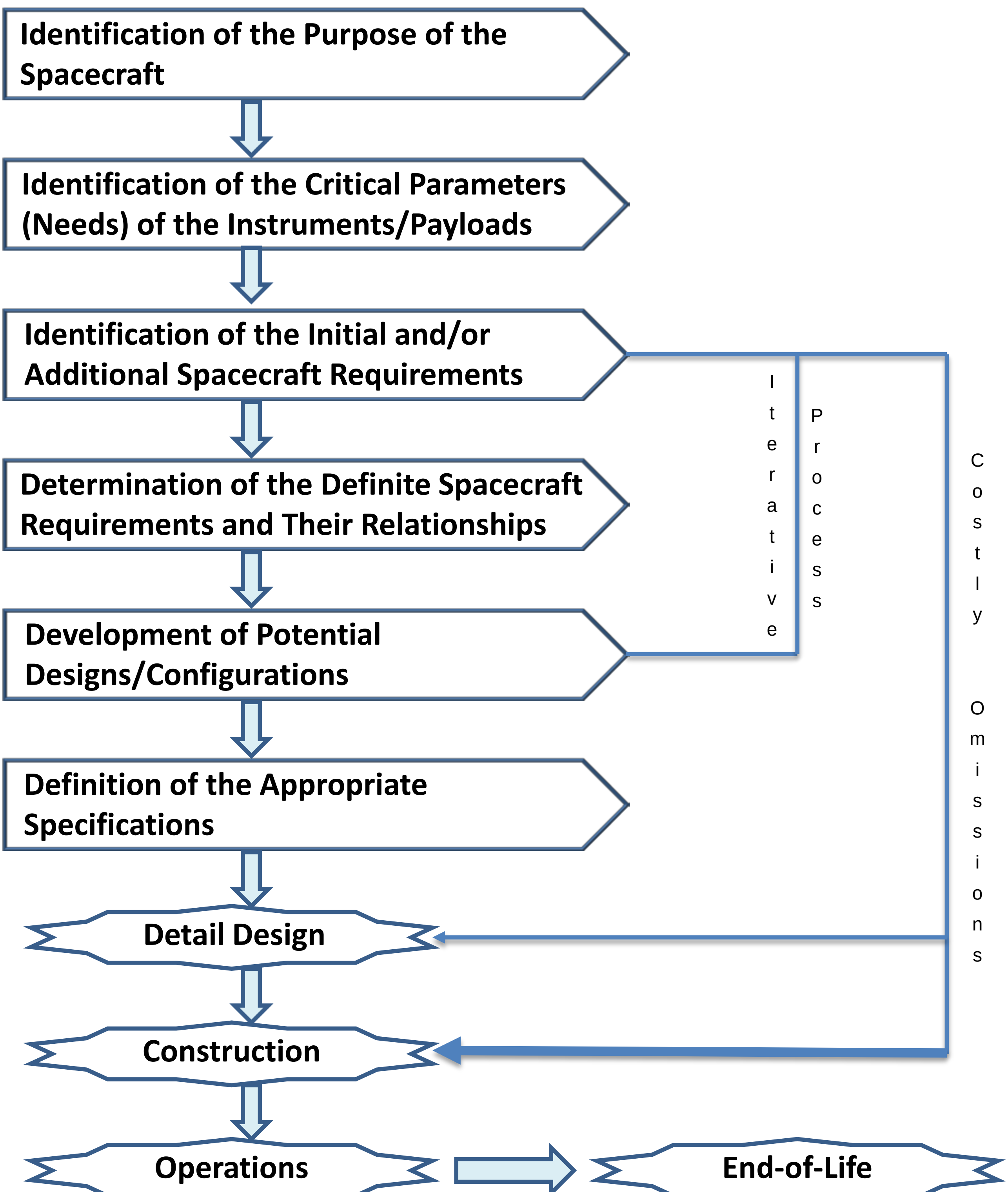


Figure 4.
Mission Requirements Outline

a) Anticipated launch date and required delivery date for spacecraft.

b) Required orbit, in-orbit position relative to earth, sky, etc.

c) Ground coverage limitations for command and data acquisition, and operational constraints on coverage.

d) Launch constraints (time, sun, etc.)

e) Basic launch vehicle (Scout, Delta, Shuttle) plus additional booster recommendations or restrictions, if any

f) Lifetime (orbital and useful data)

g) Orbital radiation environment or constraints

h) Weight and volume of payload module, with outline drawing, including center of gravity, moments of inertia about payload module, e.g., residual angular momentum, mass properties and unrestricted field of view requirements, i.e., where its field of view must not be blocked.

i) Power requirements, average, peak and orbital power profiles.

j) Clock rates required by instrument, with desired impedances, if known.

k) Data rates.

l) Analog telemetry sampling rate, subcom requirements, and TM points.

m) On-board storage.

n) Transponder required.

o) Antenna required.

p) Attitude control requirements in roll, pitch and yaw, and rates about roll, pitch and yaw for normal operations, during and after maneuvers, including thrusting if required.

q) Slew rates (degrees per minute)

r) Requirements for spin, despin, acquisition and separation

s) Requirement, if any, for orbit adjustment after initial insertion.

t) Attitude determination requirements

u) Unusual requirements which will impact the design of the spacecraft, e.g., moving parts not described under “h” above.

v) Requirements for redundancy.

Figure 5.
Advanced Earth-Orbital Spacecraft Systems Technology RTOP Participants

Name	Organization	Speciality
John DiBattista	NASA	HQ, OAST Systems
Phil Studer	716	RTOP Manager, Electromechanical
Jim Andary	712	Attitude Control
Gary Barnhard	502	Computer Model
Lou Caudill	723	Lasers
John Chitwood	727	Communications
Tom Cygnarowica	713	Cryogenics
Jack Evans	402	Mechanical
Joseph Fedor	712	Attitude Control
Ray Hartenstein	730	CMD & Data
John Hayes	728	Data Storage
Dick Hockensmith	727	Communications
H.P. Lee	713	Modeling & Analysis
Joseph Lundholm	402	Advanced Mission Analysis
Ron Muller	402	Advanced Mission Analysis
Stan Ollendorf	732	Thermal
John Osantowski	717	Optics
Joe Schepis	716	Electromechanical
Lew Slifer	711	Power
Dave Suddeth	402	Propulsion
Joe Young	731	Structures
Marvin Maxwell	920	Applications
Bob Nelson	502	Data Systems

Figure 6.
Landsat D Spacecraft System

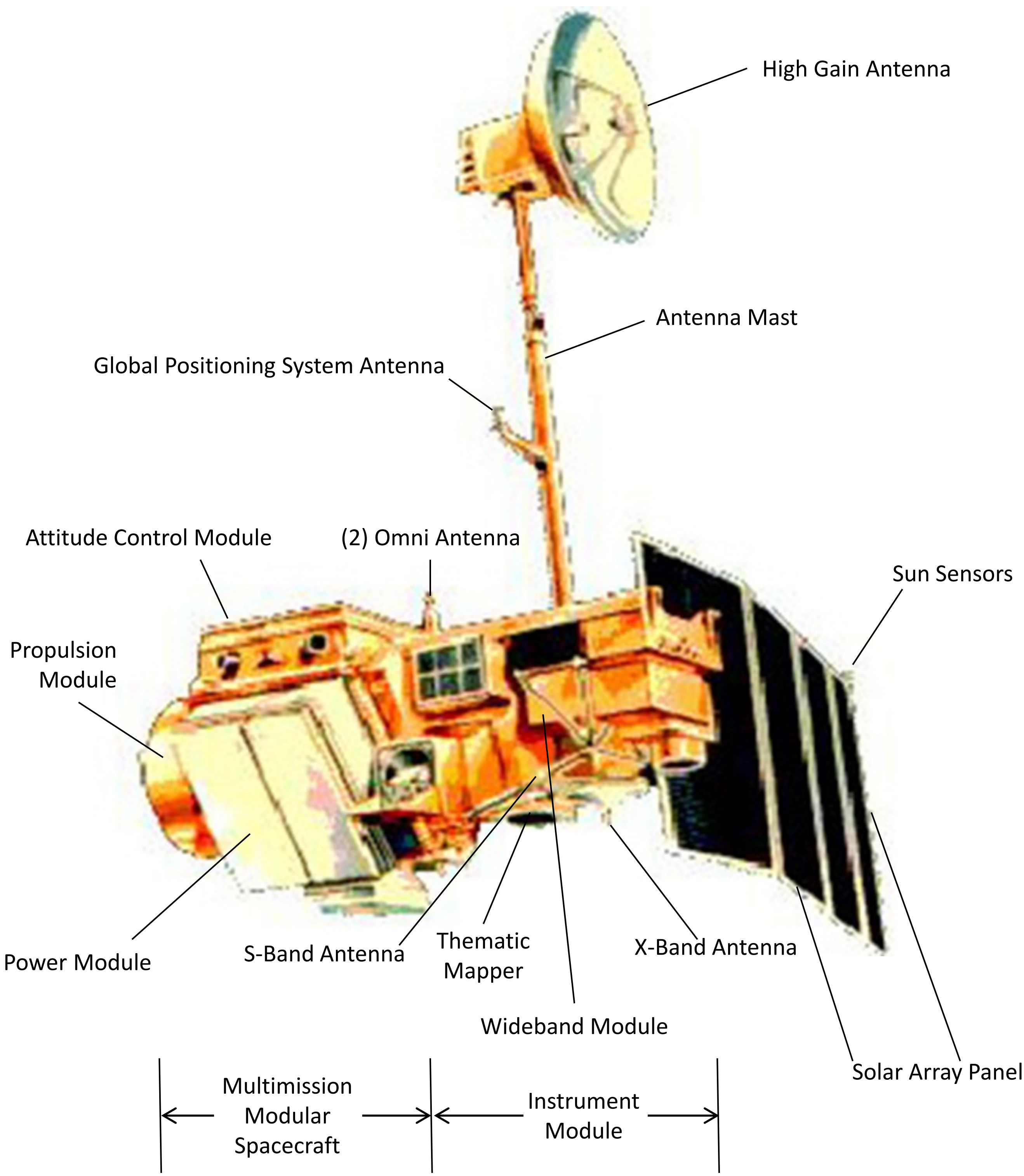


Figure 8.
Spacecraft Sub-Systems
& Key Functional Technologies

NAME	MODEL STRUCTURE
• Attitude Control • Power • Thermal • Propulsion • Communication • Command & Data Handling • Structure Navigation	Discrete
Lasers Cryogenic Optics Electromechanical	Distributed
Payload & Instruments Retro-Fit & Refurbish Environment	External Constraints

• Classic sub-system

Figure 10.
Sub-System "Flow" Taxonomy

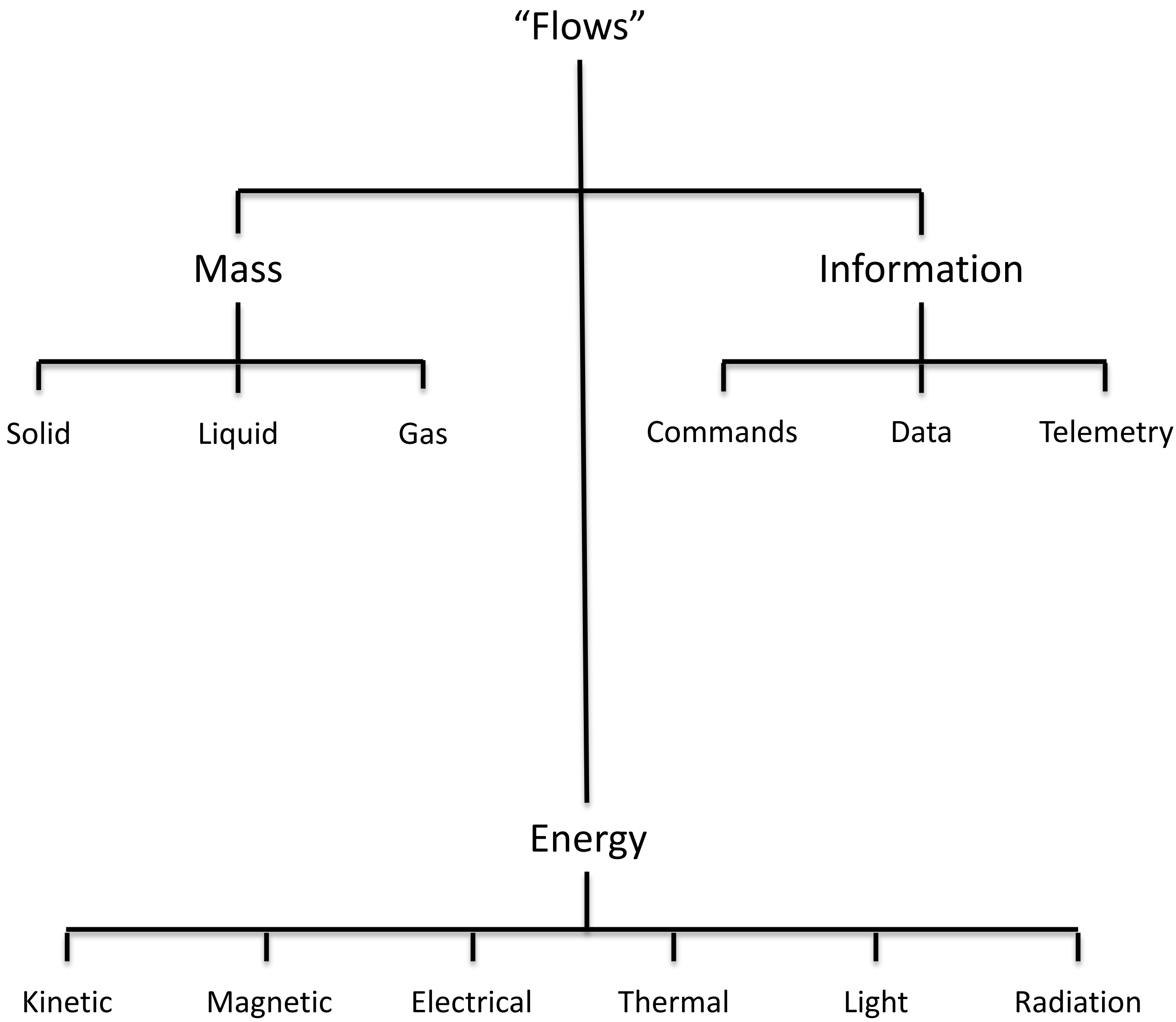


Figure 11.
Electrical Flow Specification

Electrical Flow from Sub-System: ______________________________________
to Sub-System: ______________________________________

☐ Alternating Current – Number of Cycles per Minute: ___________________

☐ Direct Current ☐ Regulated ☐ Unregulated
Voltage: ______________________ Amps: _____________________________

☐ Continuous ☐ Intermittent
Peak: ___________ Average: ___________

☐ Grounds ____

Figure 12.
Flow Matrices

<u>Flow Types</u>

Solid
Gas
Magnetic
Thermal
Radiation
Commands

Liquid
Kinetic
Electrical
Light
Data
Telemetry

A Separate Flow Matrix is Created for Each Flow Type

	Flow from Sub-System													
To Sub-System	Attitude Control	Power	Cryogenic	Propulsion	Communication	Command & Data Handling	Structure	Thermal	Lasers	Optics	Electromechanical	Payload & Instruments	Retrofit & Refurbish	Environment
Attitude Control	●													
Power		●												
Cryogenic			●											
Propulsion				●										
Communication					●									
Command & Data Handling						●								
Structure							●							
Thermal								●						
Lasers									●					
Optics										●				
Electromechanical											●			
Payload & Instruments												●		
Retrofit & Refurbish													●	
Environment														●

Figure 13.
Flow Models

1. Flow Specification Sheets

2. Functions / Rules
   A. Qualitative Relationship Rules
   B. Quantitative Relationship Functions

3. Rational
   A. Explanation & Justification of Functions/Rules

4. Documentation
   A. Information Sources (General & Specific)

Figure 14.
Model Outline

I. . . . . . Spacecraft

II. . . . . Sub-Systems

III. . . . . Interaction Matrix

IV. . . . . Flow Matrices

V. . . . . Flow Models

- Functions/Rules
- Rational
- Documentation

Figure 15.
SPaCE I Spacecraft Design Problem Boundary

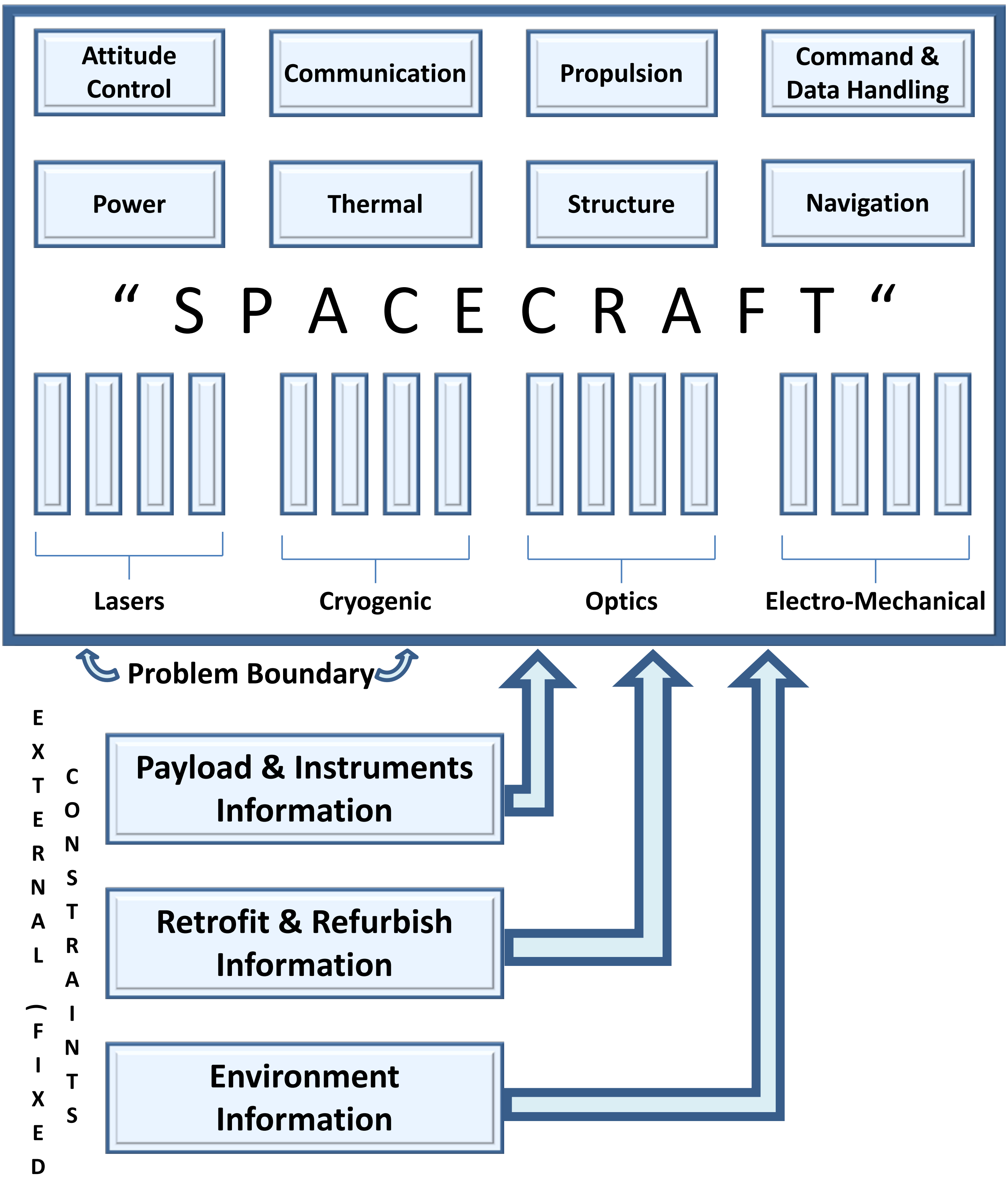


Figure 16.
SPaCE I "Expert" System Schematic

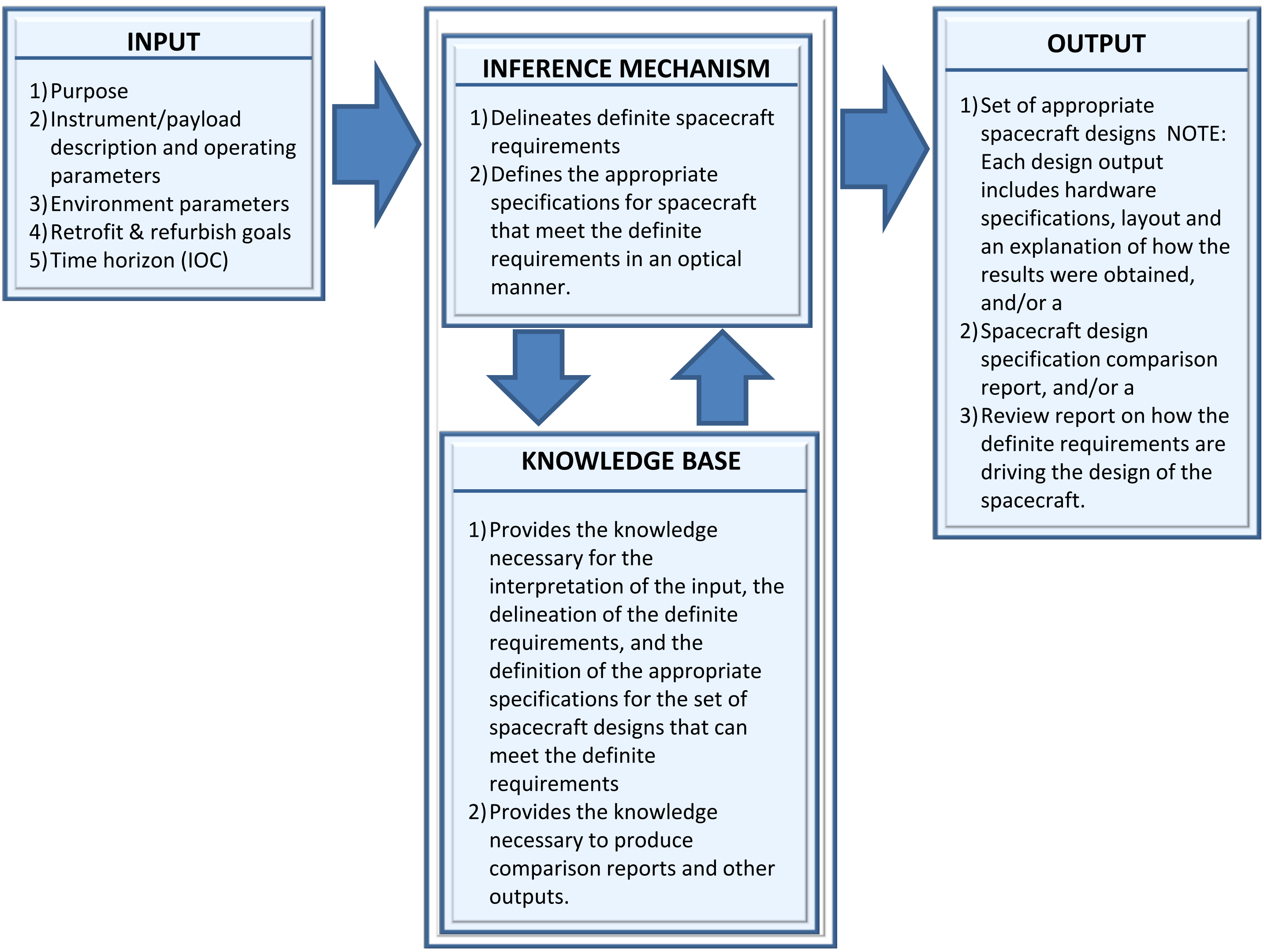


Figure 17.
SPaCE I "Expert" System Schematic

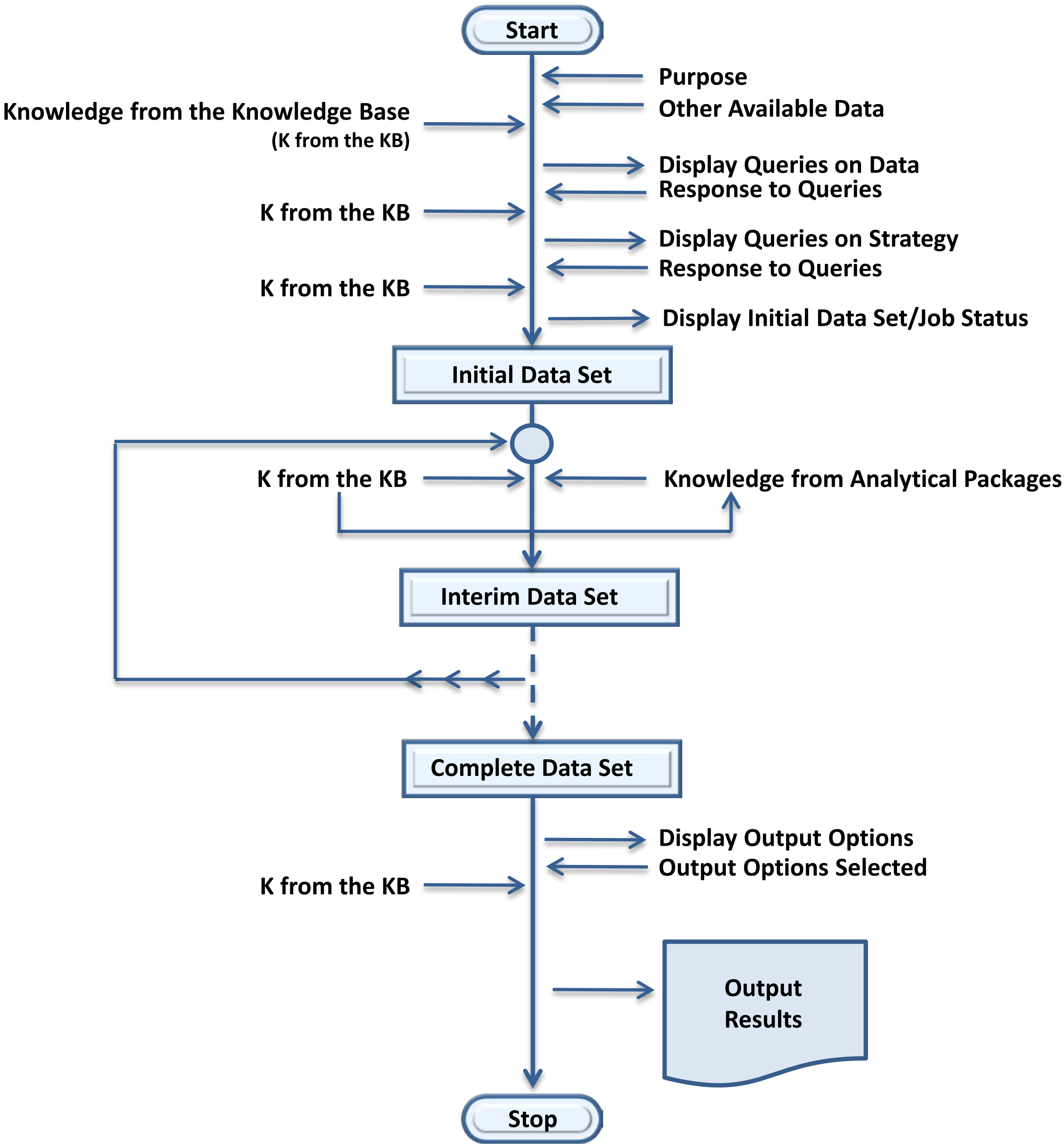


Figure 25.
SPaCE I Milestone Chart

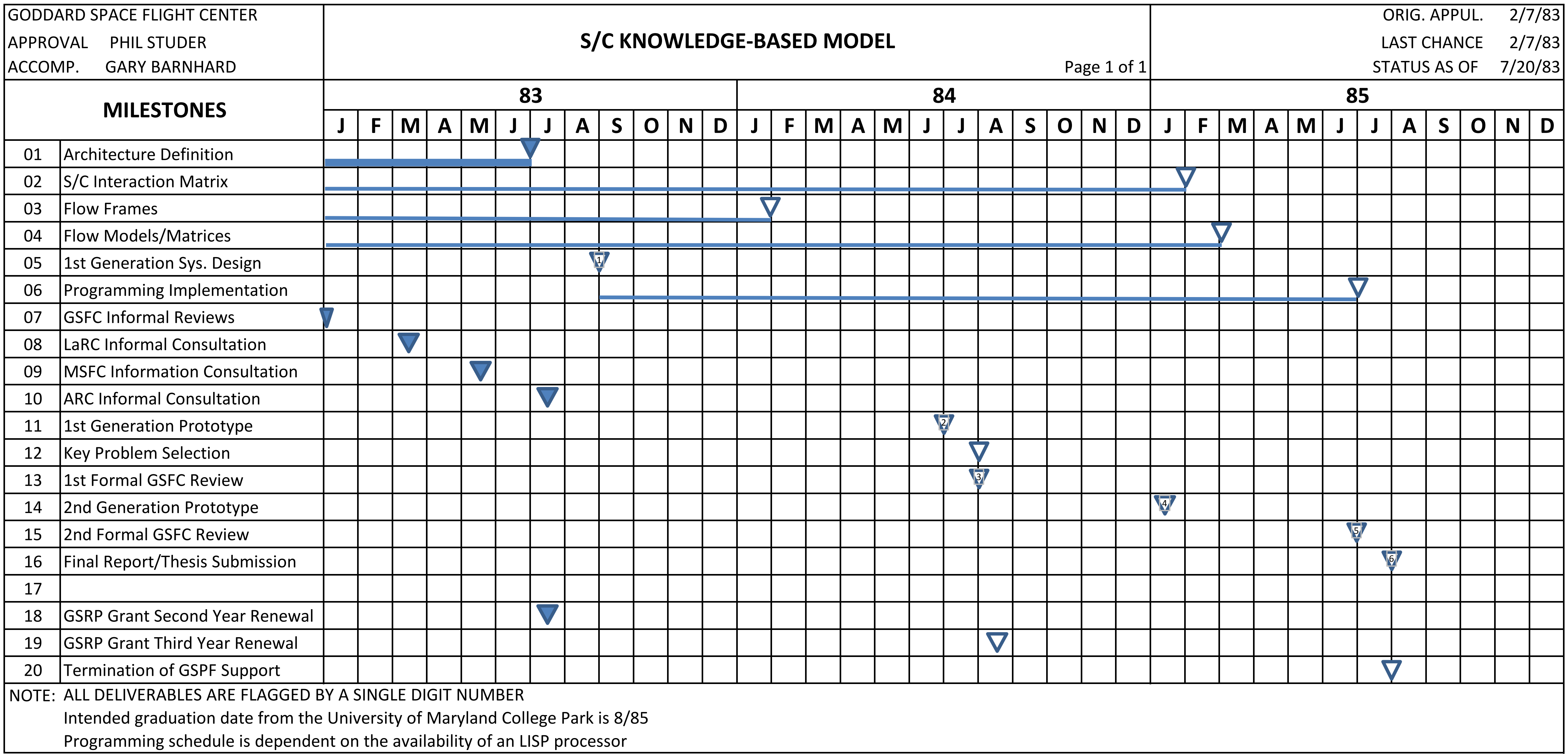


Figure 26.
Evolutionary Space Station Development, Implementation & Operations

<u>DATABASE</u>
A database consists of all available information relating to the space station system including graphics. The information is shared by NASA & industry and is updated on a continual basis.

<u>CONCEPTUAL/PRELIMINARY COMPUTER-AIDED ENGINEERING TOOLS</u>
These tools consist of software-based modeling analysis elements and relational/integrative information management systems.

<u>SUB-SYSTEM TEST BEDS</u>
Software/hardware test elements for modeling of sub-system designs.

<u>SUB-SYSTEM SIMULATORS</u>
Software/hardware test elements for evaluating and optimizing sub-system design.

<u>INDUSTRY CAD/CAM</u>
Computer-aided design/computer-aided manufacture for final component optimization and construction.

<u>SYSTEMS SIMULATORS</u>
Software/hardware test systems for modeling, evaluating and optimizing sub-system interactions.

<u>USER OPERATIONS</u>
The definition and implementation of an appropriate user operations management system.

<u>SUPPORT OPERATIONS</u>
The definition and implementation of an appropriate support operations management system.

Figure 27.
Evolutionary Space Station Development, Implementation & Operations

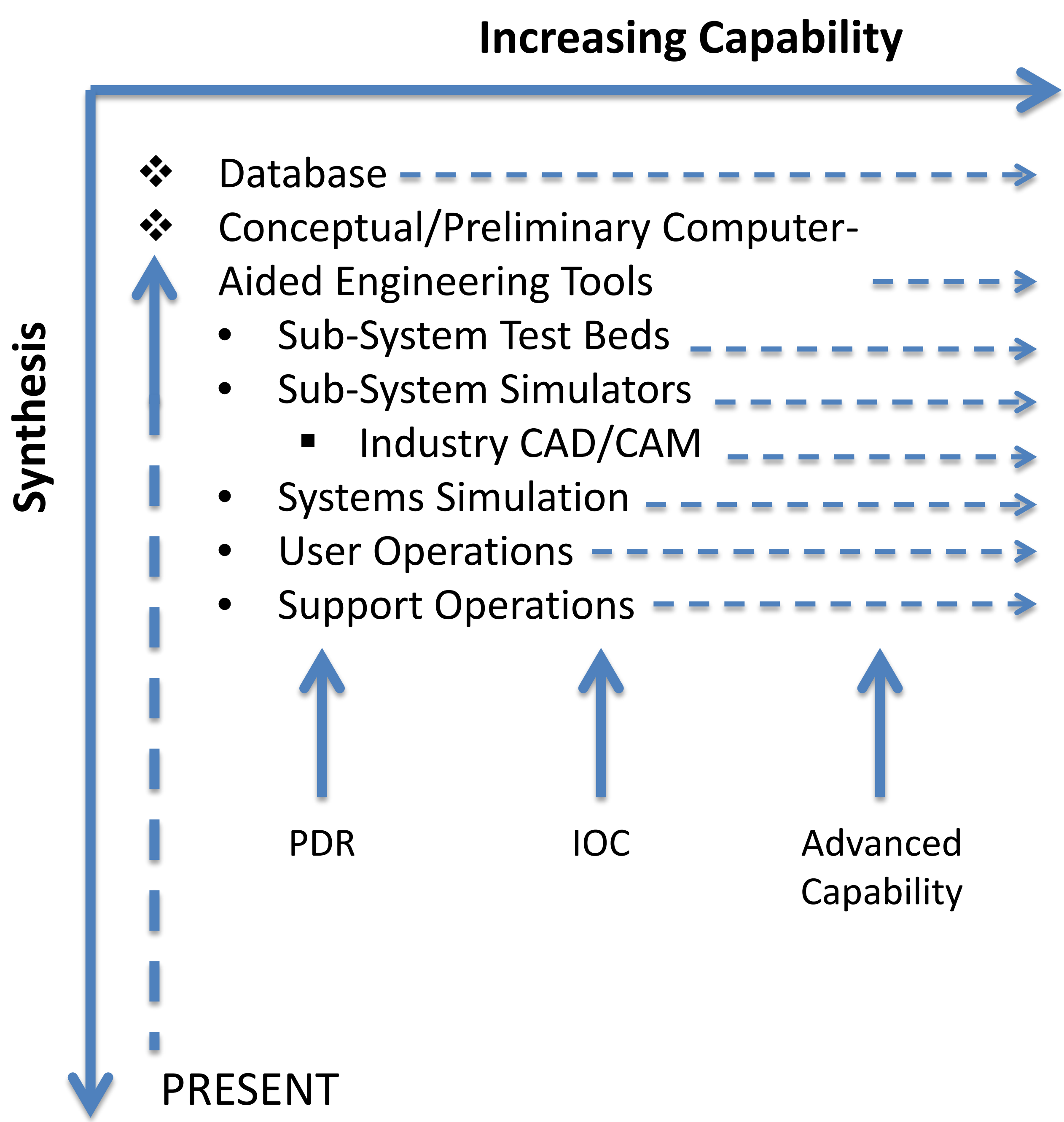
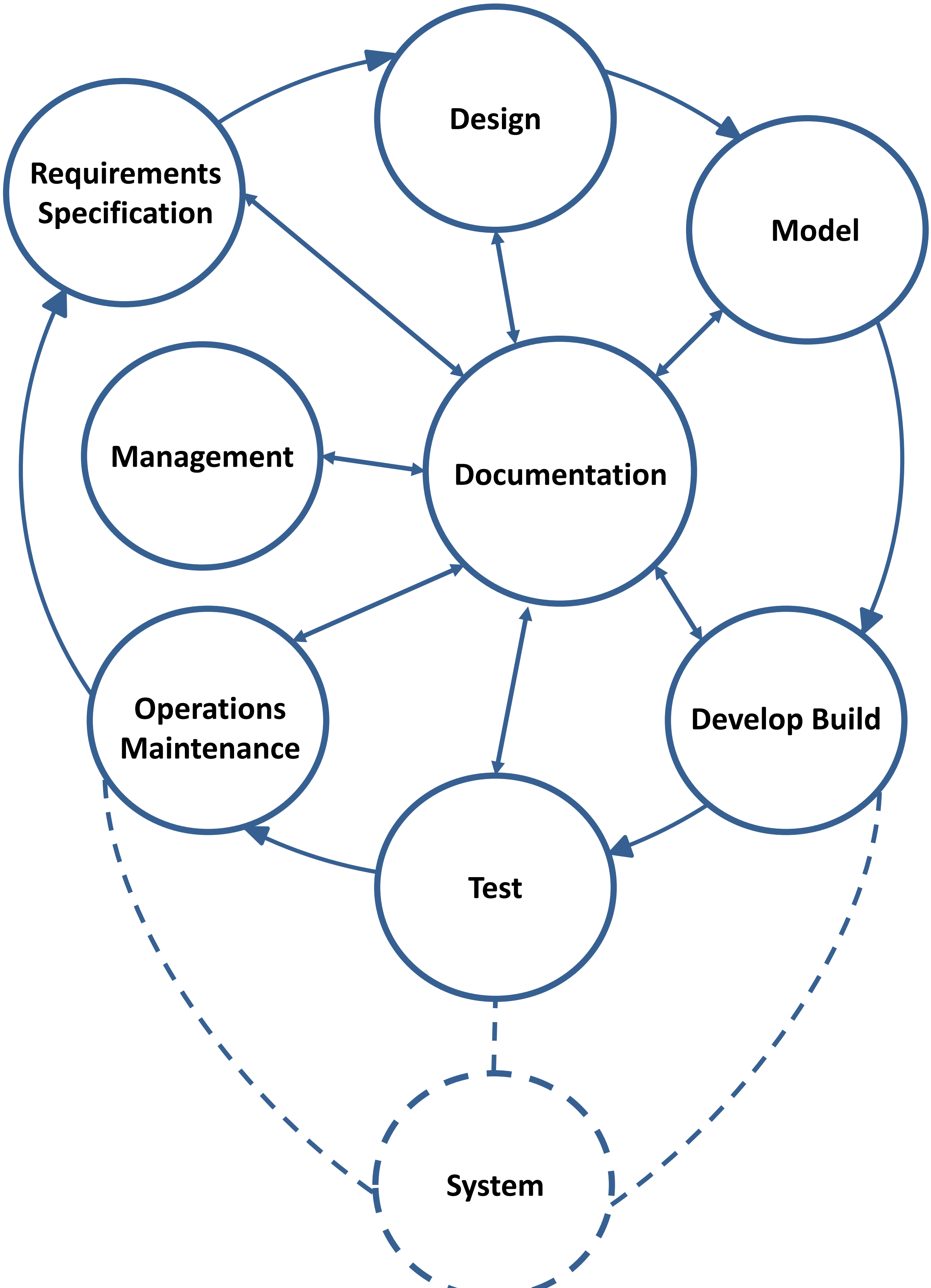


Figure 28.
Computer-Aided Systems Engineering



XII. Knowledge Base Description

The knowledge base for SPaCE I requires a significant number of elements. No attempt was made to combine elements that are always together. The breakdown was made with an emphasis on clarity and may not represent the most prudent structure for implementation. The text that follows describes the elements presumed to be required to implement SPaCE I. The following questions are addressed for each element. What knowledge is resident in the element? When is the element active? What can be said about the structure of the element? The term active refers to one of the three states that a knowledge base element can exist in. These states are: active, quiescent (flags set, job data is resident, on call) and inactive (available if called up from storage).

PROGRAM PROTOCOL – The element holds the knowledge necessary to initiate the sign-on process and to manage the activation and deactivation of other knowledge base elements. It also contains the rules which define the sequence of activities that are requested. This element is active from the moment that SPaCE I is booted by the host computer operating system until after sign-off occurs. This element consists of: several sets of strategically ordered (If <antecedent> then <consequence> production rules; rule application and activity sequence constraint frames; a list of all elements in the knowledge base; and, the control sequences for the inference mechanism and the system output.

USER PROFILES – This element contains the knowledge of: who the authorized users of the SPaCE I system are: the degree of access each individual is to be granted to the system; the experience level of the individual with the system; and, any special interactive features the individual usually requests or requires. In addition, if the user has write authority (i.e. they are allowed to update the knowledge base) their qualifications are maintained in their user profile and all additions made by them are referenced to their user profile. This element is activated by the sign-on process and enters a quiescent state after the user verifies that the profile they have entered/reviewed is accurate. If special interactive features are present, flags are sent to the JOB PROFILE element, along with read/write restrictions to be enforced. The structure of this element is that of a set of user profile frames, preceded and followed by the production rules necessary to carry out the required activities.

JOB PROFILE – This element contains the knowledge passed to it from the USER PROFILE element and the user's explanation of what activities he/she intends to carry out with the system during the current session. This information is solicited in the form of responses to various menus, or if the user prefers, free text that the system will record and attempt to parse after the job is initialized. This element is active from the time that it is initially called until sign-off. This element structure is a single frame with predefined and user-definable slots.

SYSTEM VOCABULARY – This element contains a set of all which are defined as Key Words. Key Words are words which the system recognizes. Each word is given a definition and is indexed to all unique context occurrences in the knowledge base. Each word may also function as a command by having executable codes after the free text definition. This element is in a quiet state form sign-on to sign-off unless called up. After it's called up, a search is performed to provide the requested knowledge. Once the word requested is left active and the element is returned to its quiescent state. The executable copy is erased after it is either used or put aside. The structure of this element is a large set of frames which are in the following form: index code, key word, free text definition, unique context occurrences (resident in knowledge base), warning/query (if executable rules are present), and any executable rules.

PARSER/LANGUAGE INTERFACE – This element stores the knowledge necessary for the Key Word Out of Context (KWOC) parser to parse a user-supplied sentence to the extent required to generate the appropriate entry table into the SYSTEM VOCABULARY and/or SYSTEM DOCUMENTATION elements. Should a more advanced parser capability be implemented, this element would be superceded by what is required to support the more advanced parser/natural language interface. If the user entry that is being parsed is data intended for an established data set, the input is screened for correct syntax and units, then passed on to the appropriate element. If the syntax and units are correct, the syntax and units requirements are resident in the frames of the elements that accommodate the data being supplied. If the syntax and/or the units are incorrect, then correction is attempted using the syntax correction rules resident in this element and/or the UNITS CONVERSION element. The results are displayed to the user for approval, then passed on as either the presumed data entry or as a flagged free text entry. This element also contains the knowledge necessary to decode the EXPLANATION TREE elements path description by accessing the SYSTEM VOCABULARY, SYSTEM DOCUMENTATION and other elements as required to output the text with the index codes called for by the EXPLANATION TREE. This element is active from sign-on to after sign-off. This element's structure consists of a strategically ordered set of production rules.

EXPLANATION TREE – This element contains a record of the path taken by a job through the system in order to provide a multi-level explanation capability. This element is activated at sign-on and is deactivated at sign-off. This element maintains a cumulative data file with multiple entry points which can be used to reproduce a near-natural language text explanation of the path taken between any past point and present point in the job. This element uses a special index code to access the individual pieces of the knowledge elements required to generate the explanation. This element uses the PARSER/LANGUAGE INTERFACE element to provide the near-natural language explanation output.

SYSTEM DOCUMENTATION – This element consists of all the available documentation on the SPaCE I system. This element is inactive unless called by the PARSER/LANGUAGE INTERFACE. When called, an index code entry table is sent into an active state and the rest of the element is moved into a quiescent state. The element is structured as a set of free text paragraphs that have key words and an index code associated with them. The index code entry table is cross-tabulated so that the documentation can be selectively retrieved either by keywords and/or index codes. When a documentation request is received from the PARSER/LANGUAGE INTERFACE, the paragraphs with the corresponding index codes are called up and displayed.

SYSTEM "HELP!" TUTORIAL – This element contains the system tutorial lessons and the protocol for handling random queries at any time during a job session. This element uses the PARSER/LANGUAGE INTERFACE to interpret queries, then displays the interpretation as an entry table into the SYSTEM VOCABULARY and/or the

SYSTEM DOCUMENTATION for the user's perusal. The element solicits the user's requests and returns them to the parser as an EXPLANATION TREE-like data stream. The user is then put back to the original point in the job where the query was entered. This element is inactive unless called up. The element structure consists of a set of production rules which define the protocol for handling random queries and the lesson data sets/programs.

ELEMENT TESTER - This element contains a set of substantive tests which a revised or replacement knowledge base element must pass before it can be introduced for use by the system. Each knowledge base element has specific tests which it must pass resident in this element. This element is inactive unless called. This element consists of a set of data sets/programs that are designed to test the required characteristics of each knowledge base element.

S/C FLOW FRAMES – This element contains the template flow frames for each flow type. This element is inactive unless called. This element contains twelve different template flow frames. These frames may be instantiated as often as is required to produce flow specifications for the development of a flow matrix (each instantiation is assigned 2-D array coordinates that correspond to its intended position in a given flow matrix).

S/C FLOW MATRICES – This element contains the set of all possible flow matrices. The flow matrices are filled in as flow specifications are instantiated. This element is brought to a quiescent state when the job is initialized and is fully activated whenever required by other elements. This element contains one flow matrix for each type of flow (12 have been defined). As "elements" (individual flow specifications) of each flow matrix are instantiated, the pieces of the knowledge base elements used to instantiate them are identified by their index codes. These codes are then appended to the explanation slot on the individual flow specification frames.

S/C FLOW RULES/FUNCTIONS – This knowledge base element contains all of the defined rules and functions which govern the flows in spacecraft systems. This element is inactive until called up, but once called it does not return to the inactive state until sign-off. This element consists of fourteen sets of frame-based rules/functions which are selectively activated as dictated by their activation slots. The rules/functions have, at a minimum, the following slots: index code, keywords, type (sub-system, functional technology or external constraint), activation requirements, rule/function source, rational and documentation.

COMMUNICATIONS PROTOCOL – This element contains the knowledge of how to initiate other analytical packages accessible to the system as sub-routines. This element is inactive unless called. This element consists of a set of rules which are selectively activated by inputs gleaned from the ANALYTICAL PACKAGES PROTOCOL frames.

ANALYTICAL PACKAGES PROTOCOL – This element contains frames that define which COMMUNICATIONS PROTOCOL rules should be used with a given analytical package, how it should be formatted and what results can be expected from the package. This element is inactive unless called up. The structure consists of a set of frames which correspond to each analytical package and provide the information described below.

ANALYSIS HIERARCHY – This element contains knowledge on the relative value of the different qualitative and quantitative tools available to the system and the rules which govern when each should be applied. This element is in a quiescent state as soon as a valid data set is present in the current session and assumes active status as is required by the needs of other elements. The structure of this element is that of a set of frames which correspond to, but do not duplicate the information in the ANALYTICAL PACKAGES PROTOCOL.