A HYPERTEXT TOOL TO SUPPORT DEVELOPMENT OF FLIGHT SIMULATOR SPECIFICATIONS

Edward A. Martin Ph.D
Training Systems Branch
Integrated Engineering and Technical Management Division
Wright-Patterson Air Force Base OH 45433-7800

and

John F. Lethert
Air Mobility Training Systems Division
Training Systems Program Office
Wright-Patterson Air Force Base OH 45433-7800

INTRODUCTION

Purpose of Guide Specifications

Guide specifications (formerly Military Primary or Mil-Prime Specifications) were initiated as part of an Air Force System Command effort to streamline the acquisition process and provide greater latitude for contractor innovation (Gordon, 1985; Industry News 1989). Guide Specifications are expressly designed to force tailoring to the specific application through the use of blanks inserted in the requirements, and are the vehicles of choice recommended by MIL-HDBK-248B to support the tailoring mandated by the acquisition streamlining policies and procedures of DODI-5000.2. Specification tailoring is not necessarily easy, and actually impose a greater burden on those who write and issue solicitations to thoroughly understand the operational or training environment and develop realistic requirements based upon the intended application (Gershanoff 1988). While they do provide suggested specification wording and guidance, Guide Specifications cannot serve as cookbooks for devices as varied and complex as flight simulators. The new Flight Simulator Guide Specification, which is to replace AFGS-87241A, incorporates many unique features and concepts to ease the tailoring process while supporting (and encouraging) a structured systems engineering approach to facilitate the clear definition of requirements.

The Flight Simulator Guide Specification

The new Flight Simulator Guide Specification is developed to support both government and industry tailoring of specification documents. It is envisioned that the government will primarily use this to develop the Systems Requirements Document (SRD), and that contractors will use it for generating the Type A and Type B specifications. For this reason, the Flight Simulator Guide Specification *format* is designed to produce hard-copy documents with section and paragraph arrangements that agree with the requirements of MIL-STD-490B for these types of specifications. This is done to facilitate a consistent evolution from SRD to System Specification to Complex Item Requirements Specification (CIRS), and to provide for continuity in the decomposition and location of requirements across the family of specifications for a given flight simulator.

The *content* of the Flight Simulator Guide Specification has also undergone a major change to reflect the latest Air Force simulation requirements. Guidance is now included for aircrew trainer types ranging from part-task trainers to weapon system trainers to mission rehearsal devices. Both fixed-wing and rotary-wing aircraft simulators are covered. There is an increasing demand for simulators that are affordable in large quantities[1] and which can be placed in small facilities that are not necessarily environmentally controlled. This market requires scaled-down simulators possessing something less than full fidelity simulators; therefore explicit guidance has been added for dealing with "selective-fidelity", i.e., the tailoring of fidelity to the specific application. Visual/sensor database generation system accuracy and throughput specification guidance is included in the Guide Specification to support mission rehearsal simulations. Process guidance is added to suggest processes that should be established to better define and document requirements; this guidance is intended to flag key specification interfaces with development processes.

As with predecessor Guide Specifications, recommended language is provided for function, performance, verification methods, and options. These recommendations use terminology consistent with MIL-STD-490B, and provide wording that defines each requirement in terms amenable to verification. Consistent use of both requirement and verification language is emphasized and encouraged by conventional approaches, as well as the hypertext authoring tools discussed later.

KEY PRINCIPLES

An effective Flight Simulator Guide Specification should serve as a tool that facilitates the process of translating user training requirements into specific characteristics of a simulator device. This process starts with very general requirements such as: "practice air-to-air combat, air-to-ground weapon delivery, and emergency procedures at their regular training bases and operating sites." As these top-level requirements are evolved into flight simulator devices, questions that are asked repeatedly are

- "What do you really mean?" "Is this what you really intend?" "Will this system meet your needs?" These questions are asked repeatedly during the requirements' development process, source selections, development of the device, and test of the device - and these questions are repeated when it is necessary to modify a device. The specifications should capture the answers to these questions as the acquisition progresses. Four key principles can make this happen. These are:

a. Standardized language and terminology.
b. Object oriented decomposition.
c. Use of a structural model.
d. A logical top-down specification development process.

Standardized Language and Terminology

A famous children's book uses the phrase, "I meant what I said, I said what I meant" (Suess, 1940). This principle has generally been the exception rather than the rule in specification development; nevertheless, it is fundamental to writing good specifications. Consider the following examples drawn from the current AFGS-87241A - "The DRLMS[1] shall be designed to satisfy the accuracy specified in the following paragraphs..." (3.9.1.1), "The image generator shall have the capability to retrieve and process..." (3.7.2.4) and, "The engine systems to be simulated shall include but not be limited to..." (3.3.7.2). Does the first sentence mean that the DRLMS will be designed to meet the requirements, but not fabricated to meet them? Does it mean that no testing will be required? Does the second sentence mean that the image generator need retrieve and process data perhaps one time out of 100 requests? Clearly these were not the authors' intentions. In the third sentence the engine systems are listed and its author intended "include, but not be limited to" as a contingency statement in case something was overlooked - but what the sentence really says is that even if a completely accurate list is provided, the contractor must provide something else. These comments may appear as "nit-picking" but the authors personally know of several occasions where long government and contractor discussions have revolved around such minor discrepancies in terminology. Another major language problem encountered is purely semantic - words have different meanings to different people. For example, does the word "platform" refer to an aircraft carrying a radar system or to a motion platform? What do the terms "user friendly", "high fidelity", and "adequate to train" really mean? The new Flight Simulator Guide Specification addresses these terminology and semantic problems with five basic rules.

The first rule is, "The system shall do it." In the first example above, what was obviously intended was, "The DRLMS shall meet the accuracy requirements of the following subparagraphs." In the second example the intention was, "As the simulated aircraft moves across the gaming area, the image generator shall retrieve and process..." So why not say it? In almost every instance words such as "shall be designed to" or "shall be capable of" are used to provide variety in writing style. However, this also leads to variety in interpretation. Specifications should communicate requirements - they do not have to be interesting reading.

The second rule is, "Say it once." MILSTD-490A permitted a large degree of interpretation regarding the organization of various requirements. Because of this, there has been a tendency to cover the same item in different sections of the specification. Instructor controls are one classic example. In the past, instructor controls were often discussed both in an instructor console section and in sections such as the simulated aircraft and simulated environment. When a requirement is stated in more than one place, authors tend to write the requirement differently in each place. In addition, a specification document will often be changed during the development process, and it is extremely likely that a requirement may be changed in one place and not another. If a requirement is stated in one place - and one place only - the likelihood of internal conflicts arising in the specification is reduced. The *architecture* of the Guide Specification, which is discussed later, is one mechanism used to achieve the goal of stating a requirement in only one place. In addition, the recommended language isolates discussion of simulator controls from the effects of those controls. As an example, refer to Table I and Table 2. The language

---

[1] Digital Radar Landmass Simulation

Table I

Definition of fidelity levels for crewstation instrumentation, controls, and displays.

Replicated	the correct location, and have the appearance and feel of the aircraft equipment as defined by the approved design criteria. Shall be interfaced with the simulator computational system such that the computer can identify the state of the controls, and drive the instrumentation and displays in accordance with this specification.
Depicted	Need not be in the exact location, nor have the appearance or feel of the aircraft equipment Shall be interfaced with the simulator computational system such that the computer can identify the state of the controls, and drive the instrumentation and displays in accordance with this specification.
Inert	Shall be in the correct location, and have the appearance and feel of the aircraft equipment as defined by the approved design criteria. Need not be interfaced with the simulator computational system.
Pictorial	May be a static picture of the aircraft equipment, but shall be of the size and in the location defined by the approved design criteria.
Not Required	No representation of the aircraft equipment is required.

recommended in Table 2 defines which flight controls are to be included in the simulator (and the appearance and tactual fidelity of those controls), but does not attempt to state how the simulated aircraft should respond to those controls; the aircraft response is handled in one place under "Air Vehicle Dynamics.

The third rule is, "One name - one meaning." This avoids the "platform" problem discussed earlier. To facilitate this rule, the Guide Specification includes definitions that are easily accessed using some of the Guide Specification Toolkit features discussed later.

In addition, the Guide Specification encourages the clear definition of key performance characteristics (such as "freeze", "record", and "replay") right from the outset of a specification's development. This is necessary to ensure consistent understanding and implementation of these characteristics across all simulator subsystems. In previous specifications, performance characteristics have often been called out as "one-liners", addressed differently in different sections, or sometimes not mentioned at all. Where performance characteristics have not been well defined, time has been lost in lengthy deliberations to arrive at a consensus regarding the "real" user requirement. These situations have also provided additional opportunities to arrive at inappropriate implementations and ultimate user dissatisfaction. To help avoid these problems, the Guide Specification includes the following:

a. Simulator modes are defined up front. These are defined to be, "a simulation state or collection of simulation states which represent fundamental ways of operating the simulator from the viewpoint of a crewmember or operator." Examples include normal operating mode, networked operation, maintenance mode, and part task operating modes.

Table II

Examples of recommended language, verification methods, guidance, and embedded examples extracted from the "Flight Controls" simulation requirements.

Requirement.	The ___1___ flight controls shall be ___2 ___. Gearing shall be in accordance with the approved design criteria. Simulated force feedback at the flight controls shall be___3___. ___4___shall be ___5___.
Verification.	This requirement shall be verified by inspection and test. Inspection shall verify compliance with appearance and location requirements of the flight controls and controls located upon these flight controls. The tolerances specified in this paragraph shall also apply to relevant dynamic tests conducted in accordance with...
Requirements Guidance.	Blank (1) should identify the flight controls to be included, and should be tailored to the application. This would typically be wheel, column, and pedal' for a classical transport aircraft... In lower fidelity applications all flight controls (e.g., pedal) may not be required. Blank (2) should state the required level of fidelity for the flight controls, which would typically be "replicated, and have displacement in accordance with the approved design criteria." For a very low fidelity device (where a simple joystick might suffice)... Blank (3) should define the force simulation required, in accordance with the following guidance: Full-Fidelity Simulation Of Traditional Mechanical Flight Control Systems; In this case, high quality force feedback is required. Put the following into blank (3)... Lesser Fidelity Simulators: This might apply for devices intended for uses such as procedural refresher training of mission-qualified pilots... If full-fidelity force-feedback is not required, put the following into blank (3)... and (5) are intended to capture the fidelity requirements of the switches, buttons, and any other controls located on the flight controls
Verification Guidance.	The "Verification of Flight Controls" paragraph is written for a full-fidelity simulation of a traditional mechanical flight control system, and should be tailored downward where requirements call for lesser control loading fidelity. Tests or demonstrations should be retained for any quantitative requirements such as gearing, control envelope, and control free response....
Process Guidance.	If a level of fidelity "representative of the aircraft force feedback" is specified in blank (3), a process should be defined which results in: (1) the force-feel simulation being prototyped and demonstrated with the user, and (2) the documentation of the acceptable force-feel transfer characteristic, and its incorporation into the approved design criteria.
Example.	Where a lesser-fidelity device is being acquired for supporting practice of highly procedural tasks, substantial cost savings might be realized by reducing the fidelity required of the force-feel system. For example: "Flight Controls. The stick flight control shall be replicated, and have displacement in accordance with the approved design criteria. The pedal flight control shall be inert and need not pivot, but shall be adjustable fore and aft (using the Rudder Pedal Adjustment on the Control Pedestal) over the full range specified In the approved design criteria. Gearing shall be in accordance with the approved design criteria. Simulated force feedback at the flight controls shall be representative of the aircraft force feedback, but need not provide full tactual fidelity and may be realized using passive control loading devices. Force feedback shall be in accordance with the approved design criteria. The two position gun trigger on the stick shall be inert. The following stick controls shall be replicated: a. Four-way aircraft trim switch. b. Weapon release button. c. Fore/aft/down auto-acquisition switch."

b. Up-front definitions are provided for events such as freeze, crash, halt, and malfunctions. The malfunction definition includes the requirement that, "Other systems shall respond to this change of system state in a natural manner in accordance with the design criteria... i.e., the malfunction effects shall propagate through the simulated system in a manner analogous to failure propagation through the real system." This also forces software models into closer correspondence with the real world; the benefit of this is discussed in the next section.

c. Guidance is provided so that activities such as record, replay, stabilization, crash override, set and reset of various simulator conditions, and mission time management are all explicitly defined from the start.

Simulation requirements are often very difficult to describe. Quantitative requirements are used in the Guide Specification where possible, but there are many instances where it is extremely difficult - or simply not feasible - to write quantitative requirements. Qualitative terms, such as those dealing with required levels of fidelity, are defined at the outset so that all parties will have a common understanding of their meaning. For example, the level-of-fidelity definitions provided in Table *1* appear in the Guide Specification prior to their use in lower level paragraphs. This leads to the fourth rule, "If fidelity must be described qualitatively, use a series of well-defined adjectives and define it consistently." Table 2 provides some examples regarding the use of the level-of-fidelity definitions in recommended specification statements.

The final rule is, "Distinguish between process and product." Process is very important --as evidenced by the emphasis on process throughout the acquisition community. A sound process is essential to produce the device in a manner that converges the user expectations and the final product. Previous specifications have often referred to processes in the form of requirements to conduct reliability and maintainability or integrity programs, requirements to identify malfunctions at design reviews, and the specification of tests to be performed at the contractor's plant versus on-site. The problems with this approach are:

a. There is often overlap with the Statement of Work. This leads to the risk of violating the "say it once" rule.
b. There is no closed loop; the process may change the characteristics of the device, but the changes are not captured in the specification.
c. Authors may inadvertently discuss process when they mean product; this produces ambiguous meanings.
d. Process must be tailored to the particular device, the particular program, and the particular acquisition agency.

The Flight Simulator Guide Specification deletes all process language from the specification text. It then includes process guidance in the rationale section. This rationale section is intended to support development of the Statement of Work. Fig. 1 provides one example wherein a process is recommended to update the specification once the necessary information becomes available. Table 2 provides another example wherein a process is recommended to quantify level-of-fidelity requirements and then include these in the approved design criteria.

An agency is preparing a System Specification with minimal security information available. The Statement of Work tasks the contractor to perform a System Security Engineering Program. The specification language recommended by the Guide Specification in this case would be:

3.3.9 System Security. The vulnerabilities of the simulator shall be minimized.

The Process Guidance in the Rationale Section of the Guide Specification would recommend that the Program Statement of Work require the contractor to update the specification when new information becomes available as a result of the System Security Engineering Program.

Process Guidance: A System Security Engineering Program should identify threats and vulnerability. It must also require appropriate contractor interface with the accrediting or certifying authority. It must provide appropriate information for this specification when necessary.

After the security issues are resolved, the specification is updated to read:

3.3.9 System Security. The follow vulnerabilities of the simulator shall be minimized with the countermeasures indicated:

a. Software viruses. A commercially available virus protection program shall be incorporated in the simulator such that all software files are checked on installation.

b. Access to classified data by personnel not cleared to the appropriate level. Use of the system shall require a login password and unique id with authentication data. The probability of a guess shall be .000001 or less. There shall be no group ids. A Discretionary Access Control (DAC) system shall insure that classified data is accessed on a need- to- know basis. In addition The computational system shall be declassified by 1) removable hard disc drives and 2) an overwrite algorithm. The overwrite software shall overwrite the computer memory locations a minimum of three times.

c. Physical damage or destruction. An alarm shall indicate access to the cockpit unless authorized at the instructor console.

d. Access to the KY 601 Panel. The KY 601 Panel shall be easily removable when the simulator is not in use.

The highest level of Information processed in the simulator shall be SECRET.

Figure 1. Separation of process and product in the Guide Specification. An example drawn from the System Security section.

Object Oriented Decomposition

Standardized language and terminology are essential, but the architecture of the specification itself can be just as important in communicating clear requirements. The traditional flight simulator specification architecture is a product of the evolution of the flight simulator. Flight simulators began as devices to provide instrument training. They were essentially a simulation of the dynamics of the air vehicle. As technology improved, simulators became much more complex. The industry added motion systems, visual systems, complex avionics and electronic combat simulations, Digital Radar Landmass simulations (DRLMS), and complex instructional systems. As systems were added, simulator manufacturers' engineering departments added experts to deal with each of these functional areas. Government personnel became likewise specialized. These experts tended to focus on their specific functional areas rather than the whole system; this led to specifications being organized around functional areas. As systems became more complex, this contributed to ambiguity in requirements.

The traditional specification breakout illustrated in Fig. 2 arose from this focus on functional areas. How are the requirements described under this traditional approach? The air vehicle section covers aircraft performance.

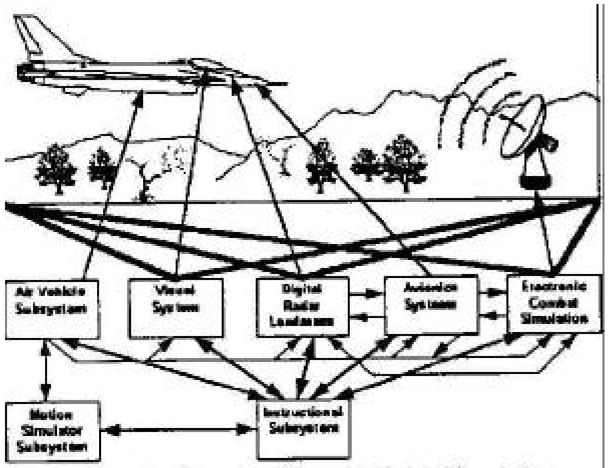

Figure 2. The traditional Flight Simulator Specification architecture.

The avionics section covers the avionics computer that interfaces with the electrical system, which is in turn covered under the air vehicle. Aircraft radar is partially covered under avionics and partially under Digital Radar Landmass (DRLMS). The visual and radar sections both deal with terrain and the same portion of the earth's surface; these requirements differ only in regard to appearance attributes at different portions of the electromagnetic spectrum. Features, such as enemy radar, must appear both on the visual system and the aircraft radar - and are covered in both sections. The electronic combat section not only covers enemy radar illuminating the aircraft, but also aircraft displays covered under avionics, visual, electronic combat, DRLMS, and the air vehicle all cover weather effects. Instructional requirements are frequently covered in multiple places, and the instructor console requirements interface with everything else. This traditional breakout leads to a great deal of overlapping requirements, and affords ample opportunities to violate the "say it once" rule. In addition, with this traditional breakout there is no direct correspondence to the real world; this makes it more difficult to relate simulator requirements directly with user requirements, which are often expressed in real world terms.

The Flight Simulator Guide Specification takes a different approach. It incorporates an architecture that corresponds more to the real world. All requirements on the primary air vehicle are in a simulated air vehicle section. This section parallels ASC's Air Vehicle Guide Specification. The remainder of the real world is represented by the simulated environment. This includes other vehicles (entities), the terrain, magnetic variation, celestial objects, and electromagnetic radiation. Interactions are natural and implicit; vehicles hide in terrain and weather attenuates radar signals. Translating the military mission into simulator requirements is simplified and straightforward, relative to the traditional breakout. The Flight Simulator Guide Specification approach is illustrated in Fig. 3.

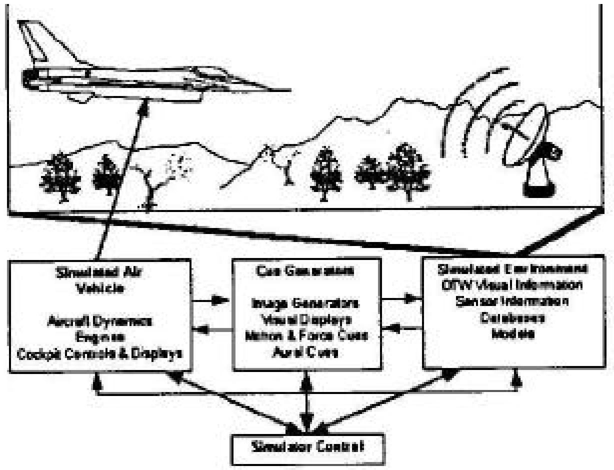

Figure 3. The architecture employed in the new Flight Simulator Guide Specification.

We might view a simulation as a series of models representing the aircraft and a series of databases and models representing the rest of the world. In nature there is natural interaction between the vehicle and the environment. For example, if a window is put into the vehicle the pilot can see the world. However, merely putting a window in the simulator gives the pilot a nice view of the closest wall. Systems are needed to produce the interaction between the simulated air vehicle and the environment. The Guide Specification calls these systems cue generators. This section of the Guide Specification includes such requirements as motion systems, image generators, and visual displays.

Traditional specifications typically include the requirement for instructor consoles or systems in one section, but disperse many of the interface and interaction requirements to the relevant subsystems. The Guide Specification avoids this scattering of simulator control requirements in the following way. In the simulated air vehicle and environment sections, the term "commanded" is used to denote that a system is to respond to an instructor, operator, or automated control input. However, details regarding interactions and implementation are all included in the simulator control section. This section also deals with the display of information an instructor may require to operate the system or evaluate trainee performance.

The final section of the Guide Specification (which does not appear in Fig. 3) deals with system support. This includes requirements such as those for a Training System Support Center or a database generation system.

Use of a Structural Model

The Software Engineering Institute or SEI (1992, p.5) offers the following definition for a structural model. "A structural model is a pattern for specifying and implementing software system functionality. It reflects system engineering decisions about partitioning and coordination."

The structural model concept was used on B-2, C-17, and Special Operations Forces Aircrew Training System simulators. The partitioning strategy suggested for a structural model is an object-oriented decomposition wherein software objects are related to real world entities (Software Engineering Institute, 1992). This inspired the object-oriented partitioning strategy discussed in the previous section. The object-oriented architecture of the Guide Specification should, in itself, facilitate the specification and implementation of software system functionality. However the "Computer Resources" section of the Guide Specification goes further, and explicitly requires a structural model as follows:

> "All simulation software shall be object oriented. The software architecture shall have three levels; they are: a) executive level, b) subsystem level, and c) component level. Each subsystem level shall model the systems on the real aircraft and each component shall model the components of the aircraft being simulated (e.g., fuel pumps, turbines, etc.). The executive level shall coordinate the subsystem level. The subsystem shall manage a group of components at the component level so that they will behave as a unit. The component level shall be concerned only with computation. Each subsystem shall have no direct knowledge of other subsystems. All information shall be transferred from the memory locations where each subsystem will place its data. The components shall have no knowledge of the outside world except through input or output parameters. Any systems outside the aircraft such as radars, missiles or other aircraft shall also be simulated using structural modeling and object oriented design."

Additionally, use of a structural model will help enforce consistency across all simulator subsystems. One of the basic concepts of the structural model is the use of common software templates. This should impose consistent software implementation by designers.

The Training Systems Program Office (ASC/YT) is developing a Structural Model Handbook with the SEI. This is intended to communicate concepts and design approaches to the flight simulation community as a means to help transition this technology. Once complete, a reference to the handbook will be included in the guidance section.

A Logical Top-down Specification Development Process

The Flight Simulator Guide Specification is a lengthy, complex product. It was started as a guide for writing:

a. A System Specification for a simulator procured as a stand alone device.
b. A Complex Item Requirements Specification (CIRS) for a simulator procured as part of an aircrew training system -- or a standalone system where there was a requirement to expand and clarify the System Specification.

The Guide Specification is intended for government or contractor use. Current policy at ASC requires that contractors write specifications and that the government writes an abbreviated requirements document called a System Requirements Document (SRD). Since an SRD will be expanded into a Systems Specification and Systems Specifications may be expanded into CIRS, the same overall organization should be used with each lower level of specification containing more detail. Thus the Guide Specification can be used as a guide for SRD preparation. It is essential$_1$ however, that the specification development process proceed in a logical manner.

Specification or SRD development should not begin until after an Instructional Systems Development process has identified the key requirements of the device. Then, the first step in developing the specification or SRD is to name the simulator and identify its purpose. This is accomplished by filling in the blanks In the Guide Specification paragraphs shown In Fig. 4. Thereafter, as also shown in Fig. 4, key physical constraints are identified. The constraints should match the specific application; for example, if the device is to be portable and fit into an office, there should be constraints imposed on size and weight.

1. SCOPE. This specification establishes the requirements and associated verification methods for _____.

Requirements Guidance: fill in the blank with device name, nomenclature (if available), and aircraft represented (if applicable).

3.1 System Definition. The___1___ is intended for use ___2___ in ___3___ at ___ 4__. It shall comply with all requirements of this specification.

Requirements Guidance*: Fill in blanks as follows:*

1. *Fill in device name*
2. *Fill in who uses and their initial qualifications*
3. *Fill in task(s) to be accomplished*
4. *State where the device is to be used*

Physical Characteristics. The simulator shall consist of ___1___ The simulator ___2___.

Requirements Guidance:

1. *This blank should describe the principal physical entities (e.g. cockpits, crew stations, instructor stations, motion bases, etc.) which make up the simulators well as the physical relationships between them.*
2. *This blank shall describe any limitations on the dimensions or weight of the principal physical entities. The sentence may be deleted if there are no dimensional or weight limitations.*

Care must be taken to avoid unnecessary size or weight restrictions Dimensions must be consistent with other requirements (e.g. the height must be consistent with visual display requirements).

Figure 4. The first steps to the top-down development of a specification.

The next step is to identify the major vehicle and environmental characteristics that must be simulated. In addition, there must be a check for consistency between the device's purpose and its physical attributes. Is the right portion of the cockpit required? Are all needed environmental characteristics provided? Are physical constraints consistent with other requirements?

The process proceeds through succeeding steps in a similar manner. As it progresses, several facts become evident:

a. The order in which paragraphs should be written becomes less obvious.

b. Multiple authors can work on paragraphs independently.
c. Certain paragraphs may be inappropriate to the level of specification being written.

The Guide Specification Toolkit, discussed in the next section, includes a "User Guide" utility to facilitate this top-down specification development process.

THE GUIDE SPECIFICATION TOOLKIT
A HYPERTEXT-BASED SYSTEM

The new Flight Simulator Guide Specification includes an authoring system toolkit that runs under Microsoft Windows. This provides a familiar graphical user interface for specification development that features the same sorts of menus, dialog boxes, and "point-and-click" controls found in other Microsoft Windows applications. To promote usability and acceptance, every effort is made to make the user interface in this toolkit as consistent as possible with other Windows-based programs so that the interface controls respond in the way expected by the user.

This Guide Specification includes the Windows-based toolkit for several reasons. First, the specification includes a large volume of specially formatted text that must be tail6red significantly. Editing with a mouse-based, graphically-interfaced authoring system can be considerably easier than working with a simple text editor. Second, Microsoft Windows is a widely-used, well-accepted, graphical mouse-based interface. As such, the interface should be familiar to most of the targeted users of this toolkit. Third, it is easy to import text from many of the other Windows word processors into this application. Finally, the Windows graphical user interface inherently supports the implementation of hypertext links.

Hypertext is useful because it can greatly expedite navigation through large volumes of text, and facilitate the access of data from a variety of directions (Midford, 1989). The essence of hypertext is computer support for links within and between documents (Conklin, 1987). Say, for example, you were reading an encyclopedia article entitled "Primates", and at the bottom of the article the text read "see Mammals." If you wanted to know more about mammals, you would flip to that article, read through it and find the pertinent information. In a hypertext environment, the computer would automatically look up the article on mammals for you when you clicked on the word "mammals" (which would be designated as a "hotword" in the text). Such hypertext techniques are widely used in Windows-based programs.

The software toolkit provides a range of productivity enhancement tools that would otherwise be unavailable in a strictly hardcopy format. Some are discussed below.

Support for Standardized Language and Terminology

Software-controlled linking is exploited in order to foster common usage and understanding of terms. Two tools are provided: (1) a glossary of definitions and acronyms, and (2) graphics to convey key concepts in conjunction with the definitions. The hyperlinked glossary, illustrated in Fig. 5, provides quick and convenient access to definitions of terms from anywhere within the toolkit. The capability will also be provided to automatically write these definitions and

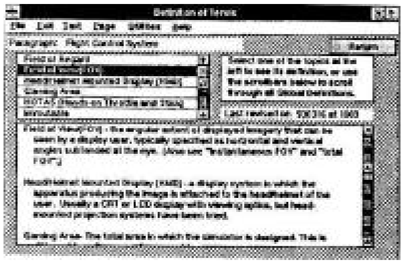

Figure 5. This glossary of terms is conveniently accessed through a pull-down menu from any where within the Guide Specification Toolkit.

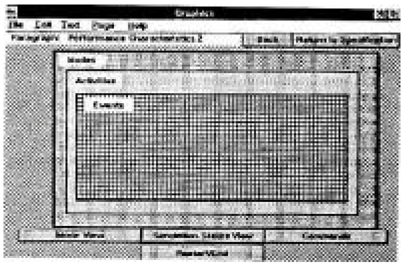

Figure 6. A graphical representation of the relationships among simulator modes, activities, and events that is displayed in response to selecting the "Mode View" button. This page is accessed by clicking an icon on the Guide Specification pages dealing with these topics.

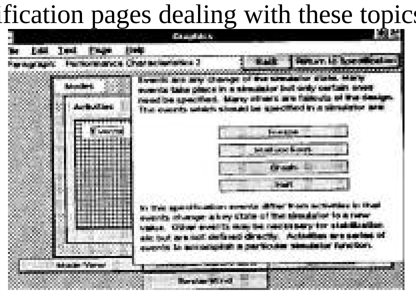

Figure 7. Clicking on the word "Events causes this definition to be displayed. Clicking on any event name will display the definition of that event.

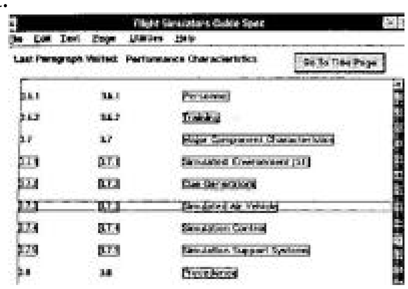

Figure 8. The uppermost level of the table of contents pages. Clicking on a paragraph name links directly to that paragraph. Clicking on a hot paragraph number links to the next level-of-detail table of contents menu.

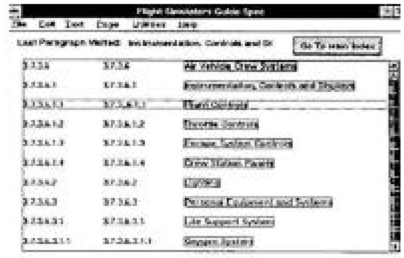

Figure 9. The table of contents menu reached by clicking on 3.7.3 at the preceding level.

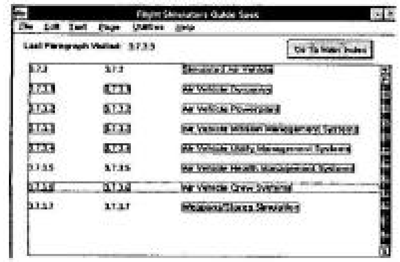

Figure 10. The highest level-of-detail table of contents menu. This is reached by clicking on "3.7.3.6" at the preceding level.

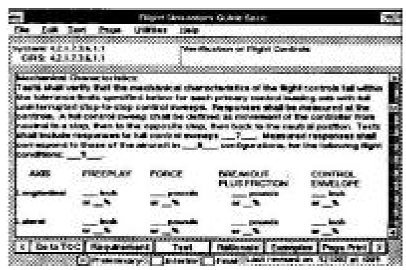

Figure 11. The Flight Controls "page" of the Guide Specification Toolkit. The page as shown is opened to the verification paragraph.

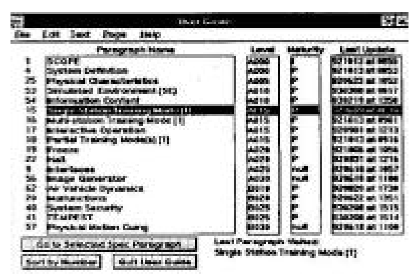

Figure 12. The User Guide tool that assists in developing specifications in a logical, top-down manner. Paragraph names are shown sorted by level. Clicking the "Sort by Number" button would order paragraph names in the order that they appear in the Guide Specification.

acronyms to Section 6 of the hardcopy specification documents in MIL-STD-490B format. Supplemental graphics are used to illustrate and clarify key underlying concepts. These are accessed by clicking an icon that appears on those toolkit pages having such graphical explanations. Fig. 6 and Fig. 7 provide an example by way of graphics used to illustrate the relationships among modes, activities, and events. An icon appearing on any of the toolkit pages dealing with "Performance Characteristics" associated with these topics will instantly link the user to a graphic such as shown in Fig. 6. Clicking on the appropriate control will then bring up further detail, as illustrated in Fig. 7. If desired, definitions for any of the events shown can be obtained by clicking on that event's name. These tools provide not only convenient and rapid access to definitions, but serve to explain concepts in ways that cannot be accomplished nearly as conveniently in hardcopy format. Use of these tools should facilitate consistency in the specification language and the interpretation of that language.

Efficient Access to Material

The table of contents is designed as a multilevel menu to hasten the process of locating a desired paragraph. A hierarchy is employed so that no paragraph titles beyond three levels of indenture appear in the main table of contents (Fig. 8). This provides the user a top-level view of the organization, and facilitates identification of the desired topic area. Clicking on appropriate "hotwords" (designated by the boxes surrounding the text) permits the user to either go to a lower level of detail in the table of contents, or directly to the desired paragraph. For example, clicking on "3.7.3" in Fig. 8 will cause the main subparagraphs under "Simulated Air Vehicle" (Fig. 9) to be displayed. In order to proceed to the subparagraphs under "Air Vehicle Crew Systems", the user would then click on "3.7.3.6" in Fig. 9 - which would bring up the screen shown in Fig. 10. At this deepest level-of-detail, only paragraph titles appear as hotwords. Clicking on a paragraph title anywhere within the Table of Contents will immediately access the designated page. Clicking on "Flight Controls" in Fig. 10 would take the user to the Flight Controls specification "page" as illustrated in Fig. 11.

Each guide specification "page" incorporates all the guidance paragraphs for a specific topic such as "Flight Controls". The paragraph numbers (both System Specification and CIRS, in accordance with MIL-STD-490B) and name appear at the top of the page. The relevant text is displayed in a scroll box in the center (this text can be edited or copied by the user). Underneath the text box is a row of buttons by which the user can display:

- the recommended requirement or verification language (buttons "Requirement" and "Test", respectively).
- rationale and guidance ("Rationale" button).
- examples of completed paragraphs.

Other buttons permit the user to obtain a hardcopy output of the contents of the entire page ("Page Print") or to navigate within the toolkit (the "<" and ">" buttons move to adjacent pages, while the "Go To TOC" returns the user to the Table of Contents). Under the buttons are three checkboxes that are used by the specification author to assign the maturity level for each page. In addition, a time stamp is provided which indicates the last time the page was revised.

Support for a Logical Top-down Specification Development Process

"User Guide", illustrated in Fig. 12, is a user-selectable utility that is provided to facilitate the top-down specification development process discussed earlier. Essentially, User Guide utilizes a hierarchical structure, and guides the user to develop the specification in accordance with that structure. The User Guide's display page provides summary information for all relevant paragraphs regarding their place in the hierarchy (or "level"), their maturity status (null, P or preliminary, I or interim, and F or final), and their last revision date. Hyperlinks are provided which allow the user to navigate directly between specification pages and the utility display page, so that the specification paragraphs can be readily accessed and edited in the proper order.

Each Guide Specification paragraph may be assigned a "level", which consists of a single letter and a three digit number. The three digit number determines the order in which specification paragraphs should be written within a "group" - i.e., lower numbers should be written first. The letter is the "group" designation. Group "A" must be developed first. Groups "B" through "Z" can then be developed in parallel. The maturity of a paragraph within a group cannot be greater than all paragraphs with lower numbered levels. The User Guide utility reads the maturity status from the checkboxes at the bottom of the relevant pages (see Fig. 11), and determines whether this rule was violated. If so, the user is provided a warning along with contextual guidance regarding how to proceed.

The User Guide is an optional tool for specification writers. It identifies a preferred order in which specifications should be written and provides indications when the order is violated. The specification authors are allowed to omit paragraphs and to edit paragraph order before using it. Its use then forces requirements' decisions to proceed logically on the basis of previous, more fundamental decisions.

PLANS AND PROGRESS

The Guide Specification Toolkit has already undergone a number of in-process reviews, and is now at a point where it can be more widely distributed for evaluation. Revision of the specification content is nearing completion, and is currently undergoing internal reviews. It is anticipated that the Guide Specification will be released for industry review during the summer of 1994. We will welcome any and all feedback from those reviewing our product - and are especially interested in feedback regarding:

a. Content.
   - Did we follow our rules? For example, were requirements stated in only one place and did we say what we meant?
   - Do the recommended requirements language and guidance makes sense?
   - Do the recommended requirements and verification language and guidance yield requirements that can be evaluated? Is there a definitive basis for acceptance or rejection? Does it make sense?

b. Architecture.

- Does the way in which the requirements were allocated against the architecture make sense?

c. Guide Specification Toolkit.
    - Is the Windows-based toolkit easy to use? Does it add value to the specification-authoring process?
    - Are there toolkit features that should be implemented in a different way?
    - Are there additional toolkit features needed, or enhancements that would improve the process?

SUMMARY

Our desire is to build a useful tool that will help both us and the simulation industry. We hope that the emendation of content and architecture truly leads to improved specifications and that the added Guide Specification Toolkit features serve to facilitate the tailoring process. It is our expectation that making this Guide Specification easy to use will enhance its application, and lead to greater standardization in specification format and language. We need your views, and look for serious feedback from the industry review.

ACKNOWLEDGMENTS

The authors would like to thank Lt. Brett Borghetti for his contributions supporting the Guide Specification Toolkit discussion.

REFERENCES

AFGS-87241A, U.S. Air Force Guide Specification for Flight Simulators. 9 Jan 1989.

Conklin, J. *Hypertext: An introduction and survey.* Computer, 21(1), pp.17-41, Sep 1987.

DOD Instruction 5000.2, Defense Acquisition Management Policies & Procedures. Part 6, Section Q. *DOD standardization program,* and Part 10, Section C. *Acquisition streamlining,* 23 Feb 1991.

Gershanoff, H. *Liar's dice.* Journal of Electronic Defense, p.14, July 1988.

Gordon, J.K. *USAF writes new specifications to ease its prime contractor role.* Aviation Week & Space Technology, pp.83-84, 5 Aug 1985.

Industry News. *AVIP demands better electronic systems.* Defense Electronics, p.34, Feb 1989.

Midford, R. *Vendors probe hypertext for repair crews.* Federal Computer Week, p.26, 20 Mar1989.

MIL-HDBK-248B, Acquisition Streamlining. Section 8. *Streamlining tools and techniques,* 9 Feb1989.

MIL-STD-490B, Preparation of Program-unique Specifications. 7 December 1992 (draft).

MIL-STD-680B, Standardization Program Requirements for Defense Acquisitions. 1Mar1990.

Software Engineering Institute. *An introduction to structural models.* Presented to Interservice/Industry Training Systems and Education Conference (I/ITESC), Training Systems Program Office (ASCIYT), Wright-Patterson AFB OH 45433, ASC 92-2719, 1 Nov 1992,

Suess, Dr. *Horton hatches the egg.* Random House, New York, 1940