INTEGRATED CBT SMART GRAPHICS: COST-SAVING GRAPHIC GENERATORS AND SIMULATIONS

Mark C. Schuetz and Thomas M. King
F22 Training Analysis & Integration
Boeing Information, Space and Defense Systems

INTRODUCTION

Better, faster, cheaper, pick any two, or so the expression goes. Boeing decided instead to do all three with new graphic development tools used for 777 computer based training. These tools, referred to at Boeing as “smart graphics”, have evolved from simple panel graphics generators to more sophisticated simulations.

For the Boeing 777, students would learn from over 100 hours of computer based training, along with classes and simulators. To assemble all this courseware on time and within budget new production techniques would be needed. Among the techniques used, were smart graphics which helped maintain consistency and configuration control yet reduced cycle time and cost.

Smart graphics were critical to the success of 777 training, and have become an integral part of subsequent Boeing training development programs. With each new development cycle, smart graphics improved and evolved. This paper discusses the benefits attained through smart graphics and looks at the evolution of the smart graphic including future enhancements.

SMART GRAPHICS DEFINED

Smart graphics are a means of more efficiently producing images for computer based training, CBT. Aircraft panels and displays typically consist of an array of simple elements, such as toggle switches, dials or basic symbols. To produce training one generally needs these panels in a variety of configurations. Without smart graphics a simple panel with 3 toggle switches and 2 four-position rotary switches, could expand into 128 possible configurations. Rather than waiting for individual graphic requests an artist and programmer can work together to create an image or code module that will produce all of these combinations. This is how a smart graphic functions as a graphic generator.

Graphic Generators

First and foremost, smart graphics are graphics generators. At the simplest level graphic generators can be created with a base graphic and overlays. The end-user simply drops pre-configured elements on top of the base elements. This can be as simple as dropping a toggle switch in the “up” position on top of a base switch in the default “down” position. While this requires nothing more than for the artist to create the base elements and the overlays, the end user must properly align the overlays (See Figure 1).

Simple “State Machines”

Smart graphics that incorporate logic are actually simple state machines. The user can click on an individual switch to cycle through the various positions. By combining groups of configurable switches and knobs, the number of states multiplies dramatically. This allows hundreds of potential configurations to be built from a few basic elements.

These simple state machines can be built in virtually any authoring tool or programming language. When we discuss second generation smart graphics however, we will see that there are certain economies of scale to be gained by selecting a tool that integrates with your general CBT development effort.

ADVANTAGES OF SMART GRAPHICS

Reduced Graphic Requests

By focusing on the larger object such as a panel, rather than a specific configuration of that panel, fewer individual art requests are needed. Furthermore, by working within a larger context the artist can see how elements will need to work together. This reduces re-work when compared to a situation where different artists might work on different configurations of the same panel.

Full-time Availability

Other approaches to graphic development may require captures from the aircraft itself, a simulator, or other training device. Typically, such devices are not readily available for training development. However, a smart graphic stored on a network file server can be available 24 hours a day for subject matter experts (SMEs) to use to generate training.

Reduced Cycle Time/Immediate Turn Around

Because fewer graphics requests are generated, there is less overhead and fewer individual pieces of work to track. Additionally, a SME who needs "just one more configuration" of a panel, can now just click and generate it himself.

Better Configuration Management

Since all configurations of a given panel now come from a single source, there are fewer opportunities for variance and updates are centralized. In the initial example there were 128 possible configurations for that panel. Instead of cataloging, revising, and searching through 128 different graphics, the SME simply checks a single smart graphic out of the library and creates the required graphic.

Cost Reduction

Because of the reduced overhead, reduced cycle time, improved availability, and simplified maintenance, costs are reduced. Further cost reductions are available when the cost of simulators and other device access is factored as an expense.

TYPES OF SMART GRAPHICS

For the purpose of discussion we will consider two categories of smart graphics, referred to as first generation and second generation. First generation smart graphics focused primarily on improving graphics production. Second generation smart graphics also address CBT delivery and stand-up instruction.

FIRST GENERATION SMART GRAPHICS

Characteristics

A first generation smart graphic has little or no logic associated with it. It is a simple state machine where a feature is either on or off, or in one of several positions. As stated earlier these smart graphics are simply graphic production tools. They allow the end user to mass-produce different standard configurations of a graphic. There are really no smarts associated with a first generation smart graphic.

First generation smart graphics stand-alone, separate from CBT. They require a screen capture utility, e.g. Paint Shop Pro for Windows, Exposure Pro for Macintosh, to produce a graphic suitable for inclusion in the CBT. They do not operate with the CBT. First generation smart graphics are not intended as teaching aids. They simply generate graphics to be used in the CBT. First generation smart graphics can be divided into two categories: "Drag & Drop" and "Toggle/Cycle."

First Generation "Drag & Drop"

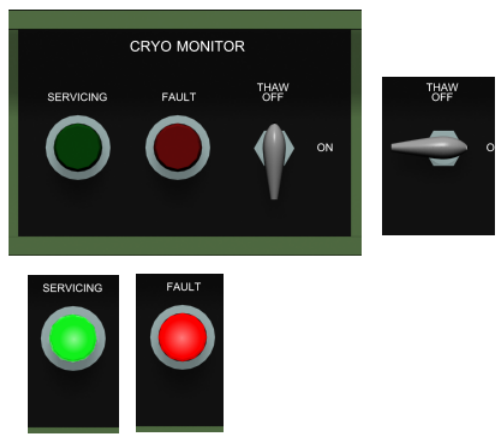

Figure 1. Drag & Drop Smart Graphic

The most elemental form of smart graphics is the "Drag & Drop". This type is not really "smart" at all, but really the approach of a clever graphic artist. The "Drag & Drop" smart graphic relies on image editing software that supports layers or multiple independent bitmap objects. These objects can then be moved and "dropped" on top of each other to create new combined images. For example, in Figure 1, dragging the "extra" switch into position can change the switch position. Then a developer can "flatten" the stacked objects by copying them all at once or using a screen capture utility to grab the combined image. The new image can then be inserted into CBT or electronic documentation.

First Generation "Toggle/Cycle"

The first step towards "smartness" in a smart graphic is adding the capability to toggle or cycle through the various states, without requiring the end-user to carefully position the individual elements. The 777 Air Conditioning Panel shown in Figure 2 is a classic example of the Toggle/Cycle smart graphic.

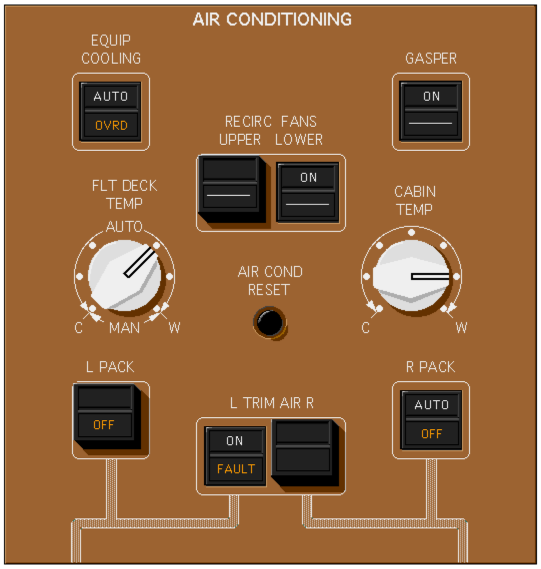

Figure 2. Toggle/Cycle Smart Graphic of 777 Air Conditioning Panel

A mouse click on a rotary switch cycles the switch clockwise one position. The last position then wraps around to the first position. The various push buttons cycle or toggle through their states, and like the switches eventually "loop" back to their original state. Although this looks like a relatively simple graphic, it is capable of generating 2.7 *million* unique states.

OBSERVATIONS ON FIRST GENERATION

Techniques and Issues

Through continuous process improvement, later versions of first generation smart graphics became even more useful than their predecessors. Based on both user feedback and experiences, two types of improvements were made: end-user enhancements and developer enhancements. The first type of improvement simply makes the smart graphic easier to operate. Some of the end-user enhancements included: alignment aids, crop marks, on-screen instructions and help.

Alignment Aids—Including a portion of the background or a nearby object for alignment helped make graphics more consistent. In addition, use of the "inverse" mode made it easier to properly position overlays by watching for a "white-out" when all pieces are aligned. This was helpful for both end-users working with "drag & drop" smart graphics and for developers creating interactive "Toggle/Cycle" smart graphics. In the case of the Toggle/Cycle smart graphics, the alignment tricks help the programmer line-up the various states more quickly.

Crop Marks—Many CBT projects need to work within hardware performance constraints. Providing crop marks for the end user helps them take consistent "snapshots" of configured graphics. It also saves a few precious pixels of screen real estate on individual CBT screens. A few saved pixels multiplied by hundreds of captures and hundreds of lessons can also save quite a few kilobytes. These savings can mean more lessons per CD-ROM, reduced file server requirements, and improved lesson performance. For an example, see Figure 3 – *777 Electrical Synoptic with Crop Marks and Instructions*.

Instructions and Help—Though an SME may be intimately familiar with a particular panel, CBT maintenance might be done by a developer who knows the CBT tools, but is not an SME. As shown in Figure 3, on-screen instructions can include a description of how to cycle an object, tips, and standard configurations used for a project. Furthermore, a dynamic help screen can include tools that reveal the otherwise hidden "hot spots" that cycle unseen objects (See Figure 4 – 777 Electrical Synoptics with Click/Touch Areas Revealed). This helps users generate graphics quickly, without hunting for hot-spots, using trial-and-error to find out what states are available, or unwittingly missing an optional hidden switch.

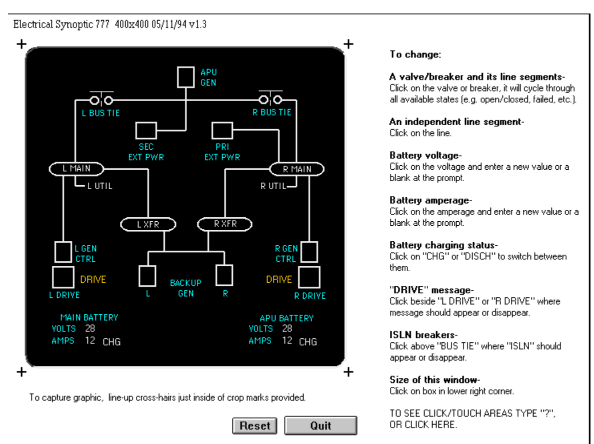

Figure 3. 777 Electrical Synoptic with Crop Marks and Instructions

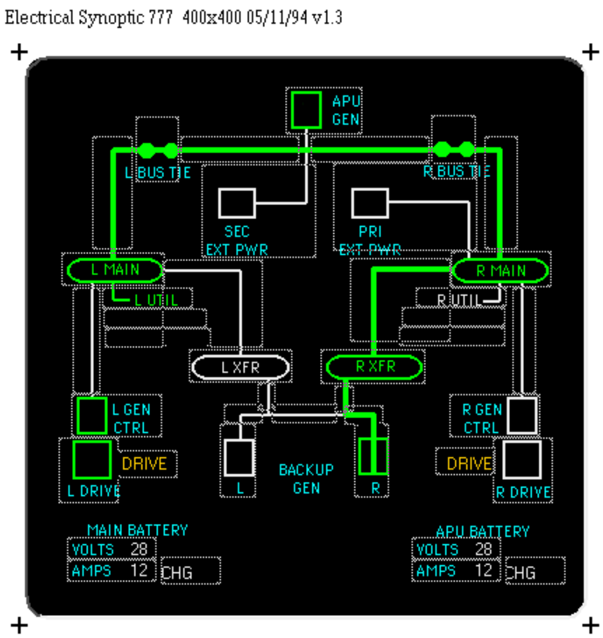

Figure 4. 777 Electrical Synoptic with Click/Touch Areas Revealed

The addition of crop marks, instructions, help functions, and improved operation are job aids that make the smart graphics more suitable to their primary task. During development, we also identified some techniques to improve programming processes and practices. As mentioned earlier, something as simple as tight cropping can make a significant difference. In one case, the drop shadows from the “up” state of a toggle went right up to the edge of the next toggle. This created a dependency where the second toggle looked different depending on the position (up/down) of the first. The solution was to shorten the shadows, so there was no interference between independent objects. Thus in some cases, efficiency and productivity will need to override graphic realism and shadows may need to be less pronounced while up/down states of buttons may need to be exaggerated for clarity within the training materials.

Sizing—Many smart graphics include text such as switch labels etc. Some also include text that is more dynamic, such as radio frequencies and so on. In such cases, be sure the artist develops the display “slots” for this text with a specific, standard typeface and point size in mind. Manually adjusting letter or line spacing is tedious and self-defeating. It will also facilitate the transition into second generation smart graphics, by allowing these fields to become variables.

First generation smart graphics are quick and easy to develop. They increase productivity, but they do not contain any logic to truly simulate device logic. As a result, one failing of first generation smart graphics is the fact that it may be possible for them to display an impossible situation, e.g. a switch is in the off position, yet the light is on. In an effort to keep these smart graphics simple, we do not insert logic that prevents an end user from accidentally creating a graphic representing an unrealistic state. Though this may initially seem imprudent, too often we have spent weeks developing the logic for a panel or display that is only used as a graphic generator. The time spent developing that logic can never be recouped if the smart graphic is only used as a graphics generator. The prudent path is to build these graphics in a scalable fashion, so what starts as a Toggle/Cycle graphic can grow smoothly into a “mini-simulation.” If the frequency of use by SMEs, developers, and instructors justify the development, then the addition of validating logic and more faithful simulation can be cost effective.

SECOND GENERATION SMART GRAPHICS

Characteristics

Second generation smart graphics incorporate logic that allows them to closely approximate simulations of actual devices. This logic has two benefits. As a job aid, it prevents the subject matter expert from creating graphics depicting impractical or impossible situations. This capability can also make the smart graphic available as a teaching aid. Because of their data validation characteristics, their scalable fidelity, and scalable use, these smart graphics can be more than simple graphics generators. Since they perform actual state checking, these smart graphics can be converted to teaching aids and used interactively by students or instructors.

Data validation

Incorporating logic into a smart graphic can assure that impossible or unrealistic scenarios are not depicted. This can be as simple as making sure that a negative fuel load cannot be displayed or as complex as putting the correct symbols on the primary flight display depending on status. Intermediate examples include changing airspeed to mach or turning off the radio altimeter indication above specified altitudes.

Types

Stand-alone

Stand-alone second generation smart graphics contain logic that makes them more accurate and makes them easier to use for their primary purpose. Stand-alone smart graphics are used solely as graphic generators because they are not integrated with other systems, and therefore their use as teaching aids is impractical.

The 777 Fuel Synoptic, shown in Figure 5, is one example of a second generation, stand-alone smart graphic. Although it appears very similar to the first generation graphics, it has several second generation characteristics. In addition to the improvements in later first generation graphics (help, instructions, etc.), this graphic incorporates system logic. For example, the “TOTAL FUEL” reading is based on the contents of the individual tanks. As on the airplane, the only way to really alter this value is to add or deplete fuel from one or more of the tanks.

Another bit of logic was added to accommodate metric fuel weights and temperatures. In this case, the “smarts” is the automatic conversion of temperatures and weights between units of measure. Degrees Fahrenheit are automatically converted to degrees Celsius and fuel readings in pounds are likewise converted to kilograms. Beyond the convenience of unit conversion, the graphic also knows the maximum temperature readings and maximum fuel loads, thus there is no way to over-load a given tank. Figure 5 also shows the unit conversion capabilities and the expanded “jettison” mode of the synoptic. The smart graphic is toggled back to the normal mode via the radio button in the upper left.

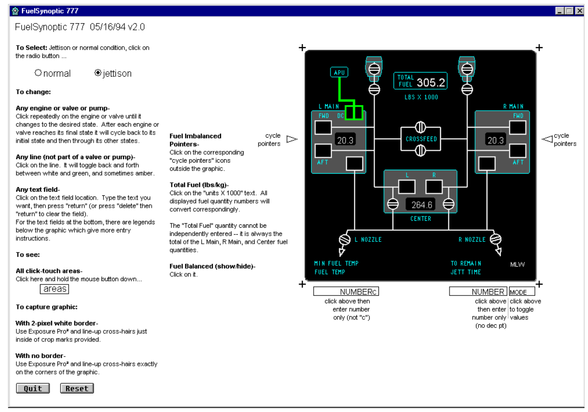

Figure 5. 777 Fuel Synoptic Second Generation Smart Graphic

Second Generation In-line

These smart graphics contain logic and hooks to other displays. They look and act like their predecessors, allowing easy configuration of a panel or display, but are also able to simulate one system or display and show the impact on another display. In some cases, two panels/displays can work in tandem.

Figure 6 shows a lesson on the F-22 hydraulic system. The schematic was created using a portion of the hydraulic system smart graphic. The captured image of the schematic was placed on the screen along with titles, topics, and highlight arrows. The student is introduced to the hydraulic system step by step, learning about all the components and how they interact.

At the point these initial captured images were created, the smart graphic lacked the logic of a full simulation. The hydraulic graphic was laid out with variables/flags attached to different components allowing simple error checking. When it was decided to embed a simulation of the hydraulic isolation system in the CBT, the required logic was added using the preplanned variables. This technique is elaborated on later in scalable use and fidelity.

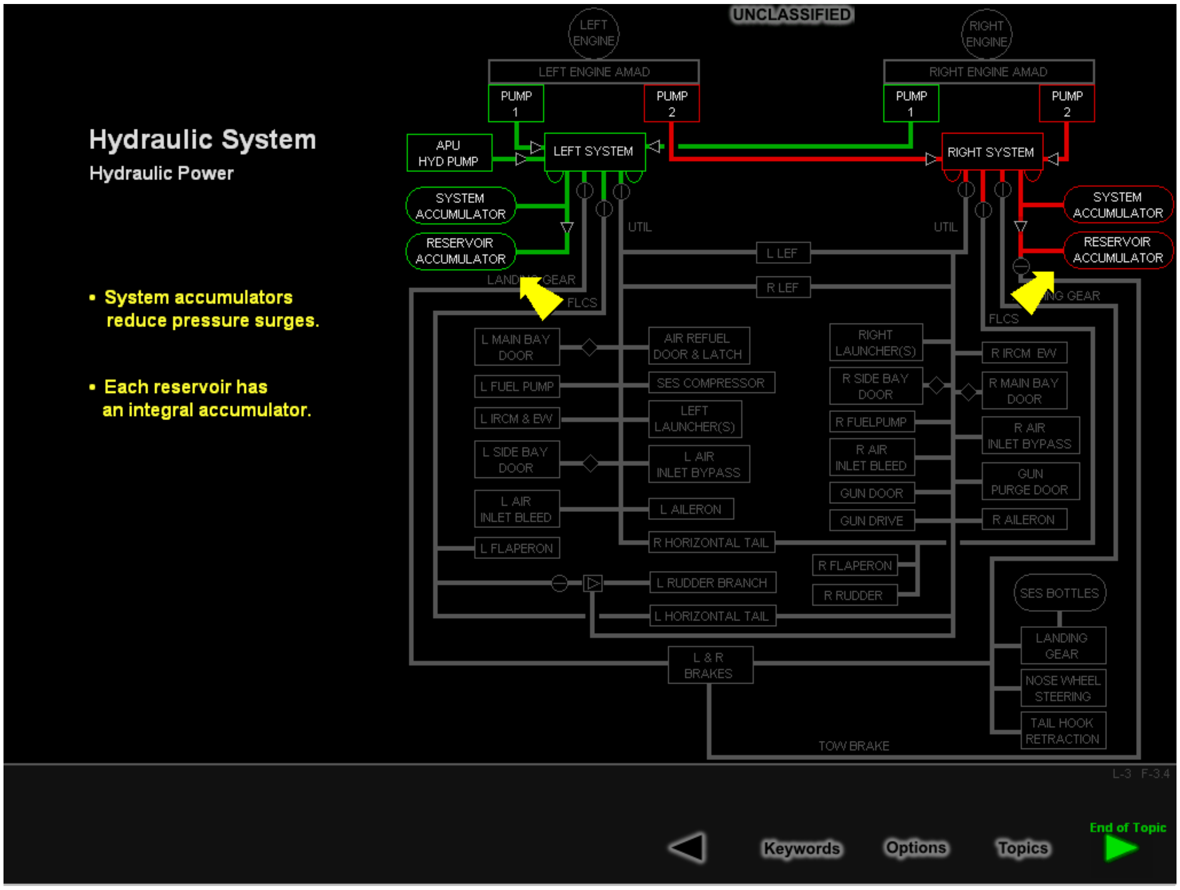

Figure 6. F-22 Hydraulic Schematic Smart Graphic, Electronic Work Book

After learning about the system, the students are allowed to experiment on their own. The same smart graphic used for the schematic is also embedded in the electronic workbook as a teaching aid (Figure 7).

When the schematic runs in the full hydraulic isolation mode, the student can click on the different isolation valves and observe the effects. As each valve closes the red or green lines fade to gray to show loss of pressure. At the same time the affected component grays out, the appropriate warning message appears as it would on the cockpit's up front display.

This is accomplished by changing the values of different variables. These variables or flags are then checked and a new graphic displayed based on the new status of the hydraulic system. Components are not turned on or off by simple clicking. The component status and warning messages are determined by which isolation valves have been turned off. (In gray scale figures the loss of pressure is difficult to see, however on a color display it is easy to tell which components are no longer active and read the warnings.)

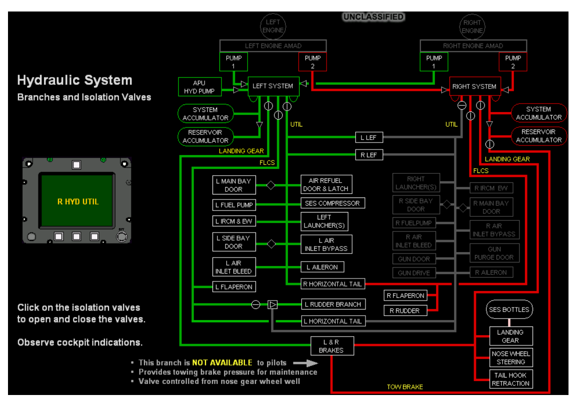

Figure 7. F-22 Hydraulic Isolation Schematic, In-line Smart Graphic as Teaching Aid

OBSERVATIONS ON SECOND GENERATION

Techniques

Source Data—Take advantage of access to modeling programs and engineering data to generate source images for an artist to "tune" into source art for smart graphics. Even something as simple as starting with a CRT display showing all the available symbols or a virtual snapshot of a panel can make the artists job easier and faster. The artists can quickly isolate, scale or otherwise adapt these elements, as opposed to drawing them from scratch.

KISS—(Keep It Simple SME) Don't overdo the smarts in smart graphics. Build the interactive graphics as the simpler 1st generation graphics, but plan on expansion. If SMEs or instructors want more data validation or a true classroom presentation, you can enhance the smart graphic at that time. Until the need arises though, you can keep it simple.

Anticipate Reuse—If you build logic to step through the frames of an animation (movie) to show a dial sweep, anticipate that other smart graphics will have similar needs. Try to write the code so that it will work with not only a given RPM range, but also an altitude or other display that may use a needle sweep.

Seek and Apply User Feedback—Smart graphics quickly become popular among SME developers. They can be a great source of ideas for both usability and simplifying your SG approach.

Anticipate Real-World Usage—Try to build in simple usability tools. For example, rather than only cycling the flaps indicator higher and "wrapping" to the base value, also allow a shift-click or control-click to cycle lower. Also, consider adding buttons off the edge of the panel or display that quickly configure the whole panel/display to common pre-sets, like take-off, cruise, or approach.

COST-BENEFIT ANALYSIS

The issue when creating these second-generation smart graphics is how much logic should be incorporated. As the system being represented becomes more complex the labor involved in creating it increases significantly. While the benefits derived from this internal logic can be rather significant, the costs can be just as prohibitive. Integrated logic requires configuration control. If airplane configuration changes, those changes now need to be incorporated in the smart graphic. Multiple configurations require the ability to mimic those multiple configurations within the smart graphic. When evaluating the cost-benefit analysis of smart graphics, there are a number of issues to consider. We have noticed several key considerations to help assess the costs and anticipated return on smart graphics.

Key Considerations

How often will the smart graphic be used? Spending hours building something that will only be used once or twice has little or no payback.

Can the smart graphic be converted to a simulation or other type of training aid? How often will it be used as a training aid? Time spent up front now can be recovered when creating the training aid.

Is complex logic required to effectively simulate the device? Easily simulated objects have almost immediate payoff. Large complex projects require a cost benefit analysis up front.

Is the system so complex that only the "experts" will be able to configure it correctly? Time invested defining acceptable behavior for the smart graphic now can save research and rework time for less knowledgeable end-users later.

Response to Cost-Benefit Considerations

We use two approaches to address these considerations: scalable fidelity and scalable use. In both cases, the approach is to develop at the low end and anticipate a migration path to the high end. Though the design and development approach needs to consider this path, it does not necessarily need to pursue it. The considerations identified above allow team managers to evaluate graphics on a rolling or case-by-case basis to determine if and when to pursue high end characteristics. The scalable fidelity and scalable use approach provides decision-making flexibility to management.

Scalable fidelity is a design and development approach whereby the accuracy and completeness of the smart graphic can easily be adapted or extended to more accurately simulate device appearance and behavior. Rather than immediately building a complex simulation, we start with the simplest acceptable second generation smart graphic. At the low end, the emphasis is on device appearance with device behavior and logic becoming increasingly important as things progress to the high end. Scalable fidelity allows a smart graphic to be developed and "frozen" at the low end if that is appropriate and cost effective. Likewise, extensive use of a graphic may warrant a migration to a more "faithful" simulation that validates the states or mimics transitional and logical behavior.

Scalable use addresses the intended role of the tool. If a smart graphic is not intended as a classroom lecture aid, then the interface can be less formal, and the simulation less realistic. Again, the key is a well-planned development approach that allows migration from a simple graphic generator on to a high-fidelity simulation suitable for classroom use or integration into student paced CBT. Techniques that allow this migration include programming logic that can hide or reveal extended control panels on the smart graphic. With the extended controls hidden, the smart graphic serves as a classroom teaching tool. Likewise, with the extended panel and job aids revealed, the smart graphic can be a more effective image generator.

The combination of these approaches with critical path scheduling provides an additional tool to help expand the benefits of smart graphics. By looking at critical paths, one can balance resources, use, and fidelity against schedules. For instance, in some projects, CBT development begins long before delivery of stand-up instruction. In such cases, scalable fidelity and scalable use allows the development team to initially concentrate on graphic generators that emphasize imagery over logic and transitions. Then, as course materials are prepared for stand-up instruction, the developers can incrementally improve the logic and fidelity of selected smart graphics. These enhanced smart graphics can now be used for classroom discussion and as teaching aids.

FUTURE ENHANCEMENTS

As with any project developers often foresee many potential enhancements that are desirable, but not immediately available due to technology, time, and resources. However, both time and technology march onward and many of these enhancements are likely to come to fruition. In fact, between the initial outline draft and final preparation of this paper several of the "future" enhancements we identified have come to fruition. Proposed enhancements include:

- Dual Mode Smart Graphics
- Auto-Capture
- Aggregation
- Integration with Devices
- Multi-screen/Multi-Machine Smart Graphics
- Re-use or Re-hosting of Production Code

We would like to share more details of these envisioned enhancements in the hope that they may inspire the reader to pursue them or generate their own unique enhancements.

Dual-Mode Smart Graphics

Dual Mode smart graphics could work both as graphic generators with instructions and control panels. By setting a flag, using a special keystroke, or pressing a button the "control panel" could be hidden or revealed. With the control panel hidden, the smart graphic would work as a presentation and simulation tool. This virtual device could be used by an instructor to foster classroom discussion, prepare for class, or even respond to student "what if" questions. Figures 8 and 9 below show a prototype "Dual Mode" smart graphic, in both a native development environment (Rapid) and as an embedded simulation tool linked to CBT done in another authoring tool (Authorware).

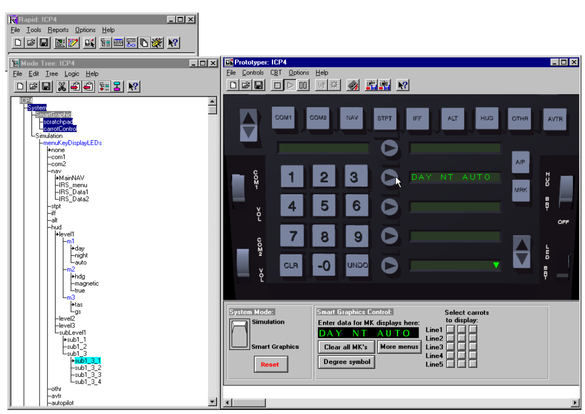

Figure 8. F-22 Integrated Control Panel (ICP) Dual Mode Smart Graphic in Smart Graphic Mode

Though it is not readily apparent in Figure 9, the displayed ICP is not a static image capture. Rather, the smart graphic is running in a separate, borderless window floating on top of the CBT window. Interprocess communication between the Authorware CBT and Rapid smart graphic allows the two to interact. If the user presses a button on the ICP, it generates a message that is sent to the CBT. At that point the CBT can judge the response, acknowledge it and allow the ICP smart graphic to process it normally. However, the CBT can also selectively filter events to intercept them with diagnostic feedback and prevent students from going too far down a wrong path. This leverages the simulation effort of the smart graphic, while keeping training more efficient and focused than pure free-play “discovery learning.”

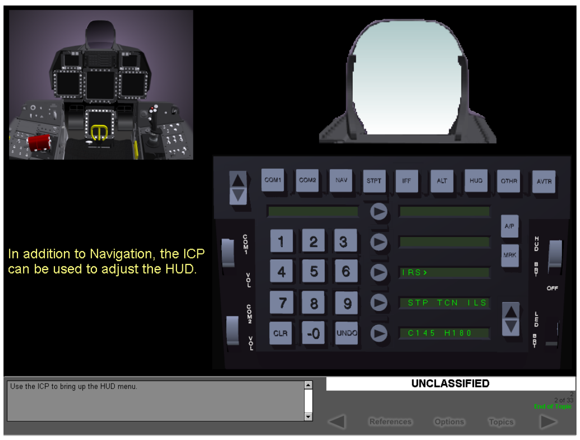

Figure 9. F-22 ICP Dual Mode Smart Graphic in Simulation Mode

Auto-Capture

Though the cross hairs and job aids incorporated in second generation smart graphic improve usability, a utility to automatically capture imagery would be even better. With “Auto-Capture” the end user could push a button to automatically send the current smart graphic configuration to either the clipboard, or a bitmap file. Additionally, common areas of the graphic could be pre-programmed to allow easy creation of overlays or enlarged call-outs to emphasize a teaching point.

Aggregation

Often several panels interact with each other or displays. Well-designed smart graphics with simulation capabilities could be combined into larger simulations by aggregating the individual components into a larger, “smarter” smart graphic.

Integration with External Devices

Potentially, smart graphics could be integrated with input devices such as keypads, joysticks, throttles and part-task training devices. The input devices could drive the displays and configuration of the smart graphics, while still allowing stand-alone use as graphic generators. Likewise, a smart graphic panel properly configured with a hardware interface could act as an input device and potentially drive a production display.

Multi-Screen/Multi-Machine Smart Graphics

Windows NT 4 and Windows 98 allow multiple monitor configurations. Many smart graphics need to function at less than full scale to fit on a standard CBT screen and still allow room for teaching points and CBT navigation controls. With multiple screens, smart graphics could be displayed at or above real size to increase realism or facilitate instruction of larger groups. Additionally, through networking protocols, two or more smart graphics could communicate and update each other.

Engineering Data

Many engineering and simulation tools are migrating from scientific workstations to standard desktop machines using Intel processors and Windows. As these tools migrate to the machines that traditionally host self-paced and instructor-led training content, it may be possible to re-host, port or reuse production code as desktop graphic generators or classroom training simulations.

SUMMARY

Smart graphics are a means of efficiently generating both images and limited simulations. Image generating capabilities reduce overhead, improve consistency, and reduce costs for preparing training and documentation. With planning, the simulation capabilities of smart graphic can be developed with only marginal incremental effort. Instructors and students can use smart graphic-based simulations as either stand-alone teaching aids or embedded in CBT.

The advent of new tools and the power of today’s personal computers provide the capability of simulating complex systems at the desktop. As more training migrates to self-paced and instructor-led CBT, the smart graphic can play a crucial role in making training production simpler and more economical. Developers interested in pursuing smart graphics should invest their resources based on the intended use of the tool and develop only the required interface and fidelity to support that usage. Development practices should anticipate expansion of both fidelity and the broader application of the smart graphic as a training tool.