Web Browser-based Architecture for Reusable Simulation Objects (RSOs) in PC Simulations

Joseph Swinski
DiSTI, Inc.
Orlando, FL
jswinski@simulation.com

Marcia Williams
DiSTI, Inc.
Orlando, FL
mwilliams@simulation.com

ABSTRACT

Highly dynamic systems such as aircraft, power plants and industrial control systems often require an advanced level of training that can be difficult to accomplish effectively with traditional Computer Based Training (CBT). PC Simulations, when applied to a training domain, can serve to provide the user with the opportunity for learning in a robust, motivating, and engaging environment, wherein the presentation of the material is optimized by a high degree of user interactivity, fidelity and immersion, and where context and practice are key to learning.

Given state-of-the-market toolsets, it is believed that web browser-based PC Simulations can provide an adequate level of fidelity, interactivity, and immersion necessary to provide robust, engaging and motivating training content. The challenge for CBT developers is to create highly interactive content in a cost effective manner, moving towards browser standards-based training products. Browser-based technologies are ubiquitous, platform independent, and are becoming increasing more robust in their delivery of media rich content. Utilizing Commercial Off-The-Shelf (COTS) development tools, it is now possible to rapidly create high fidelity 2D and 3D interactive simulations with mobile code technologies that can be easily deployed in a variety of computer based training applications and distance-learning environments.

Applying mobile code technologies for PC Simulation delivery via Web browsers additionally affords an opportunity to make use of Reusable Simulation Objects (RSO) by applying the Source Content Object Resource Model (SCORM). Starting with conceptual designs, these RSO can be reused through the various types and levels of CBT. The specific concept to be addressed in this paper is the potential for life cycle reuse of PC simulation content through RSO application. By adopting this architecture, producers and developers will not only save significant cost by not repeatedly developing the same models, it will also have the potential to improve fidelity of PC simulations.

ABOUT THE AUTHORS

Joseph Swinski currently serves as the President of DiSTI. DiSTI, founded in 1994 is a recognized leader in the field of PC Simulations for Computer Based Training. Mr. Swinski graduated from the State University of New York with a degree in Electrical Engineering. He has held positions at General Electric working on Image Generators and the Institute for Simulation and Training developing the original DIS protocol. Mr. Swinski is also a Pioneer of the Modeling and Simulation Professional Certification Commission.

Marcia Williams is a Software Engineer for Distributed Simulation Technology, Inc in Orlando, FL. She has worked on numerous projects including the Intelligence and Electronic Warfare Tactical Proficiency Trainer (IEWTPT) and the Deployable Intelligence Simulation for Collaborative Operations (DISCO). Marcia is currently a member of the DiSTI modular simulation development team working on high fidelity simulations of avionics systems. The most recent of these systems models the AH-64 MPD: an interactive display through which the aircraft's systems are controlled. She received her B.S. in Computer Science from the University of Central Florida and is currently pursuing an M.S. in Computer Science.

Web Browser-based Architecture for Reusable Simulation Objects (RSOs) in PC Simulations

Joseph Swinski
DiSTI, Inc.
Orlando, FL
jswinski@simulation.com

Marcia Williams
DiSTI, Inc.
Orlando, FL
mwilliams@simulation.com

INTRODUCTION

Until recently, military training simulations primarily took the form of specialized hardware and software. Examples include part-task trainers and full electrical/mechanical simulators. Technological advancements in software, personal computer (PC) hardware and new design toolsets have made it possible to migrate towards less expensive personal computers for certain types of simulations. No longer do we require significant investments in computing power with high-end graphic sub-systems. PC Platforms now afford us the same or better levels of quality.

This new environment, known as PC Simulation or PC SIM, is defined as: "A desktop or laptop computer software program that strives to mimic a phenomenon, experience, equipment or environment that is based on reality. The PC Simulation, when applied to a training domain, serves to provide the user with the opportunity for learning in a robust, motivating, and engaging environment, wherein the presentation of the material is optimized by a high degree of user interactivity, fidelity and immersion, and where context and practice are key to learning" (Scanlon, 2003). Recent investigations demonstrate that personal digital assistants (PDAs) and Tablet PCs also lend themselves to running certain types of PC Simulations.

WHY PC SIM?

There are many benefits to implementing PC Simulation over other training technologies. Although not appropriate for all training situations, it is believed that a desire to provide robust, engaging and motivating training content can be met with the use of PC Simulations.

PC Simulation:
- Can significantly reduce training time and can mimic real time or non-real time events;
- Can better mimic the actual work environment, thus producing better retention of learned skills;
- Can provide a safe environment for exercising 'what-if' scenarios or learning from one's mistakes;
- Can allow learning to take place without the need for using expensive operational equipment, thus reducing life-cycle costs;
- Can allow the practice of hazardous procedures – avoiding the loss of life or tactical assets.

Users of PC Simulations are given the opportunity to repeatedly practice skill sets, resulting in better skill retention where context and practice are key to learning. If equipment operation or maintenance training is the goal, PC Simulation can, in most cases, replace actual equipment used for training. Training equipment can be expensive to procure. Additionally, replacing actual training equipment with PC SIM reduces life-cycle costs since expensive and sometimes impossible maintenance does not have to be performed on the training equipment. Also, a lack of sufficient quantities of training equipment assets could sometimes limit throughput (availability to students).

If situational training is desired, the use of PC SIMs allows the learner to be taught in a controlled and safe classroom environment. The learner may practice 'what-if' scenarios and can safely learn from mistakes that may have been harmful in real-world situations (Scanlon, 2003).

Timely student feedback is yet another benefit of PC Simulation. Feedback can significantly add to a student's experience, which should be at least as rich as the corresponding experience would have been with traditional education (Thomas, 2001). Today's technologies provide a means of presenting robust, engaging and motivating training through PC Simulations. Feedback can be given immediately with PC Simulations, since each student can work individually. The student using a PC SIM will not have to wait for another student to finish with a piece

of training equipment or for the instructor to be present to observe the training event. Feedback is crucial for recognizing safety violations and / or bad practices when brought immediately to the student's attention. The student can be given another chance to attempt the training event.

An instructor's assessment of student progress also benefits from using PC Simulations for training. In a traditional learning environment, student assessment is limited to the usual matching, multiple choice and ordering types of questions. Adding a simulation to the testing allows for new types of questions to be asked. For example, a student can be graded on repairing a malfunctioning system or on placing the simulation in some state that represents a specified behavior (Thomas, 2001).

A final benefit of PC Simulation over other training technologies is the portability of the simulation environment. Portability reduces the operational costs of training by allowing the training to come to the students. The current paradigm requires the students to come to the training center. The savings in travel costs alone of sending hundreds or thousands of students to the training center is a benefit.

PC Simulation Example – Generic Skill Trainer

The PC Simulation of a Basic Electronics Computer Aided Training System (Device 6F24) is, shown in Figure 1. The PC Simulation was based on the existing hardware trainer from Nida Corporation. It can be used as an introductory or refresher course in basic electronics, and provides an interactive method for learning about electronic circuits using a high fidelity photo-realistic simulation.

Figure 1. Electronic Circuit Trainer PC Simulation created using DiSTI's GL Studio

When the actual equipment represented in this simulation is set up in a classroom, a learner has only a limited amount of time to spend on the equipment. The learner would not receive interactive feedback from the actual equipment either. In addition, equipment in a learning lab is often abused and in need of costly repairs. All of these issues are avoided by using the PC Simulation in place of the actual equipment. In fact, NAVAIR analysis has shown that the PC Simulation alternative would reduce the time to by at least 30% and the Return-on-Investment (ROI) would be significant (*Trainer Analysis*, 2004). With a PC SIM, a learner can have frequent access to unbreakable equipment and is provided with feedback during training.

Once this PC Simulation is accessible through a Learning Management System (LMS), any number of people may use it from their home or classroom computers. The simulation may also be copied to a CD that learners can take to sites that do not have internet or intranet access or it can be used for "Just In Time Training". This is the type of course that is reusable by many different people including, electronic equipment service technicians, hobbyists, anyone entering a specialized electronics field, beginning electrical engineers, and universities.

QUALITY CONSIDERATIONS FOR PC SIMULATION

For PC SIMs to be most effective in the learning environment, the proper mix of fidelity, interactivity and immersion, is required. Each simulation differs in the levels of fidelity, interactivity and immersion required to meet training objectives. The simulation's developers will need to make decisions based on their requirements in each of these areas and also on cost, time and manpower constraints. Fidelity, interactivity and immersion, as defined below, describe the quality of a simulation's representation of a real-world experience (National Institute of Standards and Technology, 2004).

Fidelity

There are three types of fidelity we are concerned with: audio, visual and the functional behavior of simulated objects. Fidelity is a measurement of how closely a real-world sound, image or behavior compares to its PC Simulation representation. A

higher fidelity PC Simulation generally has a higher development cost.

Audio Fidelity is a function of:
- Sampling rate of the original audio
- Compression / Decompression techniques
- Mono or Stereo presentation to end user

If a high level of audio fidelity is required, stereo presentation may be used and the frequency of sampling should be increased. 8KHz is sufficient but not always ideal, while 44KHz is used for CD quality music and will lead to large audio files. The higher the sampling rate, the less hiss, noise or abrupt changes in sound will be heard during playback.

With a high level of audio fidelity there are tradeoffs that must be taken into consideration. Throughput, bandwidth and storage space must all be considered when determining the level of audio fidelity for a PC Simulation.

Visual Fidelity is a function of:
- Color depth
- Size (storage)
- Image Resolution
- Shading fills and light focus
- Frame rate (if motion is required)
- Level of detail

Again, a higher level of fidelity in images for the simulation will usually result in greater amount of data generated. Throughput and bandwidth limitations can affect the allowable level of visual fidelity. Image and video compression can be used to help meet training objectives for a PC Simulation (National Institute of Standards and Technology, 2004).

The Functional behavior of simulated objects is a function of:
- Response to user input
- Physics model (for example: gravity, friction)
- Reaction to other simulation components (trees, equipment, users, etc.)
- Capabilities requirements (for example, are all of the required knobs functioning as they are supposed to)

Functional fidelity is the degree that a simulated situation or object behaves like its real word counterpart. A relatively large part of the design and development of a PC Simulation is usually spent in achieving a high level of functional fidelity. The functionality of a PC SIM is encapsulated in a behavior model, which is usually labor-intensive to create. The cost of creating this behavior model has not changed much over the years, as opposed to the cost of creating 3D graphics, which has dropped considerably. The three toolkits discussed later in this paper can be used to create the behavior model, however, they are best suited for creating graphics and a high level of visual fidelity.

A PC Simulation's behavior can be input from a third party that specializes in generating behavior for that type of simulation. For example, Simulation Program for Integrated Circuits Emphasis (SPICE), a powerful analog circuit simulator, can be used to provide the behavior for a PC Simulation of a circuit board, for example. SPICE accurately represents the real-world behavior of an analog circuit simulator and can be used to verify and / or predict the circuit behavior (University of Pennsylvania, 1995).

Immersion

Immersion is the act of being absorbed or engrossed, giving complete attention to the task or training scenario presented. A PC Simulation engages a learner by providing an atmosphere of "learning by doing". Usually, the more engaging the simulation is, the higher the levels of immersion, fidelity and interactivity.

PC Simulation immersion technologies include "traditional" computer applications and limited multi-sensory output devices. In the former system, the monitor is used to display images and no other sensorial output, besides sound, is used. In the multi-sensory system, specialized equipment, such as tactile or force feedback devices, are used to meet specific requirements of the training objectives.

PC Simulation immersion technologies (for purposes of this paper) do not include stereoscopic images used with head tracking equipment nor do they provide the illusion of being inside the simulated environment (National Institute of Standards and Technology, 2004). The above exclusions help to differentiate PC SIM from other types of simulation technologies: (e.g. MicroSIM devices, part-task trainers and full-up simulation systems).

Interactivity

A high level of interactivity in a simulation tends to correspond to the value and quality of the learning (Thomas, 2001). In an interactive system, users do not

simply receive information, but instead engage with objects that are responsive to the user's stimuli. There are several levels of interactivity that a PC Simulation can use to guide the communication between computer and learner. These levels, listed below, are not mutually exclusive within a simulation (Sims, 1997).

- Object
- Linear
- Support
- Update
- Construct
- Reflective
- Simulation
- Hyperlinked
- Non-Immersive Contextual
- Immersive Virtual

Figure 2. TACLANE KG-175 PC Simulation created using DiSTI's GL Studio

COST OF DEVELOPMENT

If PC Simulation techniques can produce an even more robust and engaging learning experience, why don't we make more frequent use of these technologies? Make no mistake; PC simulations can be very expensive to produce. Focus groups chartered by Brandon-Hall have estimated that typical development time for a generic web-based e-learning course is around 220 hours of development for every hour of instruction. The development time required for PC Simulation is estimated to be at a ratio of 750-1300:1 (Brandon-Hall, 2003).

Reported Brandon-Hall development times not withstanding, experience shows that high audio/visual fidelity does not have to be associated with high development costs. The cost of PC Simulations is rapidly decreasing with the development of new graphics toolsets. Using state-of-the-market toolsets (e.g. DiSTI, Inc - GL Studio™, Macromedia, Inc. - Flash®, NGRAIN® Corporation – NGRAIN® software tools), high levels of audio/visual fidelity can now be very cost effective. It is the level of fidelity for functional behavior of simulated objects that still has potentially significant development cost ramifications. Figure 2 shows an example wherein fully interactive 3D graphics represents only 20% of the development cycle. The majority of the development time and costs are associated with the functional behavior of the simulation. In the past, greater then 50% of the development time would have been spent on 2D or 3D graphics for the PC Simulations.

A PROPOSED PC SIMULATION ARCHITECTURE

Although not appropriate for every type of PC Simulation product, it is believed that an adequate level of fidelity, interactivity, and immersion, and the ensuing desire to provide robust, engaging and motivating training content can be met through the use of Reusable Simulation Objects (RSOs) interacting with the Learning Management System (LMS) (see Figure 3). The PC Simulations will be delivered to the students on browser-based technologies. We propose that PC Simulation vendors migrate towards this standardized architecture and the use of RSOs wherever possible.

The move towards this standardized PC Simulation architecture has significant benefits:

- Browser technology is ubiquitous, addressing some cross-platform issues
- Supports multiple media deployment options
- Supported by the SCORM
- Promotes content re-use
- More easily integrated into Learning Content Management Systems (repositories)
- More easily integrated with multiple LMSs

A Learning Management System (LMS) allows learners to access courses. Learners can register for courses, take assessment tests and perform other tasks related to completing courses. Instructors can access learners' records and assessment information (Carnegie Mellon, 2003).

RSOs can be created using the toolsets, discussed later, and then stored in the Learning Content Management System (LCMS), which is a repository for RSOs and other learning assets. RSOs can be extracted from the

LCMS by developers and used to create PC Simulations. Job sheets for current hardware trainers can be used when creating a new PC SIM for existing hardware based trainers.

Mobile code technology is then used to deliver our browser-based solution to students. Students at sites with no internet or intranet capabilities can receive the PC Simulation via a CD Rom. Web based students and classroom based students will interact with a LMS for the delivery of the PC Simulation. Student results can then be recorded and tracked by the LMS.
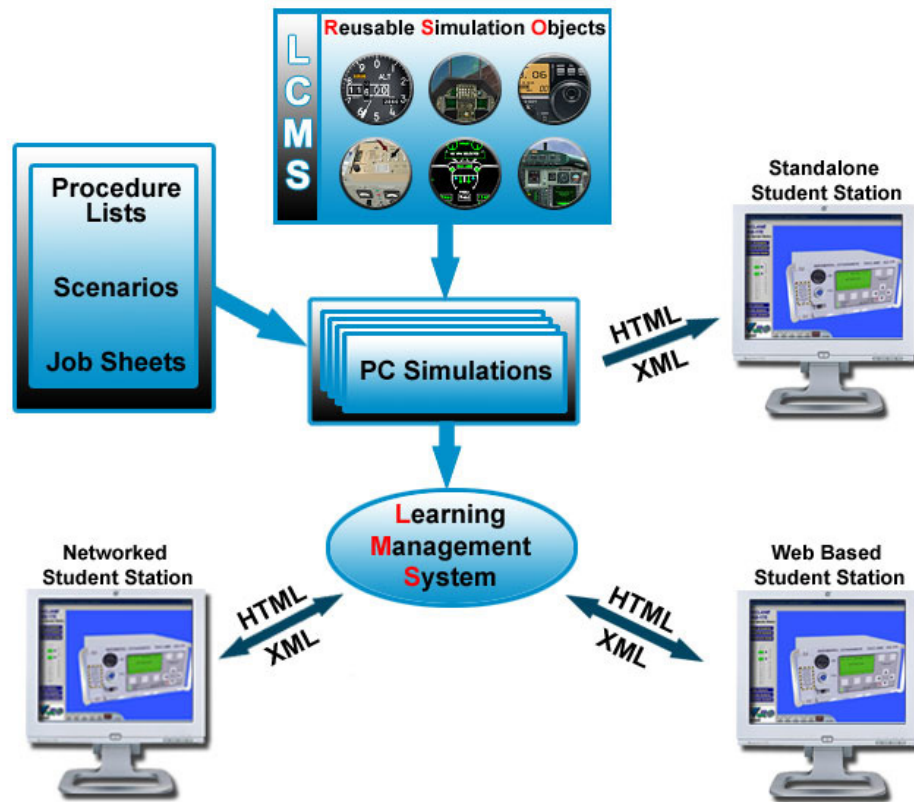
Figure 3. Proposed PC Simulation Architecture

RSO

A Reusable Simulation Object (RSO) is a modular, encapsulated simulation that is packaged for incorporation into a larger application framework. RSOs can be quickly and easily deployed in computer based training applications, courseware development tools and distance-learning environments. A RSO contains compiled or interpreted code, which encapsulates component simulations in a form that promotes reuse. RSOs conceptually can contain strictly simulation logic (without interfaces to displays or input hardware), however, most of the RSOs that DiSTI implements are high fidelity 2D and 3D interactive simulations. RSOs can contain other RSOs. This helps to make each RSO modular and reusable. In Figure 4, a RSO representing an altimeter

has been created once and reused in multiple cockpits, which can also be RSOs themselves.

RSOs can handle user inputs, such as mouse or keyboard events, and / or pass them on to children RSOs for handling. RSOs can also emit events which are passed up the chain to parent applications to be processed.

The internal data of a RSO is exposed in a generic and dynamically discoverable way. This gives a RSO the ability to provide data to other RSOs and to parent applications. Given the current simulation time, an RSO can calculate its internal state and coordinate actions with parent and peer RSOs. In addition, a

RSO can render a graphical representation of its current state (Andrews, 2004).

Figure 4. Altimeter RSO included in multiple cockpit RSOs

One possible use for RSOs is the dynamic creation of complex waveforms and other mathematical functions. If the display shown in the oscilloscope (see Figure 5) was not dynamically created, it would have to be created with a set of images that represent each possible state. This would be very time consuming to recreate and maintain. Instead, the code to draw the display is written once and changes or new functionality is easy to implement. Using RSOs to create dynamic functions vs. creating a set of images results in a reduction in file size, which results in faster download times over the internet.


Figure 5. RSO - Oscilloscope

RSOs could be created using any of the three state-of-the-market toolsets described at the end of the paper. NGRAIN allows users to create SCORM-conformant 3D content that is durable, interoperable, accessible, and reusable. NGRAIN uses the concept of Knowledge Objects, which can also be considered RSOs. Flash can be used to make browser based RSOs since Flash content can be used in almost any browser after the plug-in is installed. Another benefit of Flash is that since it is widely used, training material for developers is plentiful. The use of RSOs in GL Studio is described in the next section.

GL Studio RSOs

Within the DiSTI GL Studio product, the term RSO is used to describe, on a conceptual level, the data that is contained within and generated from a GL Studio design file. This includes the data necessary to fully represent and model an object within a computer simulation. In practice, this consists of geometrical data needed to render objects in 3D using OpenGL and C++ code necessary to handle user interaction events and simulate the object's behavior over time. The graphics and behaviors defined in the GL Studio editor are compiled into a single reusable object, the RSO.

RSOs created with GL Studio support several binary delivery mechanisms: ActiveX™, Dynamically Linked Libraries (DLL), or Dynamic Shared Object (DSO). GL Studio RSOs can also be rapidly interfaced to a number of current text based interface standards, such as XML, and are easily integrated into Java based environments. CORBA, DIS and HLA are all supported by GL Studio RSOs.

The GL Studio RSO concept supports RSO reuse in a highly diverse set of environments with no RSO reengineering. A RSO can be included in other RSOs (see Figure 6), which can then be deployed to PC Simulations, maintenance and part task trainers, CBT, instructional aids, job performance aids and full virtual simulators.
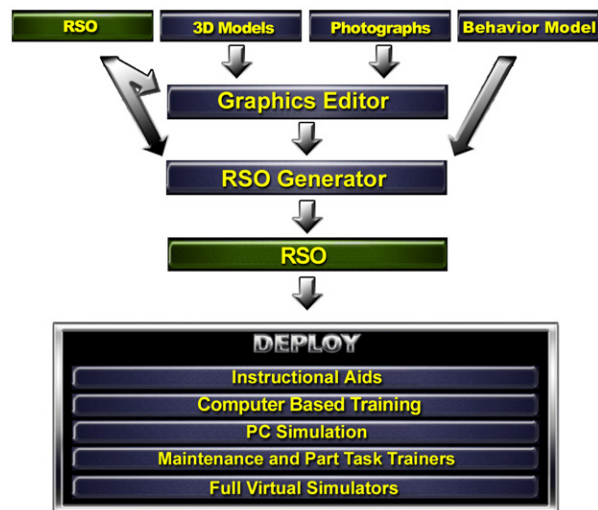
Figure 6. RSO Reuse

Browser-based Technologies for Delivery of RSOs

Browser-based technologies are ubiquitous, supported on multiple platforms, and are becoming increasingly

more robust in their delivery of media rich content. Using browser-based technologies, PC Simulations can be delivered, albeit with varying granularity of fidelity, via many different media: stand-alone personal computers, intranets, the Internet, synchronous or asynchronous electronic classrooms and even Personal Digital Assistants (PDA). The secret to achieving a media-rich environment within browser-based products is to deploy mobile code technologies.

Mobile Code Technologies

The secret to achieving a media-rich environment within browser-based products is to deploy mobile code technologies. Mobile code is software that can be transmitted across the network and executed on the other end, usually without any action from the end user. With mobile code, we can add robust multimedia that motivates learning through simulation, animations, practical embedded applications, etc. The overall reason mobile code technologies should be applied to browser-based training products is to better communicate to students the intended learning objectives through multi-sensory stimulation and interactivity. These techniques are intended to maintain student interest and the motivation to learn outside of the traditional classroom. Mobile Code offers a better means of engaging the learner. Recent experimentations shows that robust PC simulations can be compiled into mobile code technologies and delivered within a browser environment to meet these desired learning outcomes.

Proprietary Runtime

Proprietary Runtimes offer several disadvantages when being used to develop RSOs. Developers using tools that create RSOs with proprietary runtime need to proceed with caution. Proprietary runtimes need to be downloaded and installed on the local PC. This is not always possible like in the case of the NAVY NMCI computers. In order to have the proprietary runtime installed the user needs to gain permission from the network administrator which is not always possible. Macromedia's Flash and other similar tools require a learner to install a plug-in before using their proprietary software. An alternative is DiSTI's toolset GL Studio, which uses ActiveX technology that is already built into the operating system. The use of an open runtime technology allows RSOs to easily be included in webpages and other course material. The learner can simply download the course material and

use it. More generic examples of non-proprietary runtimes would be RSOs created to use the Microsoft .NET or J2EE development environments.

Where Do RSOs Fit Into SCORM?

RSOs typically take the form of SCORM assets, which are "electronic representations of media, text, images, sounds, Web pages, assessment objects, and other pieces of data that can be delivered to a Web client" (Carnegie Mellon, 2003). RSOs can be used as a component of a Sharable Content Object (SCO), which is a "collection of assets that becomes an independent, defined piece of instructional material." Currently, SCOs should be standalone objects since they cannot directly access other SCOs. Inter-SCO communication is done through the LMS (Carnegie Mellon, 2003). RSOs can contain other RSOs and they can communicate directly.

Using RSOs as components of SCOs can greatly reduce the development time to create a SCO and reduces the level of effort in maintaining course content. If the RSO represents real hardware, changes to the hardware can be handled by a simple update to the RSO. The RSO is then redeployed with little to no updates to the SCO. Different RSOs can have different levels of fidelity. PC SIM developers can select RSOs from the LCMS to meet fidelity requirements for their PC SIM.
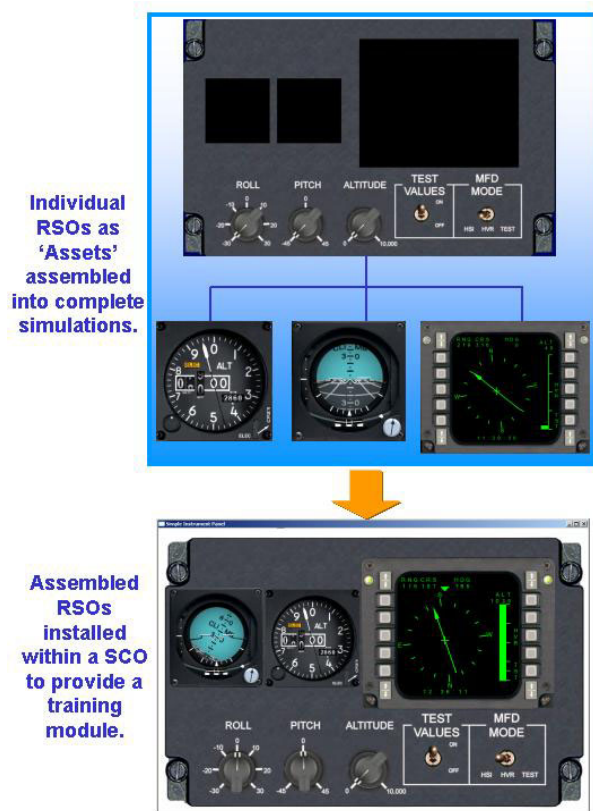
Figure 7. RSO Components Allow for Rapid Assembly of Complex Models

STATE-OF-THE-MARKET-TOOLSETS

The rest of this section holds a discussion on three toolsets that can be used to create PC Simulations. Deciding which of these toolsets is right for a given situation depends on the situation. GL Studio is perfect for PC Simulations that require high fidelity, photo-realistic 3D graphics. If narrative and animation can be used to present your instructional content, and 3D realism is not required, then Macromedia's Flash would be a good choice (Shank, n.d.). NGRAIN would be useful for PC SIMs that require a learner to primarily assemble and disassemble equipment.

DiSTI - GL Studio

GL Studio™ is a rapid virtual prototyping tool that allows you to create photo-realistic interactive objects for real time simulation. Object oriented OpenGL C++ source code is generated from the GL Studio design

suite to create RSOs. Then the RSOs are linked into your simulation in real time.

GL Studio Features include:
- Create your own object behaviors or use behavior libraries supplied for common 2D and 3D objects: buttons, knobs, sliders, thumbwheels, linear and non-linear gauges, odometers, and more
- Easy to use tool tray for the most common tasks used in object creation and manipulation: lines, arcs, spheres, and more
- Realistic mouse and touch screen behaviors for mimicking knobs, switches, and other devices.
- Supports textured geometry allowing the use of digitized photography for creating photorealistic instruments
- Create reusable C++ objects with embedded behavior

Use GL Studio to create ActiveX Components for easily web deployable interactive applications. As an ActiveX control, the GL Studio RSOs can easily be integrated with PowerPoint, VisualBasic, MFC, Macromedia Authorware, CiTect and any other ActiveX capable systems.

Figure 8. T-39 cockpit created with DiSTI's GL Studio

Macromedia, Inc. – Flash

Video, text, audio, and graphics can be integrated using Macromedia Flash into "immersive, rich experiences that deliver superior results for interactive marketing and presentations, e-learning, and application user interfaces. (Macromedia, 2004)" A Flash player exists for almost all platforms with a web browser (even handheld devices). Flash is used on

more than 97% of all Internet-enabled desktops across the world and on many other types of devices. Flash content can be played in a standalone application or in another container, such as a web page. Flash only supports pseudo-3D objects and should not be used for real-time applications.

Figure 9. "Teaching Matters, Inc.: The Scientific Process" created using Macromedia's Flash
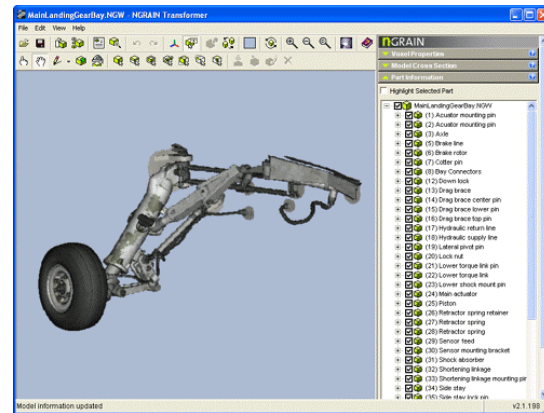
NGRAIN

NGRAIN is focused on creating compressed, interactive 3D knowledge objects from your existing 3D content (e.g. CAD files). Your content is converted to .NGW (NGRAIN World) format. These knowledge objects allow learners to view, assemble and disassemble representations of 3D equipment in real time (Brandon-Hall, 2003). These 3D models can be enhanced with animation, hyperlinking and disassembly logic and can be packaged as SCORM SCOs. Using the NGRAIN™ Mobilizer, learners may move, disassemble, reassemble, and draw on 3D models. Mobilizer can be embedded into HTML content, Microsoft Word documents, PowerPoint presentations and WinForm-based applications (Gibbs, 2004). Mobilizer must be installed on the learner's machine to view NGRAIN Knowledge Objects.

The NGRAIN suite of tools is composed of:
- Transformer – Imports existing 3D data into the NGRAIN format;
- Knowledge Module – adds links, logic and animations to 3D models, creating a Knowledge Object;
- Mobilizer (a .NET control) – enables end user to view previously created Knowledge Objects;
- Constructor – provides a set of libraries, application utilities, documentation and source

code to help developers create robust and interactive 3D applications for deployment on common PCs and laptops.

Figure 10. 3D Landing Gear model converted using NGRAIN® Transformer

SUMMARY

Over the years the cost of creating high fidelity visual simulations has dropped dramatically through the use of toolkits like the ones described above. On the other hand, the cost of creating high fidelity behavior models to drive the visual simulations has remained static. GL Studio is the choice of PC Simulation developers who wish to create true 3D, real-time, photo-realistic PC SIMs and RSOs in a cost-effective manner.

The PC Simulation architecture defined above can be summarized as follows: by using RSOs and mobile code technology to deliver a browser-based solution to students, developers can create an engaging and motivating training environment that can be used in a classroom, through the internet or standalone. PC Simulation vendors and developers should consider moving towards this architecture and the use of RSOs wherever possible.

REFERENCES

National Institute of Standards and Technology, (2004). *PC Modeling and Simulation Guidelines: Volume 1 – Overview.*

Trainer Analysis of the AV "I" Level Course. (2004).

Scanlon, Steve. (2003). *PC Simulation: Proposed Categorization, Architecture and Business Case Analysis (version 0.9).*

Sims, Roderick. (1997). *Interactivity: A Forgotten Art?*, Retrieved June 14, 2004, from http://www.gsu.edu/~wwwitr/docs/interact/.

Thomas, Ruth. (2001). *Interactivity & Simulations in e-Learning.* Retrieved June 14, 2004, from http://www.jelsim.org/resources/whitepaper.pdf

Andrews, William & Baker, Ben. (2004) – *Centered Encapsulating Simulation Behavior with 3D Visual Models: Reusable Simualtion Objects (RSO).*

Carnegie Mellon. (2003). *SCORM: Best Practices Guide for Content Developers (1st Edition)*, Retrieved June 14, 2004, from http://www.lsal.cmu.edu/lsal/expertise/projects/developersguide/

Shank, Patti. (n.d.). *Macromedia Flash MX: Serious Instructional Authoring Tool?*. Retrieved June 21, 2004, from http://www.macromedia.com/resources/elearning/article/flashmx_authoring/

Brandon-Hall. (2003). *Learning Simulations: Tools and Services for Creating Software, Business, and Technical Skills Simulations.*

Gibbs, Mark. (2004). Realistic 3D delivery with NGRAIN. *Network World Web Applications Newsletter,* Retrieved June 21, 2004, from http://www.nwfusion.com/newsletters/web/2004/0126web2.html

University of Pennsylvania – Department of Electrical Engineering, (1995). *SPICE – A Brief Overview*, Retrieved June 23, 2004, from http://www.seas.upenn.edu/~jan/spice/spice.overview.html#INTRODUCTION

Macromedia, (2004). *Macromedia Flash MX 2004,* Retrieved June 23, 2004, from http://www.macromedia.com/software/flash/

NGRAIN, (2004). *NGRAIN Products Overview,* Retrieved June 23, 2004, from http://www.ngrain.com/products/index.html

The products Flash, GL Studio, Transformer, Knowledge Module, Mobilizer, and Constructor are registered trademarks of their respective companies.