Real-time Feedback Modules to Enhance User Learning in Surgical Simulation

Abstract

The objective of a surgical simulation is to

a) Simulate a real surgical procedure as precisely as possible

b) Gauge user performance based on a set of metrics

c) Augment user learning through accurate and intelligent feedback

Advancements in (a) and (b) suggest that surgical simulators today are viewed more as a practicing tool for those already familiar with surgical procedure complexities, than a teaching tool. Progress in (c) has been confined to basic textual and visual feedback, both of which are not utilized as instructional tools but as directional ones. Their remains an untapped potential for surgical simulators to teach novice surgeons and medical students the intricacies of surgical procedures. This 15-month research study focuses on the innovative design and construction of three unique feedback modules: step-by-step monitoring system, audio-assisted commands, and an intelligent video assessment system, with an objective to augment the learning of an individual. Formative research on a working prototype has validated a positive combinatorial effect of feedback configurations on end user performance and learning. Summative research with 40 subjects is expected to be completed by December 2007. The prototype is highly versatile and can be leveraged across a multitude of surgical procedures as well as other industries.

Real-time Feedback Modules to Enhance User Learning in Surgical Simulation

1 Introduction

Feedback in a surgical simulator is a method of relaying complex information in a simpler manner by targeting the five human senses. The end goal of a feedback mechanism is to enhance user learning of a particular task. In addition, it is important to have this feedback in real-time as this not only improves human-computer interaction but also user performance. Currently, the use of feedback in most surgical simulators has been limited to textual (e.g. viewing a list of errors in text), visual feedback (e.g. color-coded objects), and force feedback rendered by haptic devices. Also, most surgical simulators today focus on providing a "practicing" environment, which expects users to be familiar with the procedure being manipulated and to have prior experience with surgical simulators. In this "practicing" environment, therefore, many features of the simulator are primarily used as practicing guides rather than teaching guides. Hence, novice users (e.g. medical students, novice surgeons) utilizing surgical simulators have a hard time acclimatizing to this new virtual reality (VR) environment and learning surgical manipulations in the absence of a "teaching-oriented" feedback system.

1.1 Purpose

The purpose of this research is to develop a "teaching and learning" environment for medical students and novice surgeons, who have little experience in using surgical simulators and performing rudimentary surgical manipulations. This is achieved through the design and construction of three unique real-time and intelligent feedback modules – step-by-step monitoring, audio-assisted commands, and a video assessment system. The hypothesis is that the combinatorial impact of all these three modules will positively impact end user performance and

learning. Key design factors considered during this research also relate to the versatility and cost-effectiveness of the solution. Versatility primarily relates to an ability to deploy these modules across a multitude of commercially available surgical simulators used across different types of surgical procedures – endoscopic and laparoscopic surgeries. Cost-effectiveness primarily relates to the use of open-source software platforms to the maximum extent possible.

1.2 Commercial Surgical Simulators in the Market

Significant background research has been conducted on the use of feedback mechanisms in leading commercial surgical simulators. The simulators researched include LapSim by Surgical Science, LTS3e by RealSim Systems, LapMentor by Simbionix, ProMIS by Haptica, and SurgicalSim by METI. Following is the summary:

In the LapSim simulator [1], textual feedback is employed to give directions to the users and to display the final results, and visual feedback is used to color-code specific targets. The LTS3e [2] is a physical surgical simulator, which utilizes textual feedback, and video feedback comprising of pre-defined video clips to show users how to do a particular subtask within the simulation. However, in this case the video feedback is used as a tool for direction and guidance rather than teaching and learning. Another simulator, the LapMentor [3], has been positioned in a way where it is strictly made for meeting the training and practicing needs of surgeons; thus its focus is on replicating realism and detailed evaluation and metrics rather than teaching-oriented feedback mechanisms. The ProMIS [4], like the LTS3e, utilizes textual and video feedback for the same purposes but in addition to those modules, the ProMIS also utilizes audio feedback. The ProMIS is one of the few physical simulators, which have gone beyond the conventional textual and visual feedback with its audio features, but this transition remains elusive in modern virtual-reality based surgical simulations. Lastly, the SurgicalSim [5], in addition to the conventional

textual and visual feedback, introduces physiological feedback (e.g. blood pressure, heart rate, etc.) to support their training methodologies.

1.3 Primary Research in Feedback Systems in Surgical Simulators

Current research being conducted in the surgical simulation feedback mechanisms revolves around haptic force-feedback, virtual assistants, preliminary forms of audio feedback, and terminal feedback. Haptic force-feedback [6] exploits the sense of touch in surgically simulated environment and minimally invasive surgery (MIS). Currently, there are many commercial haptic devices which can be configured to work with commercial surgical simulators; some include the Bimanual Haptic Workstation by Immersion Medical and the PHANTOM Omni Desktop by Sensable Technologies (the one used in this study). Virtual assistants are a mature technology in other consumer-oriented software applications and are now being leveraged in guiding surgeons as they perform surgical procedures within a surgical simulation [7]. Preliminary uses of audio feedback in the form of signals are being leveraged for a limited scope of activities like tool navigation for which an audio signal is triggered when the user is deviating from the most efficient tool pathway [8]. Terminal feedback, as opposed to real-time feedback, is used to report a final performance score summarizing the metric components.

Based on background research with medical students, haptic force-feedback aims to improve the "practicing" environment for experienced surgeons by enhancing realism. Also, virtual assistants are used as a tool for guidance rather than teaching. Hence, the user might complete the surgical simulator task with minimal problems but could be hesitant in performing an actual surgery. The scope of audio feedback is limited and is also used primarily for guidance purposes (e.g. signals, warnings). Lastly, terminal feedback, in its numerical-based score report, fails to inform the user on how and why they made their mistakes thus overlooking the "teaching" element.

2 Tools and Methods

2.1 Introduction

This research study comprises of the design and construction of three unique real-time and intelligent feedback modules. It is targeted at novice surgeons and medical students who are new to VR surgical simulations and/or surgical procedures and have a desire to learn the intricacies of the same on a VR surgical simulation instead of a cadaver or a manikin. The purpose of these feedback modules is to foster an instructive environment where the goal is to enhance user-learning and information retention, rather than a practicing environment, where the goal is to refine one's already established skills. The three feedback modules designed and implemented in this study consist of:

a) Step-by-step Monitoring System – the system breaks down the end goal of a particular surgical task into several intermediate steps. It then leverages workflow diagrams that aid the user in pursuing the modeled step-by-step approach to perform the surgery. As a user performs the surgical manipulation, each completed task in the monitoring system changes color (red, orange, green) based on whether the user has completed that task, is in the process of completing the task, or has not completed the task respectively.

b) Audio-Assisted Commands – as the user performs manipulations, appropriate voice commands are generated (e.g. computer will say "Stop!" if a user is proceeding in the wrong direction). Audio commands are also generated from the text in the monitoring system and the voice is played based on the user's current stage within the procedure. This is different from conventional audio feedback in this area because instead of signals/ warnings, the audio feedback here mimics the instructions and subtasks of a procedure thus potentially engraining the information in the user's mind for future operations.

c) Video Assessment System – once a user has achieved the simulation goal, the complete procedure with all the user's manipulations is replayed. During the replay, all the user's mistakes are highlighted as the user made them thus giving the user the power not only to understand what mistakes were made but how and why they were made.

2.2 Research Timeline

This study, commenced in Nov 2006, is a 15-month project with a target completion time of Feb 2008. It is composed of five phases, which are discussed in detail in the following sections.

a) *Phase I:* Understand the surgical simulation development environment

b) *Phase II:* Initiate construction of a simulated scenario

c) *Phase III:* Design, develop, and embed feedback modules in the simulated scenario

d) *Phase IV:* Create a testing plan and formative research

e) *Phase V:* Extensive testing, statistical analysis, and summative research

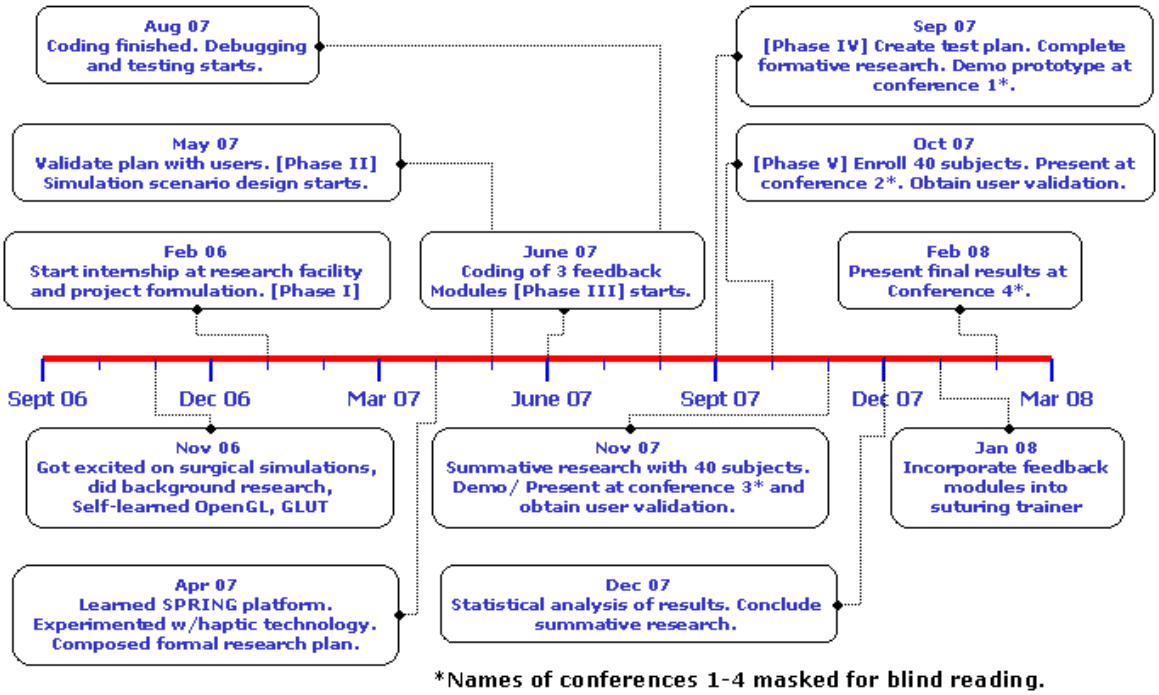

Figure 1: Project Timeline

2.3 Phase I – Understand the surgical simulation development environment

The development environment utilized by this study consists of the SPRING platform [9,10], an open-source soft-tissue simulator platform written in C++, and Microsoft Visual Studio. Key features (Fig 2) of the SPRING platform include:

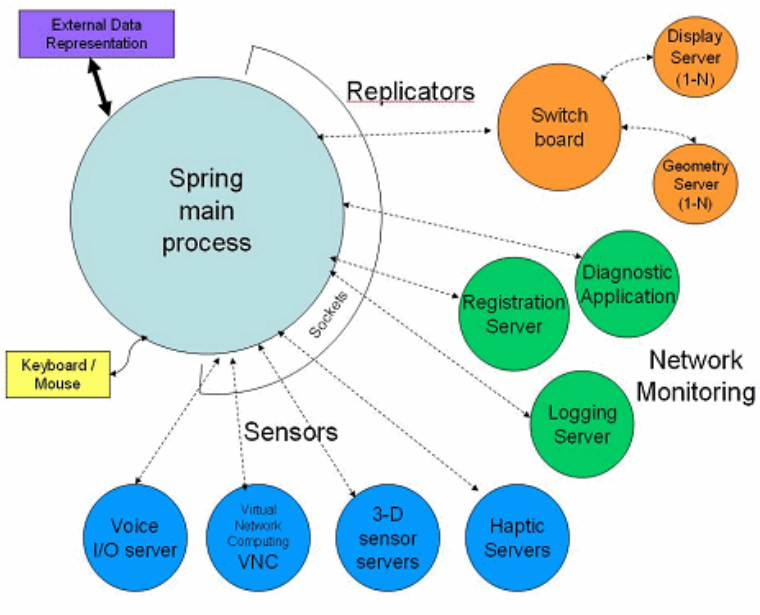

Figure 2: SPRING External Platform

a) Functionality with multi-platform and multiprocessor systems – SPRING can be compiled under Windows 2000/XP, LINUX, and UNIX

b) 3-D sensor servers and haptic servers which allow SPRING to connect to haptic devices via TCP/IP connections to exchange 3-D positioning and force-feedback information

c) External data representation methodologies to store 3-D objects and scenario-specific information in the form of .OBJ, .SOBJ, .WRL, and .DESC files

d) The main SPRING process – performs background calculations in collision detection, physics computations, and numerical solutions

Other open-source libraries used in this study include GLUT, OpenGL's Utility Toolkit and GLUI, the foundation for all user-interfaces within SPRING.

2.4 Phase II – Initiate the construction of a simulated scenario

This phase relates to the development of content and architecture for a scenario that has been used in this research for novice surgeons and medical students.

This research study takes a novel approach to identify the content of a scenario because unlike modern simulations [11], which replicate specific surgical procedures holistically, this scenario replicates the common characteristics in most procedures in a more simplistic manner. Fundamental manipulations in surgery today include exploration (comprising of visualization and palpitation), injection, incision, evacuation, scarification, extraction, excision, closure, and transplantation [12]. Of these, this scenario replicates exploration, incision, evacuation, and extraction, based on an objective to achieve the right balance of complexity (see Figure 3). Increased complexity is required so that the effects of the feedback modules could indeed be recognized in the results; if the scenario is easy to perform, then the increase in a user's performance due to repeated trials could merely be caused by user practice and the feedback modules could play no significant role in the improvement. Decreased complexity is required due to the fact that the experiment uses human subjects, and it is vital that their frustration does not dictate their end user performance as this may nullify potential benefits the feedback modules provide. Hence, the right balance of complexity in a scenario is a crucial decision in the study to ensure consistent and relevant results.

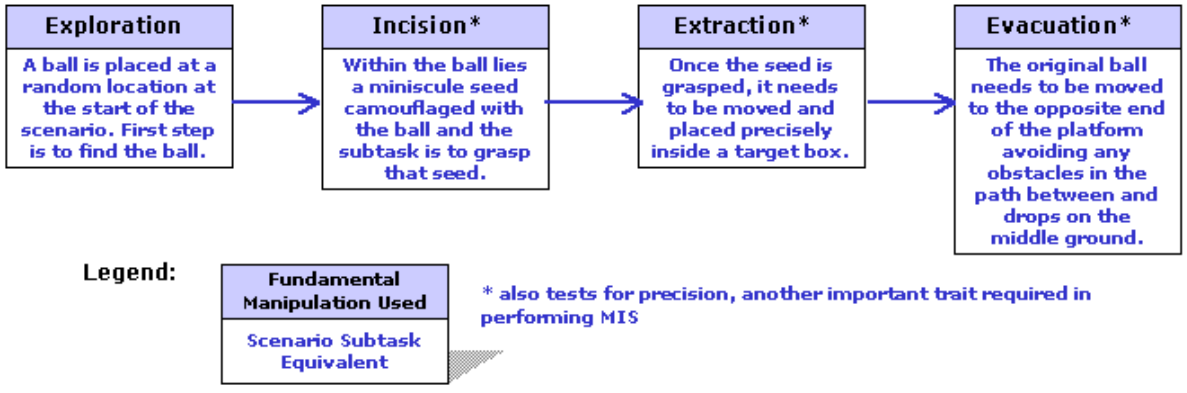

Figure 3: Fundamental Manipulation Used vs. Corresponding Scenario Subtask Equivalent

The following UML chart (Figure 4) shows the object-oriented implementation of the above scenario in the SPRING platform. This chart shows only the important classes and methods.

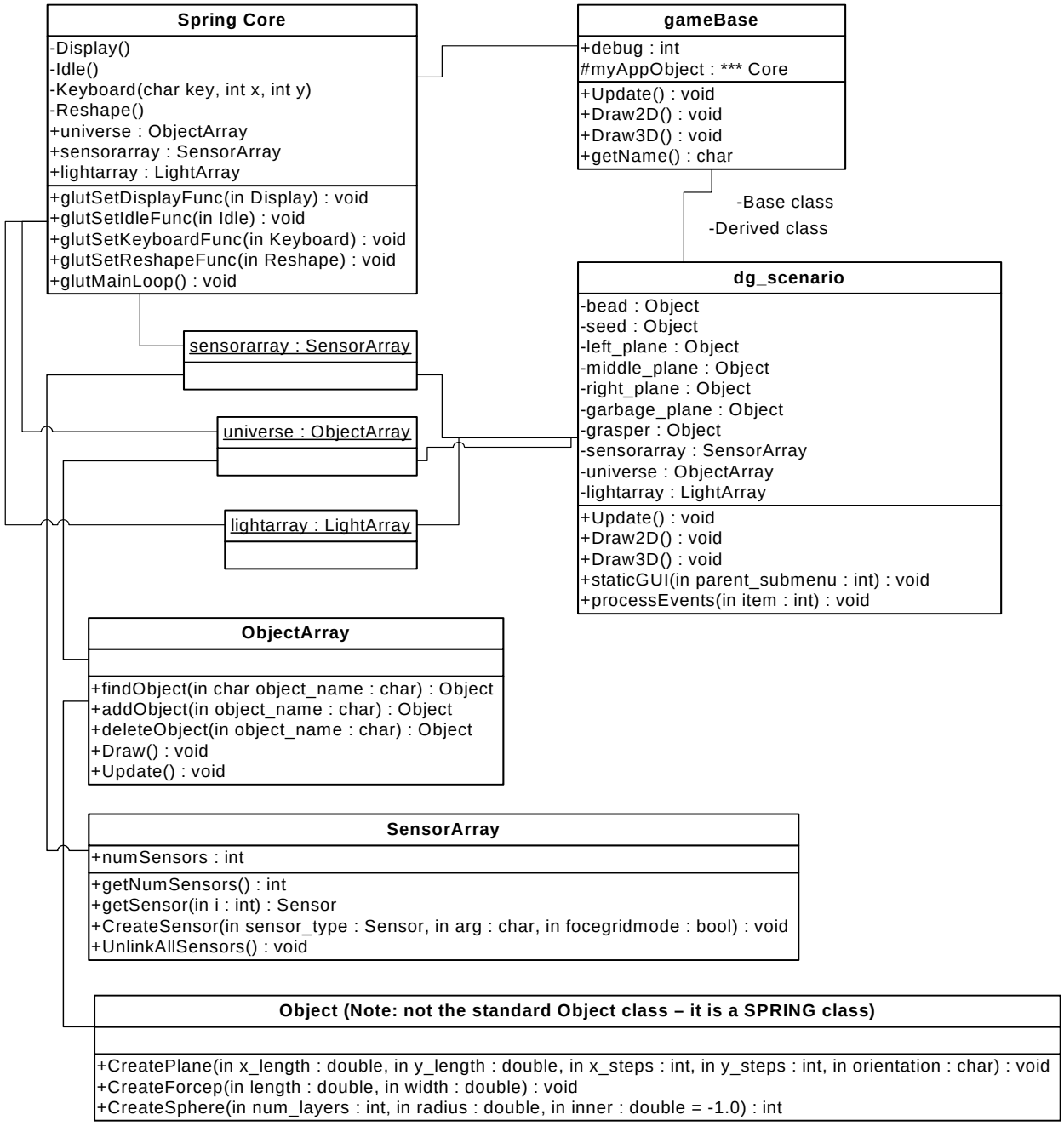

Figure 4: UML Diagram # 1 – Architecture of a custom scenario within SPRING Platform

The main SPRING process consists of a glutMainLoop() which has pre-registered callback functions to the application's Idle function, Display function, Keyboard function, and Reshape function. The Idle function is called every $1/30^{th}$ of a second and it first updates all the 3-D objects in the universe by handling their behaviors and collisions. If SPRING is running a scenario, it also calls the scenario's Update function. The Display function configures the camera position and angle, lighting intensities, and then draws all 3-D objects in the universe, by calling the Draw2D and Draw3D functions of the scenario, in their respective positions. The Keyboard function handles all keyboard commands.

2.5 Phase III – Design, develop, and embed the feedback modules

After completing the basic framework for the scenario, the next step is to design and incorporate the three feedback modules – a step-by-step monitoring system, audio-assisted commands, and a video assessment system.

2.5.1 Step-by-Step Monitoring System – Architecture

The step-by-step monitoring system is based on a linked-list data structure in which each node (part of the "wNode" class – see figure 6) contains its unique identifier, information on whether or not it is a decision node, its enumerated completion stage (incomplete, completing, and completed), and a brief text describing its contents. The "workflow" class contains a pointer to the first node in the linked-list and contains a Draw2D function, which processes the linked-list through an interpreter to print the nodes and their contents onto a window. The interactivity in the monitoring system arises from printing nodes' borderlines with different colors based on their completion stage in real-time. The completion stage of a workflow node is updated in the Update function of the scenario class based on logical reasoning unique to the scenario's objects,

states, and tools. Two components forming the basis for the logic are the user's sequence of completing the subtasks and the scenario's tool-object interactions. Figure 5 below provides a correlation between each workflow node and the circumstances within the scenario that triggers that node to be "completed." Please refer to Figure 3 to understand the scenario steps in detail.

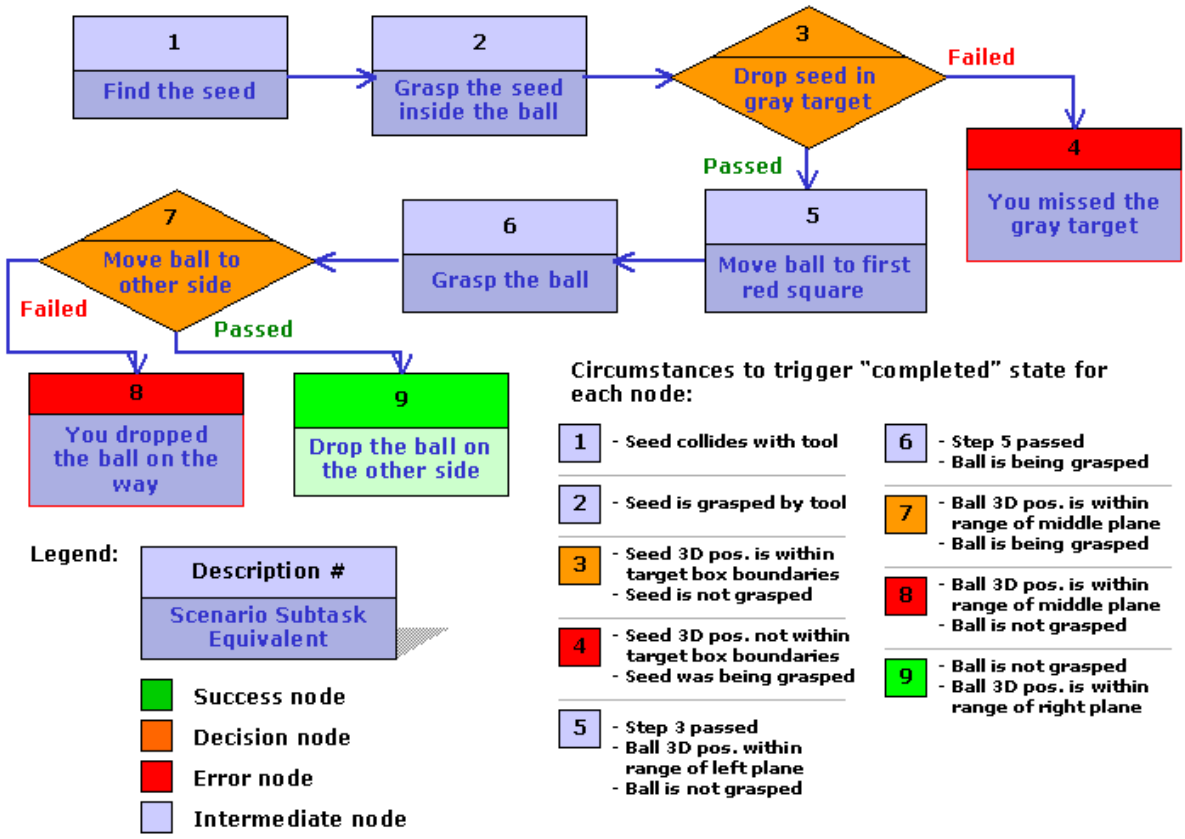

Figure 5: Workflow Node Breakdown vs. Circumstances to Trigger "Completed" State

2.5.2 Audio-Assisted Commands – Architecture

Real-time audio-assisted commands are based on task-specific instructions and standard warnings. In addition to reading instructions, users can hear the same through audio commands, which consist of task-specific prompts in .avi file format and universal warnings like "Stop!" and "Good Job!" The WavPlayer class in the SPRING platform based on OpenAL, and open-source audio API, is used to direct "play", "stop", and "isPlaying" commands. Audacity, an open-source

audio recording application, is used to record appropriate .avi files that can be played within the scenario application [13].

Another layer of logic is added to the Update function of the scenario to incorporate audio-assisted commands. This logic is directly correlated with the states of the workflow nodes which could be received using the getState function in the workflow class – as soon a node "n" is changed to a "completed" state, the audio file corresponding to node "n+1" is played. Additional triggers have been added to alert a user proceeding in the wrong direction within to scenario through standard universal commands (e.g. "Stop!"). Major checkpoints have been defined in the scenario, and when "completed", the "Good Job!" audio file is played.

Based on initial user validation, the architecture incorporates precautions to avoid ambiguity during run-time. These include a check using the "isPlaying" command to avoid any audio overlaps, and a multi-threaded environment, so long audio files don't stall the overall application.

2.5.3 Video Assessment System – Architecture

The video assessment system is based on two components, user error storage/identifier and visual 3-D replay.

User error storage/identifier – Each scenario has its own array of potential errors that a user can commit. An error check function inside the Update function checks every $1/30^{th}$ of a second whether an error has been committed in the scenario. This error check function is embedded into the logic previously developed by the step-by-step monitoring system and audio-assisted feedback modules. An error is logged when a workflow node, pre-identified as an error node, turns into its "completed" state. At this time, a description of the error and the sensor position of the main tool used at the time (in this scenario, a forceps is the only tool used) are written out to

an error file. The error file records the error description, the tool position in 3-D coordinates, and other related intelligent information.

Visual 3-D replay – This component is responsible for replaying all the 3-D objects accurately based on user manipulations during the actual scenario. All the initialized static "world" and dynamic data are written to a replay file in the scenario's Update function. Static data includes the initial positions of all the objects in the universe, and world properties (e.g. gravity). Dynamic data consists of the 3-D data (e.g. position, rotation-matrix, states) for dynamic objects and is written to the file every $1/30^{th}$ of a second. In this particular scenario, there are three dynamic objects, one of which is directly controlled by the user's manipulations (the forceps tool). Instead of keeping track of the necessary 3-D data of each object, which would be highly inefficient and data-intensive, only the 3-D data of the object being directly manipulated by the user is tracked. The 3-D details of the forceps collected dynamically every $1/30^{th}$ of second include the 3-D coordinates of the tool, the state regarding whether it is open (0) or closed (1), the four-by-four rotation matrix, which determines its angular position, and other intelligent information.

Once the actual scenario is complete, the video assessment system feedback module can be used to read in data from the replay file. First the "world" is initialized based on the file's static data. Then the dynamic data is read every $1/30^{th}$ of a second and:

a) the 3-D position is read and is used to set the position of the sensor attached to the forceps tool using the sensor's setPos function

b) the16 floating-point numbers are inserted into the forceps tool's rotation matrix

c) the forceps tool's "being grabbed" state is given an open (0) or closed (1) value

The efficiency of this algorithm arises from its object-oriented design – while the forceps is being replicated as it was previously manipulated by the user, SPRING platform's built-in physics models and numerical computations automatically allow the forceps to interact with the other two dynamic objects as if this replay was an actual user-manipulated scenario. Meanwhile, the 3-D data being read in from replay file are also compared to the 3-D data read in from the error file and if there is a match, then the replay pauses for ten seconds and the error description is displayed on the screen highlighting to the user that this is when and how the error occurred.

2.5.4 UML diagram for three feedback modules

An overview of the architectures for the three feedback modules in relation to the scenario (discussed in Section 2.4 - Phase II), is shown in Figure 6 on the next page.

2.6 Phase IV – Create a testing plan and formative research

The aim of this testing plan is to validate the hypothesis by evaluating the combinatorial effectiveness of the three real-time feedback modules on end user performance and learning. Initial user validation (formative research) has been done with 7 subjects, many of whom have also provided inputs from the conception of this project. In addition, this project has also been presented/demonstrated at conference 1 (actual name masked for blind read) of laparoscopic surgeons. While no statistical conclusions can be drawn by this formative research, these inputs have been useful in improving project UIs, streamlining information flow, fixing bugs and designing/refining detailed test plans for summative research.

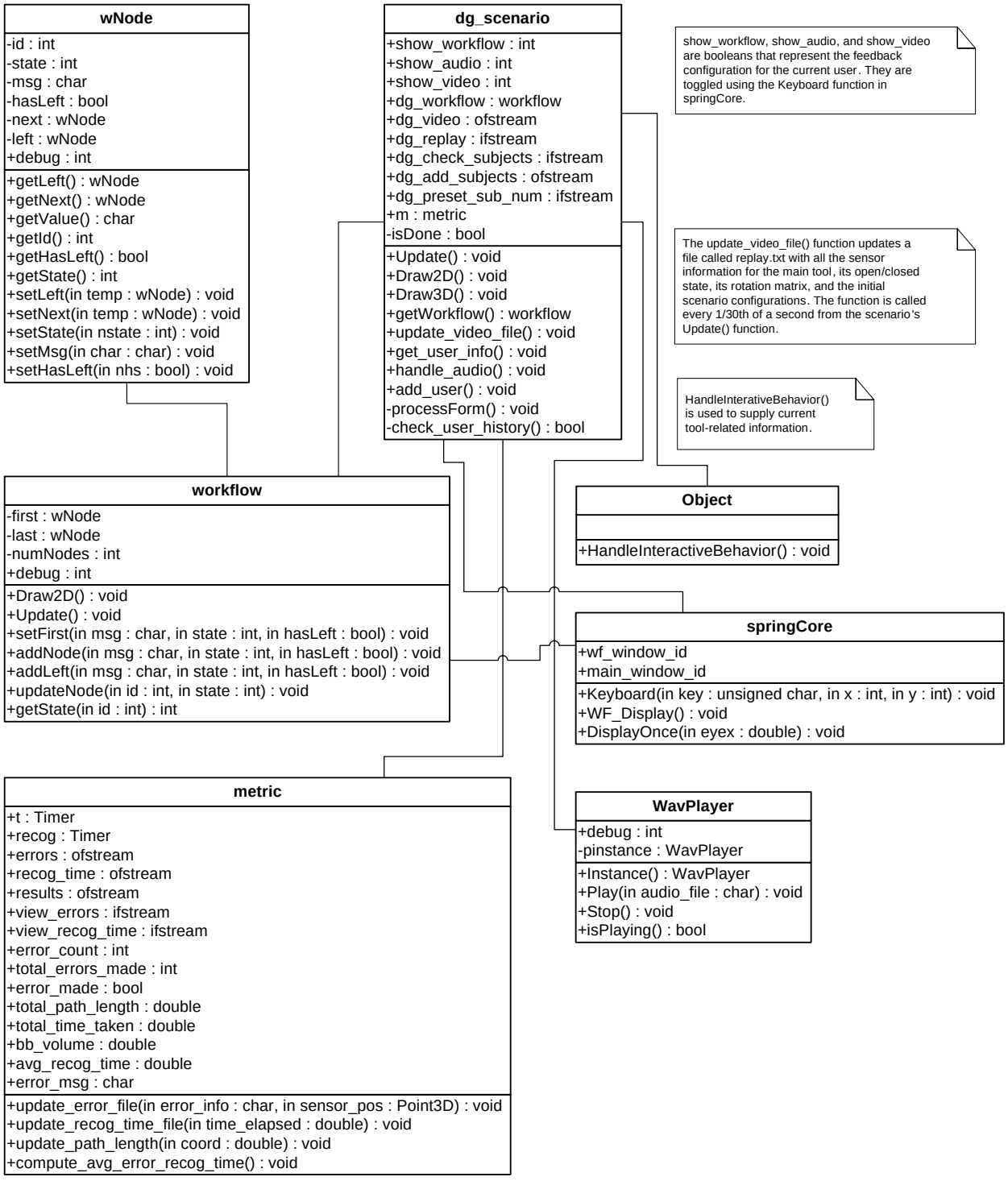

Figure 6: UML Diagram # 2 – Architecture of a custom scenario within SPRING attached to the three feedback modules implementations

2.6.1 Test Plan Details

40 medical students/novice surgeons with minimal cadaver and simulated surgical experience with would serve as subjects. Each of the three feedback modules, when toggled on/off, result in 8 possible groups (2 x 2 x 2). Based on this, subjects would be randomly divided into these 8 groups, one of which would be a control group. Each subject would do 10 tests over a period of 3 weeks with the expected time for each test expected to be around 15 minutes.

The study would collect several measures including user demographics, simulator scores, and surgical competency rating. In addition, 3 surgeons would independently grade the overall performance of the each user on a 9-step differential rating scale. The results would be then be analyzed for a statistical correlation.

Simulator scores and competency ratings consist of a set of metrics [14] including total time (seconds) the user took to complete the procedure [15], tool path trajectory comprising of total tool path length (mm) [16] and size of operating site (mm^3), number and type of errors made, and average error recognition time (seconds). All these metric computations are handled by a metric class, which has been built as a part of this study (see Figure 6 – metric class).

User demographics would be derived from a questionnaire filled out by the user prior to the scenario. The questions include preferred hand (left, right, ambidextrous), number of hours/wk playing video games, number of hours/wk using computer software, years of experience with haptic devices, years of experience with surgical simulations.

Incorporating the test plan in the overall application required some additional changes to the scenario and the SPRING platform. First, three keyboard commands have been added to the Keyboard function to toggle the feedback mechanisms on/off before the user began the simulation. Second, two additional data files have been added, one storing the metric

performance and background data of all subjects on all their trials, and the other keeping an account of all subject IDs. Third, the scenario has been configured to operate with a PHANTOM Omni Desktop (made by Sensable Technologies) haptic device which is similar to a robotic arm that allows the user to navigate is 3-D space and perform basic tool-tissue interactions like grasping and probing. For this configuration, the GHOST 4.0 API of the PHANTOM Omni Desktop is leveraged for communicating information with SPRING platform.

2.7 Phase V – Extensive testing, statistical analysis, and summative research

40 medical students/novice surgeons required for summative research are currently being recruited from the fall 2007 class at the research facility. Phase V is expected to start during Nov 2007.

3 Results

Since the conception of this project in Nov 2006, the study has resulted in the creation of a working prototype comprising of three feedback modules, leveraging open-source software. Based on this, approx 3000 additional lines of code have been added to the open source SPRING platform. Figure 8 on the next page shows a screen shot from prototype of the intelligent video assessment module.

During formative research, this prototype has been demonstrated and tested by 7 participants at the research facility. It has also been validated by novice surgeons at the conference 1 (actual name masked for blind read) of laparoscopic surgeons. Results collected during this phase are summarized below. This could however change based on exhaustive summative research planned for Nov 2007, with 40 subjects.

Figure 7: Formative Results Summary

Metrics	Control (no feedback)	With all 3 feedback modules
Mean total time (seconds)	186.6	145.2
Mean total path length (mm)	4144	401
Mean size of operating site (mm^3)	11451	9887
Mean number of errors	12	5
Mean error recognition time (seconds)	3.54	2.21

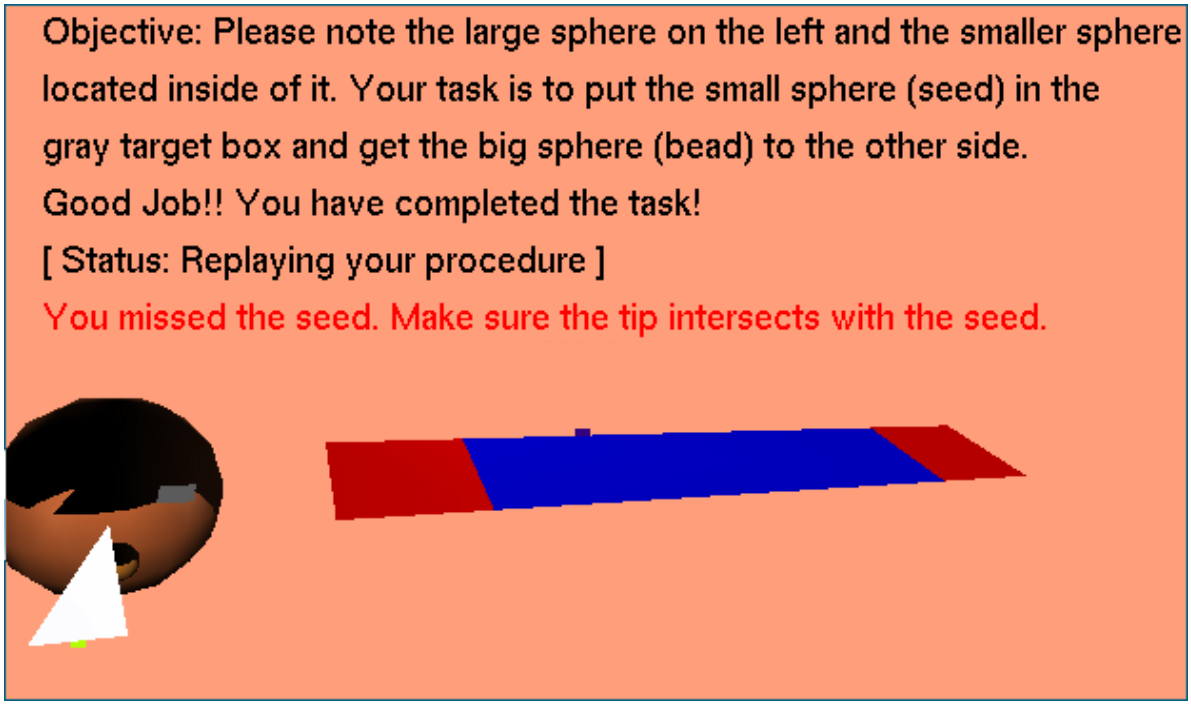

Figure 8: Screen Shot of Intelligent Video Assessment Feedback Module (Replaying Procedure)

In addition, formative research participants provided subjective feedback, which is highlighted in the Discussion section below.

4 Discussion

Findings based on formative research outlined above suggest that the subjects' (medical students/novice surgeons) performance improves significantly when subjects use all the three

feedback modules. Subjective feedback gathered from laparoscopic surgeons, endoscopic surgeons, and pediatric orthopedic surgeons, at multiple research facilities has resulted in further refinements to the prototype, especially relating to user-interfaces, streamlining information flow, and debugging.

The prototype also adequately satisfies the two key factors outlined earlier - versatility and cost effectiveness. Versatility is achieved through object-oriented design principles and the packaging format of the feedback modules, which allows the system to serve as an add-on feature to a wide range of surgical simulators. Cost effectiveness is achieved through extensive use of open-source software like the SPRING platform and Audacity audio recording software.

Unforeseen number of bugs inherent in the SPRING platform along with lack of documentation posed a challenge during the initial few months of this study. However, efforts to debug advanced our understanding of key computer science concepts such as memory referencing and pointers and also helped in comprehending the intricacies of this platform.

5 Conclusion

While the summative research is yet to be completed (Nov 2007), the outcome based on the formative research is extremely encouraging and this prototype fills a strong need for real-time feedback systems in surgical simulators. There is already a strong interest in this work and the same is therefore being exhibited in three forthcoming international conferences (actual names masked for blind read). Based on user inputs, however, the video assessment system needs to be improved further. It should not only show how and why a user made an error in the procedure but also play a brief video clip suggesting the correct way to perform that particular subtask. There are also plans to patent the three feedback modules after the completion of the summative research.

The feedback modules developed in this study have a huge potential for use by the medical students. USA alone has around 69,600 medical [17] students enrolled at any time and use of these feedback modules would result in improved quality of learning along with reduced cost of teaching/learning. Better quality of learning would then translate into better patient treatment as well.

The feedback modules developed in this study have a huge potential not only in surgical simulators but also in many other sectors across a wide range of industries, especially gaming, virtual reality technologies, and diverse applications leveraging robotics.

References

[1] "LapSim Basic Skills", Surgical Science, [Online document], Retrieved Feb 22, 2007, Available HTTP: http://www.surgical-science.com/index.cfm/en/products/lapsim_basic_skills_/

[2] "LTS3e," RealSim Systems, [Online document], Retrieved Feb 22, 2007, Available HTTP: http://www.realsimsystems.com/lts3e.htm

[3] "Lap Mentor", Simbionix, [Online document], Retrieved Feb 23, 2007, Available HTTP: http://www.simbionix.com/LAP_Mentor.html

[4] "ProMIS™ surgical simulator," Haptica Web site, [Online document], Retrieved Feb 24, 2007, Available HTTP: http://www.haptica.com/id11.htm

[5] "SurgicalSIM," METI, [Online document], Retrieved Feb 24, 2007, Available HTTP: http://www.meti.com/products_ss_lts.htm

[6] H. Maass, B. B. A. Chantier, H. K. Çakmak and U. G. Kühnapfel, "How to Add Force Feedback to a Surgery Simulator" in Surgery Simulation and Soft Tissue Modeling, Springer Berlin / Heidelberg, vol. 2673, 2003.

[7] V. F. Munoz, C. Vara-Thorbeck, J. G. DeGabriel, J. F. Lozano, E. Sanchez-Badajoz, A. Garcia-Cerezo, R. Toscano and A. Jimenez-Garrido, "A medical robotic assistant for minimally invasive surgery," IEEE International Conference on Robotics and Automation, vol. 3, pp. 2901 - 2906, 2000.

[8] K. Wegner, "Surgical navigation system and method using audio feedback" presented in International Conference on Auditory Display, Glasgow, Scotland, 1998

[9] K. Montogomery, C. Bruyns, J. Brown, S. Sorkin, F. Mazzella, G. Thonier, A. Tellier, B. Lerman and A. Menon, "Spring: a general framework for collaborative, real-time surgical simulation." Medicine Meets Virtual Reality., ISO Press, 2002.

[10] C. Corneilus, "SPRING Architecture in SPRING Open-Source Surgical Simulator," [Online Document], Retrieved Feb 16, 2007, Available HTTP: http://spring.stanford.edu/ArchitectureMain.html

[11] C. D. Bruyns, and K. Montgomery, "Generalized Interactions Using Virtual Tools within the Spring Framework: Probing, Piercing, Cauterizing and Ablating," Medicine Meets Virtual Reality., ISO Press, vol. 2, pp. 79-85, 2002.

[12] W. L. Heinrichs, S. Srivastava, K. Montgomery and P. Dev, "The Fundamental Manipulations of Surgery: A Structured Vocabulary for Designing Surgical Curricula and Simulators," The Journal of the American Association of Gynecologic Laparoscopists, vol 11, no. 4, pp. 450-456, 2004.

[13] "The Free, Cross-Platform Sound Editor," Audacity, [Online document], Retrieved July 22, 2007, Available HTTP: http://audacity.sourceforge.net/

[14] W. L. Heinrichs, B. Lukoff, P. Youngblood, P. Dev and R. Shavelson, "Criterion-based Training with Surgical Simulators: Proficiency of Experienced Surgeons," in Journal of Society of Laparoendoscopic Surgeons, 2007.

[15] S. Cotin, N. Stylopoulos, M. P. Ottensmeyer, P. F. Neumann, D. Rattner and S. Dawson, "Metrics for Laparoscopic Skills Trainers: The Weakest Link!", in Medical Image Computing and Computer-Assisted Intervention, MICCAI 2002: 5th International Conference, Tokyo, Japan, September 25-28, Proceedings, Part I, pp. 35-43, 2002.

[16] M.R. Satava, A. Cushieri and J. Hamdorf, (2003). "Metrics for objective assessment: preliminary summary of the Surgical Skills Workshop" Surgical Endoscopy, vol. 17, no. 2, pp. 220 - 226, 2003.

[17] "U.S. Medical School Enrollment Continues to Climb," Association of American Medical Colleges, [Online document], Retrieved Sept. 25, 2007, Available HTTP: http://www.aamc.org/newsroom/pressrel/2006/061018.htm