DATA MANAGEMENT WORKFLOW AND ARCHITECTURE FOR SIMULATION-BASED MEDICAL TRAINING

by

Nisrin Abou-Seido

A thesis submitted to the Department of Computing
in conformity with the requirements for
the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
(August, 2015)

Abstract

Purpose: Research is ongoing on how best to integrate simulation-based surgical training and the associated quantitative metrics into competency-based medical education. In a review of the literature, study data management tools and methods were lacking. Specifically, we were not able to find any open-source tools for managing data associated with simulation-based education. Our objective was to develop a practical data management workflow and architecture for simulation studies and implement the workflow as a free and open-source software tool. **Methods:** We designed the required data management to be compatible with the Perk Tutor (www.perktutor.org) free open-source medical simulation platform, developed primarily for image-guided intervention training. In order to determine key requirements, we consulted medical training experts who routinely use the Perk Tutor for trainee evaluation. With this information, we documented the current workflow and identified its limitations. Software design methods were followed to design the required data management module. In the analysis phase of the project, possible storage options were compared in order to choose one that would fulfill the requirements. Options included: XML, comma-separated values, e-learning platforms and database tools. **Results:** A prototype module was implemented as designed within the Perk Tutor platform. The module downloads, and parses data in CSV format from a learning management system (Moodle). Then, the simulation data generated and session metadata is saved together to the local file directory. The local directory is synchronized with a remote password-protected file server and trainee results are uploaded to Moodle. This system of data storage was found to be secure and user-friendly. The implemented module facilitates acquiring metadata and ensuring simulation results are securely stored and accessible for analysis. **Conclusion:** The new workflow design allowed us to implement a prototype module within the open-source Perk Tutor image-guided intervention training platform. Future work includes surveying users to evaluate the usefulness and performance of our data management solution.

Co-Authorship

The work described in this thesis was completed under the supervision of Dr. Gabor Fichtinger and Dr. Tamas Ungi.

Some of the work presented in this thesis was described in an extended abstract accepted for oral presentation at CARS 2015, the 29th International Congress and Exhibition for Computer Assisted Radiology and Surgery, June 24-27 2015. (N. Abou-Seido, T. Ungi, A. Lasso, R. McGraw, G Fichtinger, "Data Management Workflow and Architecture for Simulation-Based Medical Training", CARS 2015 Proceedings, Supplement of the International Journal of CARS, accepted Feb 12, 2015).

Acknowledgements

I am grateful to my co-supervisors, Dr. Gabor Fichtinger and Dr. Tamas Ungi for their support and assistance during my time in the Laboratory for Percutaneous Surgery and while working on this thesis. I thank Dr. Fichtinger for accepting me as a Master’s student and trusting me with this project. It’s been a wonderful learning experience and I am so thankful for the opportunity.

I am grateful to have met, and to have worked alongside, my peers and friends in the Perk Lab, who have been an invaluable source of advice and motivation. Special thanks to Dr. Tamas Ungi and Dr. Andras Lasso for their advice and guidance.

To my friends that I met at Queen’s, I am so glad to have met you and thank-you for being a welcome source of distraction and support. Finally, I want to acknowledge and thank my mom, dad, sister, and brother for always being there for me and supporting me no matter what.

Table of Contents

Glossary

3D Slicer	Software for visualizing medical images
CSV	Comma-separated values (a data interchange format)
CBME	Competency-based medical education
GUI	Graphical User Interface
MRML	Medical Reality Markup Language (XML-based data storage)
OR	Operating room
Perk Tutor	An image-guided needle placement training system
PLUS	Public software library for ultrasound imaging research
SFTP	Secure File Transfer Protocol
US	Ultrasound
XML	Extensible markup language

List of Figures

List of Tables

Chapter 1

Introduction

Computer-assisted simulation-based medical training and skills assessment has been receiving steadily growing attention over the past decade as a necessary component of competency-based medical curricula. In particular, as ultrasound-guided minimally-invasive interventions become more common, simulated ultrasound-guided needle placement training becomes necessary due to the steep learning curve. Central line needle placement, for example, must now be performed using US-guidance, and thus, simulators for this specific purpose are needed [1]. Typically, in order to assess trainee skill and competency, simulators compute and record a variety of performance metrics. For effective simulation-based medical education, simulators should record performance and provide quantitative trainee feedback, using valid metrics and accurate performance benchmarks [2]. However, establishing the validity of the metrics is an ongoing research problem. In a review of simulator validity studies, Michael *et al* found that "there was no standardized protocol for the methodology within the studies. Seven studies conducted pre-and post-simulator tests […] whereas the remainder only performed post-test analysis, which weakens the evidence for the usefulness of the simulator at teaching a particular skill as the baseline performance was not established" [3].

Through the documentation of the current workflow, this thesis will design and document a new workflow and architecture for data management in simulation-based training, to facilitate the integration of a surgical training platform with trainee management software. We hypothesize that this thesis will provide a practical method of systematically managing data about trainees and their performance, for use by educators in medical education.

1.1 Motivation

A literature review (Chapter 2) has shown that most simulation studies focus on publishing results with much less focus on the methodology and software used for recording and managing study data. This practice is problematic for multiple reasons. Primarily, without documented data recording and management methodology, it is unlikely that a given study's results will be reproducible. Secondly, the complexity of managing large volumes of data encourages researchers to reduce sample sizes to simplify data collection, thus making it difficult to draw conclusive statements. Thirdly, without a suitable alternative, the majority of researchers tend to rely on entering data manually into simple spreadsheets - a method that is both time-consuming and error-prone. Finally, lack of study data management facilities prevent rolling out simulation-based training methods into institution-wide curricula with self-guided training and testing. Throughout our investigations, we were not able to find any open-source tools for managing data associated with simulation-based education programs.

The Perk Tutor, an open-source, simulation-based application for ultrasound-guided intervention training was developed in recent years (www.perktutor.org). However, the effectiveness of the teaching method has been difficult to assess due to the limited data stored about each training session. The proposed solution is to create a standard workflow for associating metadata with a session. For example, each session has a participant or trainee and an associated study; we need to store data about the trainee and the study and use this data to sort and retrieve relevant sessions. The solution would be implemented as a software extension to the Perk Tutor that would facilitate carrying out research studies with medical residents for testing the validity, effectiveness, and usability of the training software.

1.2 Thesis Objectives

The overarching objective of this thesis is thus the design and documentation of a workflow and software architecture for management of trainee progress and study data within an open-source simulation training platform.

1.3 Thesis Contributions

The documentation of the current data management workflow and the design of an improved workflow enabled the development of an initial prototype that integrates an open-source simulation training platform with a learning management system to demonstrate the feasibility of the architecture and workflow. With further implementation this work will be beneficial for researchers in medical education and surgical simulation technology and will facilitate larger scale studies in these fields.

1.4 Thesis Outline

The rest of the thesis is divided into chapters as follows: **Chapter 2** of this thesis provides the background literature on competency-based medical education, simulated-based training and assessment, simulation study data management, and tools for managing data in general. **Chapter 3** describes the methodology for developing the workflow and software architecture, including requirements analysis and design. **Chapter 4** discusses results, describing the resulting workflow and prototype implementation. Finally, **Chapter 5** provides the conclusion and a brief discussion of possible future work.

Chapter 2

Background and Literature Review

2.1 Overview of Simulation-Based Surgical Training

Simulation-based surgical training, consisting of both physical simulators and virtual reality simulators, is commonly used for introducing medical residents to general surgery skills outside of the operating room. In lieu of the risky and ethically dubious practice of novices performing invasive procedures on real patients, simulation-based education, usually using a physical or virtual manikin, allows trainee to practice and receive feedback in a risk-free environment [4], [5].

Depending on the simulator, there is a wide array of advantages of simulation systems and reasons for their use in training. The availability of virtual reality trainers at any time, without expert supervision, is key. The ability to track hand and instrument movements allows computer-based simulators to automatically calculate trainee performance based on objective measures of skill. With standard simulation modules, it is possible for curriculum and training to become more standardized, less ad hoc and less reliant on the availability of real patients for practice. In general, ultrasound guided surgery and other minimally invasive surgical techniques have a steep learning curve and simulators aim to decrease the time it takes for a novice to achieve competency.

The increase in simulation-based training is also directly related to the adoption of competency-based approaches to medical training by medical schools across North America. Competency-based medical education has gained steadily increasing attention over the past decade. Its aim is to deemphasize traditional knowledge-based and time-based training and focus more on the learner's progress, evaluating medical residents on the basis of observable competencies and skills [6]. Internationally, an increasing number of medical boards have adopted competency-based medical education as their recommended approach. In Canada, the Royal College of Physicians and Surgeons in Canada (www.royalcollege.ca) is transitioning to a competency-based medical education (CBME) approach to residency education and specialty practice in Canada through its “Competence by Design” initiative for implementing CBME [7].

One of the strengths of simulators for testing purposes is that they are usually programmed to be able to present evaluation problems in the same way to every trainee and therefore examiners do not have to account for the variability of actual clinical scenarios. This ensures reproducibility, which is necessary both for studies as well as in assessment, especially in high-stakes assessments for certifications and licensing. To introduce medical students and residents to surgical skills, simulators are already widely used in most medical schools and teaching hospitals. For some skills, simulators have been approved for use in evaluation as well. The Royal College of Physicians and Surgeons of Canada has recently incorporated computer-based and mannequin simulations into their national Internal Medicine certification oral exams [7].

In current practice, an expert physician must be present during the simulation to observe and provide feedback or assessment. Otherwise, the simulated procedure may be captured on video and viewed by the expert at a later time. In order to reduce the demand on expert time, hand and instrument motion analysis techniques have been explored for recording movements and calculating numerical assessments automatically. However, determining which measurements and metrics to use, to accurately assess competence, is a non-trivial task. Expert knowledge is required for ensuring metrics are valid and significant measurements. This knowledge may be used for computationally evaluating a trainee’s proficiency at a given skill.

For effective surgical simulation training in a competency-based medical education program, further research is needed on several dimensions. For each skill or procedure, we need to determine the quantitative benchmarks for determining competence, how often a skill must be practiced to achieve and retain competence and how or whether simulation competency translates into operating room ability [8].

2.2 Clinical Motivation and Application

Laparoscopic surgery, a minimally-invasive surgical technique which uses only a small incision in the skin to insert a camera and tools for operating, is a surgical technique for which simulation-based training has become standard. Commercial training systems such as the Simbionix LapMentor and the SurgicalScience LapSim are in use across medical schools in North America with established training curricula and performance benchmarks. In fact, surgical residents must log a certain number of hours on these trainers in order to become qualified to perform the procedure. These systems provide trainees and instructors with quantitative feedback on their performance and competency.

Ultrasound-guided surgical training is another area where simulation-based education has seen growing acceptance. Facet-joint injection, nephrostomy and central line placement are examples of procedures for which ultrasound is useful for facilitating precise needle placement. Central line placement in particular is a procedure for which ultrasound guidance has become mandatory in Canada.

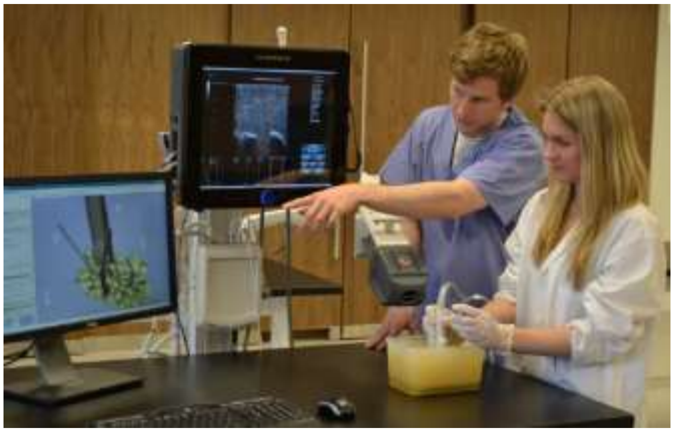


Figure 2.1: Facet-joint injection training [24]

Figure 2.1 shows an example set-up of facet joint injection training. The trainee uses the probe to scan the gel manikin and two monitors are set up to show the ultrasound image and a corresponding 3D image to assist the trainee with the needle injection. The problem, however, is that these image-guided needle interventions have a steep learning curve, caused by the difficulty of coordinating both the needle and the probe, tracking the patient, and interpreting noisy ultrasound images.

Lumbar puncture is an example of a fundamental procedure that is taught at the medical school level with a manikin and realistic simulation. A lumbar puncture consists of an injection in the lower back between two vertebrae to extract a sample of cerebrospinal fluid. It is important to clinical practice in many specialities for both diagnostic and therapeutic purposes. It is also a difficult procedure, requiring proficiency in sequencing of steps, positioning of the patient and correct needle placement. Training software for this procedure uses hand and instrument motion analysis to provide objective measures of performance. Competence in lumbar puncture is measured based on the following expert-validated metrics: total procedure time, total path distance travelled by hands, number of translational movements and number of rotational movements [9].

The competency based education model requires quantitative measures of performance and benchmarks in order to accurately evaluate whether a trainee has mastered a given skill or procedure. For lumbar puncture, studies have determined that each trainee requires between 10 to 50 practice sessions to achieve competency and skill retention. Curriculum design requires this type of data about other procedures and simulation systems as well.

2.3 Simulator-based Surgical Training Systems

The majority of simulation-based surgical training systems available today and discussed in the literature are commercially manufactured and distributed. These manufacturers include: Simbionix (www.simbionix.com), Surgical Science (www.surgicalscience.com) and CAE Healthcare (www.caehealthcare.com). Commercial simulator manufacturers often makes substantial claims of the benefits and effectiveness of training using their simulators. However, these claims must be studied for scientific validity, in particular to determine whether the simulator demonstrably models real-life interventions and whether simulator skills transfer to the corresponding skill in the operating room.

Surgical training simulators vary widely in their level of realism and technological advancement. Some low-fidelity model-based or computer-based simulators only teach basic tasks while integrated procedure simulators normally combine a manikin or other 3D object with a computerized system in order to create a high fidelity setting that models a whole procedure rather than an individual task.

Michael *et al* conducted a literature review on the validity of the state-of-the-art in integrated procedure simulators in 2014 and found that a large number of simulation systems have not been proven valid over all dimensions. The domains considered for assessing simulators include: feasibility, validity, educational impact, reliability, acceptability and cost-effectiveness. Validity testing attempts to determine the effectiveness, accuracy, and usefulness of the simulator as a training tool. Michael *et al* use a common "types of validity" framework which looks at five types of validity: face, content, construct, concurrent, and predictive validity. Face validity represents the degree of similarity between task performed and real situation setting. In high fidelity 3D simulators, face validity is higher. Content validity is the degree to which the task performed is being assessed and evaluates whether the simulator actually allow us to assess the trainee's skill at the given task. Construct validity is ability of the task to differentiate among different levels of expertise. On the other hand, concurrent validity is the degree of similarity in outcomes between the task performed and the gold standard task. Finally, predictive validity measures the degree to which the task can indicate future performance in a specific skill. A simulator with predictive validity has value in competency-based medical education because a high score on the simulator would imply competency with the actual surgical task. Only two of the surgical simulators reviewed demonstrated predictive validity – MIST-VR and LapSim. Studies that showed both predictive and face validity were rare. This review concluded that further research is required in order to categorically demonstrate the usefulness of these simulators. [3].

A literature review conducted in 2012 found that 97% of commercial simulators have not in fact been proven to be valid educational tools [10]. They may in fact be useful and valid tools, however they have not been studied sufficiently. Overall, these simulators are also expensive and proprietary which restricts their availability for studies.

The following subsections go into detail about the advantages and disadvantages of some of these surgical training systems.

2.4 Virtual Reality Simulators

2.4.1 MIST-VR

The MIST-VR (Minimally Invasive Surgical Training – Virtual Reality) is a computer-based system first presented in 1997 for teaching various laparoscopic surgical skills. Assessment is provided through a simulator generated score as well an observer's score of the videotaped exercise. In studies, MIST-VR performance was found to have predictive validity, where competence on the simulator may predict competence in reality, but minimal educational value as training on the simulator did not provide the trainee with a statistically significant improvement over their peers who did not receive simulator training. However, when combined with the LTS-2000 box trainer, the effect of combining these trainers led to a high educational impact. [3]. Recent literature on this simulator cannot be found and it seems to have been abandoned in favor of more modern virtual reality simulators.

2.4.2 LapSim (Surgical Science)

LapSim (Surgical Science) is a commercial virtual reality laparoscopic training system, with some published evidence that skills learned on this simulator will transfer to the OR It advertises curriculum integration and assessment and performance metrics for real-time onscreen instruction and off-site review. Exercises are digitally recorded with metrics, statistics and video debriefing for providing immediate and long-term feedback (www.surgicalscience.com). Its data management features include a results page which shows overview of results and allows student/trainee to add comments, the ability to directly import students from Excel files and a user-friendly graphical user interface.

A 2011 review of LapSim literature by Fairhurst *et al* found that LapSim is promising due to its use of a combination of measured parameters to produce an overall comparative performance score. One study, using LapSim to assess trainees with objective simulator-calculated scores and a questionnaire, showed construct validity for the simulator. However, it also found that LapSim does not in fact have proven significant advantages over other video trainers in terms of basic skills instruction [11]. In addition, a more recent study on predictive validity did not find that LapSim practice transferred to the OR [12]. More work is needed to determine standardized learning curves and proficiency criteria for different levels of trainees. As it is a commercial system, detailed information about the software and methodology behind the technology is not available.

2.4.3 Simbionix LAP Mentor

The LAP Mentor is a commercial high fidelity (virtual reality) laparoscopic trainer manufactured by Simbionix with some literature on its educational value. With a Likert-scale questionnaire administered to surgeons and surgical trainees, Lap Mentor was found to have face validity. That is, a large percentage of the respondents said that they expect this simulator would improve laparoscopic skills and shorten the learning curve [13]. Additionally, Zhang *et al* found that the LAP Mentor's performance parameters can be used to distinguish between subjects with various levels of laparoscopic experience thus demonstrating the simulator's construct validity[14]. Later papers have studied the validity of the LAP Mentor as a training tool. McDougall *et al* found that in a comparison between those trained on the virtual reality LAP

Mentor and those trained on a low-fidelity pelvic trainer with a silicone model, there was no statistically significant difference in scores [15].

Overall, the LAP Mentor is a valid training tool and clinical studies have demonstrated its value especially in introducing novices to basic laparoscopic skills. However, as a commercial product it does have a high cost of acquisition which may not be affordable for all institutions if less costly alternatives are available.

Simbionix MentorLearn -- Training Management Software

Simbionix MentorLearn is a web-based curriculum management system which integrates educational content and trainee management with Simbionix simulators. It includes user and group management, curriculum design tools, online course material and integration with the simulators for tracking trainee progress and exporting performance reports into files for statistical analysis. It is also intended for trainees to use offsite for viewing training objectives and their performance.

As a Simbionix product it is commercial and closed-source and only integrates with Simbionix simulator software. Detailed implementation details are not available; the above information on product features was found in the product brochure (www.simbionix.com).

2.5 Hybrid and Augmented Reality Simulators

2.5.1 CAE Healthcare VIMEDIX Ultrasound Simulator

VIMEDIX is a hybrid physical and virtual training platform for echocardiography using computer-generated ultrasound simulation. Using a manikin and mock ultrasound transducer, the trainee practices while viewing the patient anatomy and ultrasound scan in real-time on a monitor. This platform aims to shorten the path to ultrasound competency. It has a “Smart Metrics” feature which captures the learner’s movements in order to provide feedback to the trainee and the instructor. The software also comes with different pathology and patient configurations for specialized training.

A pilot study evaluating whether trainees can achieve ultrasound competency using this simulator found that students were overall more successful on the simulated scenarios after a period of training [16]. Transfer to real-life ultrasound competency was not studied. Further research is required to establish this simulator as a cost and time-efficient training tool that may be integrated in medical curricula.

2.5.2 Perk Tutor and 3D Slicer

The Perk Tutor (www.perktutor.org) is a computer-based surgical training platform. It is considered augmented reality because it displays a 3D image on the screen of the physical model and tracked instruments. It can simulate image-guided needle-based interventions and calculate certain metrics for evaluating the trainee's performance. It is an extension of the open-source and widely used medical imaging software 3D Slicer (www.slicer.org) and is freely available to be downloaded and installed through the Slicer Extension Manager. The Perk Tutor itself is also open-source and was developed primarily at Queen's University within the Laboratory for Percutaneous Surgery. It is configurable for conducting studies on hand and instrument motion analysis of trainees learning a variety of ultrasound-guided surgical skills.

Ungi *et al* [16] tested the configurable design of the Perk Tutor in three training scenarios for targeting, lumbar needle placement, and prostate biopsy tutoring showing the ability of the Perk Tutor to accommodate different clinical applications and different learning objectives without software or hardware design work. It has the advantage of providing continuous quantitative feedback for the trainee during practice, which is not available in the traditional apprenticeship approach. As well, the Perk Tutor allows trainees to practice at their own pace without the stress of a real clinical situation. This makes transition to clinical procedures easier for the trainees, when they have the necessary procedures practiced to the point of automatic execution [17].

The Transform Recorder module of the extension is used for recording instrument motion while the Perk Evaluator module calculates a set of metrics based on the XML file generated by the Transform Recorder. It works by recording needle movement and the trainee’s hand motions, using the PLUS toolkit for transmitting ultrasound and tool position data. Performance metrics are displayed continuously and summary statistics may be created at the end of an evaluation.

Feedback from supervisors is hindered by subjectivity, strengthening the need for objective performance measures. The Perk Tutor addresses this issue by providing computer-generated metrics and performance benchmarks. Though there has been significant interest in simulators for US-guided needle interventions, there are currently three major drawbacks to existing simulators. These issues are procedural specificity, lack of access for education researchers, and high cost. The Perk Tutor addresses these shortcomings. The configurable design of the Perk Tutor allows users to customize the system to serve their own particular training objectives. The Perk Tutor is an ongoing project, and new features are continuously being added.

It should be noted that further investigation is required to assess the specific effects of the Perk Tutor on skill development and retention. In particular, translational research is required for translating the Perk Tutor to actual use in medical education and producing quantitative evidence of its utility and efficacy in ultrasound-guided needle placement training. For example, medical educators require data on how many times and how often trainees need to practice in order to achieve and retain competence. This is facilitated by the openness and accessibility of the Perk Tutor. The Perk Tutor is distributed under a Berkley Source Distribution (BSD)-style open-source license that does not restrict use of the software or documentation. Open-source promotes collaboration and standardization among educational researchers and encourages usage of the software in both open-source and commercial applications. Collaborative development is supported by the Perk Tutor website, where source code, issue tracking, wiki documentation, events, and meetings are available at www.assembla.com/spaces/sligerigt.

In deploying the Perk Tutor, technician training for setup calibration and operating the graphical user interface may take a few days for new users. This is in some ways a tradeoff between simplicity and flexibility. To maintain the configurable design of the Perk Tutor, users need to invest some time in initial training. The steep learning curve may, however, be offset by some of the overall benefits of a configurable system. Configuring the Perk Tutor for new applications does not involve software development making it accessible to a broader range of users. In addition, is also less time-consuming and less costly than implementing a new system for each procedure.

2.6 Literature on Integration of Simulators into Medical Curricula

As new simulator technology becomes available, simulator-based assessment grows as a useful tool for competency-based medical education. However, a review of the literature on medical education has found that the quality of the research in the field is lacking. The majority of the published research studies are limited in size. More than half of the reviewed studies used less than 30 subjects or study participants [18]. Small study size limits the statistical significance of studies conducted. An increasing number of large-scale studies would contribute to a greater understanding of the validity of a given simulation technology.

Many studies have shown differences in the performance between small groups with large expected differences in skill (for example: experienced surgeons versus medical students). However, large studies are required for assessing if the simulator metrics can identify smaller differences in skill such as between trainees in different years of residency training.

Metrics used to evaluate technical skills need to be assessed methodically to ensure that they are reliable, valid, practical, and meaningful. To be useful for assessing competency, simulator metrics should objectively evaluate both efficiency (e.g. an experienced surgeon takes less time to insert a needle) and quality (e.g. limiting the amount of tissue damage). The measurements should also be readily available to learners to provide immediate feedback [2].

The importance of feedback in education is well-known and there is evidence that the use of medical simulations leads to improved educational outcomes. In a comprehensive overview of high-fidelity medical simulators, it was found that the top three simulator features for effective learning are: trainee feedback (in 47% of journals), repetitive practice (in 39% of journals) and curriculum integration (in 25% of journals) [19]. Traditionally, trainee feedback is delivered by an expert after an in-person evaluation. With objective metrics, computer-based simulation would ideally provide automatic trainee feedback during practice sessions.

Successful integration of simulation technology into training programs requires significant planning and decision-making. Decision include whether simulation should be voluntary or mandatory, how much supervision is necessary and how much simulation training is required to be beneficial. A multi-step approach is suggested by McClusky *et al*, wherein the simulation training is preceded by instructor-led teaching and testing in order to establish a base level of knowledge. Then the trainee practices on the simulator to achieve a certain level of competency before progressing to the OR. In this curriculum model, the simulation is integrated with traditional teaching that involves feedback and supervision. The paper does not discuss data management but the model may benefit from integration with learning management software and computer-based simulators[20].

2.7 Metadata Management

Metadata or meta-information, is generally defined as “data about data”. It provides context to the data that is stored. It may include, among other things, information about the conducted survey, the sampling subjects and method used, the population studied, the clinical protocol enforced, and the eligibility criteria [21]. Often, metadata for studies is simply stored in the researcher's personal lab notebook; however, for data preservation and longevity, it is much more useful if stored in a format that can be easily understood by others in the future. One way to do this is to ensure that the files and directories containing the data have clear and descriptive names. An Excel file with abbreviations for headers or a folder full of images with default filenames like "IMG_5473" will quickly become useless. In addition, metadata should include information about who has permission to use the data and for what purposes, especially in human studies, where ethics and confidentiality are key.

A thorough review of the literature concerning simulation studies has revealed a lack of standardized methodology for handling study metadata. The terms searched included “simulation study metadata”, “metadata management”, and “medical education technology”. Results returned include some studies which have explored metadata management in the context of clinical drug studies. However, comprehensive software tools for managing simulation-based training were not found.

2.8 CSV (Comma-separated values)

One of the key advantages of CSV files is their portability over various operating systems and software programs. Unlike Excel, which uses proprietary non-human-readable symbols for storing data in a particular format, comma-separated value files can be read by any application. “To increase reusability, tabular data should be stored in a format that can be opened by any type of software, i.e. text files. These text files use delimiters to indicate different columns” [22]. Comma-delimited text files, i.e. files with the .csv extension, are the most commonly used. Text files are preferred over proprietary formats such as .xls and .xlsx (Excel) because those formats can be difficult to import and export between programs.

In addition, CSV files can be imported and exported by almost any software designed for string or manipulating data, including relational database systems, spreadsheet software and statistical analysis software. CSV is generally the preferred format for interchange in many contexts because it is non-verbose, yet human-readable and easy to process.

CSV is advantageous due to the fact that is a simple, standard format which is both human-readable and easily processed. It is also a lightweight format; file sizes are small even for large amounts of data. It has no external dependences and can easily be imported or exported from or to spreadsheets.

Some of the possible drawbacks of using CSV include its limited ability to represent metadata (at most, a header row may be used to label each column) and the necessity of using a different separator if the actual data happens to contain commas.

2.9 Learning Management System (LMS)

Learning management systems, also known as online course management or e-learning, are software applications for managing courses and sharing course materials with students. Integration of an LMS with a surgical simulation training system is discussed as a data storage option in Chapter 3. The three most widely used systems at the post-secondary level are Blackboard (www.blackboard.com), Moodle (www.moodle.org), and D2L (www.d2l.com).

One of the key advantages of all of these learning management systems is that they are web-based and thus accessible from any web browser using a graphical user interface. As well, they have security and authentication built-in thus ensuring user privacy and data confidentiality.

Moodle (www.moodle.org) is the only one of these three systems which is an open-source learning management platform. Some of the advantages of Moodle are that it is free with no licensing fee, open-source, and highly flexible and customizable due to its open-source nature allowing users to create plugins and integrate it with external applications. Additionally, Moodle’s file repository system is a unique feature that simplifies the process of adding files to the system. For this project, it is advantageous that Moodle is currently used campus-wide at Queen’s University and many other Canadian universities.

Chapter 3

System Analysis and Design

In order to develop a new workflow that would effectively handle the data that is generated in training, as well as be usable, reliable and an improvement on the current system, we followed a systematic process. This process is roughly based on the common software development life cycle, where the main tasks are: requirements elicitation, requirements analysis, design, implementation and testing. However, we aimed to follow an iterative process such that if the project does not satisfy requirements after testing, the next step would be to return to the analysis phase in order to make changes.

Figure 3.1 shows the complete iterative process and the types of design decisions at each stage of the process. For this thesis, the scope of the project is the design and the documentation of the architecture, thus the final stage of the process in Figure 3.1 is intended for future work. Refining the requirements did, however, occur at the design stage and during the implementation of the prototype. The discussion at the end of Chapter 4 outlines in detail some of the new requirements discovered while prototyping.

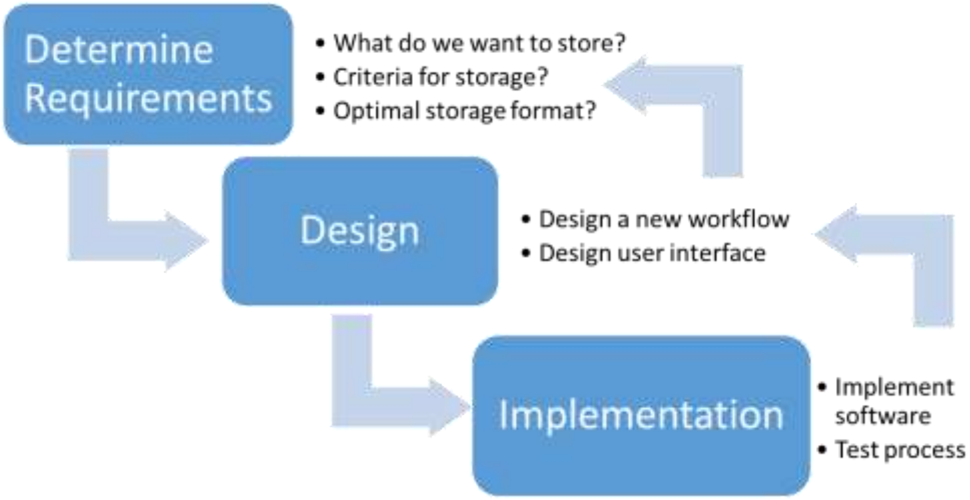


Figure 3.1: Iterative system design and development process

The following specific steps were followed:

1. Determine requirements and document the current workflow
2. Identify areas where it can be improved or automated
3. Investigate possible solutions
4. Design new workflow and system architecture which resolves the issues identified
5. Review design and requirements and add/remove requirements as necessary
6. Implement initial prototype using the design documentation
7. Demo and discuss the workflow and software with key users
8. Repeat applicable steps of the process if system does not satisfy user requirements

3.1 Analysis and Documentation of Current Processes

Ongoing studies on the effectiveness and validity of Perk Tutor were examined for their methodology. Through consultations with course facilitators and Perk Tutor users, it was possible to determine the current accepted workflow. We found that, although the process is not standardized, a given user is likely to follow approximately the same steps most of the time. The table below (Table 3.1) outlines the necessary tasks of the system and what the currently accepted process is for carrying out the given task.

TASK (What)	PROCESS (How)
Select and load scene	Find scene on local computer and drag into Slicer or open using file dialog
Authenticate user	Instructor or course administrator monitors has access to Slicer computer
Save study data locally	Write a note on paper then enter into spreadsheet
Save files locally	Click “save”, choose a directory and name the file (ideally something descriptive)
Collect trainee feedback	Give trainee hard copy form, type up results later
Copy data from local computer to global server	Use a USB flash drive to move all required files
Store metadata (data about study and trainees)	Maintain a Word document or an Excel spreadsheet
Reload scene into Slicer	Locate file by referring to memory or personal notes on where files were stored
Data Mining/ Creating a visual representation of data	Use Excel to create a spreadsheet and create graphs/tables

Table 3.1: Workflow tasks and current processes (how they are carried out)

3.1.1 Current workflow

Using the information in Table 3.1 and other observations, we documented the current workflow and identified its limitations using data flow diagrams. While UML Activity diagrams are similar to the data flow diagram, they do not allow us to easily see information flow and data dependencies. Representing the workflow with data flow diagrams was useful as it captures the overall process flow, what activities are performed and the relevant input, output and triggers for each activity. Additionally, the diagrams outline where information is stored, where it comes from, and where it needs to go.

In order to accurately document the workflow for conducting Perk Tutor studies, we define the following terminology in order to ensure clarity:

Trainee: a medical student or resident, of any skill level, who is being trained using the simulation software. In this paper, trainees may refer to actual trainees or study subjects.

Study: an experiment conducted over a particular period of time where a given set of trainees perform a task where their skill level is measured with the Perk Tutor.

Training Session: one trainee performing a given task once and saving the results of their work.

Figure 3.2 shows the first part of the workflow. The steps approximately progress from left to right, with the "Conducting Study" subsection occurring first. The **blue boxes** represent data and metadata. This includes data on trainees, the study, session video, and trainee feedback. In the current workflow, this data is manually entered into spreadsheets and manually copied as needed. The **orange boxes** are used to represent a process or an action carried out by an instructor. Actions include recording, saving and copying data onto various storage devices. Finally, the **green cylinders** are used to represent storage. Storage locations in the current workflow include the Slicer computer, the data analyst computer (for reloading data and calculating metrics) and a data server which is accessible to all administrators.

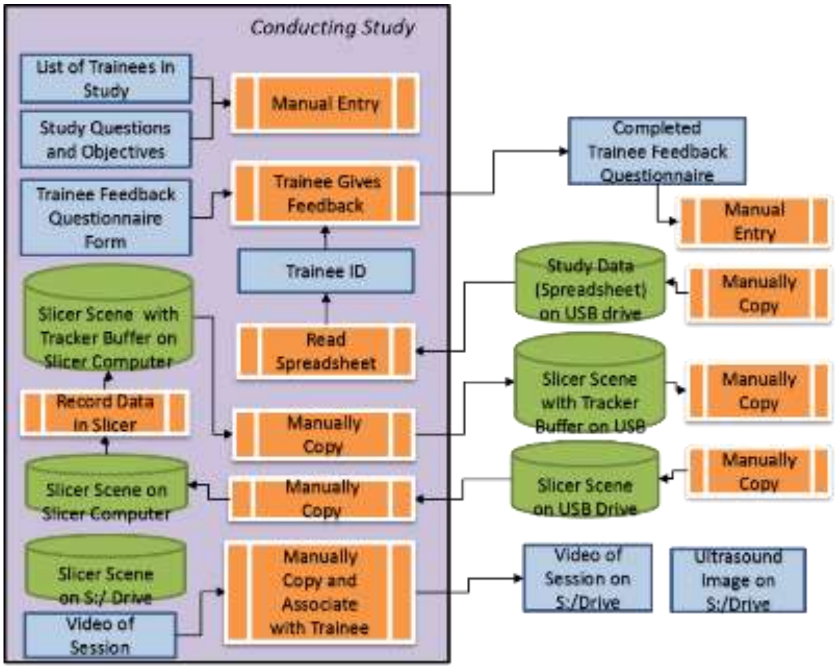

Conducting Study
List of Trainees in Study
Study Questions and Objectives
Manual Entry
Trainee Feedback Questionnaire Form
Trainee Gives Feedback
Trainee ID
Slicer Scene with Tracker Buffer on Slicer Computer
Read Spreadsheet
Record Data in Slicer
Manually Copy
Slicer Scene on Slicer Computer
Manually Copy
Slicer Scene on S:/ Drive
Video of Session
Manually Copy and Associate with Trainee
Completed Trainee Feedback Questionnaire
Manual Entry
Study Data (Spreadsheet) on USB drive
Manually Copy
Slicer Scene with Tracker Buffer on USB
Manually Copy
Slicer Scene on USB Drive
Manually Copy
Video of Session on S:/Drive
Ultrasound Image on S:/Drive

Figure 3.2: Part 1 of current workflow -- conducting a study and copying data

The data collected at this stage includes the list of trainees in the course, the course questions and objectives, the trainee feedback questionnaire form and most importantly, the ID of the trainee performing the operation. All of this data is manually entered into one or more spreadsheets ideally before the practice sessions are conducted. For each session, the instructor reads the spreadsheet, opens the correct Slicer Scene on the Slicer computer, has the trainee complete the task then saves the Slicer Scene with the Tracker Buffer on the Slicer computer. All of the data that is stored on the Slicer computer is then manually copied using a USB flash drive to another computer for long-term storage and analysis. Finally, the trainee provides feedback on their experience in the form of a hard-copy feedback form and the information on these forms is later typed up into a spreadsheet.

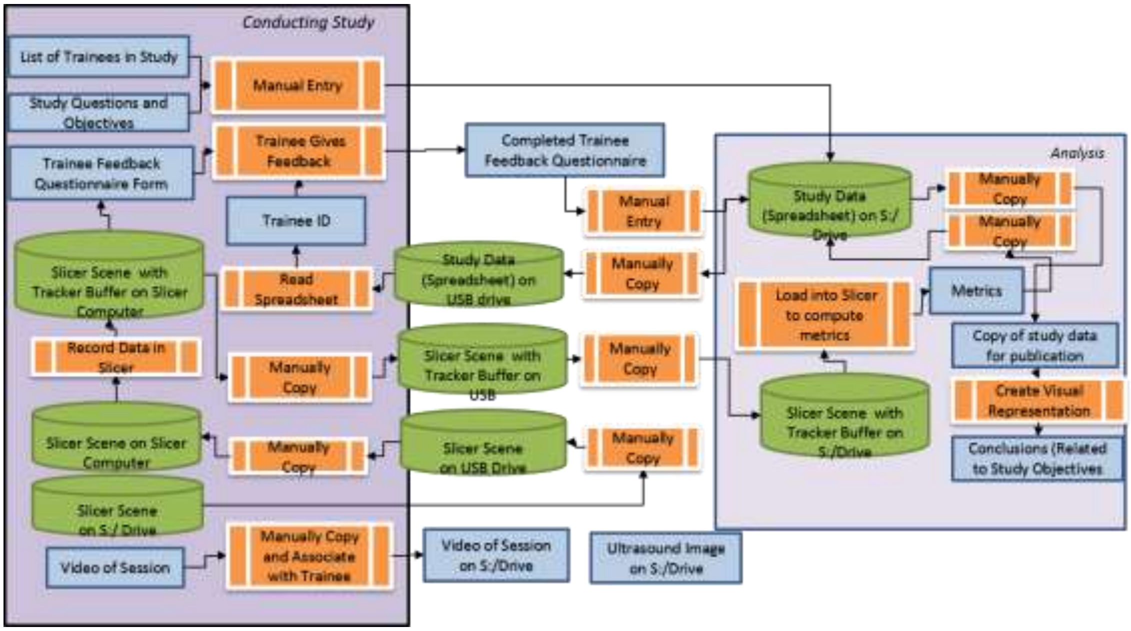

Conducting Study
List of Trainees in Study
Manual Entry
Study Questions and Objectives
Trainee Gives Feedback
Trainee Feedback Questionnaire Form
Completed Trainee Feedback Questionnaire
Trainee ID
Analysis
Manual Entry
Study Data (Spreadsheet) on S:/ Drive
Manually Copy
Manually Copy
Slicer Scene with Tracker Buffer on Slicer Computer
Read Spreadsheet
Study Data (Spreadsheet) on USB drive
Manually Copy
Load into Slicer to compute metrics
Metrics
Record Data in Slicer
Manually Copy
Slicer Scene with Tracker Buffer on USB
Manually Copy
Copy of study data for publication
Create Visual Representation
Slicer Scene with Tracker Buffer on S:/Drive
Slicer Scene on Slicer Computer
Manually Copy
Slicer Scene on USB Drive
Manually Copy
Conclusions (Related to Study Objectives
Slicer Scene on S:/ Drive
Video of Session
Manually Copy and Associate with Trainee
Video of Session on S:/Drive
Ultrasound Image on S:/Drive

Figure 3.3 Overall current workflow diagram (including analysis)

Figure 3.3 expands the workflow to include the analysis subsection. For analysis, data is stored on the "S:/Drive", a server used amongst members of the Perk Lab for storing shared data, images, and files. To restrict data access on this server, a particular user or group can be granted read/write access, read-only access, or blocked entirely. However, the file structure on this server is not standardized and users are free to create and name files as needed. Thus, the status quo has the course administrator creating a directory for their study and copying files to it manually. Files that may exist on this server include the study data (in a spreadsheet or a text file), and the Slicer scenes with tracker buffer for each subject. The course administrator then loads each tracker buffer file into Slicer, calculates metrics, copies the data onto their personal computer, creates visual representations, and draws conclusions based on their analysis.

3.1.2 Identifying Areas of Improvement

By examining the above diagrams and looking for possible areas of improvement, we identified the following flaws in the workflow which require improvement.

1) Arbitrary file storage

When a training session is complete, the trainee or the course administrator saves the file anywhere on the file system. The file does not save automatically. It is possible to forget where the file was saved or what it was called, thus requiring the session to be repeated, or at minimum, time is spent searching for the missing data. Finally, the file may also be located on a local, personal drive that is not accessible to other researchers who may be able to reuse the data.

2) Informal study metadata management

Typically, trainee identification data and other notes about the study or sessions conducted is recorded and- maintained in a personal spreadsheet or notebook.

3) **Manual copy**

A recurring issue in this workflow is that most processes are manual and thus error-prone. In particular, “Manually Copy” is one process that ideally should be minimized or eliminated if possible as it describes the unreliable and insecure process of using a USB flash drive to copy and paste data between machines.

4) Paper feedback forms

After concluding a training session, trainee feedback is collected through hard copy forms. For analysis, the data is then typed up later. This is time-consuming and error-prone and would not be difficult to improve with a web form or similar tool.

3.2 Determine System Requirements

One of the initial requirements of this application is that it should be compatible with 3D Slicer software (www.slicer.org) and the Perk Tutor (www.perktutor.org). The Perk Tutor is used by other lab members in the Laboratory for Percutaneous Surgery in the School of Computing at Queens University for measuring surgical skill in various studies. To determine other key requirements, we consulted medical training experts who routinely use the Perk Tutor for trainee evaluation.

In the Clinical Simulation Centre in which most medical training activities are conducted in the Queen's School of Medicine, constant and reliable network connectivity cannot be guaranteed. Handling the problem of intermittent internet access while limiting the use of USB flash drives for transferring files between computers was one of the goals of the new workflow.

Another required feature of the new system was to facilitate the calculation of metrics for determining a medical trainee's proficiency in the image-guided interventions. The simulation software lacks a tool for collecting and organizing the data generated for offline analysis. Thus, standardizing the process of associating the trainee metadata with the trainee's work became a key requirement. Metadata that must be stored about each study consists of the list of trainees in a study, the questions and objectives of the study. Information about the trainees and the study need to be available to Slicer when data is saved. Ideally, a user authentication system would also be involved to ensure data integrity and access control.

Requirements consist of feature requirements and non-functional quality requirements. Non-functional requirements are system specifications that are constraints on the system or qualitative attributes of the software rather than module features [23]. In this application, reliability, security and usability are non-functional system requirements which have an effect on the quality of the software. Reliability is important because this system will be used for collecting and managing study data and if there is a high chance of failure to save the data, users (course administrators) will be reluctant to use it lest they need to repeat experiments. Security is also a relevant requirement for this application due to the sensitive nature of the data we are storing. In this case, we aim to ensure that data is password protected and access to the data is limited to those who need to use it. In addition, for privacy purposes, the participants' identifying information will be stored separate from their results. Finally, we include usability as well as a key quality requirement due to the fact that users may not be technically proficient and may not use the software or the new workflow often.

Non-functional System Requirements

Quality Requirements:

1. Reliability
    1.1. Data is stored in more than one location in order to minimize loss of data.
2. Security (password protected, secure user data, access to data is controlled)
    2.1. Only administrative users can remove or modify saved data.
3. Usability
    3.1. User interface is graphical and is designed for non-technical users.

Environment

4. Compatibility with Perk Tutor and 3D Slicer
    4.1. The module is an extension to 3D Slicer.
    4.2. The module works with the Perk Tutor simulation training software.
5. Limited network and internet connectivity
    5.1. Data can be saved despite lack of Internet access.

Functional Design Requirements

1. Saving Files
   1.1. File location should be consistent and non-arbitrary.
   1.2. Files should be accessible to anyone who needs to access them.
   1.3. File location is recorded and this information is easy to find.
2. Managing study and trainee metadata
   2.1. A session is associated with one trainee ID and name in a *one-to-one* relationship.
   2.2. Trainees are associated with a study in a *many-to-many* relationship.
3. Copying data between computers
   3.1. Secure copy over a network with password-authentication.
4. Managing trainee feedback
   4.1. Trainee feedback is collected in digital form to minimize paperwork and transcribing errors.

3.3 Investigate Possible Solutions

To determine the ideal tool for data storage, an objective comparison of storage options was conducted by ranking each of the options on a scale of 1 to 5 in terms of how well each one satisfies the given criteria.

3.3.1 Comparison of Data Storage Options

The options considered for long-term data storage are: spreadsheets (Excel), comma-separated value files (CSV), XML files, SQLite (www.sqlite.org), MySQL (www.mysql.com), and learning management systems (for example: Moodle).

Both CSV and XML are widely used plain-text file formats which create human-readable files that can also be edited programmatically. XML is more verbose, but its tags are useful for indicating headings and subgroups.

SQLite and MySQL are SQL-based (standard query language) database tools. SQLite is more lightweight, server-less, and allows data to be stored locally, while MySQL is more traditional but one of the more commonly used data storage tools.

Web-based learning management systems (LMSs) in general are considered as an option. The features that would be required from any given LMS are its backend database for storing trainee data and files and a user interface for retrieving and editing data. In particular, they should provide the ability to upload files and download a list of trainees. These features are found in all of the most common LMSs (Moodle, Blackboard and D2L).

To decide on the most appropriate method for data storage, we compiled a set of criteria and considered how well each of the given options satisfied each of the criteria.

The following table (Table 3.2) outlines the subjective ratings determined by consensus with other software engineers in the lab. The ratings are a qualitative assessment of whether the storage format has the criteria as a built in feature or not.

If the option has a rating of 1 (and is colored in pink), this implies that the format does not have this feature by default, and non-trivial extra implementation work would be necessary in order to satisfy the criteria. On the other hand, a ranking of 5 implies that this criterion is already a feature of the storage tool by default. In the middle, with a rating of 3, are the formats which may be workable but are not ideal. Excel, as the status quo and current default format, was ranked in the middle for all of the categories.

Criteria	Excel	CSV	XML	SQLITE	MySQL	Moodle
GUI for editing data	3	1	1	1	1	5
Access controls	3	1	1	1	1	5

Add/edit data programmatically	3	5	5	5	5	3
Synchronization (offline, concurrent access)	3	1	1	1	1	1
Data integrity	3	3	5	5	5	5
Input/output for data mining	3	3	1	5	3	3
Accessibility for IGT tools	3	3	5	5	5	1
Open/platform-independent	3	5	5	5	5	5

Table 3.2: Data storage options and criteria ranking

All of the options explored had some drawbacks. In particular, a tool with offline synchronization, to provide a method of offline data access and synchronization when the system comes back online, was not readily available in any of the tools. Ultimately, although the database tools had advantages such as more complex data relationships and more granular access programmatically, we decided that learning management systems, due to their graphical web-based interfaces would be much more usable by less-technical end users. In addition, LMS student/instructor authentication access controls provide a significant advantage as security is a key requirement. For authentication, most formats permit basic read/write access control, while LMSs implements role-based permissions allowing for advanced control. Moodle, in particular, is currently used by the greater university population which ensures medical students, residents, and instructors can be added to it easily.

The two most complex requirements to satisfy were the ability to handle intermittent network access and the ability to reliably handle multiple concurrent users. Synchronizing local and file server data when network connection resumes, while maintaining data integrity, was not fully satisfied by any of the data storage options considered. For this, we propose a local cache, for storing a current copy of the Moodle data, which can be used when the training computer does not have internet access.

3.4 Design New Workflow

The following black box diagram (Figure 3.4) documents the input and output that the system must handle and provide, without focusing on the specific system implementation. It facilitated the design of the user interface, as it provides a visual representation of the variables that the interface must take into consideration.

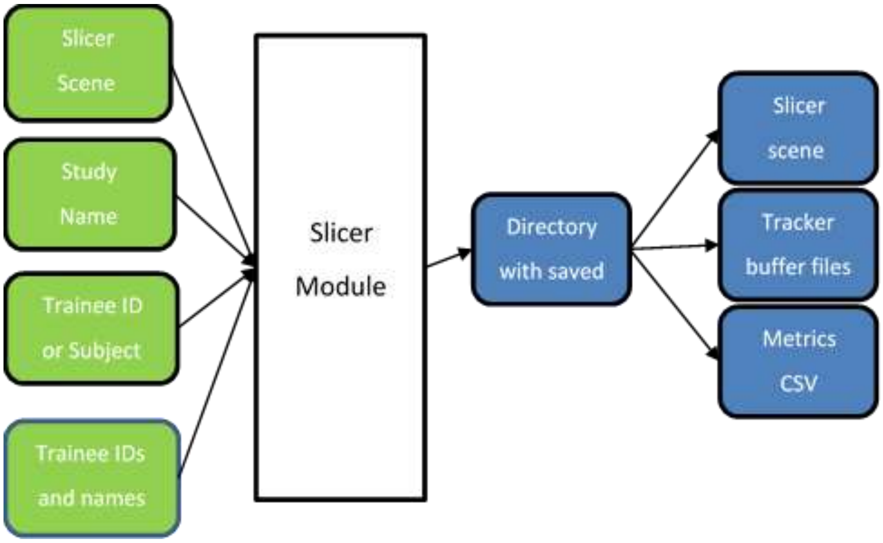


Figure 3.4: Black box diagram (depicting necessary input and output)

Sample Scenario for Data Collection:
Purpose: Study the effectiveness of needle injection practice with the Perk Tutor [1].

Number of participants: 40

Sessions per participant: 3 (Pre-testing, practice session, post-testing)

Metrics collected per session: 6

Data Collected

- Trainee ID data
- MRML Scene (one per session)
- Perk Tutor file (XML file with tracker data)
- Comments on trainee performance
- Objective results (set of metrics and values)

Using the diagrams of the current workflow (Figure 3.3) and the black box diagram of the data (Figure 3.4) handled by the proposed module as a starting point, we design and propose a new workflow that would ideally handle the issues identified and satisfy the given requirements. This was an iterative process, wherein potential users were consulted periodically to determine if the new workflow was practical and reasonable. Figure 3.5 shows the overall architecture of the proposed system and how the five different parts of the system are connected. The course administrator uses the Client computer to conduct the training sessions. The 3D Slicer software must be installed on this computer along with the Perk Tutor and Slicer IGT extensions. Additionally, PLUS Server (www.plusserver.com) software should be installed and running in order to access tracking data from hardware such as an ultrasound probe and a needle.

For each session, after the trainee performs the simulated procedure, the Perk Tutor saves data locally because it is not guaranteed to always be network-connected. Raw data files (usually large files) are uploaded for longer term storage to a secure file server over SFTP when network

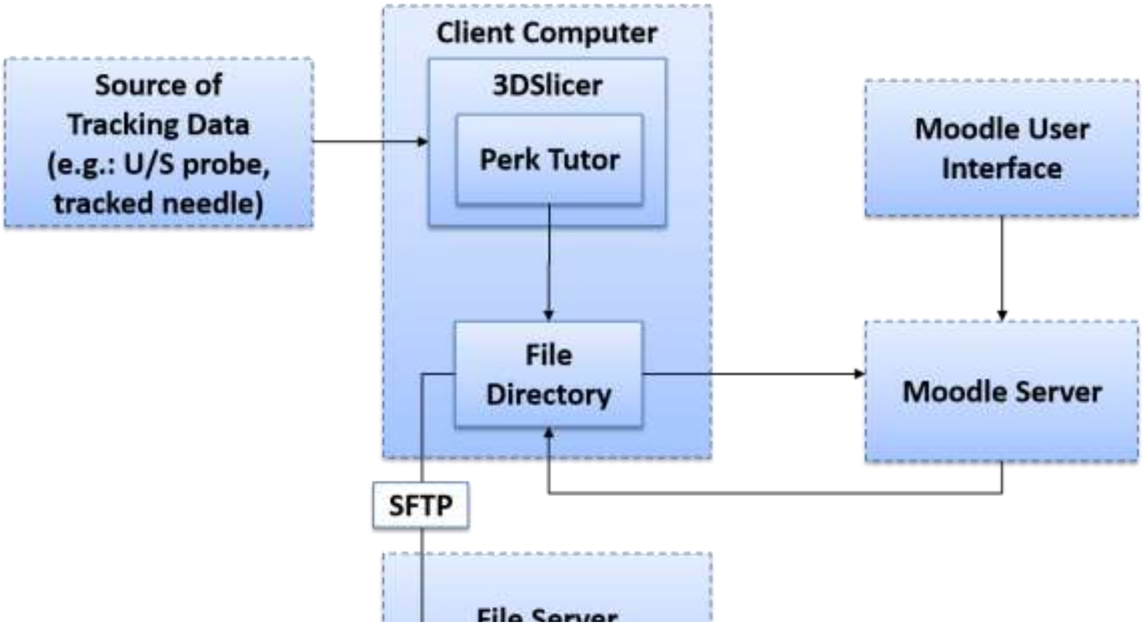


Figure 3.5 Overall architecture of the system

connectivity is available. Trainee data is uploaded and downloaded from the client computer to Moodle.

To provide more detail to the architecture diagram, a data flow diagram (Figure 3.6) was created using the “current workflow” diagram as a starting point.

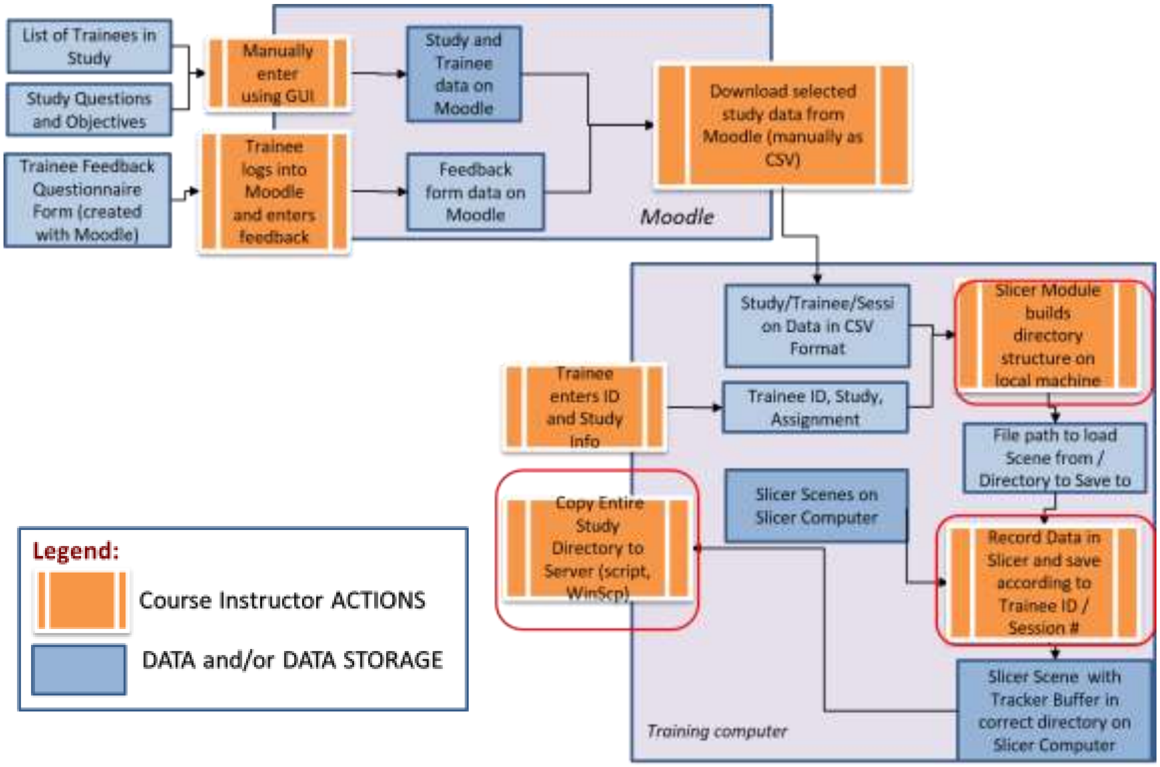

List of Trainees in Study
Study Questions and Objectives
Trainee Feedback Questionnaire Form (created with Moodle)
Manually enter using GUI
Trainee logs into Moodle and enters feedback
Study and Trainee data on Moodle
Feedback form data on Moodle
Download selected study data from Moodle (manually as CSV)
Moodle
Study/Trainee/Session Data in CSV Format
Slicer Module builds directory structure on local machine
Trainee enters ID and Study Info
Trainee ID, Study, Assignment
File path to load Scene from / Directory to Save to
Slicer Scenes on Slicer Computer
Copy Entire Study Directory to Server (script, WinScp)
Record Data in Slicer and save according to Trainee ID / Session #
Slicer Scene with Tracker Buffer in correct directory on Slicer Computer
Training computer
Legend:
Course Instructor ACTIONS
DATA and/or DATA STORAGE

Figure 3.6 Data flow diagram of Moodle and Client Computer

In Figure 3.6, we started with Figure 3.3 and removed the green cylinders that represent storage locations. Instead, we represent the storage location by the light blue boxes. The darker blue boxes still represent data types and orange boxes still represent an action or a process. The data flow was designed such that all processes could be either carried out manually or automated by software in the future. The orange boxes circled in red represent the areas of the workflow that we chose to implement as part of a new 3D Slicer module.

In general, the new workflow can be summarized as follows: The training software downloads and parses data from the online learning management tool Moodle. Then, the practice session occurs on the training computer and simulation data is recorded and automatically saved to the correct directory (on the training computer) based on session metadata. Finally, the saved data and metadata are uploaded to the password-protected File server and may later be downloaded for analysis (Figure 3.7).

Figure 3.7 is an extension of the first data flow diagram. After data is uploaded to the File server it can be downloaded by any authorized user for analysis. Since the files are packaged together (Slicer-specific file extension .mrb which does this), clicking on this file opens it in Slicer along with all the trainee and study attributes. Perk Evaluator can now be run to compute and save metrics.

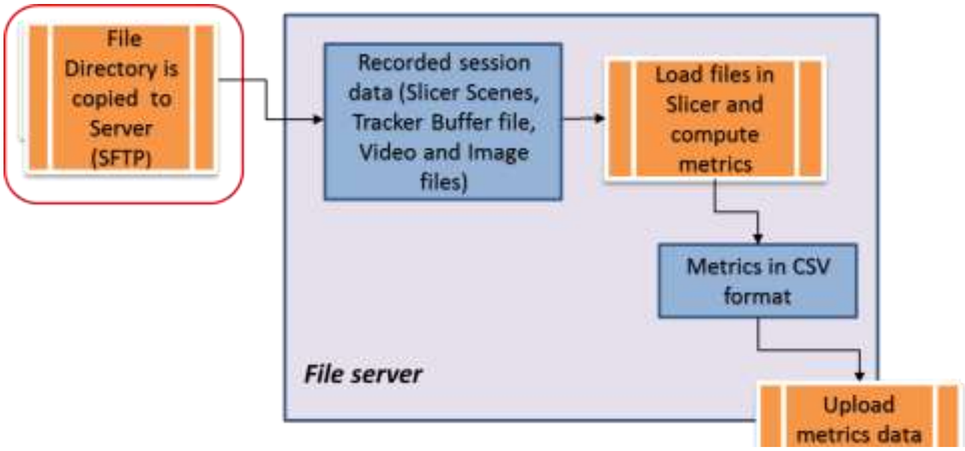

File Directory is copied to Server (SFTP)
Recorded session data (Slicer Scenes, Tracker Buffer file, Video and Image files)
Load files in Slicer and compute metrics
Metrics in CSV format
File server
Upload metrics data

Figure 3.7 Data flow for analyzing trainee performance on the File Server

The final stage of the workflow (Figure 3.8) is data mining and curriculum development. To do this, trainee performance (metrics data) is downloaded as spreadsheets. As this data is in CSV format, we can use any data mining software or graphing software to plot learning curves and make quantitative observations and conclusions.

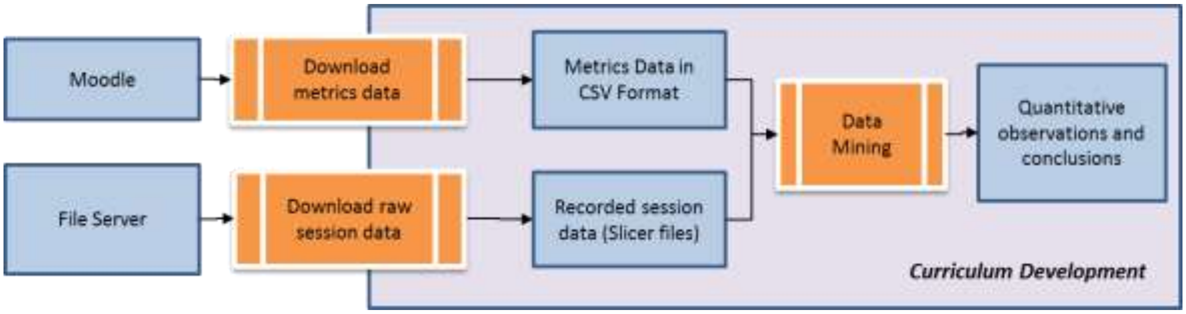

Moodle
Download metrics data
Metrics Data in CSV Format
File Server
Download raw session data
Recorded session data (Slicer files)
Data Mining
Quantitative observations and conclusions
Curriculum Development

Figure 3.8 Data flow for curriculum development. Download data from two sources.

Chapter 4

Implementation

The following chapter describes the implementation section of the project, including data storage, application logic and user interface. Using the system requirements and workflow design outlined in Chapter 3, the workflow was partially implemented as a Slicer module.

4.1 Development Environment

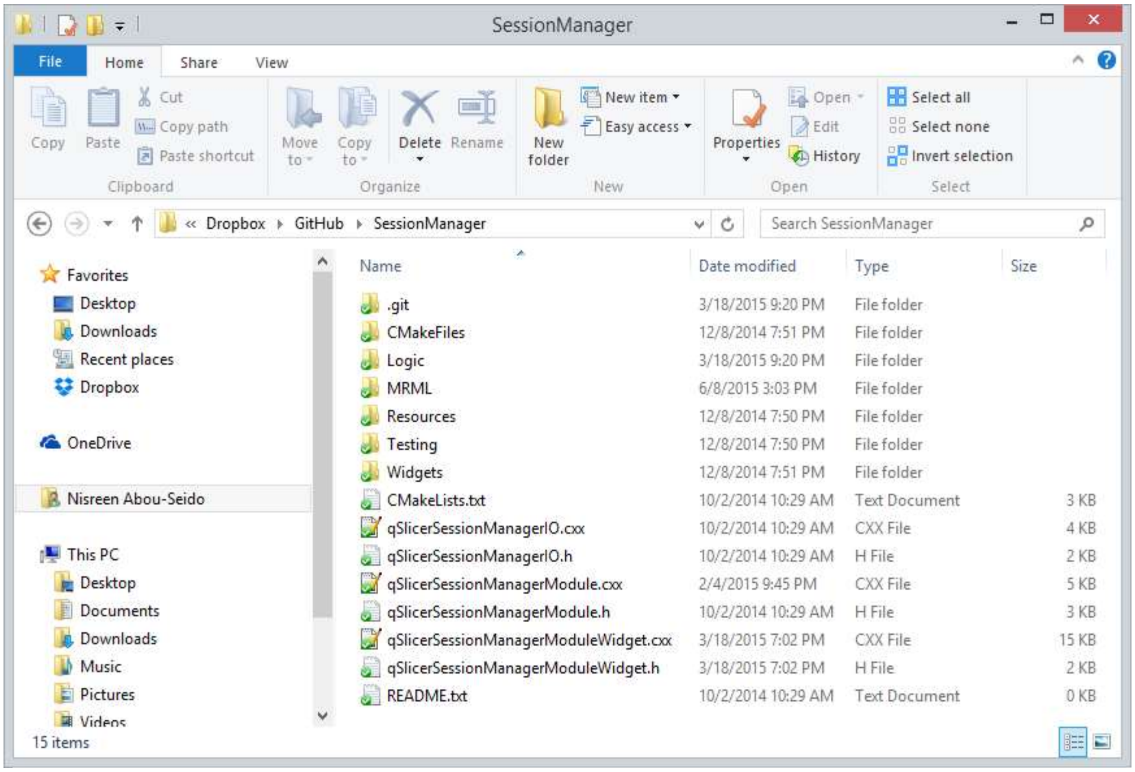
The primary outcome of this project was the design and implementation of a software extension for Perk Tutor simulation training. Therefore, the data management module was designed to be compatible with the Perk Tutor; a free, open-source, medical simulation platform (www.perktutor.org), developed primarily for image-guided intervention training [3]. Perk Tutor is based on the 3D Slicer (www.slicer.org) and Public Library for Ultrasound Research open-source platforms (www.plustoolkit.org) [4]. In Perk Tutor, a number of simulation-based training systems have been implemented as 3D Slicer plugins. Our data management module, hereafter referred to as the Session Manager, is combined with data analysis modules in the Perk Tutor extension, allowing for the implementation of a complete open-source simulation training tool that is freely available through the 3D Slicer extension manager.

To create an extension compatible with 3D Slicer, the online Slicer Wiki for Developers was consulted (www.slicer.org/slicerWIKI). In 3D Slicer, extensions can be one of three types: *command-line modules*, which have restricted access to the internal Slicer API; *scripted modules,* which have limited access to Slicer internals and are implemented in Python; and *loadable modules,* which have full access to Slicer internals, are implemented in C++, and can create subclasses of VTK/ITK/MRML classes to extent the C++ API.

The Session Manager was implemented as a loadable module so that it would be possible to create new data storage types, and later integrate it with the rest of the Perk Tutor extension with minimal limitations. As a C++ Loadable Module application, the Session Manager was developed in a Windows 7 environment using Visual Studio 2008 as seen in the following figure. However, it was designed to be cross-platform compatible, not Windows-specific, and as such it may be used within a Mac OS or UNIX environment as well.

4.1.1 C++ Loadable Module Extension Structure

In 3D Slicer, C++ Loadable Modules are expected to follow the Model-View-Controller pattern in order to systematically separate data storage, user interface and logic code. The Model is implemented as classes that extend vtkMRMLNode, the View as QT Widget classes, and the Controller as a set of Logic and IO classes. Figure 4.1 is a screenshot of the class files for the Session Manager, demonstrating the file structure of the module and how its classes are organized.



Name	Date modified	Type	Size
.git	3/18/2015 9:20 PM	File folder	
CMakeFiles	12/8/2014 7:51 PM	File folder	
Logic	3/18/2015 9:20 PM	File folder	
MRML	6/8/2015 3:03 PM	File folder	
Resources	12/8/2014 7:50 PM	File folder	
Testing	12/8/2014 7:50 PM	File folder	
Widgets	12/8/2014 7:51 PM	File folder	
CMakeLists.txt	10/2/2014 10:29 AM	Text Document	3 KB
qSlicerSessionManagerIO.cxx	10/2/2014 10:29 AM	CXX File	4 KB
qSlicerSessionManagerIO.h	10/2/2014 10:29 AM	H File	2 KB
qSlicerSessionManagerModule.cxx	2/4/2015 9:45 PM	CXX File	5 KB
qSlicerSessionManagerModule.h	10/2/2014 10:29 AM	H File	3 KB
qSlicerSessionManagerModuleWidget.cxx	3/18/2015 7:02 PM	CXX File	15 KB
qSlicerSessionManagerModuleWidget.h	3/18/2015 7:02 PM	H File	2 KB
README.txt	10/2/2014 10:29 AM	Text Document	0 KB

Figure 4.1: Module file structure with Logic, MRML, Resources, Testing and Widget subdirectories

Features of Module

The primary purpose of the module was to automate some of the steps outlined in the data flow diagram. The following are the processes that were automated through the implementation of a data management extension to 3D Slicer:

1) Create directory hierarchy on local machine
2) Record data in Slicer and save according to Trainee ID and Session ID.
3) Copy entire directory from local computer to server

4.1.2 Create directory hierarchy on local machine

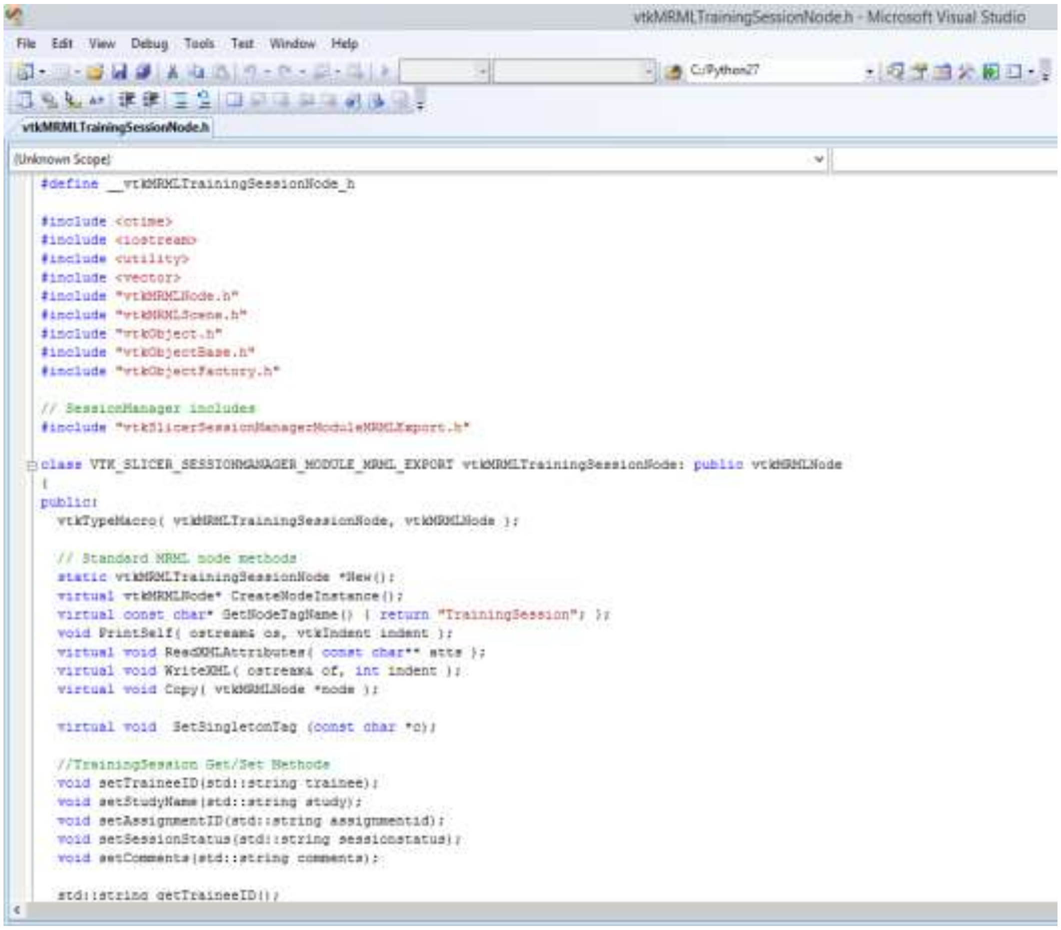
The structure of a course on the Moodle learning management system is such that each course has an instructor, other administrators and a list of students. The list of students and their grades can be accessed by any individual with administrator privileges. An administrator then has permission to both view the student list and download it as a plain-text CSV (comma-separated value) file containing all student names and their student numbers. Moodle also provides the ability to download the data in other formats, such as XML or Excel spreadsheet.

Our implementation reads and parses a given CSV file to create a directory structure on the local file system where the top level folder is the name of the course (in this thesis referred to as a “study”). Each study folder then contains a subdirectory for each of the students (also referred to as “trainees” or “subjects” in the context of this thesis). Therefore, when data is saved for a particular trainee and study, it is in a standard, easily-searchable, location.

4.1.3 Record and save data according to Trainee and Session metadata

As described earlier, a training session is a single instance of a trainee performing a given task and then saving the results of their work. The data and metadata involved in a training session includes: the trainee ID, the study name or ID, the date, whether the session was completed or not, where the session should be saved and, of course, the actual simulation data points and metrics (in Perk Tutor, this is the Tracker Buffer XML file).

In order to preserve the Model-View-Controller pattern required in Slicer modules, a new Model class was created to capture this data. The new class, vtkMRMLTrainingSessionNode, is a subclass of a vtkMRMLNode. When information about the new training session is collected from the user, a new node is created is added to the Slicer scene. Then, when the user clicks “save” this node is saved with the Slicer scene in the appropriate local subdirectory. Therefore, when the Slicer scene is reloaded in a new Slicer session, the metadata also loads in the form of this vtkMRMLNode.


vtkMRMLTrainingSessionNode.h - Microsoft Visual Studio
File Edit View Debug Tools Test Window Help
C:/Python27
vtkMRMLTrainingSessionNode.h
(Unknown Scope)

```
#define __vtkMRMLTrainingSessionNode_h

#include <ctime>
#include <iostream>
#include <utility>
#include <vector>
#include "vtkMRMLNode.h"
#include "vtkMRMLScene.h"
#include "vtkObject.h"
#include "vtkObjectBase.h"
#include "vtkObjectFactory.h"

// SessionManager includes
#include "vtkSlicerSessionManagerModuleMRMLExport.h"

class VTK_SLICER_SESSIONMANAGER_MODULE_MRML_EXPORT vtkMRMLTrainingSessionNode: public vtkMRMLNode
{
public:
  vtkTypeMacro( vtkMRMLTrainingSessionNode, vtkMRMLNode );

  // Standard MRML node methods
  static vtkMRMLTrainingSessionNode *New();
  virtual vtkMRMLNode* CreateNodeInstance();
  virtual const char* GetNodeTagName() { return "TrainingSession"; };
  void PrintSelf( ostream& os, vtkIndent indent );
  virtual void ReadXMLAttributes( const char** atts );
  virtual void WriteXML( ostream& of, int indent );
  virtual void Copy( vtkMRMLNode *node );

  virtual void  SetSingletonTag (const char *c);

  //TrainingSession Get/Set Methods
  void setTraineeID(std::string trainee);
  void setStudyName(std::string study);
  void setAssignmentID(std::string assignmentid);
  void setSessionStatus(std::string sessionstatus);
  void setComments(std::string comments);

  std::string getTraineeID();
```

Figure 4.2: Screenshot of vtkMRLTrainingSessionNode header file.

4.1.4 Copy directory from local computer to remote server

To move files between the local directory, files are transferred between machines when network-connectivity is available by running a command-line batch script. The script uses the WinSCP application to automatically synchronize the local directory with the corresponding directory on the file server using SFTP (SSH file transfer protocol). SFTP is secure, requiring username and password in order to access the server over SSH, and is therefore used instead of FTP (file transfer protocol).

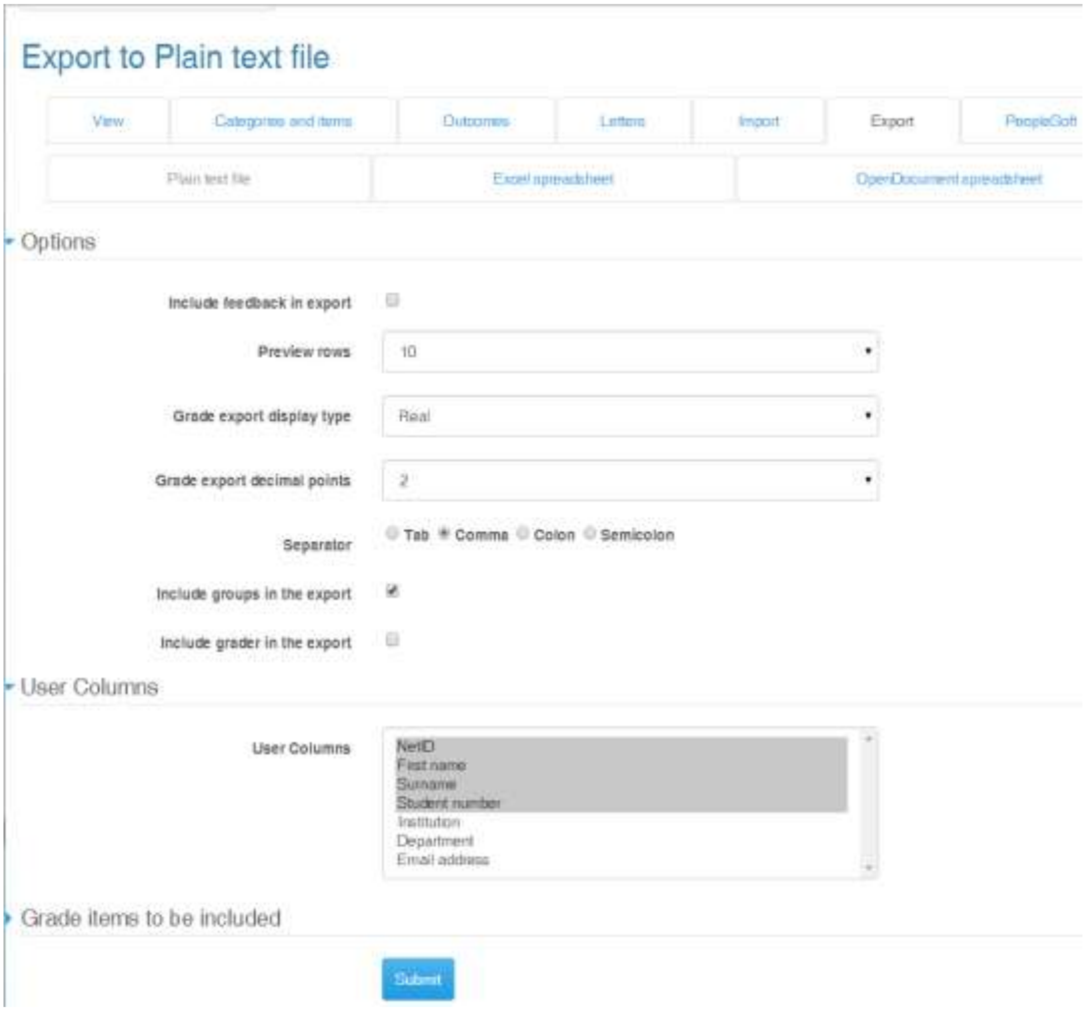
4.2 Data Storage

4.2.1 Moodle

As discussed in Chapter 3, Moodle was chosen as the framework for managing trainees, courses, and for long term storage of metadata. Moodle is a Learning Management System used extensively by many universities for managing courses, assignments and exams, including confidential materials, in a manner that provides two-way protection against tempering by trainees and unauthorized persons. It is an ideal platform for sharing material, evaluating students and managing grades.

It is also an open-source, no cost, configurable system that makes it easy to set up a server for a small department of lab regardless of whether it is used at the given institution. Its key advantages over other options are the built-in authentication for controlling access to data and the graphical web interface for easily viewing, editing and downloading data. For example, a list of student grades and other student data can be easily exported in a number of formats through the web interface. This is demonstrated in Figure 4.3, which is a screenshot of the web page for exporting “Grades” data as a plain text file.

One of the limitations of Moodle is its lack of support for offline use, synchronization with the training computer (which does not always have network connectivity) needs to be implemented independently. Another limitation is that Moodle does not, by default, support third-party access and thus is not directly compatible with Slicer IGT tools. That is, we can’t access Moodle from Slicer. Moodle is not a standard file format or database and there are no tools to read and write directly from it. The workaround that was implemented for this was the implementation of a local cache that stores data in a supported format which the training software will read and write to. To use a learning management system like Moodle in the context of surgical training studies, the trainee is a “student”, and the supervisor or administrator is the “instructor”. A “course” is created for each new study.



Export to Plain text file

Options

Include feedback in export □

Preview rows 10

Grade export display type Real

Grade export decimal points 2

Separator ○ Tab ● Comma ○ Colon ○ Semicolon

Include groups in the export ☑

Include grader in the export □

User Columns

User Columns: NetID, First name, Surname, Student number (selected); Institution, Department, Email address

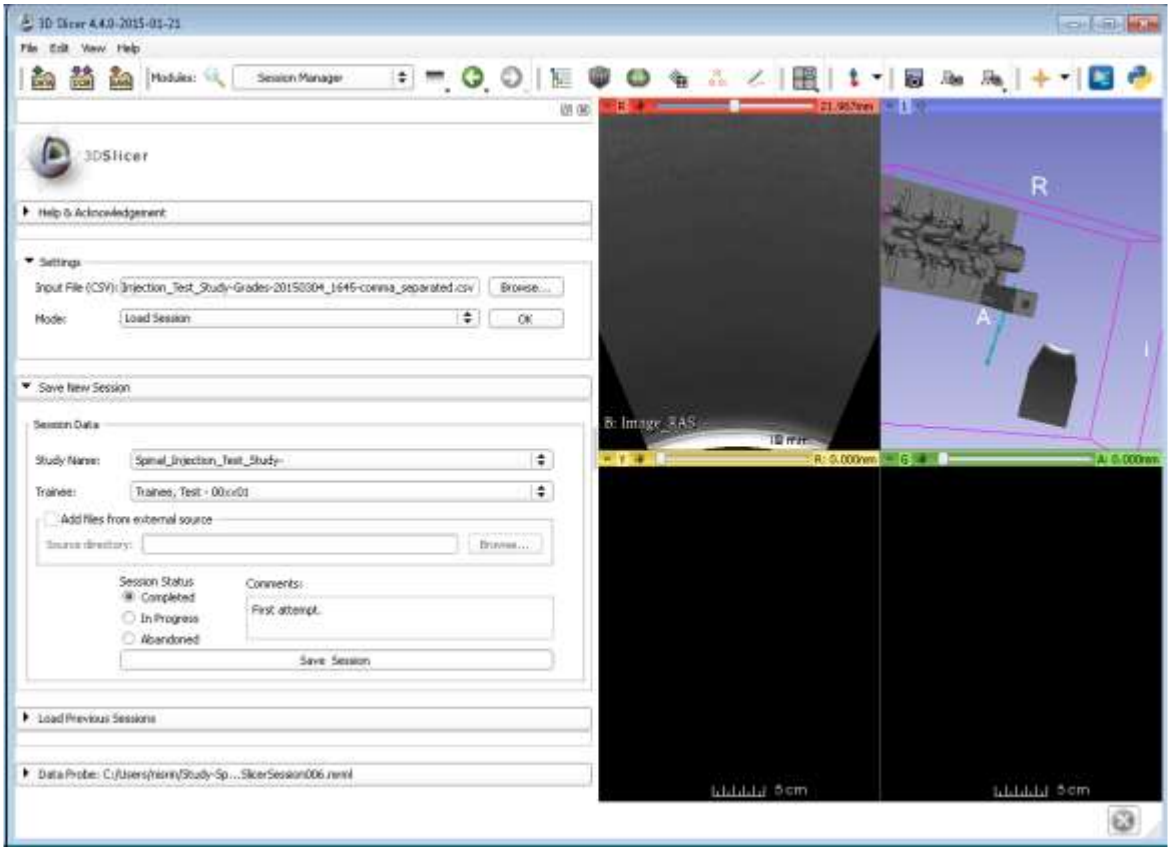
Grade items to be included

Figure 4.3: Downloading the trainee list CSV from Moodle (web interface)

4.3 User Interface

The user interface was designed to have the same look and feel of the current modules and extensions in Slicer. It was created iteratively using QT in QT Designer.

Save Session



3D Slicer 4.4.0-2015-01-21

Modules: Session Manager

Settings

Input File (CSV): Injection_Test_Study-Grades-20150304_1645-comma_separated.csv

Mode: Load Session

Save New Session

Session Data

Study Name: Spinal_Injection_Test_Study-

Trainee: Trainee, Test - 00cc01

□ Add files from external source

Source directory:

Session Status

● Completed

○ In Progress

○ Abandoned

Comments: First attempt.

Data Probe: C:/Users/nisrin/Study-Sp…SlicerSession006.mrml

Figure 4.4: The Save Session section of the Session Manager User Interface.

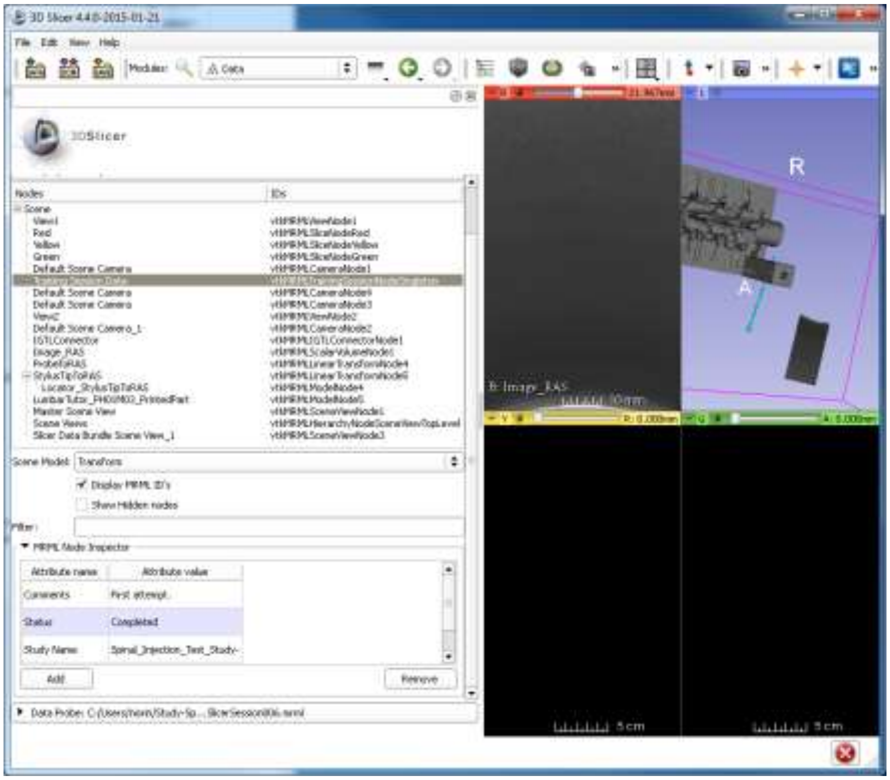
After the CSV has been loaded, the populated drop down boxes for study name and trainee names are populated and therefore allow the user (the instructor) to select from possible options, minimizing the likelihood of typos, incorrect spelling, and other errors. Other features of the save process include the session status option and the comment section for adding other metadata in text format. The session status can be either Complete, Incomplete or Abandoned helping the course administrator to avoid evaluating incomplete sessions.

When the user clicks on “Save Session”, the module displays the name of the new directory where all data will be saved to confirm file location. The scene is saved as a MRML file. Other objects in the workspace are packaged with the scene and saved as an “.mrb” file, which is a Slicer format that packages files together. Possible data types that may be saved include Transforms, Transform Buffer, MRMLNode, and Scenes. Opening the ".mrb" file later will restore the Slicer workspace as it was saved. Renaming the .mrb file to a .zip file and unpacking it is also an option in order to see all saved files individually.

Training Session Data Node (viewable in 3D Slicer Data Module)

The Training Session Node is added to the MRML scene. The format of a Slicer MRML file is XML. If the MRML file is opened in a text editor, Training Session data can be found amongst other scene data. Therefore, when the user reloads the scene, the Training Session Node attributes are reloaded as well. The following code is an example of the attributes as saved in the XML file:

```
<TrainingSession id="vtkMRMLTrainingSessionNodeSingleton"  name="Training Session Data"
hideFromEditors="false"  selectable="true"  selected="false"  attributes="Comments:First
attempt.;Status:Completed;Study Name:Spinal_Injection_Test_Study-;TraineeID:00xx01" ></TrainingSession>
```



Nodes	IDs
Scene	
View1	vtkMRMLViewNode1
Red	vtkMRMLSliceNodeRed
Yellow	vtkMRMLSliceNodeYellow
Green	vtkMRMLSliceNodeGreen
Default Scene Camera	vtkMRMLCameraNode1
Training Session Data	vtkMRMLTrainingSessionNodeSingleton
Default Scene Camera	vtkMRMLCameraNode4
Default Scene Camera	vtkMRMLCameraNode3
View2	vtkMRMLViewNode2
Default Scene Camera_1	vtkMRMLCameraNode2
IGTLConnector	vtkMRMLIGTLConnectorNode1
Image_RAS	vtkMRMLScalarVolumeNode1
ProbeToRAS	vtkMRMLLinearTransformNode4
StylusTipToRAS	vtkMRMLLinearTransformNode6
Locator_StylusTipToRAS	vtkMRMLModelNode4
LumbarTutor_PH00M02_PrintedPart	vtkMRMLModelNode5
Master Scene View	vtkMRMLSceneViewNode1
Scene Views	vtkMRMLHierarchyNodeSceneViewTopLevel
Slicer Data Bundle Scene View_1	vtkMRMLSceneViewNode3

Scene Model: Transform

☑ Display MRML ID's

□ Show Hidden nodes

Filter:

MRML Node Inspector

Attribute name	Attribute value
Comments	First attempt.
Status	Completed
Study Name	Spinal_Injection_Test_Study-

Add   Remove

Data Probe: C:/Users/hwm/Study-Sp... SlicerSession006.mrml

Figure 4.5 Screenshot of the Data attributes for a Training Session Data Node

Load Previous Session

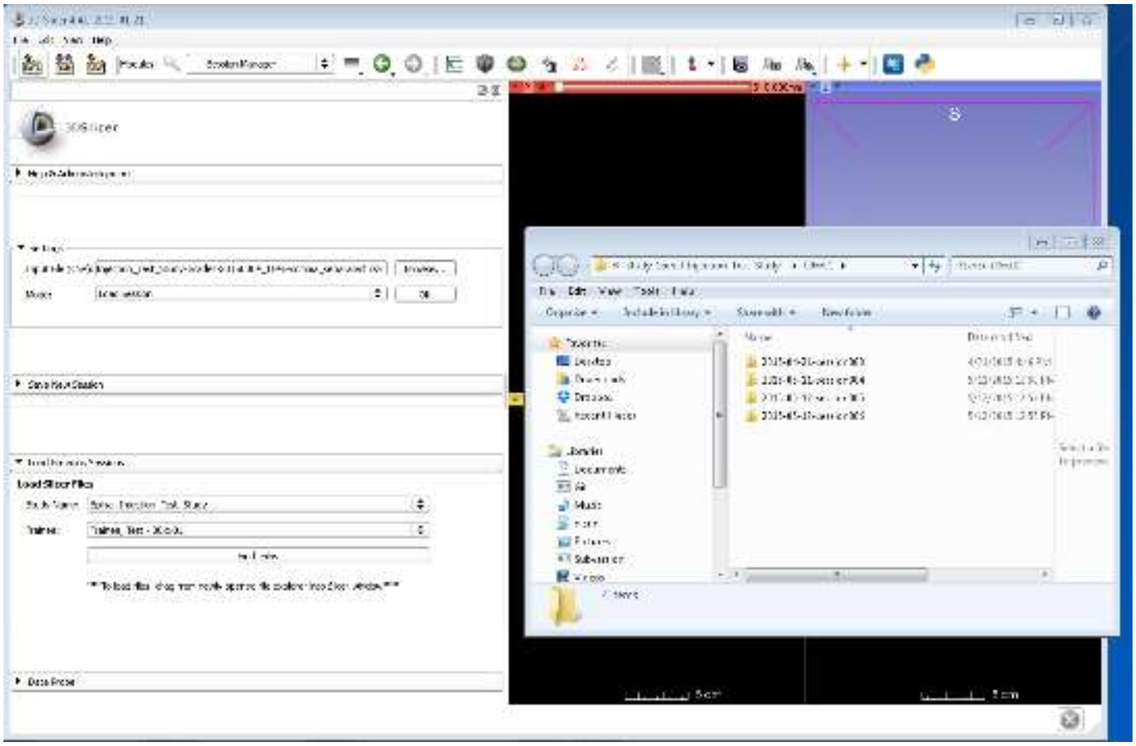


Figure 4.6: Screenshot of loading a session using the Session Manager

In the Load Session section of the user interface, the Trainee dropdown box is populated based on the CSV file selected in the Settings. When the user clicks Find Files, the Windows File Explorer opens at the directory where the sessions for the selected trainee should be saved. The user can now drag and drop the appropriate file into the Slicer window to load it.

Overview of how to use the Session Manager

For each study/training program:

1) Create a Moodle course
2) Add trainees to the course. Each trainee is a “student”
3) Export student data as a CSV file

For each session:

1) Open 3D Slicer
2) Open Session Manager module
3) Import CSV file of student/trainee/subject names and IDs
4) If new session: import Slicer file necessary for experiment
    a. Trainee performs experiment
    b. Enter trainee metadata and click “Save Session”.
5) If loading prior session: import file using the Load Session feature

4.4 Discussion

For the system analysis and design (Chapter 3, Figure 3.1) we outlined a set of steps to follow in order to methodically design a usable system. The first two stages, determining requirements, identifying areas of improvement, and exploring possible solutions -- were successfully carried out and described in this document. The final stages, implementation and testing, were not fully carried out, as the scope of this project was to document and design a new workflow and architecture. Informally, however, frequent consultations with users of the Perk Tutor allowed for minor adjustments to be made while implementing the prototype.

One new requirement that was implemented was a suggestion to add a comment box in the user interface, for the trainee or instructor to add qualitative comments as part of the metadata stored with the session. This was added to the save session user interface of the prototype.

In the latter stages of the project, Queen's University announced that it would be transitioning to D2L (www.d2l.org), another learning management system, next year. Thus, a new requirement of the workflow was to ensure limited inter-dependency with Moodle in order to facilitate the transition. The workflow was therefore designed to be compatible with any LMS that has the following basic course management features:

1) User interface for entering student information and grades
2) Ability to download list of students and their grades as a plain-text CSV file
3) User interface for uploading files and associating them with a particular student
4) Different privileges for different user roles (student, instructor, teaching assistant) and password-protection for data confidentiality.

The new workflow and prototype successfully separates Moodle-specific tasks from the Perk Tutor module so that making the transition would not require any changes to the code.

4.5 Documentation and Deployment

The documentation in this thesis allows future developers to understand design and implementation decisions. In addition, the code is open-source and well-commented to facilitate modifications. The code repository is on GitHub at www.github.com/13nas2/SesseionManager. For deployment, after testing, the module will be packaged with Perk Tutor and available for download through the 3D Slicer Extension Manager.

Chapter 5

Summary and Future Work

5.1 Summary

This thesis found that the current data management methods in simulation-based education are not efficient and that improving the quality of surgical simulation education is necessary in order to implement competency-based medical education.

Our contribution was to methodically document a new workflow and architecture for data management that would eliminate some of the current issues. The criteria and requirements for the designed workflow, architecture, and prototype were determined in consultation with Perk Tutor users. The new workflow eliminated each of the limitations of the training platform outlined in section 3.1.2. The new workflow provides the following features and advantages over the previous status quo:

1) Slicer session data is packaged together and saved automatically based on metadata.
2) Trainee data, course content and trainee feedback is stored on a secure online learning management system.
3) Session attributes and metadata are saved along with the raw hand and instrument motion data and calculated metrics.

The list of requirements identified in section 3.2 were fulfilled in the design and, by extension, in the prototype module. Thus, the resulting workflow design and architecture facilitated the implementation of an initial working prototype compatible with the Perk Tutor training platform and integrated with an online learning management platform. With further implementation it may be used in ultrasound-guided needle placement training conducted using the Perk Tutor.

5.2 Future Work

The new workflow and module sets the groundwork for multiple avenues of future work. The first stage would be to test the prototype and make any necessary changes in order to allow it to be used in courses and workshops for trainee management. Testing should include using the module in real-life scenarios, using a large number of trainees of various skill levels, and multiple instructors/administrators. In addition, it is necessary to test the system with the new learning management system after the transition.

Our workflow design designates some steps in the process to be carried out manually. Ideally, more of these steps would be automated. For example, syncing the local and remote repositories requires that the instructor or an administrator run the sync script. It would be an improvement to explore automating this process and integrating it into the 3D Slicer module. Future work could also include implementing a system of authentication in which authentication details are sent to the LMS to allow trainees to practice independently. However, for this to be feasible, consistent internet access at the simulation center would be necessary.

The groundwork is set for using the module for significant data collection. In our module, we collected trainee ID, metrics values and text feedback. It is straightforward, using QT, to add more user interface widgets to collect any other necessary attributes. For example, one could implement a widget for noting how many times a trainee attempted a given simulation.

For data mining and curriculum development, some more implementation work may be necessary so that data collected can be uploaded to data mining software in order to explore trends in learning curves and ultimately improve surgical training.

Finally, the proposed workflow also serves as a framework which could be adapted by other researchers at other institutions to work with surgical simulation and training applications other than Perk Tutor.

Bibliography

[1] C. T. Yeo, T. Ungi, P. U-Thainual, A. Lasso, R. C. McGraw, and G. Fichtinger, "The effect of augmented reality training on percutaneous needle placement in spinal facet joint injections.," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 7, pp. 2031–2037, Jul. 2011.

[2] J. D. Michelson, L. Manning, M. D, L. Manning, and M. B. A, "Competency assessment in simulation-based procedural education.," *Am. J. Surg.*, vol. 196, no. 4, pp. 609–615, Oct. 2008.

[3] M. Michael, H. Abboudi, J. Ker, M. Shamim Khan, P. Dasgupta, and K. Ahmed, "Performance of technology-driven simulators for medical students--a systematic review.," *J. Surg. Res.*, vol. 192, no. 2, pp. 531–43, Dec. 2014.

[4] D. Clinkard, C. Davison, R. Mcgraw, E. Moult, M. Holden, C. Davison, T. Ungi, G. Fichtinger, and R. Mcgraw, "Assessment of Lumbar Puncture Skill in Experts and Nonexperts Using Checklists and Quantitative Tracking of Needle Trajectories : Implications for Competency-Based," *Teach. Learn. Med.*, vol. 27, no. 1, pp. 51–56, 2015.

[5] R. J. Scalese, V. T. Obeso, and S. B. Issenberg, "Simulation technology for skills training and competency assessment in medical education," *J. Gen. Intern. Med.*, vol. 23, no. 1 SUPPL., pp. 46–49, 2008.

[6] J. R. Frank, R. Mungroo, Y. Ahmad, M. Wang, S. De Rossi, T. Horsley, S. D. E. Rossi, and T. Horsley, "Toward a definition of competency-based education in medicine: a systematic review of published definitions.," *Med. Teach.*, vol. 32, no. 8, pp. 631–637, 2010.

[7] "Royal College of Physicians and Surgeons of Canada." [Online]. Available: www.royalcollege.ca.

[8] A. G. Gallagher, E. M. Ritter, H. Champion, G. Higgins, M. P. Fried, G. Moses, C. D. Smith, and R. M. Satava, "Virtual reality simulation for the operating room: proficiency-based training as a paradigm shift in surgical skills training.," *Ann. Surg.*, vol. 241, no. 2, pp. 364–372, 2005.

[9] L. Bartha, A. Lasso, C. Pinter, T. Ungi, Z. Keri, and G. Fichtinger, "Open-source surface mesh-based ultrasound-guided spinal intervention simulator," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 8, pp. 1043–1051, 2013.

[10] J. Stunt, P. Wulms, G. Kerkhoffs, J. Dankelman, C. van Dijk, and G. Tuitjthof, "How valid are commercially available medical simulators ?," *Adv. Med. Educ. Pract.*, pp. 385–395, 2014.

[11] K. Fairhurst, A. Strickland, and G. Maddern, “The LapSim virtual reality simulator: Promising but not yet proven,” *Surg. Endosc. Other Interv. Tech.*, vol. 25, no. 2, pp. 343–355, 2011.

[12] A. Alwaal, T. Al-Qaoud, R. Haddad, T. Alzahrani, J. Delisle, and M. Anidjar, “Transfer of skills on LapSim virtual reality laparoscopic simulator into the operating room in urology,” *Urol. Ann.*, vol. 7, no. 2, p. 172, 2015.

[13] I. D. Ayodeji, M. Schijven, J. Jakimowicz, and J. W. Greve, “Face validation of the Simbionix LAP Mentor virtual reality training module and its applicability in the surgical curriculum,” *Surg. Endosc. Other Interv. Tech.*, vol. 21, no. 9, pp. 1641–1649, 2007.

[14] A. Zhang, M. Hünerbein, Y. Dai, P. M. Schlag, and S. Beller, “Construct validity testing of a laparoscopic surgery simulator (Lap Mentor®): Evaluation of surgical skill with a virtual laparoscopic training simulator,” *Surg. Endosc. Other Interv. Tech.*, vol. 22, no. 6, pp. 1440–1444, 2008.

[15] E. M. McDougall, S. B. Kolla, R. T. Santos, J. M. Gan, G. N. Box, M. K. Louie, A. J. R. Gamboa, A. G. Kaplan, R. M. Moskowitz, L. a. Andrade, D. W. Skarecky, K. E. Osann, and R. V. Clayman, “Preliminary Study of Virtual Reality and Model Simulation for Learning Laparoscopic Suturing Skills,” *J. Urol.*, vol. 182, no. 3, pp. 1018–1025, 2009.

[16] A. R. Parks, P. Atkinson, G. Verheul, and D. Leblanc-Duchin, “Can medical learners achieve point-of-care ultrasound competency using a high-fidelity ultrasound simulator?: a pilot study.,” *Crit. Ultrasound J.*, vol. 5, no. 1, p. 9, 2013.

[17] T. Ungi, D. Sargent, E. Moult, A. Lasso, C. Pinter, R. C. McGraw, and G. Fichtinger, “Perk tutor: An open-source training platform for ultrasound-guided needle insertions,” *IEEE Trans. Biomed. Eng.*, vol. 59, no. 12, pp. 3475–3481, 2012.

[18] E. R. Petrusa and R. J. Scalese, “A critical review of simulation-based medical education research : 2003 – 2009,” *Med. Educ.*, no. 44, pp. 50–63, 2010.

[19] S. B. Issenberg, W. C. McGaghie, E. R. Petrusa, D. Lee Gordon, and R. J. Scalese, “Features and uses of high-fidelity medical simulations that lead to effective learning: a BEME systematic review.,” *Med. Teach.*, vol. 27, no. 1, pp. 10–28, 2005.

[20] D. a. McClusky and C. D. Smith, “Design and development of a surgical skills simulation curriculum,” *World J. Surg.*, vol. 32, no. 2, pp. 171–181, 2008.

[21] M. Vardaki, H. Papageorgiou, and F. Pentaris, “A statistical metadata model for clinical trials’ data management,” *Comput. Methods Programs Biomed.*, vol. 95, no. 2, pp. 129–145, 2009.

[22] E. P. White, E. Baldridge, Z. T. Brym, K. J. Locey, D. J. Mcglinn, and S. R. Supp, “Special Issue — Data Sharing in Ecology and Evolution,” vol. 6, no. April, pp. 1–10, 2013.

[23] J. Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, *Non-functional requirements in software engineering*. 2012.

[24] E. Moult, T. Ungi, M. Welch, J. Lu, R. C. McGraw, and G. Fichtinger, “Ultrasound-guided facet joint injection training using Perk Tutor,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 8, pp. 831–836, 2013.