A technological platform to support education in Regional Anaesthesia with patient-specific virtual physiological human (VPH)-based models

G. Christodoulakis[1], K. Marias[1], G. Notas[2], N. Kampanis[2], and S. Sfakianakis[1]

[1] Computational BioMedicine Laboratory, FORTH-ICS, Heraklion, Greece
[2] Numerical Analysis and Scientific Computing Group, FORTH-IACM, Heraklion, Greece

Abstract— **RASimAs (Regional Anaesthesia Simulator and Assistant) is a EU FP7 project that aims at increasing the application, the effectiveness and the success rates of regional anaesthesia by developing two independent but complementary systems, one system for training by using patient-specific computer models, and one for guidance in the assistance of nerve's location during the actual intervention. In this context, the present document focuses on the training system, which will be deployed in multiple participating hospitals that will be connected to a central information system. In particular, this paper deals with the software architecture of the aforementioned integrated environment and the components that constitute it. We present indicative key components and functionalities such as the user authentication and authorization service, the user profile and performance metrics management service, the role based access control system, the VPH (Virtual Physiological Human) library, and the synchronization between the training centres and the central information system.**

Keywords— **software architectures, integration, distributed systems**

I. Introduction

Regional anaesthesia (RA) has been used increasingly during the past four decades. RA has several advantages over classic anaesthesia practices including reduced postoperative pain, earlier patient mobilization, shorter hospital stay, and significantly lower costs. RA is performed by local injection of anaesthetic in the proximity of a peripheral nerve. Clinically this is achieved either with the visualization of the injection needle close to the peripheral nerve with the use of ultrasound, or with the induction of muscle contraction when a needle tip connected to an electric nerve stimulator is very close to the nerve. Furthermore, these two techniques can be combined. However, it is a delicate technique and requires significant theoretical, practical, and non-cognitive skills in order for trainees to achieve confidence in performing regional anaesthesia and to keep complications to a minimum. Current training methods for regional anaesthesia include cadavers, video teaching, ultrasound guidance, and simple virtual patient modelling. These techniques have limited capabilities and do not consider individual anatomy. However, the Virtual Physiological Human (VPH) creates the possibility to generate patient-specific computer models that can be used for RA training sessions [1].

[1] http://www.rasimas.com/

The RASimAs EU project[1] aims to provide a training environment (Fig. 1) that, through the development of patient-specific VPH models for anaesthesia, will increase the application, the effectiveness and the success rates of RA, leading to broader clinical use of this method. Research results from the project group in advanced physics and soft tissue deformation modelling will contribute to a unique haptics enabled simulation prototype for real-life-like ultrasound- or electrical nerve stimulation-guided RA in different parts of the body. The prototype will be evaluated within the project in controlled clinical trials targeting both expert and novice trainees.

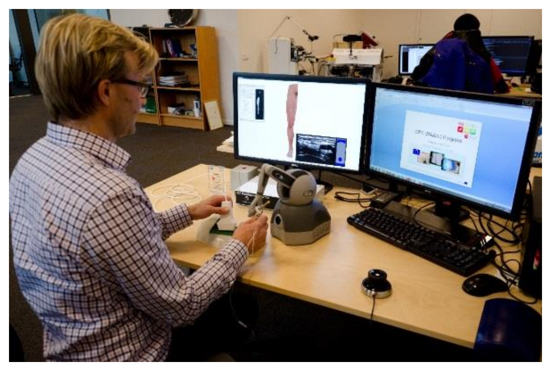

Fig. 1 The RASimAs training environment. The user interacts through the haptic devices with 3D patient specific models to perform a simulated RA operation [Photo courtesy of SenseGraphics[2]]

II. System description and requirements

The RASimAs project aims to develop a training environment for regional anaesthesia. Fig. 2 shows the RASimAs training environment, also referred to as RASim (Regional Anaesthesia Simulator) where the user interacts with the system through a workbench client computer. The system consists of three components, namely the input interface, the output interface, and the core simulation module.

[2] http://sensegraphics.com/

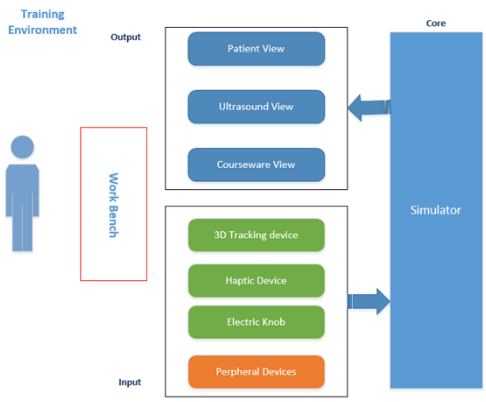

Fig. 2 Simulator training environment system overview [Image courtesy of SenseGraphics[2]]

The core simulation module is responsible for metadata exchange between the other modules and simulates information such as the scene being rendered, the needle position, the probe position, the knob status and haptic feedbacks.

The input interface connects the input devices that include the haptic apparatuses (simulating the needle), the 3D tracking devices (simulating a dummy ultrasound probe) and other devices like a rotating knob used for adjusting the current during the simulated electrical stimulation.

The output interface is responsible for showing updated views to users. The output views are divided into three panels. The first one is the patient view, which displays the scene being rendered including the virtual patient and the needle. The second one is the ultrasound screen displaying in real time the ultrasound image as the trainee moves the ultrasound tracker probe. The third view is the course-ware view, which is the main component that connects the simulator with the server side database and resources. The course-ware view displays the users' authentication/profile data, available scenarios, training material, users training history (performance metrics, training time), etc.

The application as a whole consists of a large distributed system. This is due to the fact that RA training facilities can be dispersed in multiple hospitals and universities all over Europe. Synchronization between every local training environment and a central information server will allow uploading of trainees' profile and history to the central system, and downloading of files required by the course-ware (i.e. training material, VPH models) to all local workstations. Furthermore, the trainees may have access to their profile and training history from a location outside their training environment. Thus, the need for development of a local and a central information service that will provide all the aforementioned functionalities to the course-ware module and, complemented to that with user access outside the training environment becomes apparent.

At a high level, as shown in Fig. 3, a researcher or an administrator of the system can build new training material by integrating different VPH models and uploading them to a repository located in a central administration point of the system. The training material can then be automatically synchronized to the multiple training environments where the trainees will have access to the needed infrastructure (e.g. haptic devices, ultrasound monitors, etc.). At these local installations, training material such as videos, guides, 3D scenes, model related resources, etc. will be used in the "courseware" module of the system and will be integrated with the training infrastructure and hardware.

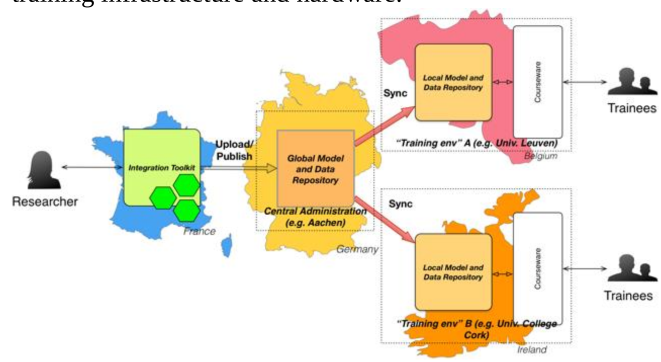

Fig. 3 The distributed nature of the RA simulator system

III. System Design

A. System context and user roles

In the simulator environment we have two important groups of users: the trainees and the mentors. The trainees are the users who participate in the simulation in order to initiate, maximize and maintain their RA practicing abilities, while the mentors are experienced anaesthesiologists supervising the training sessions. In this context, Fig. 4 shows the user visible interactions with the system. Basically the trainees participate in the Training use case whereas mentors are able to monitor and have access to the training statistics. The users of the system also include other groups such as administrators, consultants, etc. that have more specific roles. All users should be authenticated as legitimate users of the system and authorized based on their credentials and the granted access rights.

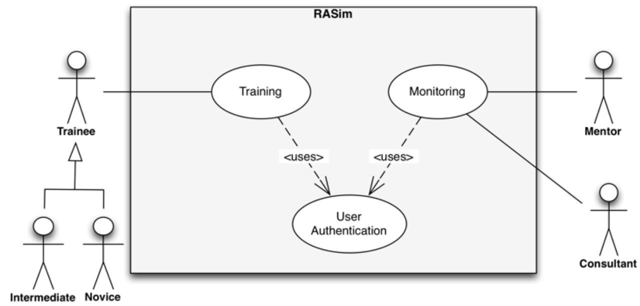

Fig. 4 The users and the main use cases of the system

B. Logical architecture

Based on these functional requirements, a two level decomposition of the RASimAs architecture has to be considered as shown in Fig. 5. The "training environment" is located in the participating hospitals' premises where the RA training takes place. Of course this environment is "replicated" in every hospital that is part of the "RASimAs system". On the other hand, there are certain components and services that are hosted in the "Central Information Server" (Global system) for the completion of the RASimAs platform. This is essentially a "star network" architecture scheme where the central components are installed once and provide the information "hub" while the distributed "training environments" are connected to a common central system. Communication between the central and the local (i.e. hospital's) components is facilitated through the use of a system-wise "software communication bus" that allows bi-directional exchange of data.

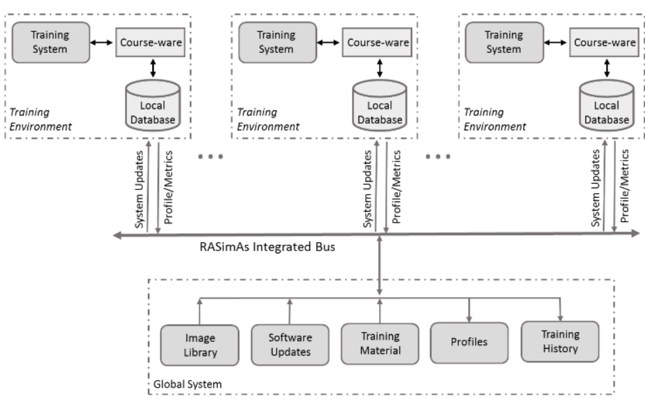

Fig. 5 Architectural design of local and global systems communication

C. Components and responsibilities

At the local environment the system is represented by the course-ware module shown before in Fig. 3. This module primarily supports the local training system by offering user registration and authentication services, providing the training material and the VPH models, and receiving performance metrics. Additionally, it performs bi-directional communication with a central server to receive system updates (software, image library, training material) and transmits updated users' data (profile/training history). The synchronization is based on the timestamp and originates from the local system: Periodically (once in a week) or on demand (i.e. when a local administrator requests it), the local system checks whether updates to the trainees' profiles or training sessions happened after the previous synchronization timestamp and then uploads any new information. Moreover, it downloads the system updates from the central system, checks whether new files were added or existing ones were deleted, and then it proceeds to consolidate these changes to the local database.

At the global site, only the administrators and other authorized users are able to provide system updates (training material, VPH models, etc.) that will then be automatically transmitted to the local environments. Additionally, the central system provides read-only access to the training history and session information of all the trainees, but this access is subject to various authorization rules. For example, all trainees may have access to their own data but a mentor is allowed to see trainee specific information only for the trainees s/he supervises in his/her local training environment.

IV. IMPLEMENTATION

The end-user applications are available as web applications so that universal access (e.g. from mobile devices), portability, and instant updates are supported. They have been implemented in Java using the SparkJava web framework[3], FreeMarker templates[4], and the Bootstrap framework[5]. Fig. 6 shows a screenshot of a trainee's training history.

The backbone of the system is a set of web-based microservices [2] for the exchange of messages and the integration of the software components. These services are implemented as REST/HTTP(s) based endpoints that support Remote Procedure Calls (RPCs) messaging patterns [3]. The payload of the packets exchanged are JSON[6] formatted text messages [4].

[3] http://sparkjava.com/
[4] http://freemarker.org/
[5] http://getbootstrap.com/
[6] JavaScript Object Notation, http://json.org/

Both the local installations and the global one use PostgreSQL[7] as the relational database for the persistence of data and training information.

The security of the system is strengthened by employing Transport Layer Security (TLS) in the HTTP communication, and the use of Public Key Infrastructure (PKI) and X.509 digital certificates [5]. In particular, every local installation and the global system have their own RSA "key pair" of public and private keys [6]. The global system is also preconfigured with the public keys of the local systems so that it can validate their "signatures" when they try to contact it.

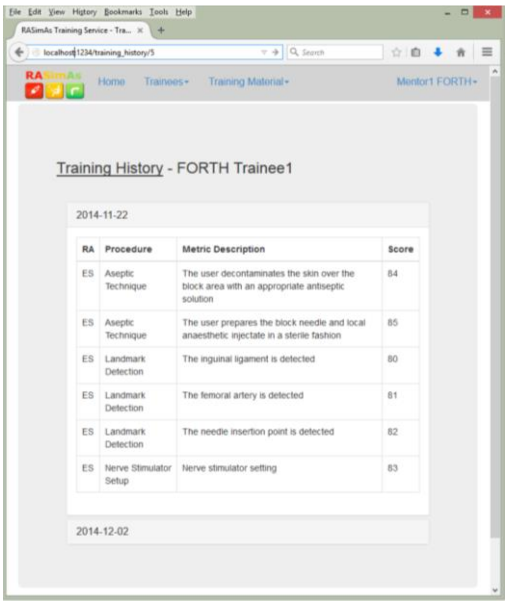

Fig. 6 View of a trainee's training history

The authentication of the communicated (sub)systems is handled through the use JSON Web Tokens [7]. JSON Web Tokens (JWT) is an open standard open source to support the exchange of "claims" between two software parties. The local training environments construct these tokens and digitally sign them using their RSA private keys when try to contact the central system for their synchronization. The central system is then able to verify the requester (client) by the public keys it has access to and subsequently return the new or updated information.

The whole system is currently deployed in a private cloud computational environment on multiple GNU/Linux virtual machines, in order to provide elasticity and flexibility over varied workload and demand.

[7] http://www.postgresql.org/

V. CONCLUSIONS

The RASimAs project aims to design and build a distributed environment for the training of regional anaesthesia. The envisaged platform comprises two systems, a local one, where the training system is installed, and a global one that acts as a central server that synchronizes periodically with all the local ones. Its primary aim is to provide an application programming interface (API) for users (credentials, profile, metrics, etc.) and system (software updates, training material, etc.) data storage/retrieval. In the case of synchronization, local systems upload user data (profile, metrics, etc.) to the central system in order to make them accessible outside the training environments, while they can also download system's updates from it (software, training material, etc.). It encompasses its own interface for accessing, locally and globally, personal information and training material via a browser after a secure login with personal credentials. The service has the capacity of uploading and storing large files (more than one gigabyte) and it is fully implemented by free and open-source software. The possible effects of this system over current training practices, in the successful application of RA by anesthesiology residents will be evaluated during the last six months of the project. Three clinical sites for RA training with RASim local systems will be deployed at the University College Cork (Ireland), at Uniklinik RWTH Aachen (Germany) and at KU Leuven (Belgium). The results of this clinical trial will be reported in a follow up publication.

ACKNOWLEDGMENT

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 610425.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

1. Grottke O, Ntouba A, Ullrich S et al. (2009) Virtual reality-based simulator for training in regional anaesthesia. Br J Anaesth. 103(4):594–600
2. Krause Lucas (2015) Microservices: Patterns and Applications. Lucas Krause
3. Fielding R. T., Taylor R. N. (2002) Principled design of the modern Web architecture. ACM Trans. Internet Technol. 2(2):115-150 DOI=http://dx.doi.org/10.1145/514183.514185

4. "RFC 7159 - The JavaScript Object Notation (JSON) Data Interchange Format" at https://tools.ietf.org/html/rfc7159
5. Menezes A. J., van Oorschot P. C., Vanstone S. A. (1996) Handbook of Applied Cryptography. CRC Press, FL
6. Rivest R., Shamir A., Adleman L. (1978) A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21 (2): 120–126 doi:10.1145/359340.359342
7. "RFC 7519 – JSON Web Tokens (JWT)" at https://tools.ietf.org/html/rfc7519

Author: Georgios Christodoulakis
Institute: Computational BioMedicine Laboratory, FORTH-ICS
Street: N.Plastira 100
City: Heraklion, Crete
Country: Greece
Email: gxristod@ics.forth.gr