A Low-cost Distributed IoT-based Augmented Reality Interactive Simulator for Team Training

Pietro Piazzolla[1], Marco Gribaudo[1], Simone Colombo[2], Davide Manca[2], and Mauro Iacono[3]

[1]Dipartimento di Elettronica, Informazione e Bioingegneria
[2]CMIC Chemical Engineering Department
Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milano, Italy
{pietro.piazzolla, marco.gribaudo, simone.colombo, davide.manca}@polimi.it
[3]Dipartimento di Matematica e Fisica, Università degli Studi della Campania "L. Vanvitelli"
Viale Lincoln 5, 81100 Caserta, Italy
mauro.iacono@unina2.it

KEYWORDS

Distributed simulation, Real time, Virtual reality, Education, Immersive training, Firefighting

ABSTRACT

The performance over cost ratio of last generation off the shelf devices enables the design of heterogeneous distributed computing systems capable of supporting the implementation of an immersive Virtual Reality, Internet of Things based training support architecture.

In this paper we present our work in progress on a low cost distributed immersive simulation system for the training of teams by means of Virtual Reality and off the shelf mobile and prototyping devices. In this case, performance prediction is crucial, because the generation of the scenario have to be performed in real time and synchronization problems may disrupt the result.

The approach is demonstrated by a prototypical case study, that consists in a distributed simulator for the interactive training of groups of people that have to coordinate to face a fire emergency, and features advanced immersivity thanks to CGI-enabled stereoscopic 360 degree 3D Virtual Reality and ad-hoc devised interaction interfaces. In particular, we focus on the subsystem that is related to a single trainee, providing a reference implementation and a performance evaluation oriented model to support the design of the complete system.

INTRODUCTION

The performance-over-cost ratio of last generation off-the-shelf devices enables the design and implementation of cost effective complex interactive cyberphysical systems for non-critical applications. The consumer market offers: powerful smartphones with many cores processors, gigabytes of RAM and native power management and communication features; low cost augmented reality or virtual reality apparels; customizable prototyping oriented computer systems, such as Raspberry Pi and Arduino; Internet of Things (IoT) enabled smart location and motion aware sensors. Conversely, the most of these devices should not be considered reliable, due to the fact that they are meant to be sold on the consumer or makers market.

This equipment is anyway a resource to build heterogeneous distributed computing systems capable of supporting the implementation of non critical applications oriented to interactive simulation in a real environment, integrating a sensor part, by means of the IoT enabled components, an Virtual Reality (VR) part, supported by smartphones or prototyping platforms, and a coordination part, that is implemented by means of a peer to peer or server based distributed layer. A possible application is the interactive training of groups of people that have to coordinate to face an emergency in a given fictional location or on the field. In such a scenario the coordination part also includes a non trivial distributed workload that should generate in real time the VR elements: this implies a local generation of the point of view of a single user and a global management of the virtual objects and what is connected to the positions and the actions of all other users. Although such workloads are not a problem for modern high performance computer systems, the nature of the target reference architecture, the available computing power, the low reliability of common consumer devices, the interconnection problems and the real time requirements make the design of the system non trivial.

We propose an immersive VR IoT based training support architecture, based on off the shelf components, for low cost real time implementations. In this paper we focus on the subsystem that is related to a single trainee, providing a reference implementation and a performance evaluation oriented model to support the design of the system, with reference to a fire extinguisher use training case study.

This paper is organized as follows: in the next Section we present a quick survey of related literature, highlighting the main features that characterize our single-trainee subsystem compared to other similar solutions, proposed in the literature or currently on market. In the subsequent Section the technical details of our proposed simulator are presented, focusing on both the hardware both the software solutions adopted.

Proceedings 31st European Conference on Modelling and
Simulation ©ECMS Zita Zoltay Paprika, Péter Horák, Kata Váradi,
Péter Tamás Zwierczyk, Ágnes Vidovics-Dancs, János Péter Rádics (Editors)
ISBN: 978-0-9932440-4-9/ ISBN: 978-0-9932440-5-6 (CD)

Then, this single-trainee subsystem is placed inside the wider perspective of an IoT collaborative system, which is described into details. Finally, a performance evaluation model of our system for the support of the design process is discussed.

RELATED WORKS

The use of virtual reality for training and learning has been largely explored by many papers (See e.g.:[1]). In this section we highlight the main features that characterize our simulator compared to other similar solutions, proposed in the literature or currently on market. While all solutions strove to provide the best real time 3D immersive computer graphics available at the time of their development, not all of them concerned about maximizing visual quality while containing the final application cost, like our proposed simulator. Other main differences between these products and our solution: the type of sensors used to detect user gestures and inputs (from ad-hoc devised controllers to on-market solutions like Microsoft Kinect II or Nintendo Wii controllers), and the hardware adopted to visually display the 3D environment (from Cave systems to Head Mounted Displays like Oculus Rift or Google Daydream). Recently in [2], the authors propose firefighting scenarios based training system on virtual reality platform. Their simulator software engine runs on a dedicated machine instead of, as in our solution, on smartphones. Our solution tend to be more cost effective with a lower physical space requirement to deploy the simulator. In [3] the authors aim for a solution that use software and hardware already on market to implement their simulator. The final product leverages on a cave system for providing immersivity, a solution which may not be widely adopted due to space constraints. One of the most comprehensive works on virtual reality for fire extinguisher use training is [4]. The author implement a simulator relying on two infrared cameras for tracking user gestures. Empirical evidence taught us that camera tracking alone (i.e.: not coupled with other sensors) may tend to be of limited effectiveness in some circumstances when the process of marker detection introduces delays.

The several on-market solutions can be sorted in two big categories: those using physical world video footage to present the virtual environment for training, like e.g.: [5], [6], [7], [8] and those leveraging on 3D computer graphics to generate it, like e.g.: [9], [10]. The latter solutions are intended for fire-fighters training, not just for the training of fire extinguishers.

About Internet-Of-Things main aspects and challenges: [11] offers a comprehensive review of the functional complexity of connecting real-world objects to the Internet with tiny sensors, while in [12] a Cloud centric vision for worldwide implementation of Internet of Things is presented and the key enabling technologies are discussed.

ARCHITECTURE OF THE SIMULATION SYSTEM

The system is composed of personal nodes, that collaborate on a peer to peer basis to enact the overall real time simulation. The hardware architecture of each node includes a smartphone, a device coordinator, a smart sensing subsystem and a AR/VR subsystem. The device coordinator, that may be implemented as an Arduino system as in our case study or by another analogous technology, is in charge of managing the smart sensing subsystem, that is composed of one or more smart sensors; the smartphone is in charge of running the distributed simulation and of coordinating with the other nodes, eventually managing dependability issues of the interconnection, by using the services that are provided by the device coordinator; according to the complexity of the AR/VR workload, either the smartphone or the device coordinator is in charge of piloting the AR/VR device. In Figure 1 the the software architecture

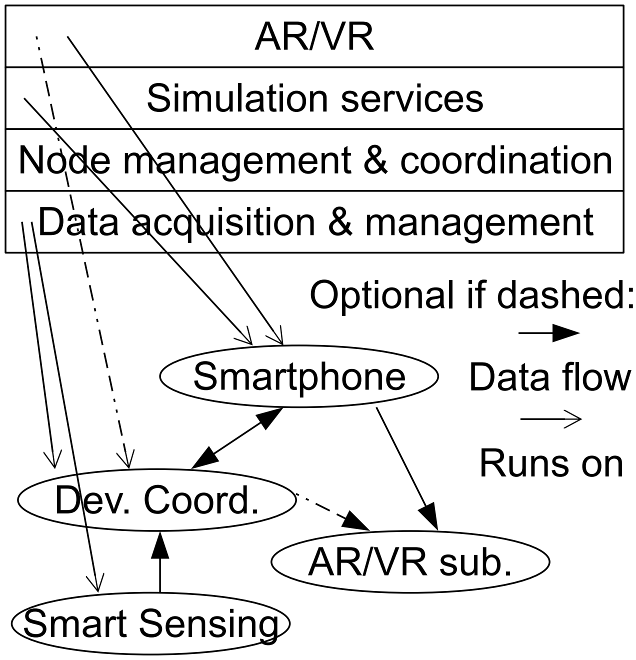

Fig. 1. The software architecture of the simulation system

of the system is presented. It is structured into 4 layers. The bottom layer is the data acquisition and management layer, that is distributed among the smart sensors and the device manager: smart sensors provide a first level of data processing, while the device manager provides a post processing that is based on a synthesis of data from smart sensors. The second layer is node management and coordination: it is executed on the smartphone and it is responsible of running a node and providing the integration into the distributed system. The third level is the simulation services layer: it runs the distributed simulation and generates and maintains the internal abstract description of the environment and of the participants and all needed synchronization. The fourth layer is the AR/VR layer, and is responsible of generating what needed for visualization (eventually supporting, besides the devices of the AR/VR subsystem, external monitors).

IMPLEMENTATION

In this Section we present the technical details of a node of the proposed simulation system. Its main goal is to allow trainees to learn the correct use of a fire extinguisher in a 3D

Virtual Reality (VR) environment. Trainees benefit of the use of a virtual environment in avoiding them all the risks and costs associated with physical world training.

A VR head-mounted display (VR HMD) enable an immersive, stereoscopic, 360 degree view of the fire emergency scenario, while a specifically developed interface system that can be mounted on a physical fire extinguisher manage the interaction between the user-trainee and the virtual simulation.

As seen, from a technical perspective, each node of the simulation system is composed by a 3D *software application* containing all the simulator's logic and its 3d graphic engine, and a *hardware system* that is in charge of handling a trainee's input gestures and providing them to the application. In Figure 2 we present the overall structure of the real implementation of a node.

Node hardware

The core element of the hardware system is an *Arduino Leonardo*[13], a microcontroller board that appears to a connected computer as a mouse and keyboard. It can be easily programmed using a dedicated programming language, in order to adapt its behavior to users needs. Connected to the Arduino Leonardo input pins there are two devices: an *accelerometer* with 9 degrees of freedom and an *analog control stick*. In particular, we use the first one to determine the direction of the fire extinguisher nozzle and the latter for handling the movements of the trainee in the simulated environment. For the purposes of our simulator prototype development, for the accelerometer component, we use a SparkFun 9DOF Sensor Stick [14], a very small sensor board with 9 degrees of freedom that includes an accelerometer and a magnetometer. In order to solve the challenge of real-time 3D orientation tracking of the nozzle, we resort to the implementation of an AHRS (Attitude and Heading Reference System): a 3-axis sensor system that provides attitude position (e.g.: pitch, roll, and heading) by fusing accelerometer data and magnetometer data.

The analog control stick used is a Nintendo Nunchuk [15] compatible controller. Its relative small size allow to easily position it over a physical fire extinguisher without incurring in handling problems. The trainee is able to move inside the virtual environment intuitively using the thumb of same hand that carries the extinguisher. The controller allow forward/backward movement along the user's viewing direction, while the left/right axis allow strafe accordingly.

Since our training goal can greatly benefit from immersivity, we resort to an head-mounted display for the trainee to visualize the virtual environment. We decided to avoid vendors lock-ins, so we opt for a generic HMD mount able to house an any-brand smartphone.

We exploit the smartphone operating system and resources to host and run the application implementing the simulator logic. Thanks to their displays and graphics adapters an high visual quality can be obtained, able to satisfy immersivity requisites about visual realism. The vast majority of newest generation smarthphone are equipped with an internal accelerometer that is used to track trainee's head movements to determine his or her looking direction in the virtual environment. The Arduino Leonardo sends data from the input devices it handles to the smartphone by means of a standard micro USB cable, but a wireless solution is currently under study. Since the Arduino board is actually powered through the USB by the phone, renouncing to the cabled solution will introduce a new challenge. Figure 3 shows how the hardware components of

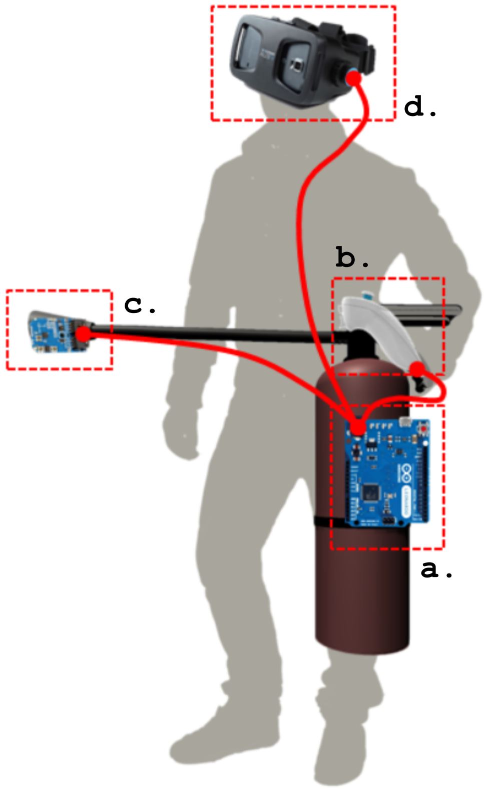

Fig. 3. Simulator' input and output devices wear by the user. The connected devices are: a. The Arduino Leonardo; b. Analogic Control Stick; c. Accellerometer for nozzle rotation tracking; d. Smartphone housed in the HMD mount.

the proposed node fit on the trainee. The main advantage of this setup is in its independence from external sensors (e.g: Microsoft Kinect II) or computational resources.

Node software

The proposed simulator logic is implemented as an application developed using the Unity cross-platform game engine[16] and has been tested both on Windows and Android operating systems. It features different scenarios of possible fire related disaster that challenge the trainee acting to avoid fire escalation, by using in the proper way the fire extinguisher.

According to our in-the-field experience, a fire extinguishing system simulator should cover and include a number of features. Based on experimental training, the trainees should get trained to wear some protective devices and should experience the side effects when they forget to. The burning substance could be different and consequently also the typology of extinguisher both in terms of compounds (e.g., foam, powder,

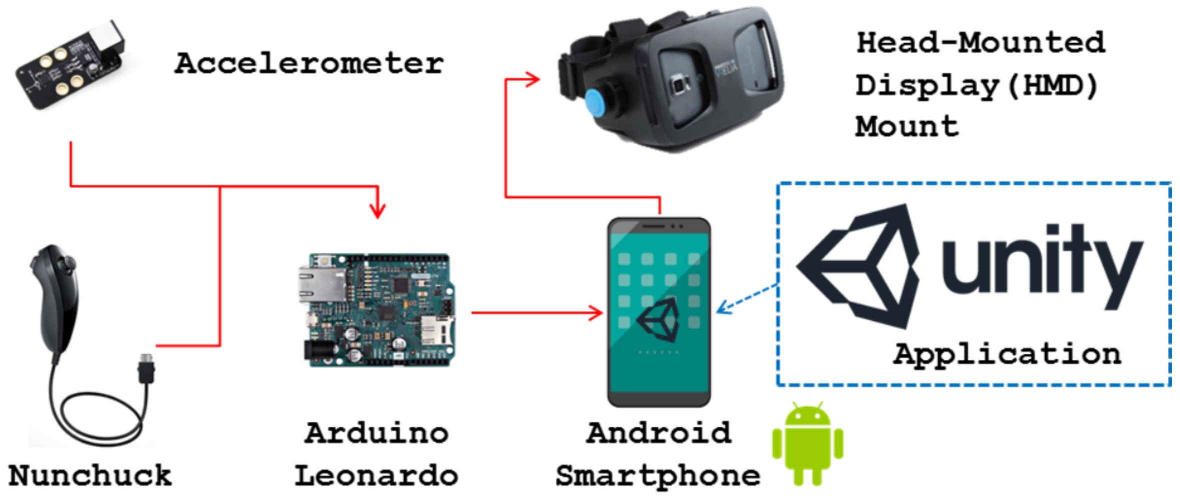

Fig. 2. Simulator structure, highlighting hardware interaction.

gas, liquid) and dimensions. The fire produced by the burning substance would consequently be different and its behavior subject to the surrounding environment. In case of outdoor fires, the flame should bend to winds. For the sake of training purposes, the trainee should take advantage from a virtual augment reality (AVR) feature. AVR would provide details about the heat radiation and the thermal load, which is the amount of heat received after exposure, i.e. the absorbed dose. A real-time graph would also provide details about the physical thresholds for first, second, and third degree burns to inform the trainee about the exposure risk to flames as a function of the donned self-protective devices and the distance from the fire.

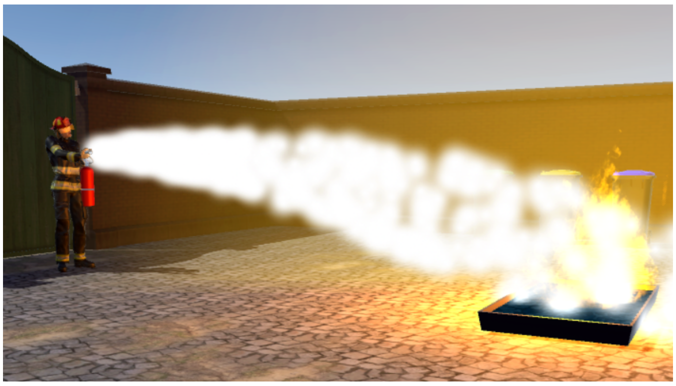

Fig. 4. Particle system used for fire extinguisher spray.

Each user is represented by an avatar handling the extinguisher, as can be seen from Figure 4. We invested a great deal of effort making the particle effects reproducing fires and sprays in a realistic way.

Note that during the training simulation, the trainee will see the environment through the avatar's eyes to ensure the immersivity.

In Figure 5 the stages that a user encounters during his interaction with our simulator are presented. When the application is launched the first time by a new learner it is necessary to go through a quick *calibration* process. It is necessary to correctly align input sensors to the virtual character and extinguisher for direction and pointing at purposes.

Fig. 5. Simulator stages.

When the technical equipment is set up, the learner can access a first *training scenario* (See Figure 6) to familiarize with the basics of moving in the virtual environment and the nozzle pointing, as well as with the first very basic notion of extinguisher handling. These information are displayed to the trainee as a sequence of tutorial-like steps to be followed. After the training scenario, the learner can *select* one of the several fire disaster scenario, different both in terms of parameters of fire diffusion and fire intensity, both in terms of extinguisher typology and in terms of the environment virtually reproduced. The selected scenario is *simulated* and the training session can begin. The third layer introduced in the previous Section is in

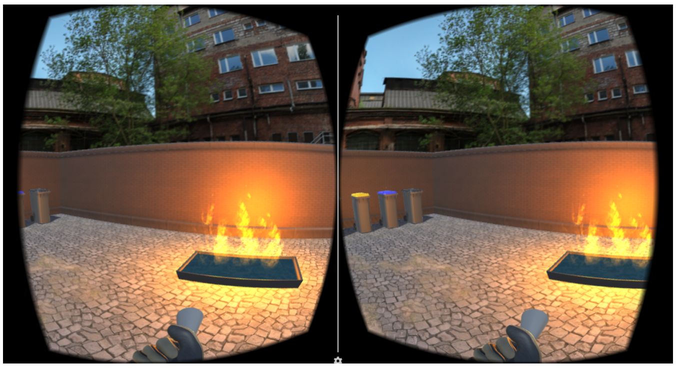

Fig. 6. Stereoscopic 360 view of the training yard level of the simulation.

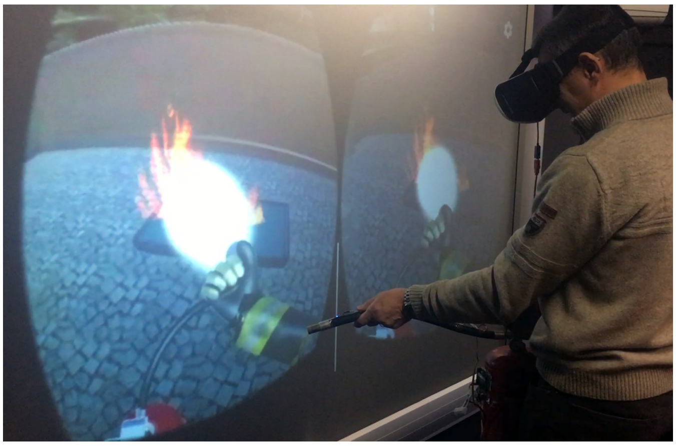

Fig. 7. Testing our simulator, with a screencast of the image as seen by the user-trainee.

its development stage and actually not yet used to interconnect the nodes. It is instead implemented on a stand-alone machine that, by means of a web browser, connects to an App (Screen Mirror[17], in particular) on the same smartphone on which the simulator is running. In this way, during training session it is possible to follow trainees' progresses by web-casting the visual result of the simulator on a screen (See Figure 7). After the session it is also possible to receive a *feedback* from the simulator in the form of collected statistics about trainees' performances.

The performance assessment of the trainees is accomplished by tracking and processing their decisions and actions. These include their distance from the fire, the impact angle of the gas/liquid/foam/powder jet on the flame, the emission time, and the total time spent to extinguish the fire. These bits of information, once automatically processed at the end on the experiment, provide some valuable details for either a self or a trainer assessment. The AVR feature can be activated or deactivated by the user or the trainer depending respectively on a training or assessment session.

MODEL AND EVALUATION

In order to support the design process of a simulation setup, it is important to predict the performances of each single node, in order to be able to verify that real time requirements are achievable, before experimenting on a prototype (in the system development phase) and before setting up a simulation (in the application of the system to a certain simulation scenario). A convenient approach is the use of a queuing network based model.

The queuing network needed for the task is composed by three modules per each node. For each node, two modules describe the device coordinator and the smartphone, respectively. These two modules are structured after the main tasks that they have to accomplish in the system, and form a pipeline. The third module, that is parametric in the number of IoT sensors, models the workload that the device coordinator has to manage from the IoT sensors. Figure 8 describes a possible implementation of the queuing network for a node of the simulation system.

With reference to Figure 8, from left to right and from top to down, the first depicted module is the IoT module. This module is composed of one queue per IoT sensor that is connected to the node ($IoTsens_i$, with i that assumes values between 1 and the number of connected sensors N). All the outputs are directed to the device coordinator module.

This module is composed of a queue that accounts for the management of the communications with the IoT sensors, a queue that accounts for the workload generated by passive sensors (that need a local processing of data, differently from IoT sensors, and that may need to be explicitly activated and controlled by the device coordinator, e.g. for polling) and a queue that accounts for the workload due to the software tasks connected to local data processing needs. The output of last queue is directed to the smartphone module.

The smartphone module is composed of a queue that models the local sensors (i.e. the ones integrated in the smartphone, such as accelerometers), a queue that models the data management operations that allow the interactions with the device coordinator, a queue that accounts for the tasks that allow the integration in the distributed simulation system by processing the events from the other nodes, and the related networking workloads, a queue that executes all the local tasks that allow to perform the simulation, including the generation of the additional VR/AR workloads to be executed by the dedicated hardware and the management of feedbacks, and a queue that models the behavior of the dedicated hardware.

The overall queuing network model is then obtained by a composition of one such model per node, and the interactions between the nodes are modeled by means of the dangling arcs showed in the figure, that are to be interpreted as one per kind towards and from any node to all others. This approach seamlessly allows to model a simulation system composed of heterogeneous nodes, that differ in the configuration and in the hardware parameters; moreover, the performances of a node may be assessed in isolation, by substituting the queuing networks representing the other nodes with a single fictional queue, that can be easily reconfigured to study system scaling problems. This approach has been designed as a support for overall simulation system assessment, node design parameter

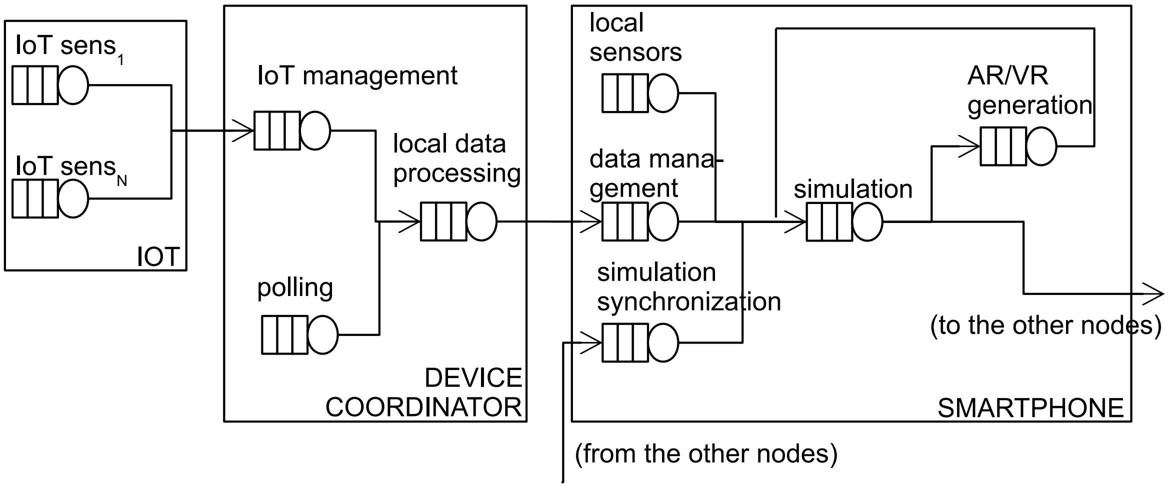

Fig. 8. The queuing network parametric model of a node

exploration and scaling problems.

To show the effectiveness of the approach, we applied the description in Figure 8 to the described case study, and simulating the presence of the other nodes by means of a single queue.

CONCLUSIONS AND FUTURE WORKS

In this paper we presented an immersive VR IoT based training support simulator architecture, which, by exploiting off-shelf components, can be implemented at low costs. Even if the presented results are very interesting, our work on the simulator is still in progress. In this paper we focused on the subsystem that is related to a single trainee, providing a performance evaluation oriented model to support the design of the system. In the future we plan to extend our study by adding the interaction of several single trainee subsystem, as well as performing a test campaign using fire department personal to asses the validity of the proposed architecture.

REFERENCES

[1] A. Craig, W. R. Sherman, and J. D. Will, *Developing Virtual Reality Applications: Foundations of Effective Design*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009.

[2] P. Vichitvejpaisal, N. Yamee, and P. Marsertsri, “Firefighting simulation on virtual reality platform,” in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, July 2016, pp. 1–5.

[3] F. Poschner, “Fire fighting and related simulations in a cave using off-the-shelf hardware and software,” in *Proceedings of SIGRAD 2014, Visual Computing, June 12-13, 2014, Göteborg, Sweden*, no. 106. Linköping University Electronic Press, Linköpings universitet, 2014, pp. 33–40.

[4] M. Maschek, “Real time simulation of fire extinguishing scenarios,” Master’s thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Favoritenstrasse 9-11/186, A-1040 Vienna, Austria, 2010. [Online]. Available: https://www.cg.tuwien.ac.at/research/publications/2010/maschek-2010-rts/

[5] iBG fire simulator. [Online]. Available: http://www.feuersimulator.com/index_en.html

[6] ViFeLoe. [Online]. Available: https://www.vrvis.at/research/projects/vifeloe/

[7] Pyrosoft - fire safety training. [Online]. Available: http://www.pyrosoft.ca/

[8] Bullex laser-driven fire extinguisher training. [Online]. Available: http://bullex.com/product/bullseye-publiceducation/

[9] ETC Training Systems - ADMS-Fire. [Online]. Available: http://www.trainingfordisastermanagement.com/products/adms-fire/

[10] Ludus firefighter services. [Online]. Available: http://www.ludus-vr.com/en/portfolio/firefighter-services/

[11] M. Swan, “Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0,” *Journal of Sensor and Actuator Networks*, vol. 1, no. 3, p. 217–253, Nov 2012. [Online]. Available: http://dx.doi.org/10.3390/jsan1030217

[12] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645 – 1660, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000241

[13] Arduino Leonardo webpage. [Online]. Available: https://www.arduino.cc/en/Main/ArduinoBoardLeonardo

[14] Spark Fun 9DOF Sensor Stick. [Online]. Available: https://www.sparkfun.com/products/retired/10724

[15] Nunchuk Wii Remote controller. [Online]. Available: https://en.wikipedia.org/wiki/Wii_Remote#Nunchuk

[16] Unity cross-platform game engine. [Online]. Available: https://unity3d.com/

[17] Screen mirror androids app. [Online]. Available: https://play.google.com/store/apps/details?id=com.ajungg.screenmirror