2011 IEEE/ASME International Conference on
Advanced Intelligent Mechatronics (AIM2011)
Budapest, Hungary, July 3-7, 2011

VirCA as Virtual Intelligent Space for RT-Middleware

Péter Galambos*†, Péter Baranyi*‡

*Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary
†Department of Manufacturing Science and Technology, Budapest University of Technology and Economics
‡Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics
E-mail: {galambos, baranyi}@sztaki.hu

***Abstract*—In this paper, VirCA [1] (Virtual Collaboration Arena) is introduced as a Virtual Intelligent Space that is organically connected to the RT-Middleware framework. The paper presents the conceptual background of the VirCA system and its relation to RT-Middleware. The scope of possible applications are discussed via working examples.**

***Index Terms*—Intelligent Space, 3D Internet, Augmented reality, Collaboration platform, Middleware technologies, Networked robots**

I. Introduction

In the last decade several frameworks appeared to facilitate component-based system development and code reusability. Some of the most significant systems are RT-Middleware (openRTM-aist) [1], [2], [3], ROS (Robot Operating System) [4], MRDS (Microsoft Robotics Developer Studio) [5], Player and Orocos [6]. A detailed discussion on state of the art architectural solutions for distributed robotics has been published by Amoretti and Reggiani in [7]. William D. Smart in [8] discourses on the demand and the possibility of a common middleware for robotics. Mohamed et al. reviewed the existing middleware technologies for robotics [9] at that time. This area is dynamically developing recently so it is hard to give an overall and up-to-date survey.

All the listed environments follow slightly different philosophy but there are attempts to find the way to bridging from one system to the other making heterogenous systems of components. For example Geoffrey Biggs has developed a patch to OpenRTM-aist establishing communication channel between components written for OpenRTM-aist and nodes written for ROS.

Most of the component-based frameworks involve the 3D visualization. Some of them loosely or inherently integrates 3D visualization module: ROS - RVIZ, RT-Middleware - OpenHRP3, Player - Stage/Gazebo. They can be used for several obvious purposes e.g. visualization of sensory and environmental information, kinematic, dynamic and sensor simulation. However, non of them exploits the possibilities of the virtual 3D space in an extensive manner. Over the before mentioned functionalities VirCA implements a novel approach for using the 3D virtual space to manage the collaboration and data exchange between the system components and the human operators. This philosophy is in harmony with the intelligent space (iSpace) concept introduced by Hashimoto [10], [11] in the early 2000's. In the philosophy of VirCA, the virtual space is considered like a 3D operating system over RT-Middleware providing a shared virtual reality for the components and the users. Special RT Components (Cyber Devices) which are representing themselves in this virtual arena can use the advanced UI, HMI facilities (speech recognition, input devices etc.) provided by VirCA and can gather various information from the virtual environment via simple queries or virtual sensors. This complex service become beneficial in reality when a real scenario (at least partly) is tracked and updated in the shared virtual environment. Although, VirCA basically lean on RT-Middleware it is possible to develop gateway components to other platforms. Different aspects of VirCA is discussed in previous works [12], [13], [14], [15], [16]. In this paper the relationship of VirCA and RT-Middleware is discussed and the perspective of possible applications is shown via example scenarios. The terminology of RT-Middleware is used within this paper. Detailed explanation of such expressions can be found on the RT-Middleware website [1].

The paper is structured as follows: Section II explains the vision behind VirCA with its four most characteristic paradigms. Section III introduces the structure of VirCA system, its relationship to RT-Middleware (openRTM-aist) and the web-based VirCA System Editor. Four example assemblies are presented to illustrate the scope of potential applications in section IV and section V summarizes the paper.

II. Motivation

In this section the aspects of demands are discussed according to the recently emerging concepts of cutting edge technology. The following four paradigms characterize the goals what we would like to reach with VirCA and RT-Middleware.

A. Activating 3D

VirCA aims to provide a platform where users can build, share and manipulate 3D content, and collaboratively interact

[1]VirCA - **Vir**tual **C**ollaboration **A**rena, developed in MTA SZTAKI (Computer and Automation Research Institute, Hungarian Academy of Sciences). VirCA is free to use for academic purposes and available online at the www.virca.hu website.

with processes in a 3D context, while the participating hardware and software devices can be spatially and/or logically distributed and connected together via IP network. For example, engineers and researchers from different countries can put a mobile robot and an obstacle avoidance software module together to test and tune it in a semi-virtual manner. They do not have to move their devices to the same place and take effort in the system integration. Or even, an architect can discuss the design of the future house with a customer, or a robot expert can train a technician how to program an industrial robot. The VirCA 3D virtual reality component can work together with stereoscopic 3D display technologies including NVidia 3D Vision and immersive 3D CAVE systems [17]. VirCA compatible 3D objects (OGRE mesh) can be converted from prevalent CAD formats or generated by advanced 3D surface measurement methods [18].

B. Augmented Collaboration

The 3D content and processes in VirCA can be synchronized with the real world, which allows the combination of reality and virtuality in the collaboration arena. VirCA users can virtually interact not only with other users, but also with existing, remotely operated hardware and software such as robots, sensors or control algorithms. This type of semi-virtual interaction let the users to build distributed systems consisting real and virtual parts at the same time.

C. Knowledge Plug and Play

As VirCA follows the uniform, modular, RT-Middleware framework, the components from different sources can be easily organized into a working system using a browser-based graphical programming environment. This allows the plug and play exploitation of the knowledge that is embedded into the community developed components, and simplifies the creation of new state of the art solutions. In our vision, RT-Middleware components and VirCA-enabled software pieces (Cyber Devices) implementing generic or customized interfaces could be collected in a community website. System integrators could use these functional building blocks for free or even on commercial basis similarly to the popular online application stores. In this way the knowledge that is integrated into the downloadable components can be easily added to any VirCA based system in a plug and play manner.

D. Proactive Development for the Future Technologies

In the virtual environment everything can be ”measured”, even quantities that are yet unmeasurable in the real world. Virtual sensors are thus could be much powerful than real ones. The case is similar with actuators and algorithms, which makes VirCA able to accommodate virtual experiments using existing or not-yet-existing ”Virtual Future Technology” actuators, sensors and algorithms. As a practical example, a mobile platform can be equipped with virtual sensors what real counterpart would be too costly for experimental purposes (e.g. laser range finder, depth camera etc.). In this manner, the controller software can be built and tested in the same way whether using real or virtual devices. In case of a robot navigation algorithm, the real mobile robot is moving in a room while its virtual manifestation doing the same in the virtual environment and collecting sensor data from the virtual world using virtual sensors. The navigation software using the virtual sensor data similarly as real world sensor input would have been processed and command the mobile platform according to the implemented algorithm.

III. Architecture of VirCA

VirCA follows modular building strategy. It is considered as a collection of components and utilities providing useful services and a flexible virtual reality infrastructure for openRTM-aist applications. This section details the building blocks and basic concepts related to VirCA.

A. VirCA Core Component

VirCA Core is the central and indispensable component of any VirCA-based modular application. It is responsible for the 3D visualization, physical simulation and acts as a database which handles and registers the external objects and the related events in the virtual reality. Additionally, the core component manages the interaction with the users providing various forms of communication channels from 3D menus to speech recognition and text to speech. VirCA core integrates several open source, community developed softwares: VirCA uses the OGRE [19] graphics engine for 3D visualization and scene management of the shared virtual environment while Bullet [20] physics engine is used for real-time multibody dynamics simulation. VirCA allows the users to open web browser within the 3D scene. This feature is implemented using the open source Berkelium library.

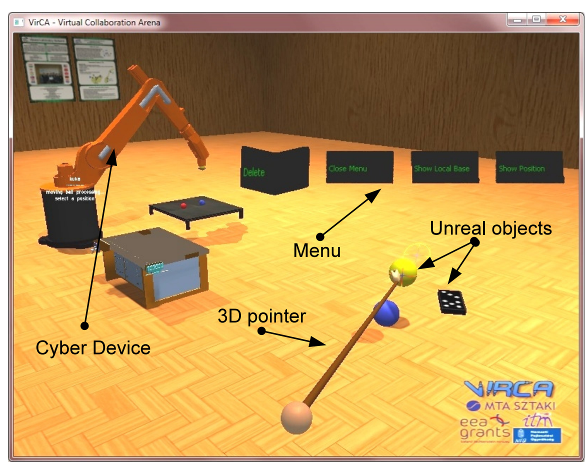

Fig. 1. Objects in the VirCA scene

To the core, further external components can be connected realizing arbitrary functionalities or services (e.g. input devices, speech recognition, text to speech, etc.) which are accessible for other VirCA enabled devices. For example, Windows SAPI (Speech API) is adapted to VirCA as external service. The most important type of connectible component is the Cyber Device that represents an active entity (any intelligent agent or DIND) in the virtual space. Figure 1 shows a snapshot form a VirCA session with captions on the different object types. VirCA core component appears to the user as a special RT-component which communicates via the VirCA interfaces.

B. Cyber Device, RT-Component

Cyber Devices (CD) are special RT-Components (RTC) what are prepared for representing a real or virtual entity in the shared space of VirCA. Cyber Devices implements special type of service ports (VirCA interfaces) dedicated to receive from and send messages to the VirCA Core component. VirCA interface is briefly described in the next subsection. The relationship between Cyber Devices and other RT-Components is illustrated by figure 2.

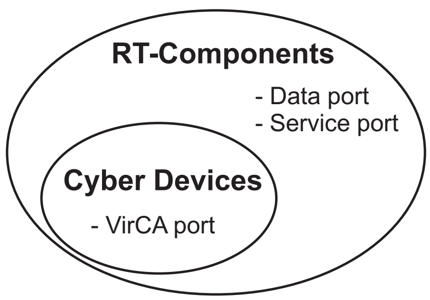

Fig. 2. Relationship of Cyber Devices and RT-Components

A Cyber Device can communicate with other (non Cyber Device) RT-Components via data ports and service ports of openRTM-aist. Typically, a Cyber Device represents a robot or a robot component (drive, sensor, tool etc.) or any item what the robots interacts with. CDs are characterized by their geometry and other relevant properties. Depending on the goal of a VirCA -based application, a Cyber Device can even act as 3D GUI of a robot system or as a simulated part of a semi-real, semi-virtual RT-Middleware assembly. For the better understanding examples are presented in section IV.

C. The VirCA Interface

The RPC-like (Remote Procedure Call) communication channel between the VirCA Core and the Cyber Devices is called VirCA interface. In RTM sense it is implemented as two service ports. The VirCA $\rightarrow$ CD dataflow is realized as an asynchronous method invocation on the application service interface. This is provided on CD side while consumed by the VirCA Core. VirCA sends messages to the CDs whenever an event occurs in the virtual arena what has effect on a given Cyber Device. The messages inform the CDs about user actions (key stroke, input device action, drag and drop, etc.) and physical events (collision, interferences, etc.). The messages contain the type of the event and its properties e.g. in case of a mouse click, the message contains the x,y, and z coordinates in the 3D virtual space.

In the other direction (Device $\rightarrow$ VirCA) the structure is similar, but the interface is provided on the VirCA Core and

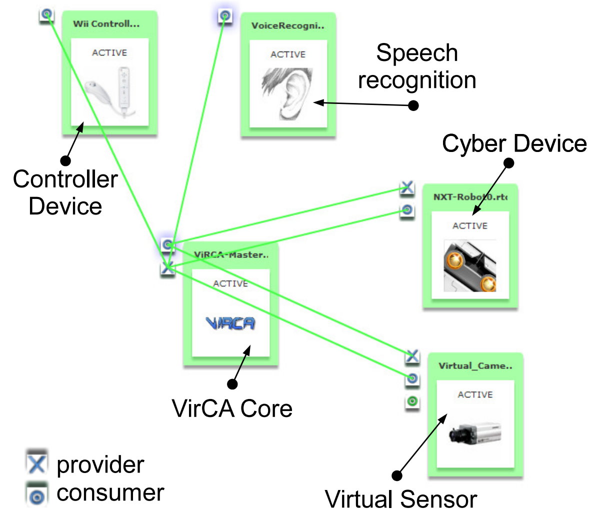

Fig. 3. Different type of components connected to VirCA

consumed in the connected external devices. This service port is a compound of several interfaces according to the different functions provided for the different devices. These functions are responsible for the manipulation of the 3D objects, the 3D pointer and the camera. External speech recognition and menu manipulation is also connected via this interface.

In recent version of VirCA, the above described service ports are implemented as ICE [21] Ports using the ICE extension for RT-Middleware [22]. In the upcoming releases of VirCA, the CORBA service port version of the interfaces will also be included what is natively supported by openRTM-aist. Detailed description of the interface functions can be found on the VirCA website.

D. VirCA System Editor

VirCA application assemblies can be managed in a web-based System Editor [Fig.4] that is able to handle openRTM-aist compatible connections and the RTM-ICE connections as well. The editor is developed in a multi-user client-server structure so that users can manage, save, reload and share their component assemblies. It can run in any Flash enabled web browser, even in which is opened within the 3D scene of VirCA.

IV. Examples

In this section, examples are presented to show possible areas of VirCA applications.

A. Remote Commanding of an Industrial Robot

In this toy example, VirCA is used as an advanced user interface in a telerobotic application. The remote plant consist of an Industrial robot (KUKA KR6) equipped with a gripper, two desks and three colored balls (red, yellow and blue). A USB camera is used to observe the position of the balls. Utilizing these hardware components, a VirCA application assembly is formed in order to let a remote operator be able to

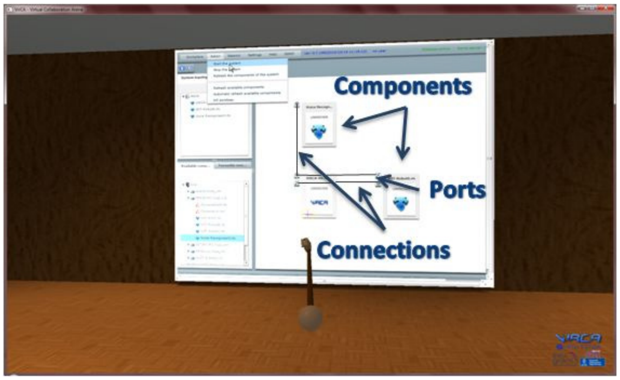

Fig. 4. Web-based system editor in a browser in the virtual arena

manipulate the colored balls with high-level user commands in a spoken dialogue. Ten components are involved in the assembly forming three functional branches around the VirCA Core [Fig. 5]. Voice recognition and a Wii controller are connected to VirCA composing the branch of user interfaces. The colored balls are represented by three Cyber Devices. They get their real positions from a position tracker RTC and update the balls position in the virtual space. The position tracker component receives video stream from a USB camera component that is installed with openRTM-aist examples. In the robot branch a Cyber Device embodies the real robot in the virtual arena and implements the logic according to the ball pick and place task. The robot CD send motion commands to the real robot using a generalized industrial robot service port. The generalized robot service is provided by an interface component that communicates with the robot controller via RS-232 line. This interface component is capable of translating generic motion commands to the robot specific language (KRL) and gathering joint and Cartesian coordinates from the robot controller. This assembly follows the iSpace concept, as the robot does not get the target positions from the image processing unit directly but query this information from the VirCA Core. In this way, VirCA take the place of the knowledge database in this partly virtual intelligent space.

After the system is started up, the industrial robot can be operated by spoken commands and controller device actions (mouse or wii controller) in the 3D virtual world. The robot is able to pick up the selected ball and place on the user-specified position. The motion of the real robot can be followed in an IP camera stream played in a browser window. The real and virtual robots moving similarly as the robot Cyber Device updates the virtual joint coordinates with a certain rate.

B. Incremental Sheet Forming Controller

In this example, the cell level control of an incremental sheet forming (ISF) process is discussed. ISF is a one of a kind or small series production technology capable of forming sheet parts by the complex motion of a simple tool rather than using positive and negative dies [23]. The experimental manufacturing cell consist of a three axis milling machine and a six DOF robot arm. The milling machine is utilized to move

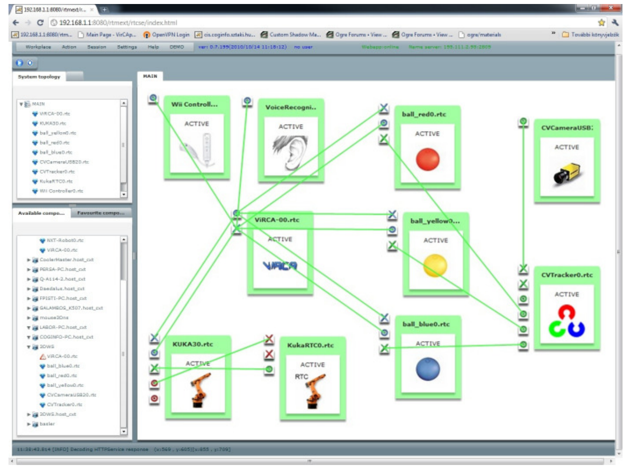

Fig. 5. Industrial robot example assembly in the System Editor

the forming tool while the robot carries a flameless heat gun to heat the plastic sheet piece locally to the optimal temperature.

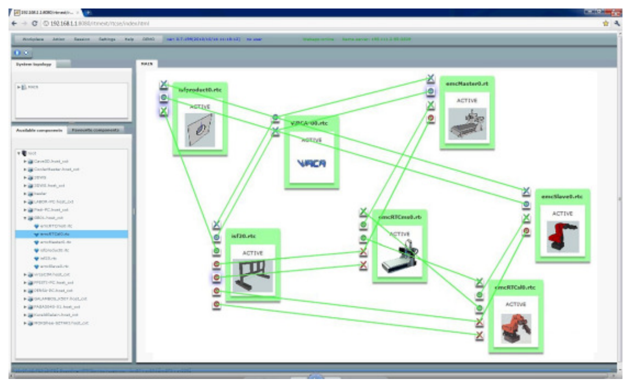

Fig. 6. Incremental Sheet Forming scenario in the System Editor

The robot and the mill are rather obsolate desktop size devices both operated using the EMC - Enhanced Machine Controller [24]. EMC is for real-time control of axis moving machines. The presented VirCA sceanrio 6 provides cell level control of the ISF process. It handles the CNC programs of the milling machine and the robot, synchronize their motion and provides process monitoring and user interface facilities. The assembly consists of six RTCs and the VirCA Core. There are four Cyber Devices representing the machine tool, the robot, the cell controller and the work piece respectively. The cell controller CD appears in the scene as the workpiece holder fixture. According to the design pattern applied in the previous example, the RT-Components of the robot and the machine tool are accessed via generic interface for the sake of reusability. The workpiece Cyber Device is responsible for updating the actual shape of the virtual workpiece according to the phase of the ISF process. The intermediate geometries (3D meshes) of the workpiece between the initial and the final status are coming from CAD or even FEM modelling of the forming process. The cell controller component commands the other cell elements in order to complete the forming process, it communicates with other components via RTM data and service ports.

The running system let the user to select a part to produce using 3D menus or even speech command as it was presented in the previous example. The 3D visualized ISF process can be observed from arbitrary camera position and a video stream from the real operation could also be inserted into the 3D scene.

C. MoCap Controlled Virtual Humanoid Robots in a Shared Virtual Space

This scenario illustrates how to use VirCA for bilateral collaboration in the virtual reality. Two users in different locations control virtual humanoid robots using full body motion capture devices. The two robots are taking place in the same virtual room and can interact with unreal objects e.g. kick a virtual ball.

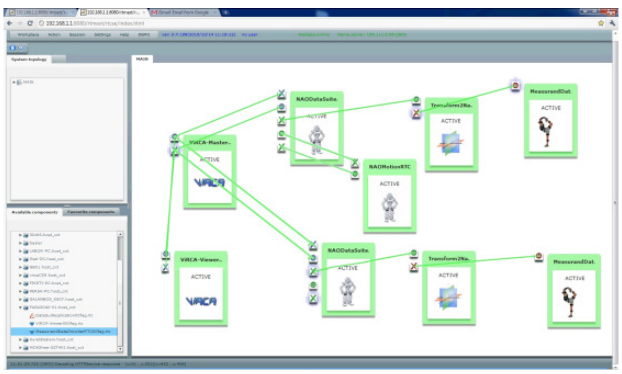

Fig. 7. MoCap controlled humanoid robots share a common virtual environment while the human operators are in different locations

The assembly contains two similar branches according to the two locations [Fig.7]. At the remote sites, instances of the same RT-Components are running. A branch consist of an RTC for the motion capture suite (Measurand), an RTC which makes transformation between the data suite data type and the input format of the robot and a Cyber Device that represents the NAO humanoid robot in VirCA. Real NAO robot can also be connected to this Cyber Device. In this case, the real robot gets commands from the virtual robot via an RTM service port (RTM wrapper for NAO API) and follows the captured motion. The two branches were connected to the VirCA core master instance, while a VirCA viewer clone was running at the other location let the remote operator to observe the shared space. In this pilot experiment the two locations was in MTA SZTAKI (Hungary) and the Narvik University College (Norway). The computers involved in the experiment were connected to the same virtual private network (VPN).

D. Virtual Sensors

Simulated sensors are quite common in robotic simulators such as MRDS or Player/Stage. Cameras, Laser range finders, Ultrasonic distance sensors can be easily modelled and applied as the real ones in a simulated scenario. VirCA provides a

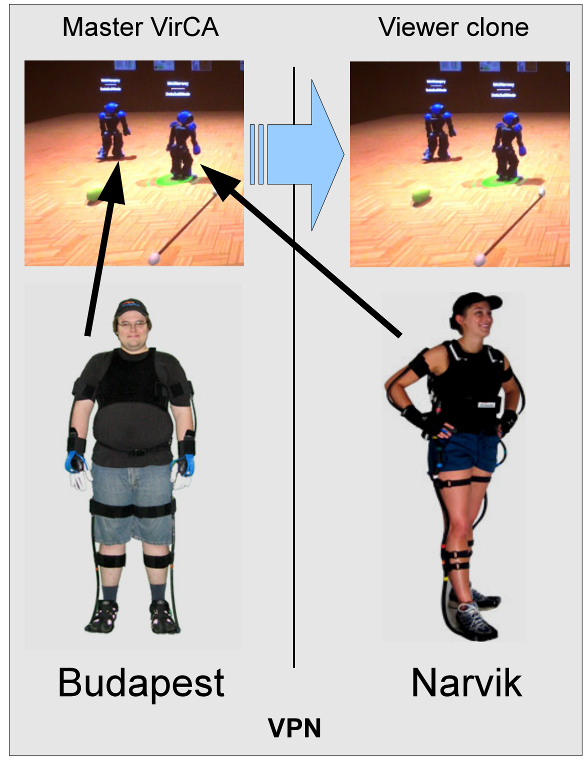

Fig. 8. Structure of the bilateral operation

native RT-Middleware background for such virtual sensing. In extant RTM-based robotic systems virtual sensorization can be implemented with minimal overhead as the processing components [25] are applicable without modification.

For example, it is a common problem in the evaluation process of mobile robot navigation algorithms, that it must be tested in various environments (corridor, office, warehouse, etc.). In VirCA, one can test a robot in any kind of virtual environment, while the real robot is simultaneously moving in an empty room and its motion is tracked by external instrumentation in order to update the position and orientation of the virtual robot in VirCA. In this case, the virtual sensors are attached to the virtual robot while the critical issues e.g. drives, wheel-floor contact, etc. do not have to be modelled in the simulation. The benefits are more obvious when the applied algorithms requires expensive (e.g. depth camera) or non-existent sensors or if the real robot cannot carry a sensor device due to its size or weight.

A further application of virtual sensors is related to Human Computer Interaction (HCI). Figure 9 shows a VirCA assembly where a virtual humanoid robot is equipped with a ”touch sensor” that capable of detecting the surface of an object what is in interference with the sensor. The robot is controlled with a motion capture suite as it is introduced in the previous subsection. The detected tactile information can be displayed using any appropriate device. In this example a cognitive infocommunication [26] method is applied to transform the tactile information into auditory icons [27].

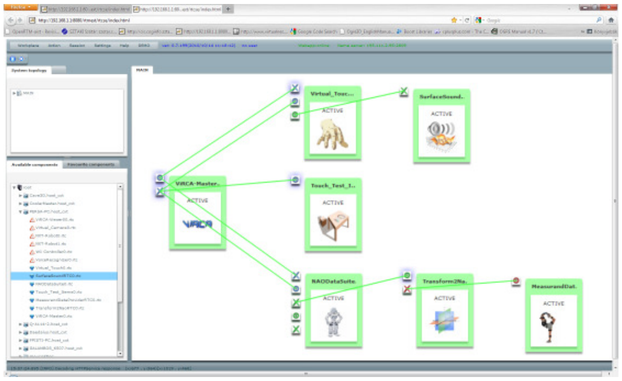

Fig. 9. Virtual tactile sensor and audio display in VirCA

V. SUMMARY

In this paper, VirCA (Virtual Collaboration Arena) is proposed as a virtual intelligent space extension for openRTM-aist. Capabilities of the VirCA system are introduced by four example scenarios giving a guideline for the potential users.

ACKNOWLEDGMENT

The research was supported by the Hungarian National Development Agency, NAP project NKTH-KCKHA005 (OMFB-01137/2008).

REFERENCES

[1] AIST, OpenRTM-aist. [Online]. Available: http://www.openrtm.org

[2] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and W. Yoon, "RT-middleware: distributed component middleware for RT (robot technology)," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alta., Canada, 2005, pp. 3933–3938.

[3] N. Ando, T. Suehiro, K. Kitagaki, and T. Kotoku, "RT(Robot Technology)-Component and its standardization - towards component based networked robot systems development," in *2006 SICE-ICASE International Joint Conference*, Convention Center-BEXCO, Busan, Korea, 2006, pp. 2633–2638.

[4] Willow Garage, Robot Operating System. [Online]. Available: http://www.ros.org

[5] Microsoft, Microsoft Robotics Developer Studio. [Online]. Available: http://www.microsoft.com/robotics/

[6] The Orocos Project, Open Robot Control Software. [Online]. Available: http://www.orocos.org

[7] M. Amoretti and M. Reggiani, "Architectural paradigms for robotics applications," *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 4–13, 2010.

[8] W. D. Smart, "Is a common middleware for robotics possible?" in *Proceedings of the IROS 2007 workshop on Measures and Procedures for the Evaluation of Robot Architectures and Middleware*, 2007.

[9] N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "Middleware for robotics: A survey," in *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*, sept. 2008, pp. 736 –742.

[10] H. Hashimoto, "Intelligent space: Interaction and intelligence," *Artificial Life and Robotics*, vol. 7, pp. 79–85, 2003.

[11] P. Korondi and H. Hashimoto, "Intelligent space, as an integrated intelligent system (keynote paper)," in *Proceedings of the International Conference on Electrical Drives and Power Electronics 2003*, High Tatras, Slovakia, Sep. 2003, pp. 24–31.

[12] A. Vámos, B. Reskó, and P. Baranyi, "Virtual collaboration arena," in *Applied Machine Intelligence and Informatics (SAMI), 2010 IEEE 8th International Symposium on*, 28-30 2010, pp. 159 –164.

[13] A. Vámos, I. Fülöp, B. Reskó, and P. Baranyi, "Collaboration in virtual reality of intelligent agents," *Acta Electrotechnica et Informatica*, vol. 10, no. 2, pp. 21–27, 2010.

[14] P. Galambos, B. Reskó, and P. Baranyi, "Introduction of virtual collaboration arena (VirCA)," in *The 7th International Conference on Ubiquitous Robots and Ambient Intelligence*, Busan, Korea, Nov. 2010, pp. 575–576.

[15] I. Fülöp, "Extending intelligent collaboration using semantic information inthe virca system," in *The 9th IEEE International Symposium on Applied Machine Intelligenceand Informatics*, 2011, pp. 237–242.

[16] ——, "Semantic services in the virca system," in *The 6th IEEE International Symposium on Applied Computational Intelligence and Informatics*, 2011.

[17] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: the design and implementation of the CAVE," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. Anaheim, CA: ACM, 1993, pp. 135–142.

[18] W. Klára, A. Ákos, M. József, T. Bertalan, and T. Péter, "New optical equipment in 3D surface measuring," *JOURNAL OF AUTOMATION MOBILE ROBOTICS & INTELLIGENT SYSTEMS*, vol. 3, no. 4, pp. 29–32, 2009.

[19] OGRE (Open Source 3D Graphics Engine). [Online]. Available: http://www.ogre3d.org

[20] BULLET Physics Library. [Online]. Available: http://bulletphysics.org

[21] ZeroC, ICE (Internet Communication Engine). [Online]. Available: http://www.zeroc.com

[22] Z. Krizsán, "ICE extension of RT-Middleware framework," in *10th International Symposium of Hungarian Researchers on Computational Intelligence and Informatics*, Budapest, Hungary, Nov. 2009, pp. 513–521.

[23] I. Paniti, "CAD API based tool path control for novel incremental sheet forming," *Pollack Periodica*, vol. 5, no. 2, pp. 81–90, Aug. 2010.

[24] EMC, Enhanced Machine Controller. [Online]. Available: http://sourceforge.net/projects/emc/

[25] G. Sziebig, A. Gaudia, P. Korondi, N. Ando, and B. Solvang, "Robot vision for RT-Middleware framework," in *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007*, Warsaw, Poland, May 2007, pp. 1–6.

[26] P. Baranyi and A. Csapo, "Cognitive infocommunications: CogInfoCom," in *2010 11th International Symposium on Computational Intelligence and Informatics (CINTI)*, Budapest, Hungary, Nov. 2010, pp. 141–146.

[27] A. Csapo and P. Baranyi, "An interaction-based model for auditory substitution of tactile percepts," in *2010 IEEE 14th International Conference on Intelligent Engineering Systems*, Las Palmas, Spain, 2010, pp. 271–276.