Virtual Collaboration Arena: research and development

István Marcell Fülöp*, Barna Reskó**, Péter Baranyi*
* Cognitive Informatics Research Group, MTA SZTAKI, Budapest, Hungary
** Alba Regia University Center, Óbuda University, Budapest, Hungary
ifulop@sztaki.hu, resko@sztaki.hu, baranyi@sztaki.hu

***Abstract*— The VirCA system (Virtual Collaboration Arena) is a component based, interactive virtual environment providing facilities for collaborative manipulation of intelligent systems. [1] In this paper the technologies, upon which the system is based are going to be introduced, then the build-up and the operation of the system itself as well as the potential of using the system.**

I. TECHNOLOGY

A. *Virtual reality: 3D visualization*

In the recent time the fast development of the visualization systems allowed the virtual reality based applications getting abroad in a wide range of use. The first step of this was the appearance of computer 3D rendering, which the laymen could met primarily in computer games.

The second step of this was the appearance of 3D graphics accelerator cards. Here we could see a beneficent interaction between the development of the technology and the utilization of this technology by the applications: in a short time accelerator cards of high performance got in the affordable category for the man in the street.

Today the actual step of this development in the history of computer visualization is the spread of "real" 3D displays. These displays generate a real stereoscopic sensation by the help of some method in the user, which enhances the experience of use. On the grounds of this method the 3D systems can be differentiated into passive and active stereo systems.

In the case of passive stereo systems the left and right image, which create the stereoscopic sense, arrive in a common channel to the user, where some passive tool sorts out the appropriate image for the appropriate eye. E.g. in the case of the color filtering method, which has been applied in the cinemas for a long while, the left and the right image are in separate color range, while the glasses worn by the user contain color filters: they pass through only the color range which belongs to the appropriate eye. The polarization method also needs to be mentioned which operates on a discipline similar to the color filtering method, but in that case the complete original color range is available.

In the case of active stereo systems the left and right image arrive in separate channels to the user, where some active tool creates the proper image. E.g. in the case of the so called "shutter glass" method the glasses of the user contain liquid crystals. These are electrically controlled in such a way, that by turns the left and the right one passes through the light. In this case the two images arrive interlaced in time from the display and the glasses are synchronized to it. It means, if the left image is visible on the display, the left side of the glasses passes through and if the right, the right side.

Figure 1. nVidia 3D Vision system

Such systems appear today already in households, the 3D Vision system of nVidia – graphics card, display, glasses – can be bought for an obtainable price. (Figure 1) In the case of this system the display is an LCD display of 120 Hz which displays the left and right image with 60-60 Hz. The so called autostereoscopic systems without any glasses should be mentioned as well, where on a special display many images are displayed at once: due to the placement of the pixels and the optical grid in front of the display, from different places different images are visible, in that way different images arrive to the left and right eyes of the user.

B. *Collaboration: modular architecture*

In the case of modern information systems modularity and compatibility are important aspects. It means, that the system does not consist of a monolithic unit, but of many modules, which can be treated as separate units. This build-up has first of all economic advantages. In that way, in the case of a new demand – if the modules are functionally well delimited –, the complete system need not be changed, but only the module which is responsible for the appropriate functionality. So in the case of a modular system, the single modules can be changed, replaced separately without changing other modules.

Of course, this can be only realized, if the modules are "compatible" with each other, which means, that they function according to given standards, provide standard interfaces to be connected to each other, in other words, they realize the standards, which provide the base of the cooperation.

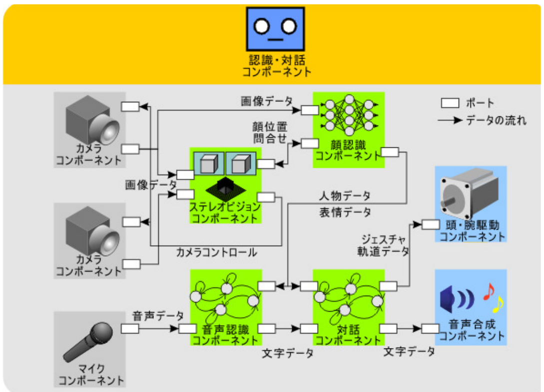

Figure 2. RT-Middleware based system

Such a standard system in the field of robot technology, which is supported by the Japanese government, is the RT-Middleware (Robot Technology Middleware), which aims the modular build-up of the industrial systems. With the spread of this standard system it can come true, that different parts of an industrial system can be ordered from different manufacturers by the user, from wich the decrease of the purchasing price can be expected.

RT-Middleware uses standard channels, so called ports for connecting different components. Ports realize one-way channels: they can be either providers, i.e. outputs or consumers, i.e. inputs. Ports can be data ports, through which simple date flows flow or so called service ports, through which functions can be accessed. Components can be connected by connecting their ports. (Figure 2) Ports can be connected, if their descriptions are identical, but their directions are opposite.

RT-Middleware uses the CORBA standard for data transfer and with the extension developed in MTA SZTAKI – "Turbo RT-Middleware" – it can use the ICE standard as well. The RT-Middleware system consists of a name server, arbitrary number of components connected to it together with an editor. The components register into the name server telling their own address. With the editor the components registered into the name server can be queried and connected graphically with each other. Therefore the connected components contact each other on the grounds of the data in the name server, the data transfer comes off between them.

II. DEVELOPMENT

A. Build-up, architecture

The VirCA system developed in MTA SZTAKI has a modular build-up. Its central part is the virtual reality handler component together with the 3D visualization. This component on the one hand operates as a database, which registers the objects in the virtual reality and the events related to them and on the other hand provides the connection to the user in the form of 3D visualization and user interaction. VirCA uses the Ogre3D graphical engine for 3D visualization and for the physical simulation of the motion of the objects the Bullet physical engine. The different components of VirCA are connected together with RT-Middleware and for the data transfer through internet the ICE communication engine is used.

To the virtual reality handler component arbitrary further components can be connected, which realize some kind of functionality. Among them one type is the "cyber device" which represents either a real or a pure virtual device in the virtual reality. In the case of a real device the cyber device realizes the connection between the "true" reality and the virtual. In the case of a pure virtual device there is no real device behind the cyber device, but only a program. The connection between the cyber device and the virtual reality is two-way: on the one hand the cyber device can manipulate the virtual room through its representation and on the other hand it can receive user commands from the virtual room.

B. Properties, fields of use

One important attribute of the VirCA system is the integration of information. It means on the one hand, that by the help of the "real" 3D visualization all the three dimensions are available for the representation of the information. On the other hand, in the case of a complex process, different informations do not arrive in separate channels to the user – e.g. in separate displays –, but according to the reality, they can be displayed on a spatial representation. This can largely increase the efficiency of the human-machine communication. In the interactive virtual environment the user can control the different devices through a standard interface. For this, arbitrary input devices can be available, which can pass the given information in the appropriate, arbitrary way to the system. E.g. moving the 3D pointer with the mouse a cyber device can be selected or with the help of the voice recognizer the cyber device can be addressed by saying its name. (Figure 3)

One other important attribute of the VirCA system is the distributed system approach. It means, that the devices represented in the virtual reality can be controlled in the same way without reference to their location in the reality. That is, VirCA conceals the complex network connections, the user can simply select the required device in the virtual reality, as if it would be available on the local machine.

The VirCA system can be used to create virtual test cases consisting of many different systems. It means, that the cooperation of devices far from each other in the room can be tested without the need of delivering them to the same place. (Figure 4) E.g. in the case of large industrial robots it would be very expensive. It differs from the simulation of the devices in the virtual reality in that way, that in this case the representations of the real devices are in the virtual reality, whose motions follow the motions of

Figure 3. Controlling cyber device from the virtual reality

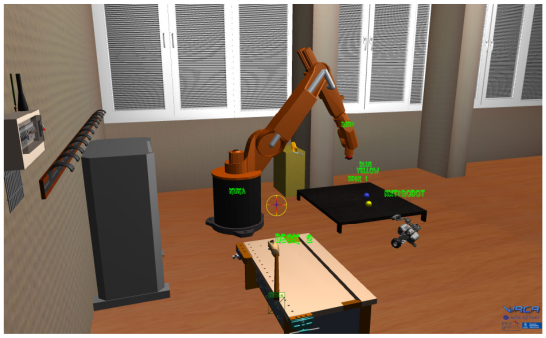

Figure 4. Collaboration of robots in the virtual reality

the real devices, so that on the one hand, with the development of simple tracker systems the development of expensive simulators can be avoided and on the other hand, the virtual representation tracked in this way corresponds better with the real device, than the simulated. So, e.g. before the purchase of an expensive device it already can be tested, whether it suits the cooperation demands of the application.

The VirCA system can be applied as a virtual information room between different devices. E.g. the state of the virtual reality can be refreshed with different kinds of sensor and tracker devices corresponding to the true reality, so that other devices on the grounds of the virtual reality can manipulate in it. So the single devices need not have all the abilities they use, but they can use the abilities of other devices through the virtual reality. E.g. a mobile robot can orientate itself on the grounds of the virtual reality refreshed by a sensor. Moreover, in the virtual reality such sensors can be simulated as well which do not exist in the reality or could be purchased only very expensive. So certain devices can be tested without the existence of other devices which are required for their usage. It facilitates the separate, from each other independent development, testing of the devices and in the case of purchase, the trial of different devices. E.g. in the case of a mobile robot it can be decided easily, what kind of camera should be selected for the realization of the orientation. But in the information room, not only sensor and tracker devices, but only in program realized intelligence can be “placed” as well. So the “plug’n’play” knowledge can be realized, i.e. the intelligence in the virtual reality can be used by other devices.

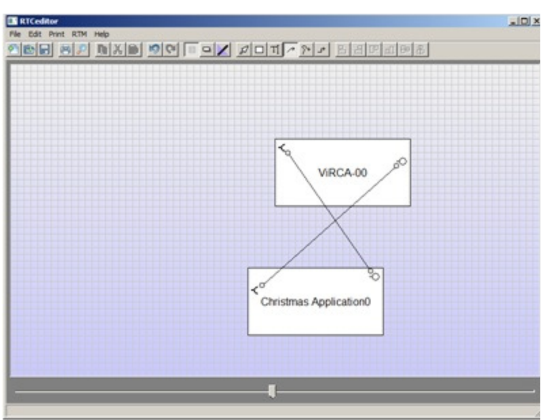

Figure 5. Using the RT-Middleware editor

C. *Usage, development of components*

The usage of the VirCA system starts with running the installer which can be found on the homepage. It installs the virtual reality handler component itself and other programs required for its usage onto the computer. The VirCA system can be used with other components in such a way, that first the RT-Middleware name server is started, to which the virtual reality handler component registers as started, then the editor is started, with which after connecting to the name server, the virtual reality handler component can be connected with arbitrary other components. (Figure 5)

To the VirCA system an own cyber device can be developed basically in three ways. Either an own RT-Middleware component is written completely, or it is inherited from the so called “cyber device base” component, or the so called cyber device API library is used. The first choice is the hardest, but has the most facilities, while the last is the easiest, but only a cyber device with limited capabilities can be developed with it.

III. Research

A. *Goals and tools*

In case of collaboration of intelligent systems it is a common problem how the functionality of the single components should be shared. The more accurately the information is shared, what the component is capable of, the more efficient collaboration can be realized. Moreover, not only the functionality needs to be shared but also the way how the functionality can be accessed. Since the functionality of intelligent systems, e.g. robot cells is related to some ideal or material reality, the formal representation of this reality seems to be inevitable.

In the conventional way, information about the functionality is shared as an informal description, e.g. natural language text. It is formulated by the developers of the component and interpreted by the developers of other components in order to access the functionality of the component. It requires a big human effort to read and interpret all the descriptions, how the communication should be realized, and cannot be automated, as natural language is difficult to be interpreted by non-humans. To make this process possible for intelligent systems, a formal representation of the reality needs to be introduced. If the functionality of a component needs to be formally represented, also the domain, i.e. the concepts required for the description of the functionality needs to be represented.

B. *Ontology handler component*

For the formal representation, a possible choice is the adaptation of ontology which is a standard representation of a domain of concepts. With ontology, the “meaning” can be standardized, so that interpretation becomes much easier. The VirCA system can also be extended in order to support this formal representation of information. As the 3D visualization component displays the content of the virtual reality graphically, an ontology handler component can display it formally represented in an ontology. This way the content of the virtual reality can be described standardized and refreshed according to the changes. This virtual reality is the domain of the functionality of intelligent systems, so the functionality can be represented in this ontology. The task of the ontology handler component is to collect semantic information of the virtual reality itself, of the functionalities of the intelligent systems, and of the requests of the user. On the basis of this information an inference can be performed, whether an appropriate collaboration of the intelligent systems can fulfill the request of the user.

C. Component wrapper

The original components of the VirCA system do not support semantic communication, but they can be "wrapped up" into semantic components using universal wrapper components. The formal, but not semantic "language" of the components needs to be matched with the formal, semantic representation of the ontology. It means, for a given semantic input, the wrapper component should send a command to the original component in its language, and as the result is returned, the wrapper component should send a semantic output back to the system. For this, the formal language of the component should be completely known, which is possible in most of the situations, but this way, also a human speech component of natural language can be wrapped up with intelligent methods if the speech is restricted to an enough simple grammar.

Acknowledgment

The research was supported by HUNOROB project (HU0045, 0045/NA/2006-2/ÖP-9), a grant from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Hungarian National Development Agency.

References

[1] Á. 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.