

WebVR

Facilitating the Development of Multi User Scenarios VR Experiences Feedback from the Industrial VR Field

Geoffrey Subileau (geoffrey.subileau@3ds.com), Fivos Doganis (fivos.doganis@3ds.com), David Nahon (david.nahon@3ds.com), Nathalie Sarrazin (nathalie.sarrazin@3ds.com)

Dassault Systèmes, France, September 21, 2016

Abstract

The more the VR technology evolves the more user scenarios become powerful and various. But for the developer it becomes more and more complex to develop VR experiences in a scalable and efficient way for a wide range of user scenarios and real world physical contexts. We would like to share our experience on that topic and ask how WebVR will adapt to the many devices to come, for the sake of consumer VR adoption.

What Dassault Systèmes does in the VR industry?

Dassault Systèmes (DS), the 3D Experience Company, is the leading computer-aided-design and product-life-cycle software provider (editor of CATIA and Solidworks). It provides business and people with virtual universe to imagine sustainable innovations.

For more than 20 years, our customers have been using VR experiences for training, collaborative mockup review and ideas sharing. It helps their R&D, design or training departments to save a lot of time and money. Those VR experiences are consumed directly into CATIA or were built from Unity-like real-time 3D engines: 3DVIA Studio and 3DVIA Virtools. They can run on several VR systems like CAVEs, stereoscopic 3D walls, HMDs...

DS serves 12 industries. VR use cases are countless, but here is one example of VR experience for each:

- **Aerospace and Defense:** Yellow dog virtual training on aircraft-carrier
- **Architecture, Engineering and Construction:** Space planning immersive review
- **Consumer Packaged Goods and Retail:** Virtual in-store shopping review and analysis
- **Energy, Process and Utilities:** Mission planning review in power plants (before doing anything dangerous in real, we verify virtually how to do it is safely, and train the field operators)

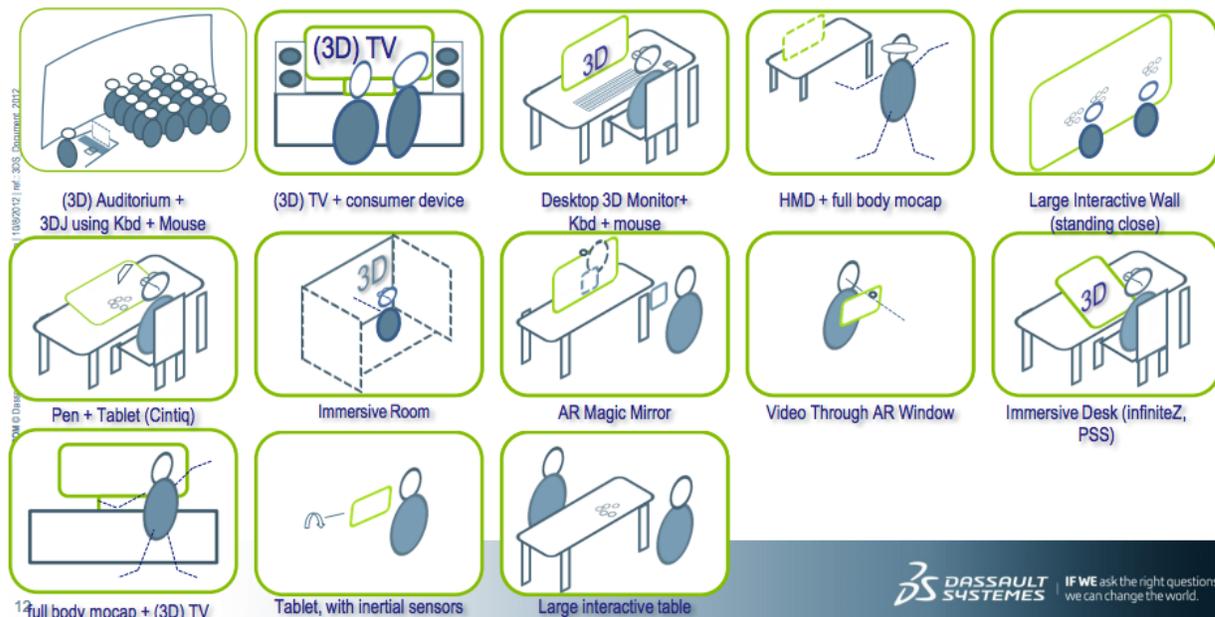
- **Financial and Business Services:** Complex and massive data visualization
- **High-Tech:** Concept ideation, design and consumer experience review
- **Industrial Equipment:** Heavy equipment immersive sales configurator
- **Life Sciences:** Collaborative session of complex molecule visualizations (see in 3D what scientists have in mind), Body motion analysis for healthcare studies
- **Marine and Offshore:** Field of view and ergonomics review of the ship's control room
- **Natural Resources:** Ground exploration review
- **Transportation and Mobility:** Driving experience

Challenge of developing for multi VR user scenarios and physical contexts

VR applications are user-centered designed. This means that the application has to know as much as possible about the user's body and the physical interaction context. We speak about *User's Body and Interaction Context Aware Computing*.

The more the VR/AR technology evolves the more user scenarios become powerful and various. But for the developer it becomes more and more complex to develop VR experiences in a scalable and efficient way for a wide range of user scenario and physical context. Theoretically the task a user is supposed to do in VR has to be compliant with the devices capacities and physical space. So the VR developer needs to adapt user scenario and interaction paradigms according to the devices and physical context.

Here is a list of user scenarios we target:



Slide from [iV Evangelist](#)

WebVR features proposal

If we consider that more and more VR devices will hit the market, we think consumer VR adoption will require the possibility to make VR devices converge easily, so developers could create new kind of user scenarios which will take user's real physical space and device capacities into account. Developers will want to make experiences run across multiple devices, and dynamically adapt interactions depending on device capacities detected by the browser. That being said, this convergence phase in the consumer VR/AR device industry is not there yet.

Speaking of now, here is a list of features or concept requests that we would love to be taken into considerations by WebVR and browser vendors.

IO Device abstraction

Each new VR device can offer new user interactions possibilities, specially crafted for certain tasks in VR (like we use to do in real life with tools).

Like it is in the VR industrial/research field, there will be a lot of consumer VR devices to support.

VR input/output SDK and a standard device abstraction layer seem necessary (and maybe more: device capacities introspection, user's body and physical space description model....)

For instance, in the industrial VR, many use the open source *Virtual-Reality Peripheral Network* (VRPN). It is a set of classes within a library and a set of servers that are designed to implement a network-transparent interface between application programs and the set of physical devices (tracker, etc.) used in a VR system. VRPN also provides an abstraction layer that makes all devices of the same base class look the same; for example, all tracking devices look like they are of the type *vrpn_Tracker*.

- Virtual Input/Output Device Class:

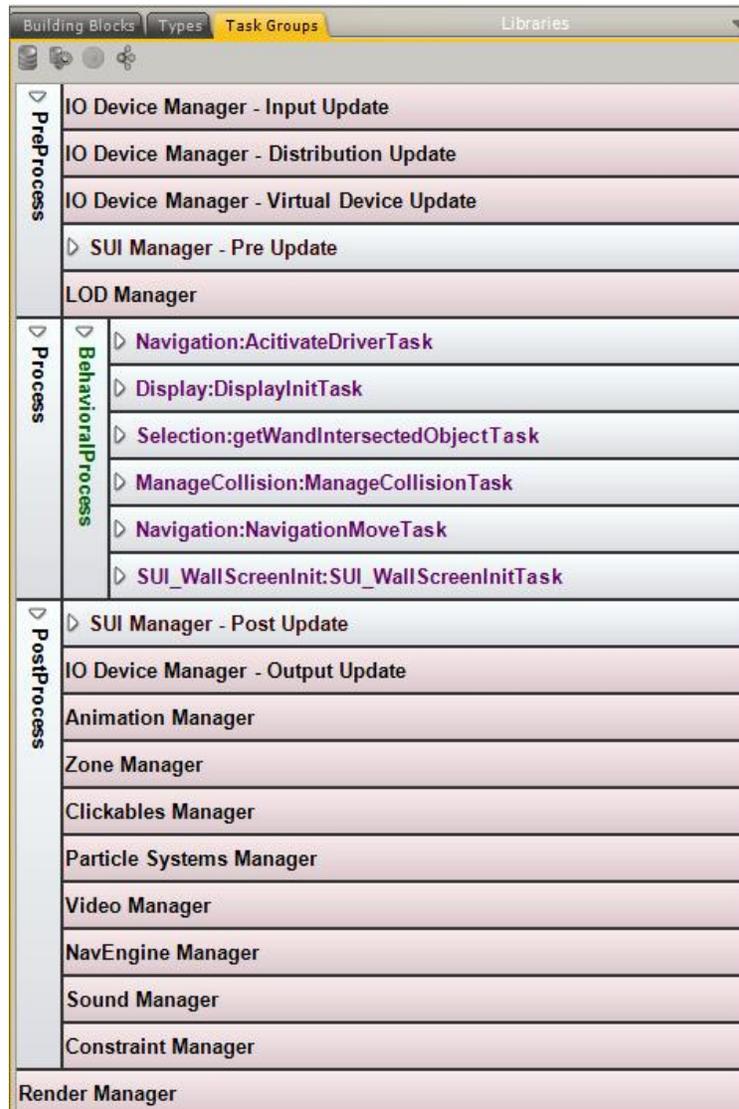
Abstract interactions code as much as possible from the hardware. You may want to change the input device without rewriting your navigation script, or only the part where you bind the input devices to a virtual IO device objects. For instance, in your interaction script, you would use values from a virtual position/orientation tracker to move a head entity, and bind the actual tracker object to the virtual one in a separated configuration script.

- IO Device SDK:

Facilitate the development and maintenance of drivers by VR hardware vendors themselves.

IO devices JavaScript callbacks:

Like the `requestAnimationFrame()` function, we would need an **input update** and **output update** callback functions: virtual input device states could then be updated at the very beginning of the frame before any other behavioral scripts. An output device callback can be useful to update a force feedback device for instance.



Example of the 3DVIA Studio Re-arrangeable Tasks Scheduler for each frame

Other requests indirectly related to WebVR specs

WebRTC (related to Mixed Reality scenario)

We've been exploring social VR scenario. For instance, [Never Blind In VR](#), is a mixed reality scenario using a HMD and depth sensors to allow a user wearing a VR headset to see his own body, perceive the real surrounding world and interact with it, as well as have social interactions with other people in the room. In order to do this scenario with distant users (immersive tele presence), we would have this kind of requests:

- get RGBD texture from getUserMedia()
- standardization of RGBD codecs

More authoring/debugging tools for VR in the developer tools

- detailed render loop profiler
- minimal WebGL scenegraph representation
- emulation / record/replay of VR session