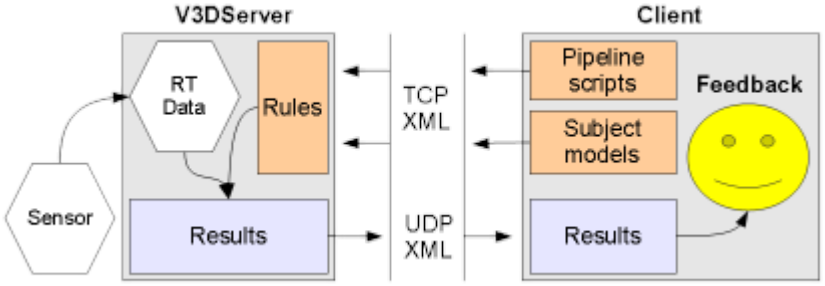
TCPIP

Overview

Visual3DServer is a desktop application that can stream data from multiple real-time hardware systems (sensors), process and then transmit that data (results) to your client application over the network. The rules for processing real-time hardware data are provided by your client application. The following diagram outlines the client server data flow.



Client Applications

You can develop your client applications one of two ways:

1) Visual3DServer Client Library (V3DSClientLib)
Use this simple C library to manage the connection and data flow between your client and Visual3DServer. V3DSClientLib works with C, C, Matlab and Python, and may also work with other languages that support C bindings. Currently, the V3DSClientLib components are installed with Visual3DServer in the **SDK** folder. **Pro:** High level, fast C functions; data 'push' callback function; hides all Visual3DServer TCP/XML protocol details; library comes with Visual3DServer.\\ **Con:** Relies on V3DSClientLib, which is only available for Win32 currently. 2) [[Other:Visual3DServer:Documentation:XML_Protocol Visual3DServer TCP/XML protocol]]
Write your own code to manage the connection between your client and Visual3DServer. You can do this on any platform (e.g., Windows, Mac OS X, Linux, iOS, Android) using any programming language that supports TCP/IP communication (e.g., C, C, objective-C, Java, Python, Matlab). Note that Visual3DServer broadcasts on TCP port **12099**.

Pro: Allows client development on any platform and with any language that supports TCP/IP.
Con: Visual3DServer TCP/XML protocol must be implemented as needed; results in longer programs, needing more testing and debugging.

Client Requests

Once connected, the your client sends messages to Visual3DServer and then waits for a response. Client messages can include a Visual3D pipeline script and subject specific model, which tells Visual3DServer how to process the real-time hardware data. Visual3DServer responds to your client messages by sending back processed data and status indicators. Your client then displays the data as

feedback to the user. You can also further process the data in your client as part of a larger solution.

Supported Pipeline Commands

(**This Section is under construction**)

Visual3DServer supports the following pipeline commands:

- Center of Pressure (COP) Path
- Center of Mass (COM)
- First Derivative
- Force
- Ground Reaction Force (GRF)
- Joint Angle
- Joint Moment
- Joint Power
- Joint Rotation
- Joint Velocity
- Multiply by Constant
- Subtract Signals

Subject Specific Models

Your client application can send a message to Visual3DServer that contains a subject specific model. Visual3DServer will then apply this model to the real-time data on a frame-by-frame basis. Different clients can provide different models for the same real-time data.

A subject model is first built using Visual3D. The model is then saved to disk in XML format, from within Visual3D, using the pipeline command:

Model Management → Save_Model_Template_XML

This XML file is then read by your client application and sent on to Visual3DServer at the appropriate time in your client workflow.

From:
https://has-motion.com/wiki/ - **Wiki Documentation Site**

Permanent link:
https://has-motion.com/wiki/doku.php?id=other:visual3dserver:documentation:tcpip&rev=1721230991

Last update: **2024/07/17 15:43**