accenture

consulting | technology | outsourcing

**Author:** Prashanth CM, CFPS
prashanth.cm@accenture.com

Accenture Services Pvt Ltd
Divyasree Techno park, Sy No 36/2,
KR Puram Hobli, Whitefield, Bangalore – 66,
Karnataka, India

High performance. Delivered.

# FP Veracity for Virtual Reality

**Presenter**: Heidi L Malkiewicz, CFPS
heidi.l.malkiewicz@accenture.com

# Agenda

- Abstract
- Application of VR
- Architecture and Different components
- Application of FP
- Application Boundary
- Data and Transaction Function
- General System Characteristics
- Recommendations and Challenges.
- Conclusion

# Abstract

- Virtual Reality (VR) refers to a high-end user interface that involves real-time simulation and interactions through multiple sensorial channels. VR is able to immerse you in a computer-generated world of your own making: a room, a city, the interior of human body. With VR, you can explore any uncharted territory of the human imagination.
- Architecture of VR system would mainly include Input Processor, Simulation Processor, Rendering Processor and World Database. Virtual reality is a form of technology which creates computer generated worlds or immersive environments which people can explore and in many cases, interact with.
- Virtual reality has its advocates and opponents which are mainly due to a lack of understanding about this technology and its capabilities. Unrealistic expectations coupled with lack of awareness regarding technical limitations means that for many people, virtual reality is difficult to grasp or even take seriously.
- Although there were indications of VR in the sixty’s, its started gaining more and more prominence only during the recent past with its application in different areas like military, medical sciences, gaming etc.

# Abstract (Continued)


- Virtual reality is often used to describe a wide variety of applications commonly associated with immersive, highly visual, 3D environments. The development of CAD software, graphics hardware acceleration, head mounted displays, database gloves, and miniaturization have helped popularize the notion.
- As it gains importance and as and when there are development needs for companies to develop VR applications, there can also be increasing need to measure the same and that’s when people might turn towards IFPUG FP sizing technique.
- This paper, as the name indicates would mainly deal with FP sizing methods application for a VR application. Here, we will see whether sizing of VR application can be a reality.

# Multimedia vs. Virtual Reality

- Closely related technologies sharing some similar hardware and data usage
- Multimedia (MM) - Computer systems allowing for integrated access to a range of data through the means of stimulating human senses using digital technologies
- Virtual Reality (VR) - Computer systems able to combine a mixture of real world experiences and computer generated material to allow for simulated real world representation

| Multimedia (MM) | Virtual Reality (VR) |
|---|---|
| Multimedia covers the integration of:<br>- images, video and graphics (both still and animated); including raster and vector data, maps, photographs , text; in a variety of forms including alphanumeric databases, sound<br>- (potentially) smell and taste | VR addresses the construction of artificial worlds, with clear spatial dimensions databases for VR can structure and store data using methods |

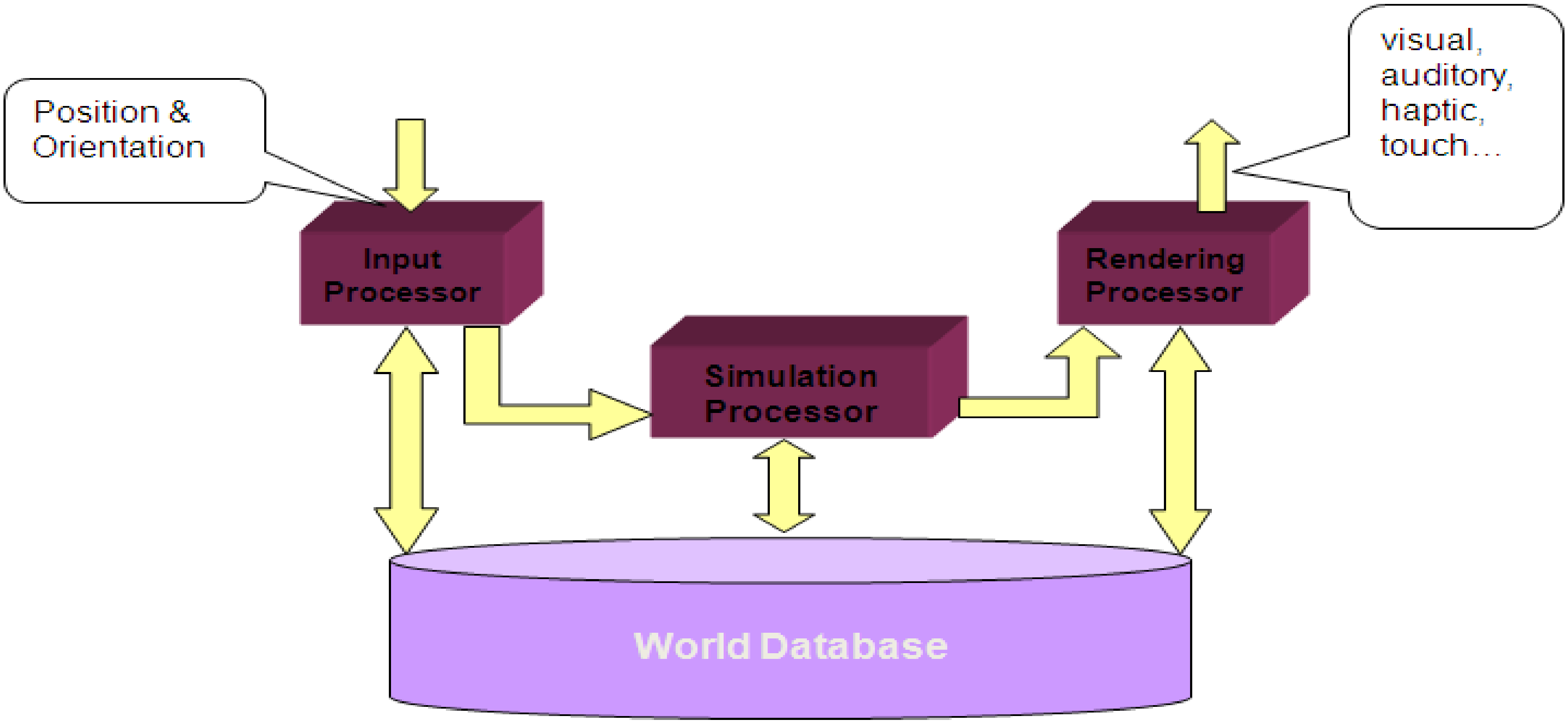
- Among the important concepts in MM and VR are database construction and integration, and user navigation and interaction
  - the former can be achieved using existing datasets or 'on-the-fly'
  - the latter can be done real-time or can be pre-determined, can be in a fixed sequence or can be interactively led

# About Virtual Reality

# Application of VR

- **Examples**: Architecture, Training, Medicine, Engineering and Design, E-Commerce, Entertainment, Manufacturing

# Architecture of VR Application



# Components of VR Application

## Input Processor

- Control the devices used to input information to the computer. The object is to get the coordinate data to the rest of the system with minimal lag time.
- Keyboard, mouse, 3D position trackers, a voice recognition system, etc.

## Simulation Processor

- Core of a VR system.
- Takes the user inputs along with any tasks programmed into the world and determine the actions that will take place in the virtual world.

## Rendering Processor

- Create the sensations that are output to the user.
- Separate rendering processes are used for visual, auditory, haptic and other sensory systems. Each renderer take a description of the world stat from the simulation process or derive it directly from the World Database for each time step.

## World Database (World Description Files)

- Store the objects that inhabit the world, scripts that describe actions of those objects.

# Virtual Reality Modelling Language (VRML)

Most exciting is the development of VRML on the World Wide Web.

In addition to HTML (Hypertext Markup Language), that has become a standard authoring tool for the creation of home pages, VRML provides three-dimensional worlds with integrated hyperlinks on the Web. Home pages become home spaces.

**Characteristics Of VRML:**

- Not a programming language like C++ or java
- Descriptive (rather than procedural) like HTML
- File formats contains human readable and editable ASCII text

# Application of FP

# Application of FP on VR applications

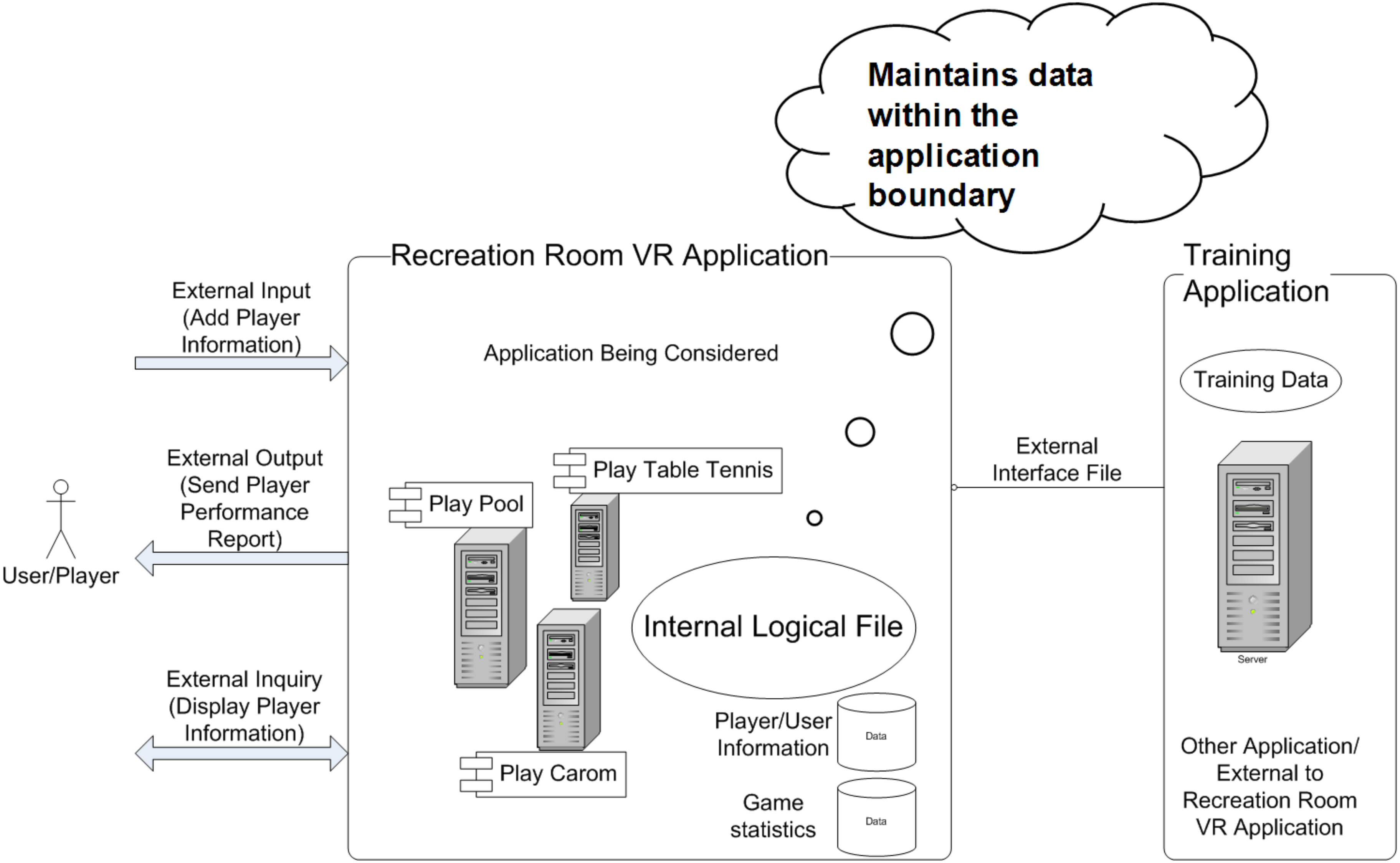
Virtual Reality (VR) has sparked many people's imaginations, but writing VR programs remains difficult. Besides the obvious problems in managing the arcane I/O devices (trackers, gloves, etc), the developer must allow the participant to operate effectively in the immersive environment.

Virtual environments present a new medium for both the participant and the author. It is believed that the best way to accelerate a new medium such as VR is to provide tools that allow people without highly technical backgrounds to create VR programs. These authors must be able to try different nuances of an idea without spending time waiting or re-implementing; the author must be able to easily ask "what if" questions.

Like it is important for the author to ask “what if” questions, the same would be applicable to the Function Point analyst to understand what the final user functions are.

It is assumed in this paper that the **counting scope** is to size all the functionalities that are present in the application and the **purpose** being to compare one VR application to the other similar VR application built in a different way. This would require an application baseline size to be created, and the next step would be to draw an **application boundary.**

# Recreation Room VR Application



# Application Boundary

**Application Boundary:** Boundary is the border between the application and its Users. Each application is conceptually enclosed within its boundary.

In this paper, we would be looking at one of the examples of a VR application and see how Function point rules and guidelines apply.

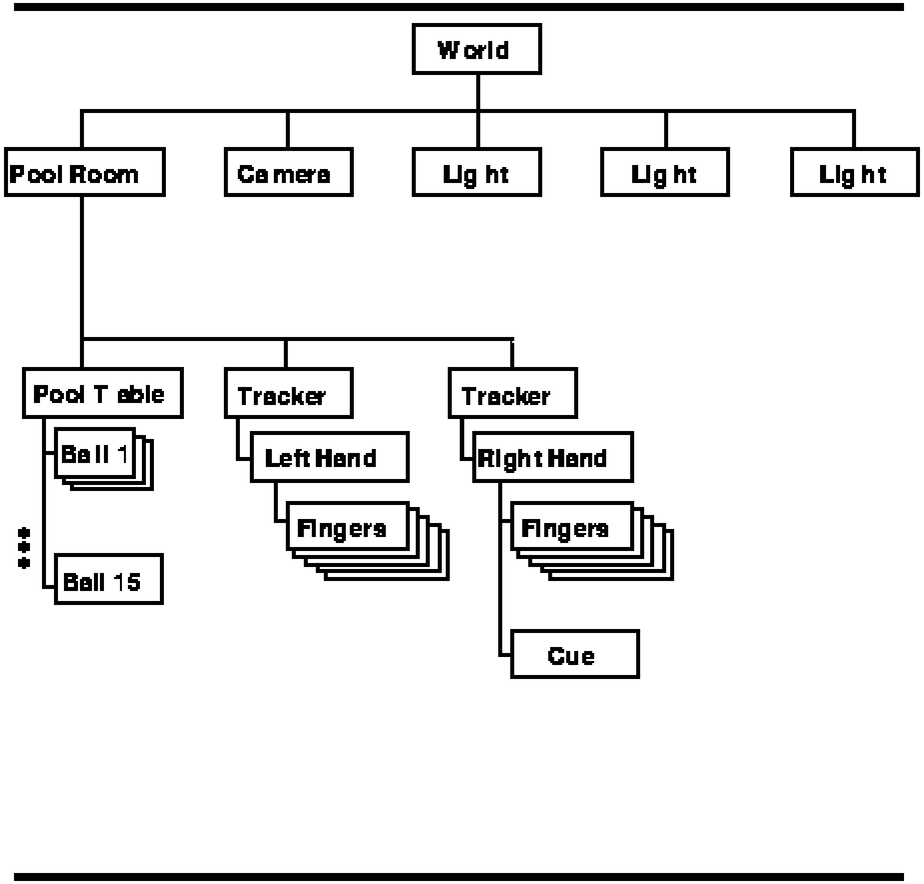
I have this fictitious VR application which I have named “Recreation VR application”. “Recreation VR application” is a boundary that I have created that would allow the users to enter into the Recreation room and choose any of the games to play. The options that the user has to play are Pool, Table Tennis or Carom.

The “Recreation VR application” will be a combination of physical components being

- Input Processor
- Simulation Processor
- Rendering Processor
- World Database

As in any other types of applications, defining a boundary can be a challenge when there are multiple applications interacting with each other to provide a solution to the business problem.

# Data/Object Structure




- This would be the hierarchy of  objects for part of the Recreation VR application with just the Pool game considered and the objects involved in delivering what is required for the user to play pool using a pool table, Cue, Cue ball, Colored balls etc.

- The Logical files will need to be determined using this hierarchical/tree structure of the objects.

# Data Functions

# Data Functions - ILF

**Data Functions:**

Data functions are the functionality given to the user to meet internal and external data storage requirements. These are part of Base Functional Components (BFC).

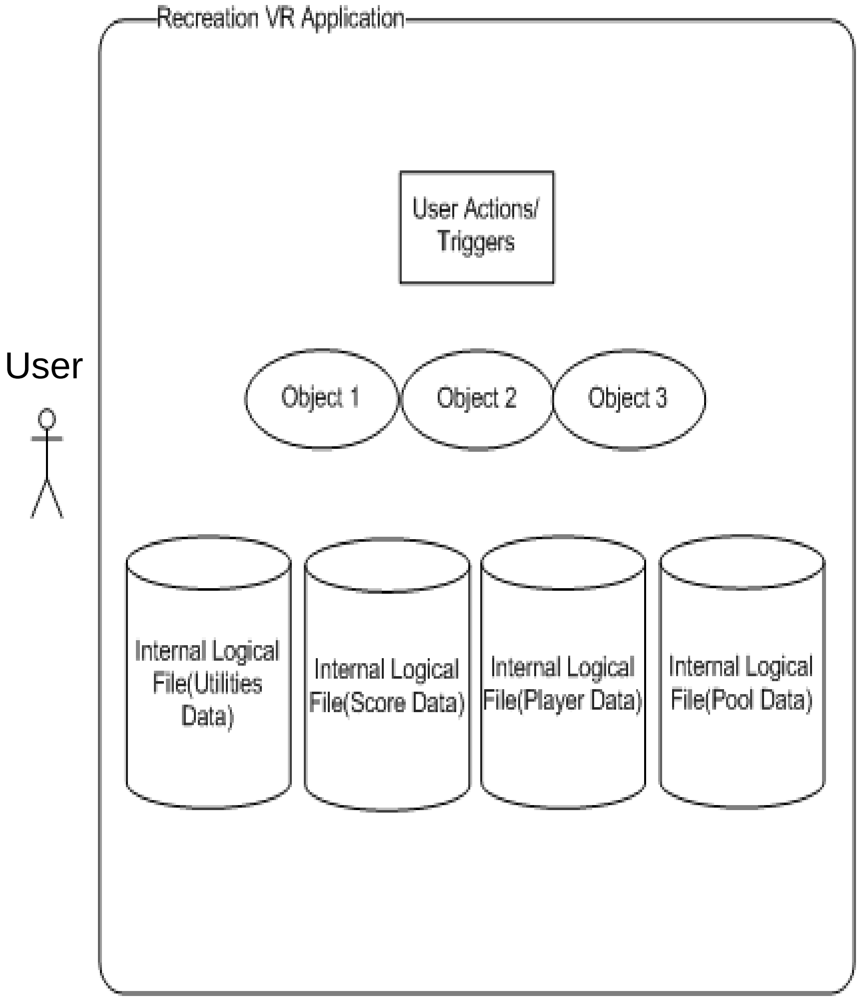
**Internal Data Storage :**

Data storage internal to an application forms Internal Logical Files (ILF).ILFs correspond to data maintained within boundary of an application. In other words, data maintained by an application through processes like Create, Edit and Delete is essentially ILF.

**Example:**

In the "Recreation VR Application, there are different details of the user/player that the application stores like the name of the player, games played, scores etc. Player data, Score data hence stored become ILF's for "Recreation VR Application".

# Data Functions – ILF (Continued)




- Users interacting with the "Recreation VR Application" would be able to use the doors of the room, the light switches, windows of the recreation room etc…which has been classified as one logical area for the application under utilities data.
- Each user/player can have different sets/combinations of utilities that he can set to and this information will also be stored for each user like the different lights switched on when a game was played, windows and doors that were open/closed etc.
- The administrator of the VR application will be able to move objects from one place to other like placing the pool table at one place in the Virtual room, moving the related components if pool table that would be stored under pool data logical file.

# Data Functions - EIF

**External Data Reference:**

External data storage corresponds to EIFs (External Interface File), data referenced by application under discussion, but maintained outside its boundary.

**Example:**

In the “Recreation VR Application”, there can be different training data that can be seen for trainings attended by a particular user/player. To view data related to the players trainings, there can be some interfaces to the training VR application to view training data. Hence the training data can be considered as an EIF.

Based on the scores gained in the Recreation VR application, the user can also be recommended to go to the Training room and get some additional trainings on the recommendations made in the “Recreation VR application”

# Transaction Functions

# Transaction Functions – External Input

**Transaction Function:**

Transaction Functions are functionality given to the user to process data. There are three types of transaction functions, which are also part of Base Functional Components (BFC). These functions can be visualized as data in motion, while data functions corresponding to stationary data of the application.

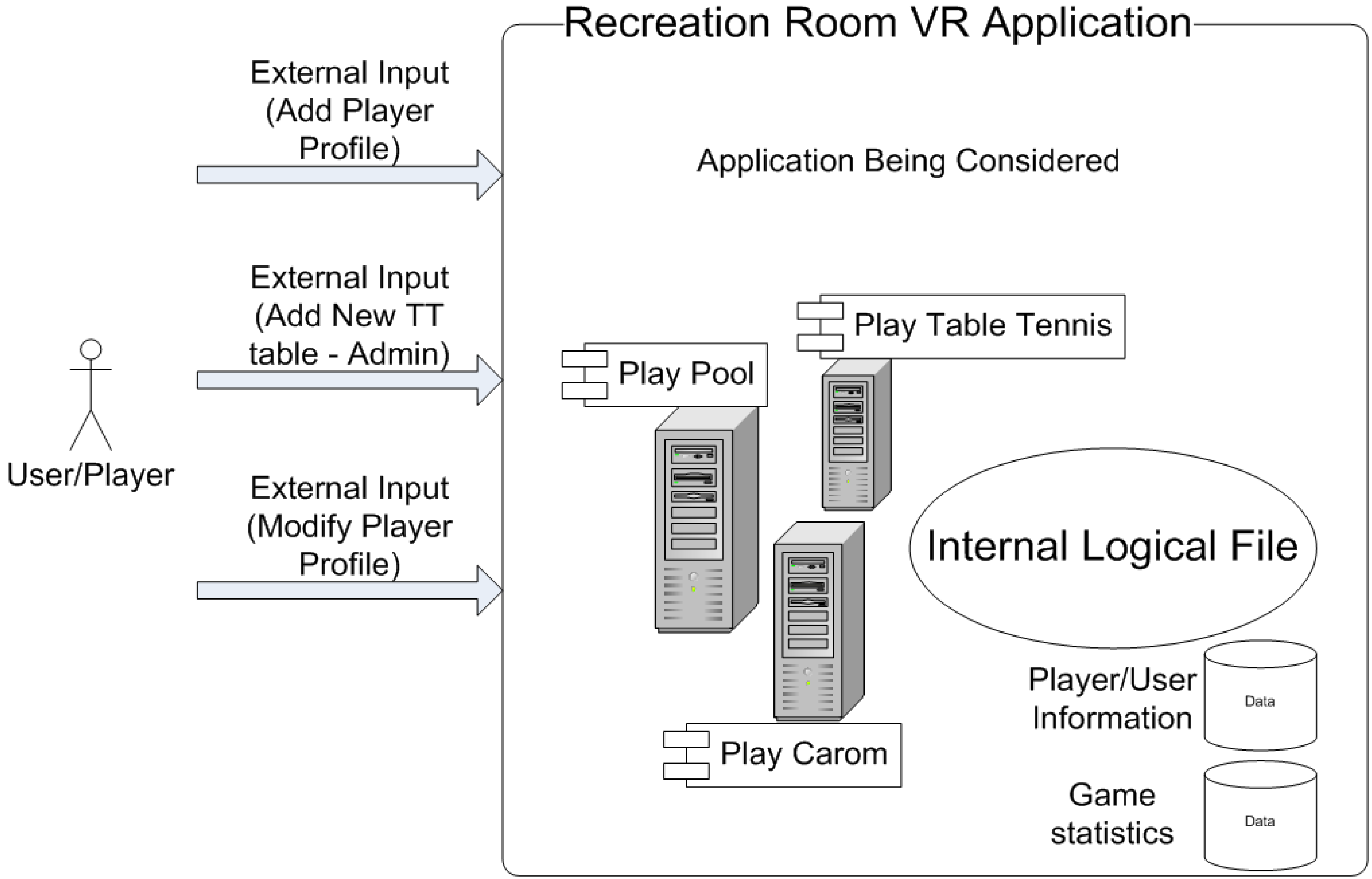
**External Input**: External Inputs are transaction functions that accept data from user and maintain ILFs.

**Example:**

- For an administrator of a Recreation VR application, there are functions designed to move objects within the virtual room, create new objects etc. These functions would be considered as an External Input as the user's action here would be to maintain Virtual Room ILF.
- For the user using the Recreation VR application, there can be options to create a player profile before entering the room or before playing a game . The user can add, modify and delete profile. These can be considered as an External Inputs.

Uniqueness rules need to be applied, where different roles can perform same functions in a different way to satisfy one requirement.

# Transaction Functions – External Input (Depicted)



# Transaction Functions – External Output

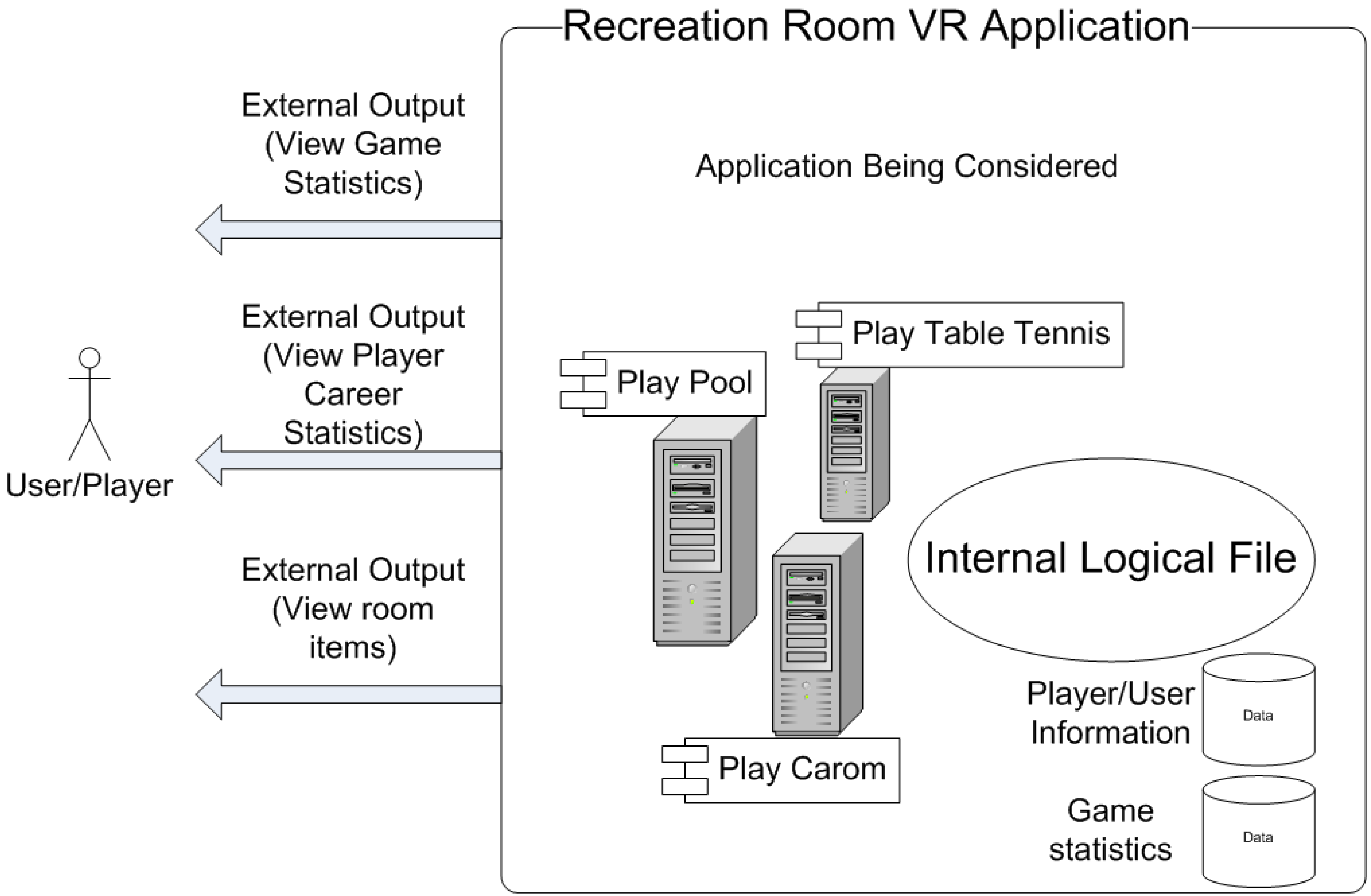
**External Output:**

External Outputs are transaction functions that present information to the user with help of additional processing logic like Calculations and generation of derived data.

**Example:**

- Game statistics that can be viewed after a game is played involves mathematical calculations and can be considered as an External Output.
- Player career statistics that can be viewed also involved mathematical calculations and can be considered as an External Output.
- Display of items in a virtual room involves calculations to be made with respect to head movements/eye movements using a head mounted displays. This function calculates what needs to be displayed thus calculating the orientation and selecting appropriate images to be rendered to give a real time experience to the user along with playing appropriate audio suitable to the scene. This can be considered as External Output.

# Transaction Functions – External Output (Depicted)



# Transaction Functions – External Inquiry

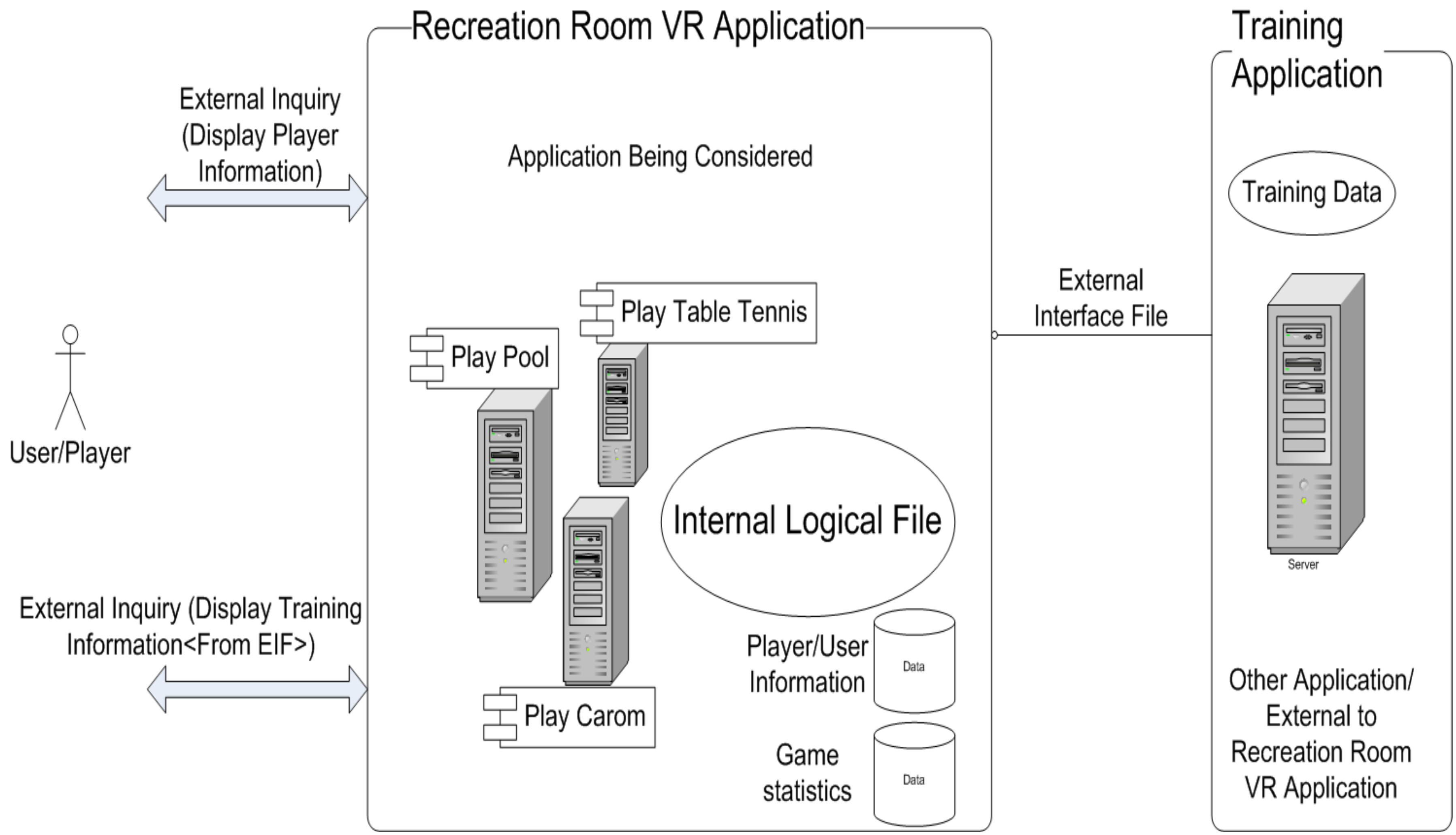
**External Inquiries:**

External Inquiries are transaction functions that present information to the user with retrieval of data from logical file(s).

**Example:**

- Viewing Player information like the Player name, no. of matches played, ranking etc need to be directly fetched from the database and can be considered as External Inquiries.
- Viewing of Training information can be considered as External Inquiry as it can fetch data from other application for information related to trainings taken by the player and the players training assessment information.
- While the Virtual room objects like the light switches, score board in the room, balls etc can be considered as business data that needs to be displayed to the user, the room itself, like the walls, the ceiling, Photo frames in the room etc will not be considered as valid fields/views, but can only be considered as code data and the views related to the same can be considered same as navigation in web applications. Hence these might not yield function points.

# Transaction Functions – External Inquiry (Depicted)



# General System Characteristics

# General System Characteristics

As we discuss on that General System Characteristics for Virtual Reality applications, we need to know the primary requirements of a VR development system.

Some of the primary requirements of a VR development system are: **Performance, Run time flexibility, Ease of Use, Distributed Environments**

Although each VR application might vary in the degree of influence of each of the general system characteristics, there are some things that can be understood so that the degree of influence set for a particular GSC is not illogical.

**Performance:**

Effective immersive environments need a high frame rate (15 Hz or better) and low latency. Poor performance is not merely an inconvenience for the end user; it can cause unpleasant side effects including disorientation and motion sickness. Therefore, a VR system should be able to take advantage of all available resources on a system, such as processors and special graphics hardware.

The development system itself should have as little overhead as possible, letting the hardware run at its maximum efficiency. Performance GSC usually will get a score of 4 or 5 based on the requirements.

# General System Characteristics (Continued)

**Run time Flexibility:** The development environment should be able to adapt to many hardware and software configurations. If the environment cannot adapt to new configurations, applications will be limited in the scope of their usefulness. A developer should not be required to rewrite an application for every new configuration. In addition, the design of the system itself should not limit the sort of applications that can be developed with it. Developers should never hit a wall where the development environment restricts them from creating an application that works the way they have envisioned it. Multiple Sites GSC usually will receive a score of 3 and above because of this need.

**Ease of Use:** The development system should be easy to configure and to learn. The Application Programming Interfaces (APIs) and/or languages used to create applications should be cleanly designed and should hide as much of the system's underlying complexity as possible. An ideal development system would score perfect marks in each category. In reality, they often conflict. An easy-to-use system's interface may limit the developer's options, sacrificing flexibility. A very flexible system might be difficult to optimize for performance because of the number of choices presented to the developer. Each of the systems we will present has chosen a different balance of these elements. End user efficiency usually will receive a score of 3 and above.

# General System Characteristics (Continued)

**Distributed Environments:**

In some VR technologies, they separate the simulation frames from the rendering frames by using a multiple process architecture, typically with each process running on a separate CPU. This allows the participant to interact in real-time, even when the application-level simulation computations become complex. This is useful because even though the changes to the environment (e.g. the position of the billiard balls) might only update twice a second, the participant's viewpoint in the head-mounted display, being fed by the tracking devices, will update more rapidly, and that is the frame rate that must be kept high at all costs, in order to maintain the illusion of presence in an environment.

Distributed data processing GSC usually receives a score of 3 and above.

- All other GSC’s need to be analyzed and degree’s of influence decided on a case by case basis for each of the application under study.

# Recommendations and Challenges

# General Rules/ Recommendations

- Count only objects in the virtual space that are of real significance for the user and plays a role in completion of a elementary process.
- Count all items that are present in the virtual space and on which the user has control over.
- Do not count images or views related to look and feel that gives the real time experience to the user in the virtual world.
- VR applications, if designed appropriately to give quality real world experience to the user in the virtual world, would require multiple processing to be done on different servers thus increasing the quality of output. This needs to be considered in GSC's to give credit.
- Each object has a list of action routines, or callbacks, which are executed with each frame of the simulation. These action routines are responsible for changing the internal state of the object (position, orientation, application-specific state, etc.) for the next animation frame. For example, the action routines for each ball in pool game could compute the physics of their collisions. Functional size analysis need to be made at elementary process level that would merge the action routines into one.

# Challenges

- There can be scenarios where it would be difficult to distinguish between what is needed in the virtual world for function identification and what is present just for look and feel.
- It can also be challenging to identify as to how the actual game being played by the user in the current application being discussed (or generally speaking, the interactive actions performed by the user) which is actually the main functionality delivered by the application.
- There are multiple images that are taken from the real world and rendered in the application to give a real world look and feel. These might not yield FP's being static images and discussions need to be made to handle these kind of scenarios in productivity measurement programs.
- For each of the action routine triggers, there can be multiple processes triggered that might run on different servers in parallel thus increasing the technical complexity of functions. Understanding the system complexity is also important to correctly identify the degrees of influence in GSC's.
- There can be very complex applications unlike the example discussed in this paper with multiple parallel processing and confusing to determine the functions being delivered. Care needs to be taken to distinguish real world experience role assigned in the virtual world.

# Conclusion

# Conclusion

In this paper, we have tried exploring how the VR application architecture would look like and realized that FP veracity for Virtual Reality is in fact a reality.

With increasing demands and focus on VR applications, there can be increasing need to embed VR applications into Web applications used by clients and at the same time an increasing need to measure VR applications.

Function point methodology can indeed be applied on VR applications and there might be additional scenarios that might come up or get uncovered in detailed counting cases because of vast spread of VR functions/scenarios. Each scenario needs to be addressed specifically without deviating from the basic concepts that we already understand.

We did cover some of the examples at a high level in this presentation and the plan is to work on VR application requirements in more detail and come up with clear guideline recommendations  where exhaustive list of scenarios are covered.

# References


1) IFPUG CPM - 4.3
2) Software Tools for Virtual Reality Application Development paper BY Allen Bierbaum and Christopher Just
3) Ten Steps to Developing Virtual Reality Applications for Engineering Education, John T. Bell, H. Scott Fogler, Department of Chemical Engineering, University of Michigan
4) A Brief Architectural Overview of Alice, a Rapid Prototyping System for Virtual Reality, article first appeared in the May 1995 issue of IEEE Computer Graphics and Applications
5) A Generic Virtual Reality Software System's Architecture and Application, Frank Steinicke, Timo Ropinski, Klaus Hinrichs
6) An Open Software Architecture for Virtual Reality Interaction, Gerhard Reitmayr, Dieter Schmalstieg
7) What's Real About Virtual Reality?, Frederick P. Brooks, Jr. University of North Carolina at Chapel Hill
8) Virtual Reality CS60-520 Presentation, Instructor: Dr. Aggarwal, Student: Yang Gao
9) Virtual Reality Presentation, Masih Farnia


Q
&
A

# Thank You

# About the Author

- Prashanth CM, Senior Systems Analyst, Accenture Services Pvt Ltd, has over 10 years of experience in the information technology industry specializing in improving the software quality and productivity through measurement. His areas of expertise include function point analysis, quality assurance, process improvement, performance management and project management. Have worked as function point consultant for varied industries measurement programs covering Retail and Telecom clients.

- As a Senior Systems Analyst, he was responsible for various process improvement and measurement initiatives. These assignments included reporting metrics to all levels of management, coordinating/implementing function point measurements and delivering training in function point methodology. Prior to his metrics specialization, Prashanth CM held a variety of programming, analyst, and management positions at Accenture.

- Prashanth CM received his MS in Software Engineering from Birla Institute of Technology (BITS). He is recognized by International Function Point Users Group as a Certified Function Point Specialist. He has also presented a paper in ISMA Cinco in Brazil Sao Paulo during September 2010.

# About the Presenter

- **Heidi Malkiewicz** currently works for Accenture managing benchmarks and supporting large-scale global clients that use Function Points, Service Level Agreements, and Benchmarking. Heidi has been an employee of Accenture for the past 15 years. Heidi has been a Certified Function Point Specialist for over 10 years and actively utilizes her certification through Function Point counting and analysis, teaching and mentoring others on Function Points, and managing productivity programs. Heidi also is serving on IFPUG's ITMAC Committee.