the Virtual Windtunnel
Design and Implementation of
How Human Factors Drove the

NASA Ames Research Center
Steve Bryson

http://science.nasa.gov/software/vwt
http://science.nasa.gov/~bryson
Outline

- Interface Design
- Run-time Architecture
- Software Design
- Implementation
  - Initial Choices
  - Primary Challenges
  - The Application Task
- Design
- Introduction
etc...
contour planes
isosurfaces
streamlines
variety of visualization techniques
pre-computed tiles
complex meshes
volumes of vector and scalar data on CFD simulations

Virtual Windtunnel: Virtual Reality for Introduc
Ease of use with many capabilities
- overall sense of flow
- pressure distribution
- vortical structure
- Things different users looking for different

Simulation phenomena in 3D volume

Exploration of spatially complex

The Application Task
• Distributed operation
• Conservative user community capabilities
• Extensibility to new interfaces and
• Very large amounts of data (> 200 GB)
• Versatile Interface – computationally intensive
• High performance

Primary Challenges
Use an object-oriented approach, allows easy
All data resident in memory
Decouple computation and interaction
Place all control within environment
Support a variety of direct manipulation

(not an historical account)
- metaphors from real wind tunnels
- groups visualizations in a natural way
- use "visualization control tools"
- many visualizations move
- very crowded scenes "everywhere"
- reflect "direct manipulation"
- grab and point
- restrict to two active gestures
- based on direct manipulation

Initial Choices: User Control
Design

Implementation: Software

- Visualizations access data objects
- Some tools control environment
  - Some tools control visualizations (tools)
- User interacts with tool objects
  - Objects
  - Specified high-level interface between
  - Each object manages all aspects of itself
- Strong encapsulation
- Implemented in C++ and OpenGL
Objects

- Human(s) and data are special global
  - Managed by environment list object
  - Environment objects

- Tools and visualizations are

Class Hierarchy
Vector and Scalar Visualization

- Numerical values
- Grid planes
- Color-mapped objects
- Cutting/contour planes
- (Local) isosurfaces
- Scalar visualization
- Streamlines, etc.

Visualizations
- Single emitter for cutting/contour plane

- Plane (2D array of emitters)

- Rake (line of emitters)

- Sample point (single emitter)

- Emitters can emit up to five visualization types

- Tools contain emitters of visualizations (Tools)
Non-Direct Manipulation

- Global Visualizations
- Pre-computed graphics data
- Controlled via menu and sliders
- Allows much larger data sets to be non-interactively viewed

\[ \infty \cdot g \cdot \text{results of batch visualizations} \]
- 1D, 2D, 3D environment/current view tool
- Sliders to control parameters of
- Depends on current view tool
- Acts on current view tool
- Hierarchical
- "Point in space" paradigm
- Pop-up menus for command selection

Other interface tools
does not need to understand rest of VMT
uses local data supplied by tools
compute and draw visualization
developer only needs to know how to
draw samples in template
-use of some obscure calls required,
class hierarchy
"automatically" managed at higher level of
interaction with tools happens
according to template
fill in compute() and draw() routines

Visualization
Extensibility: Adding a
Success

Requires somewhat deeper knowledge of WIT

Developer determines what `find` and `grab` use human object as input

According to template

Fill in `find()`, `grab()`, and `draw()` routines

Extensibility: Adding a tool
actuator
- add a command and assign it to an
  command
- Actuator is a subclass of tool which
  API is text and value based
command
- Command encapsulates action of
  Command and Actuator classes
- Future: voice control, text input
  from several
- Control commands come from several

Environment Control
Synchronization
Issues of communication and asynchronous process groups
Computation take place in two
Implications that interaction and
determined by user

Computation > 2 fps > 0.5 sec latency
Both display and control
Interaction > 10 fps, > 0.1 sec latency

Architecture
Implementation: Run-Time
Overall Run-Time Architecture

Child Graphics Process Group

Main Graphics and Interaction Process Group

Computation Group

Data Buffers

Pipeline for 2nd Graphics Process Group

Synchronization

...
Computation Process Group

so there is a parallel and non-parallel phase
- global isosurfaces
  visualizations
- allow "internally parallel"
  - no attempt to load-balance
  - streamlines, iterative isosurfaces, etc.
  - parallelize across "non-parallel"

•
Processing of user input
- current gesture(s)
- location of human "parts"
- current graphics transformations
- includes setting of human states
- includes sampling of I/O devices
- includes graphics pipelines
- multi-processed to support multiple

Graphics Process Group
operation

- Converts gracefully to distributed
- Provide API to hide this complexity

buffer

- Use various senses of time to choose
- Communication buffers for each object
- Handle on an object-by-object basis

Many demands

Communication

Computation-Graphs

Comm...
- time-critical computing
do best job in given time
Each visualization determines how to
- non-trivial
over visualizations
VWT distributes specificed frame time
rate (one hopes)
- does not effect display/interaction frame rate
- user specifies the desired
computation

Visualization Computation
– build into navigation paradigm

• Work at many scales

• Comfort Very Important

• Display quality Very Important

• "fisheye" VR display

• Immersive Workbench

– users often relate to simulation as model

• Immersion Less Important

• 3D Presence Very Important

Implementation: Display
• Fast, cheap, and in control

- Workstation/mouse mode popular
- Menus in environment even when in devices

• Maintain the same interface for all

- and 3D versions
- find() and grab() in tools come in both 2D
- trackers and mice

• Support both 3D and 2D manipulation

Implementation: User control
- explore methods of grouping of widgets
- Problem when there are a large number
- active research area in our group
- requires files in special format
- through very fast access from disk
- Remove data-in-memory limitation

Near Future
Unresolved Issues: Evaluation

- Large System Evaluation is very difficult
- can test pieces, but this does not inform how pieces work together
- problem is one of high dimensionality
- User feedback is best