2004-01-3121

Knowledge Navigation for Virtual Vehicles

Julian E Gómez
RIACS / NASA Ames Research Center

ABSTRACT

A virtual vehicle is a digital model of the knowledge surrounding a potentially real vehicle. Knowledge consists not only of the tangible information, such as CAD, but also what is known about the knowledge – its metadata. This paper is an overview of technologies relevant to building a virtual vehicle, and an assessment of how to bring those technologies together.

INTRODUCTION

A virtual vehicle refers to more than just a CAD model of a vehicle – it is the concept of everything that is known about the vehicle. The NASA Virtual Iron Bird [6] program says that virtual vehicles are really knowledge management systems for the vehicles. A central reference site for a vehicle would serve as a powerful starting point for anyone who needs to know something about it.

A requisite of such a system is that it includes everything that is known about the vehicle. This could easily mean that new information and storage constructs will have to be developed in order to model the real information. Conventional information storage mechanisms have limited information modeling capabilities, but modern technology creates potential to meet the problem, instead of forcing the problem into predefined models.

A ramification of including all knowledge is that the virtual vehicle has to be an integrating model, that is, as new knowledge is developed, the model has to be able to integrate it as a peer in the existing knowledge.

Building a virtual vehicle from knowledge models has several advantages. The first is that the vehicle is described at a higher abstraction level, making it easier to convey the meaning behind systems design. Another is that knowledge models are domain based, so the information in the knowledge base can be referred to with domain based referencing, instead having to translate the domain knowledge into the information model. Many vendors have already recognized these advantages and are currently incorporating them into their product strategies.

This paper is an overview of technologies relevant to building a virtual vehicle, and an assessment of how to bring those technologies together. Some familiarity with CAD and information technologies is assumed.

MAIN SECTION

There are number of technical areas involved in building a virtual vehicle.

VISUALIZATION / VIRTUAL REALITY

CAD and engineering

3D computer graphics has long been used in engineering and science in many different applications, which are all important in building a virtual vehicle.

Modern engineering design is almost universally done with CAD systems, although more current embodiments of CAD include PDM and PLM. In addition to providing sophisticated tools for design and analysis, these tools provide pathways directly to the manufacturing process.

Besides design, CAD provides a means for automated analysis of designed parts. Stress, thermal, and other kinds of loading can be simulated and evaluated long before a part is actually manufactured.

Visualization

Since stress functions do not have a physical existence, it's necessary to visualize them with computer graphics; this of course also applies to all other kinds of functional testing. In general, any kind of function, including those in, for example, phase spaces, requires a visualization process to see them. This likewise applies to trying to see systems that are impractical in reality, such as airflow in a thunderstorm or different models of atomic structure.

In general, visualization allows processes that are not inherently visible to become so, thereby promoting understanding of the processes. This is a fundamental principle in building a virtual vehicle, since the goals involve much more than just having the CAD data of the vehicle being available.

This is due in large part to the fact that humans can appreciate an image very quickly: "A picture is worth a thousand words." More technically, an image contains a very large amount of information, but human perceptual bandwidth is easily high enough to process that information in real time.

A separate, but very important, advantage of visualization is the capability to simultaneously present multiple dimensions of information. Given a particular set of information, it is possible to have multiple windows onscreen showing the same information from different contexts. Of course, the multiple windows might instead be showing related data. The idea is that having access to information, whether in different contexts, or through related data, promotes understanding.

A critical part of viewing data is being able to interact with it. In 2D graphics, there is generally no more information than what is presented on the screen. In 3D graphics, the screen is more of a window into another space, which means the objects in that space can be manipulated, and the window will display the results of the manipulation. Initially a user might be presented with the front view of a vehicle, but by using the mouse turn the vehicle all around, zoom in to look at details, or click in special places to bring up additional information, or even translate into a separate information space, e.g. moving from a display of the electrical system into a display of the hydraulic system.

Virtual Reality

Most 3D graphics systems render images into a 2D window on the screen, and are driven by the user moving a mouse. More sophisticated technologies are available, at higher cost.

Immersive displays

Since a screen window is actually 2D, depth information has to be indicated by different techniques. Long established techniques include shading (calculating reflected light), depth cueing (making things farther away be dimmer), and hidden surface removal (not rendering geometric elements that are in back of other elements). In all of these cases, however, the viewer is still some distance away from the screen, looking at a 2D rendering.

Immersive displays use different methods to place the viewer into the displayed image(s). Head mounted displays (HMD) are helmets with small screens in front of each eye, obscuring all vision except for those screens. A dome display places the viewer near the center of a spherical screen so that images are being presented through almost all of the peripheral vision. A CAVE-style display places the user inside of a set of approximately six foot high screens; there can be between 3 and 6 of these. Thus the user is presented with an image when looking in any direction. Other technologies, such as radial projection and layered LCD displays, have been brought out and will be on the market soon.

The use of stereo imagery promotes the effect of all of the above. Audio is an important part of the overall effect, and stereo audio is now available off-the-shelf.

Haptic input/output

One problem with visualization is that there is usually no connection between the user's input and the generated image. Flight simulators have for a long time used sophisticated mechanics so that pilots have "feel" when manipulating the controls in the simulator.

Haptic input/output devices provide ways for the visualization user to have some kind of "feel" when working. The first example is a mouse with motors that push back on the user's hand while the user is pushing on the mouse; the amount of pushback is controlled by the visualization and would be keyed to what the user is doing to give the illusion of physical feel. A very basic example would be a simulation of pushing a rock uphill – as the user pushes with the mouse, the simulation pushes back to make it feel like that rock really has weight.

Other kinds of haptic devices include force feedback joysticks, the UNC GRIP arm, the VPL Dataglove and its descendants, and the Sensable Technologies Phantom and similar, which provide six degrees of freedom input and three degrees of force feedback output.

SIMULATION

Part of the overall task of a virtual vehicle is to provide a framework to generate simulation scenarios and incorporate their results as part of the knowledge base. This topic is not being treated separately here because its integration into the knowledge base is similar to other topics already presented.

DATA DESCRIPTION STANDARDS AND TOOLS

Historically, transferring data from one domain to another has been a formidable problem. Currently, the syntax of transferring data is a solved problem, although the semantics of the process are not as far along.

In this section it should be recognized that there are many practices and even standards to accomplish the purposes presented here. This paper focuses first on ISO standards and in their absence on the most common practice.

XML

In contemporary terms, a critical requirement is that a standard be XML based. XML is the Extensible Markup Language, and provides two key capabilities:

1. Standard method of describing data

2. Standard method of describing metadata (data about data)

Because XML provides a standard format, the processes of handling and translating data are simplified. The IT industry has a large set of COTS tools available, and because of XML, a tool does not have to understand a domain while dealing with data in that domain; the tool understands the syntax of the data and is not required to deal with the semantics.

Because of this, translation becomes a smaller problem. XSLT, which is itself XML based, provides a technology for an XML based translator to be directed how to translate one XML file into another. The problem of handling semantics during the translation has been transformed into generating the correct XSLT. Since the translation specification is itself a file, the translation problem becomes scalable.

CAD data translation

The CAD domain has evolved two standards – IGES and STEP – for transferring data between CAD applications, and these are also supported in other categories of applications. CDF is a proposed standard for transferring CAD data to non-CAD applications.

CAD data translation is a critical requirement because:

1. CAD systems are expensive and require a lot of training, so it is not reasonable to expect that everyone using the data use a CAD system.
2. Most 3D graphics capabilities exist in systems that are not CAD driven. There needs to be a translation process so that the CAD data from the engineering domain can be brought into these other systems.

KNOWLEDGE REPRESENTATION

The basic construct for an information network is a directed graph. In a relational database, nodes represent records and arcs are relations to other records. A semantic web refers to adding meaning to the arcs, i.e. they carry information just as the nodes do. Frequently there will be a two-way aspect to the relationship, e.g. node B is managed by node A is one relationship, and node A manages node B is the other.

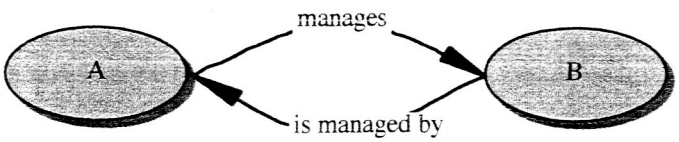

Figure 1. Example relationship.

RDF provides a standard way for describing relationships between entities. An RDF triple consists of a subject, a relationship, and the object of that relationship. A simple example might be "Brad is Vera's father.", and a simple engineering example might be "the landing gear locked sensor - is mounted on - the right main landing gear."

OWL is the Web Ontology Language, a representation specification that includes description logics. An ontology is a set of constructs describing the knowledge of a system. By knowing how information in a knowledge base is constructed, it eventually becomes possible to work with that knowledge on a meta level, including relating information between different knowledge bases because the metadata of both is well understood.

A common tool for working with ontologies is Stanford's Protégé.

VISUALIZATION TECHNIQUES

Visualization of CAD and functional data is already well understood. Visualization of information networks has been well studied as a 2D problem [Geroimenko], and there are COTS packages providing those capabilities. A newer problem is visualization of information networks as 3D graphics. Munzner [5] has studied this as a 2D projection onto a 3D surface; Dwyer [1] has done extensive experiments in this visualization problem.

When drawing a network in 2D, it is common to indicate classes of information by using different shapes, colors, sizes, dashed vs. solid lines, etc. These concepts transfer into 3D graphics, but with added capabilities. Positions of objects also have depth, and shapes become geometries. Arcs also have geometries; whereas before they were just lines, now they can be cylinders, cones, hexagonal tubes, etc., which in 2D would be indistinguishable. Furthermore, with texture mapping, all shapes can have images mapped onto their surfaces, either static or moving from a video. All of these combine to provide more bandwidth to display information about the network.

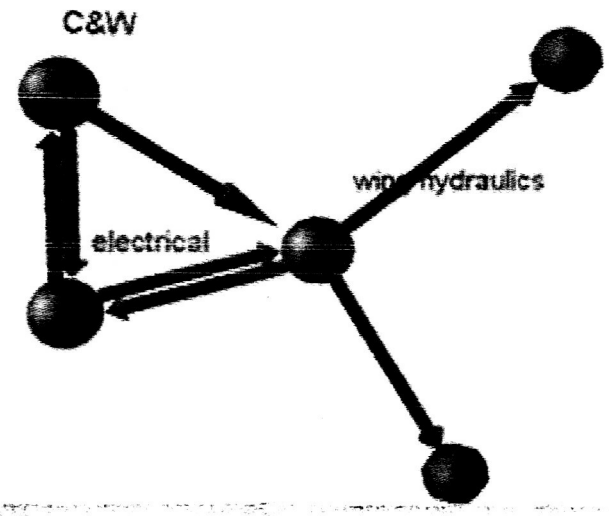

Figure 2. Example network graph in 3D. Image generated with WilmaGraph [1].

Because the objects being displayed have depth, it is possible for the user to (virtually) travel inside the network graph (Guha [4] did a very early implementaton of this idea in Hotsauce), using techniques outlined in a previous section. This means that the graph can be analyzed from an interior perspective, instead of just looking at a 2D drawing. Being immersed in the network provides a different mode of understanding, and the interactive capability enhances this. The user can directly experience how information travels from one place to another, either by standing inside the graph and watching, or traveling along with the information packet as it moves through the system.

Style sheets

The HTML 4.0 specification includes support for Cascading Style Sheets (CSS), which are sets of rules governing how each tag in an HTML document should be interpreted at display time. The same concept can apply when rendering graphics in 3D. The rules specify classes of information and how to show those classes specifically. An example rule set could be:

1. All hydraulic system components drawn in red.
2. Primary hydraulic pumps drawn as cones.
3. Secondary hydraulic pumps drawn as spheres.

The 3D rendering would then show at a glance the vehicle and its hydraulic system, and in the same image indicate the system subdivisions.

Note that the rules deal with classes of information, so it is not difficult to relate them to the queries into the database. XSLT provides the binding technology between queries and results and display.

Composition

Displays do not have to be monotonic; some powerful information presentations can come from combining different types of image generation.

As an example, consider a vehicle that has just reported a fault. In the onscreen window, there is a 3D model of the vehicle. Within that the reporting network is shown in red, so the physical locations of each node show where the physical unit is located in the vehicle. Overlaid on this display are multiple telemetry channels showing a time range of interest for nodes in the faulting network. This type of layering process [7] provides simultaneous multiple viewing contexts as described earlier.

KNOWLEDGE NAVIGATION

The ability to coherently store knowledge is the first part of the virtual vehicle problem; accessing that knowledge is the second. It is taken for granted that text queries returning text results are useful but not a comprehensive solution, and that 3D visualization is a necessary part of the solution.

Many queries into the knowledge base will yield 3D data, e.g. anything involving CAD data. In fact, queries could return many kinds of data; a partial list includes:

- 3D data
- audio
- video
- engineering drawings
- stress analyses
- photographs

Also, because of the severely heterogeneous nature of the virtual vehicle knowledge, a query could return different types of data, so the system must be able to properly display them and allow context sensitive interaction with the results.

By extrapolation, it would be useful if the queries themselves could be heterogeneous. The ideal query constructor would be one that allowed entry of any type of construct in the knowledge base. Finding "something that looks like this" or "something whose subnetwork of knowledge looks like this" are extremely powerful concepts. Most of these, for example queries involving 3D graphics as query terms, are current research topics.

A critical requirement for knowledge navigation was alluded to earlier, when discussing simultaneous displays of data. In general, the process of viewing data from different contexts falls into the Model-View-Controller (MVC) paradigm, which specifies that the concept of viewing the data is independent of the data model, and likewise control of data should not be bound up with the data model nor the views.

The direct application of this to a virtual vehicle is that the user must be able to navigate freely in context. For example, suppose one window shows a logical view of the hydraulic system while another shows its physical layout. The user could navigate in the former view and hit a key which says to transfer to the second view, but at the same point in the system that was currently being viewed in the first window. A minute later, after navigating in the second window, the user hits another key which says to transfer to a new window showing the electrical system at the point where it's connected to the hydraulic system, at the point being viewed in the second window.

The idea here is that all of the views are just that – views into the knowledge base. The user should be able to navigate freely through the knowledge base using any context and paradigm. Examples of context would be: electrical, hydraulic, water, etc., and examples of paradigm would be: schematic, logical, physical layout, etc.

FUTURE WORK

A significant issue not yet addressed is displaying information on the scale of a virtual vehicle. In an actual space vehicle there are millions of parts.

Coupled with the other kinds of knowledge, it's actually infeasible to display all this on a screen.

One area of the complexity issue is algorithms to meaningfully lay out nodes and arcs in a 3D space. Methods such as cone trees and star trees serve only part of the purpose because they require hierarchical data, and knowledge networks are not necessarily hierarchical. Dwyer [1] has looked at this problem but it is an open research area.

Another possibility for dealing with the problem is meaningful reduction. Being able to reduce the number of nodes and arcs being displayed without also reducing the information content is also a research area.

CONCLUSION

A virtual vehicle is a compendium of many areas, whose confluence produces new science and technology that enable new ways of modeling and interacting with information. The ability to have a central reference site for everything known about a vehicle, including information not normally visible and information describing the information, provides powerful pathways to designing and understanding real vehicles.

A virtual vehicle is finally technical feasible because the state of information and knowledge systems are powerful enough to deal with the required data, and visualization technologies are powerful enough to enable effective interaction with the knowledge bases.

Furthermore, it is practically feasible because the use of standards for information description and transfer makes it feasible for knowledge bases to coexist and work compatibly.

ACKNOWLEDGMENTS

Here is the Acknowledgment section. This is an optional section.

REFERENCES

1. Dwyer, Tim. *Three Dimensional UML using Force Directed Layout*. University of Melbourne TR2001/25. 2001. http://www.wilmascope.org
2. Fensel, Dieter. *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag. 2001. ISBN 3-540-41602-1.
3. Geroimenko, Vladimir and Chen, Chaomei. *Visualizing the Semantic Web*. Springer-Verlag. 2003. ISBN 1-85233-576-9.
4. Guha, R. V. *Hotsauce* http://www.xspace.net/hotsauce
5. Munzner, Tamara. *Interactive Visualization of Large Graphs and Networks*. Ph.D. dissertation, Stanford University, June 2000.
6. Proceedings of the NASA Workshop on the Knowledge Integrating Virtual Iron Bird. 2004. http://ic.arc.nasa.gov/vib
7. Tufte, Edward. *Envisioning Information*. Graphics Press. 1990

DEFINITIONS, ACRONYMS, ABBREVIATIONS

CDF: CAD Distillation Format

OWL: Web Ontology Language

RDF: Resource Description Format

X3D: Extensible 3D Graphics

XML: eXtensible Markup Language

XSLT: Extensible Stylesheet Language Transformations