Knowledge Space Visualization

Overview

Entities are the basic building block of knowledge. A *knowledge space* is a collection of entities created by one or more cooperating, peer-to-peer servers. Depending upon how the servers are configured, it is possible to create multiple knowledge spaces, each separate and distinct from each other. In general, however, it is preferable to create a single unified knowledge space that spans the server network. Every entity in a knowledge space has an entity unique identifier (EUID) that distinguishes the entity from all other entities in the knowledge space. The AvesTerra knowledge space technology is specifically designed to support trillions of entities widely distributed across a global network.

Given the potentially enormous size of a knowledge space, visualization of an entire knowledge space at one time is generally not practical. To accommodate scale, a knowledge space may often be partitioned into multiple separate (sub)spaces where such spaces may potentially overlap, with entities from the parent knowledge space residing in zero or more of these space partitions. To aid visualization, a space may be assigned a coordinate system which allows entities to be placed at a unique location within the subspace. Different spaces may use different coordinate systems, and the type of space may apply different semantics (meaning) to the notion of location. Four fundamental space types are envisioned:

- Geospace: A space which resembles the structure and shape of the Earth with the notion of entities residing at a particular latitude, longitude, and elevation, and potentially occupying a volume of that space.
- Hyperspace: A space which captures the notion of dependencies and relationships between entities as characterized by their properties and attributes.
- Cyberspace: A space which reflects the notion of electronic connectivity between the physical or virtual objects that the entities represent.
- Metaspace: A space with transcends an ordinary physical space, but which employs the notion of a coordinate system to assign meaning to entities via differences in their location.

To visualize a space of entities, a graphical user interface (GUI) tool is employed. Depending upon the type of space, the GUI will render the space differently in order in order to aide understanding. A geospace is generally rendered using a map system that reflects the physical Earth. A hyperspace renders entities based on the types of relationships entities have among themselves (e.g. a directed graph structure), typically using icons to represent entities and various connecting lines to represent different types of entity dependencies. A cyberspace rendering typically will provide an iconic representation of each entity with linkages that represent connectivity in the cyber network. A metaspace rendering uses the notion of a virtual coordinate system to position and render entities to help aid understanding of their relationships. A metaspace is perhaps the most general. For example, an automobile entity may be represented as the shape of a passenger vehicle with entities representing vehicle components such as the engine and transmission drawn as icons at their appropriate positions within this shape. Metaspaces may also employ abstract concepts that may not directly relate to physical world constructs.

Associated with every user of the GUI is the notion of an *identity*, itself an entity in the knowledge space that represents the real-world user. Optionally associated with every identity entity is the notion of a *profile*. A profile is representation of user's latest GUI state. GUI state may include a list of spaces a user is viewing and their locations on a display screen, in addition to other user interface preferences. A user's profile may be saved and later restored so that a user's different views of a knowledge space may be preserved across multiple GUI tool sessions. A user's initial profile when using the GUI tool for the first time might be blank. Alternatively, their profile might be pre-populated with various spaces and settings depending upon a specific role or type of service for which they have been enrolled.

When starting with a blank profile, a user may request an initial space to be viewed (e.g. geospace) which would allow the user to browse that space using their available pointing and scroll devices. As the user moves through the space, entities and their relationships will be depicted, often overlaid on a backdrop such as a map or schematic to provide additional context. As entities are displayed, the user may choose to "select" one or more of the entities, requesting more details of the entity to be displayed. Typically this will involve displaying a new window where the content of the window is determined by the type (i.e. class) of entity and its current state. Displayed in the new window may be another space, or alternatively an instrument panel designed for the specific class of entity being viewed. If another space, the GUI will again provide the user with the appropriate controls for browsing/exploring that space. If an instrument panel, however, the rendering of the instrument panel can be various status displays and control buttons that enable the user to directly interact with the entity. These displays and controls all involve accessing different methods that are associated with the entity as it exists in the parent knowledge space.

User interfaces often employ the notion of "copy", "cut", and "paste". The space construct may often leverage the same concepts. That is, depending upon the type of space, entities from one space may be copied and pasted into another space with their relationships with other entities within that space promptly depicted. Similarly, some space types may allow users to remove entities entirely or perhaps move them to other spaces. As these entities exist in the original parent knowledge space, the actual entities are not duplicated or deleted by these operations. Rather they may be just included or excluded from various spaces as directed by the user.

Instrument Panels

All entities in a knowledge space possess a class that indicates the type of thing the entity represents. When an entity is selected, the viewing system first examines the entity's class to determine how the entity should be visualized. If the class of the entity indicates that it is a space, as mentioned the viewing system will bring up the appropriate viewer for that type of space. However, if the entity is some other class, the viewing system will dispatch the appropriate instrument panel to handle that specific entity type.

Instrument panel dispatching can be accomplished in a very general-purpose manner. Instrument panels are simply independent applications that can be activated via an instrument panel manager. Among the simplest way to implement this manger is via an adapter. When a user selects an entity using the viewing system, a message request containing the entity's unique identifier, the entity's class, the user's identity, and the user's authorization is sent to the manager adapter. The manager uses this information to determine which instrument panel to present, and may include which physical hardware to use. Information associated with the identity and authorization that is passed is used to determine how that instrument panel should be organized and presented to the user. That is, different users operating in different roles will often need different status and controls. The instrument panel application uses this information to structure its display. Certain instrument panel controls may lead to activating other instrument panels. Similarly, some instrument panel controls may activate various space viewers.

Architecturally, it may be helpful to implement this space viewer/instrument panel construct using a panel manager. That is, space viewers may themselves be implemented as instrument panels and activated using the same instrument panel manager. For users with uninitialized panels, some sort of default or starting "home" interface must be provided. This too can be structured as a different type of instrument panel. Alternatively, the default or "home" instrument panel may be one of the space viewers (e.g. geospace). In this manner, ever major component of the knowledge visualization system can be implemented as an instrument panel, all using the same activation messaging to interact with each other, and their various current state information stored in each user's profile.

Note that in a truly distributed knowledge space environment, there need not be a single instrument panel manger. Rather, there well may be many, perhaps implemented by different teams at different sites widely spread across the Internet. What is important, however, is that the message request protocol across all mangers is identical so that all panels of whatever type can interact. Given that user profiles are represented using entities in the knowledge space, an instrument panel anywhere in a knowledge space enterprise will have access (within the limits of whatever authorizations restrictions that are imposed).

For additional information:
J. C. Smart
Chief Scientist/Research Professor
AvesTerra/ATra Program
Office of the Senior Vice President for Research
Georgetown University
37th & Ost. NW
Washington, DC 20057
smart@georgetown.edu