

DESCRIPTION LANGUAGES FOR VR

William Bricken

July 1990

There are several approaches to world modeling, but to my knowledge, they are all standard, in the sense that they have not been crafted for VR in particular. Some alternatives include:

- Graphics based simulation: the set of transformations, rotations, scalings, etc. found in most high-end graphics modeling packages.

- Object-oriented simulation: the language capacity of assigning properties and methods (dispositions) to objects, found in most object-oriented programming languages.

- Simulation languages: tools and techniques embodied in simulation languages such as Modula, Actor, and Simpack.

In VEOS, we are adopting Entity-based modeling. Entities are similar to agents in AI. Genesereth and Nilsson, Logical Foundations of Artificial Intelligence has a good technical description. Entities are objects with processing power, they can be thought of as operating systems. Each entity contains a mathematical description language (such as Mathematica), capable of pattern matching and constraint enforcement. We use equational logic for assertion and inference, so that the entire disposition is simply a set of equations. Thus, entities are defined (programmed) by inserting equations into their disposition. Time (and dynamics) is just another variable, although for efficiency it is treated specially. The other cornerstone of our architecture is self-similarity. Everything, including spaces which contain "objects", is an entity. There are then two fundamental types of relations between entities: containment and spatial juxtaposition. As it turns out, these mechanisms are fundamental to mathematics itself, function and set.

So our base language is algebra, the rest is knowledge engineering. Although we generally constrain expressability to equational first-order predicate logic, we freely use imaginary Booleans for maintenance of contradictory concurrent worlds. We are currently working on tools to make algebraic programming easy.