SEVERAL SOFTWARE PROJECTS IN ONE PAGE EACH

William Bricken
April 1993

Each project herein stands alone as a solution to an existing problem with virtual environments. Together they form an integrated solution to the entire suite of software problems for VEs.

Boundary Math Engine: This unique engine is at the silicon core running collision detection as spatial interpretation of engine and passive tracking as input interpretation, and dynamics as world modeling interpretation.

Tools for complete functionality

interaction	==>	the Wand
construction	==>	Lsystems
behavioral modeling	==>	entities
functional modeling	==>	relativity graphs
scripting	==>	narrative

Boundary Mathematics

Problem: Fast parallel computation suffers from a deep schism: numerical and symbolic processing are incompatible. This problem manifests in current serial architectures in the form of separate chips for general cpu processes and for floating-point computations. For VR and other computationally intense applications, parallelism breaks down.

Solution: We have developed a new notation for numbers and for logic called boundary mathematics. These forms are inherently compatible during computation. This advantage is relevant only for parallel computation. Boundary mathematics requires a very fine grain parallel silicon architecture to manifest its advantages.

We propose to develop simulators for fine-grain parallel Content Addressable Memory (CAM) chips that are optimized for boundary mathematical computations. We have generated mathematical simulations of the performance of these chips with excellent results.

-- 20,000 (average size) functions can be evaluated each second for *each dollar* of manufacturing cost for the CAM chips. The chips can be ganged and cascaded arbitrarily. This means that we expect to provide the performance of a Cray I for the average cost of a 386 chip, a factor of 1000 improvement of current technologies.

-- This approach is Turing equivalent. If you strip away the hardware architecture specific functions in procedural languages such as C, there is a simple parser from C-- to Boundary Math

-- Boundary math is algebraic, providing simple parsers from algebraic equations.

-- Boundary math uses algebraic logic, providing fully functional theorem proving (Prolog-- equivalent)

Entity Programming System

Problem: Object-oriented programming requires too much pre-engineering, and does not provide autonomous behaviors.

Solution: We propose to install a small, fast inference and constraint engine into an object programming system, and to link the rules that the inference engine enforces to graphic and behaviors outcomes in simulated environment.

Objects + behavior = entities

The HITL VEOS infrastructure provides distributed communications and rapid prototyping of entities. The remaining technical issue is to identify inference engine capabilities and interface. We propose to embed constraints into behaviors, so that entities exhibit satisficing as well as optimizing behavior. Using the Boundary Math engine, this capability can be provided cost-effectively in real-time.

Entities are essentially different from objects in the following ways. Each difference is critical for simulated environments.

-- Entities have two processing loops: reactive and contemplative. This separates computational intense internal activities (such as inference and statistical analysis) from reactions to world events (such as fleeing from a fire).

-- Entities are dynamically configurable, eliminating the impossible task of programming objects and interactions for a universe. Dynamic configurations can be keyed to real-time behavior of participants.

-- Entities are both object (seen from the outside) and environment (experienced from the inside). The inside/outside choice is the essential defining characteristic of simulated environments.

Fast Collision Detection

Problem: For interaction with simulated environments, collision detection is essential. We want to be able to push a virtual table against a virtual wall, and have the two align. Collision detection is currently a computationally expensive task.

Solution: We propose to develop a specialized CAM chip which maintains a spatial model. Collisions can be detected spatially in four machine cycles at 40 MH across the entire space of the model. This makes collision detection so fast that it is like reality.

In current software approaches to the problem which use similar spatial ideas, the model must be static. The Spatial CAM we propose does ordering comparisons as well as equality tests. This means that it can track moving objects as easily as static objects. Since computation of collision is so much faster than movement, the system can treat the moving world as if it were standing still.

The deeper issue with collision detection is collision response. A mechanism is needed to coordinate detections with consequences. The Boundary Mathematics CAM (see other proposal) seamless integrates with the spatial CAM, providing rule-based symbolic computational response to spatial information.

Passive Tracking

Problem: Spatial tracking of hands and head movement currently requires physical-mechanical linkages or electro-magnetic fields. These systems provide a spatial location but little else, so that additional mechanism must be added for gesture, control, etc. They are clumsy and not configurable.

Solution: The obvious solution is to use image recognition from video cameras to provide both spatial and intentional information. This is used in low fidelity in commercial systems like Mandala. Video recognition places the burden on software and computational efficiency.

We propose to develop a new set of location and tracking algorithms that perform in real-time and are easy to use. Components include

-- image recognition

-- virtual body with inverse kinematics (maps physical to database)

-- physical tool recognition and functional definitions (white objects)

The virtual body and the white object systems provide semantics for digital input into simulated environments. Spatial and physical modeling in real-time are provided by dynamics, collision, and boundary math engine (as separate proposals.)

Rapid Construction System

Problem: CAD modeling techniques require specification of each facet of an object. This is too low of a level of detail for most applications, it is time consuming, tedious, exacting, and blind to the purpose of construction.

Solution: We are developing a mathematically precise abstract construction capability based on Lindermeyer systems. The idea is to able to construct from the top down, to specify the general outline, of a ship for instance, and then have the details generated automatically.

Lindermeyer systems are graphics rule sets that generate 3D objects through term-rewriting. This technical structure permits some powerful graphics tools:

1) L-systems can model organic reality. Prusinkiewicz has built photo-realistic flowers, shells, trees, and other organic systems.

2) L-systems can be automated in parallel, creating extremely fast renderings.

3) There is an L-system algebra which permits principled mathematical manipulation. Addition is spatial concatenation; multiplication is form substitution. This means that space and shape can be treated in a formal manner.

4) L-system specifications can be compiled, optimized, parameterized, and randomized, providing a complete description language for graphics.

5) L-systems convert spatial and form information into parallel strings rewrite rules, which make them easy to manipulate computationally.

6) L-systems can be embedded into a virtual graphics environment, mapping specifications onto physical behavior (using VR techniques). Thus, they support an intuitive human interface.

Example: The designer asks for a generic ship of a particular class. The compartments are generated automatically. The designer points to a particular cabin, stretches the width to twice the size, asks for that width to propagate throughout the deck without violating bulkhead divisions. The specification of the new ship is generated automatically. The designer then asks for each cabin to be populated with generic sleeping quarters for four crewmen. The interior of the cabin is generated automatically.

Embedded Narrative

Problem: Humans are story-tellers. Computational systems are not.

Solution: We propose to develop computational systems within an entity architecture (see other proposal) that embed narrative, reactive scripting, dramatic tension, characterization, and story telling arts into the computational environment.

Seen as a global project, this objective is impossibly difficult. However, entities provide a way of modularizing narrative in the form of interactive behaviors. It is critical to recognize that specifying all two-way interactions is not a tractable solution to object interaction. Instead, entities must have internal properties and responses that are dynamically evaluated *in context.*

For example, a ship may have 500 types of screws interacting with 40 types of materials. The screw-by-materials matrix is intractable (as a general solution). Adding a new screw, for example requires 40 more materials rules/methods. Instead, when a particular screw is embedded into a particular material, the physically modeled properties of each must be dynamically computed to determine the interaction. This reduces an O(N^2) relational structure (at best) to an O(N) property structure plus a small set of general rules, the classical technique in scientific law formulation. The dynamic computation is tractable using the Boundary Math Engine and the Dynamics subsystems (in other proposals).

When this approach is extended to biologic interaction, the rules and effects of narrative can be explored.