SEMINAR ANNOUNCEMENTS
William Bricken
compiled 2004

Asterisks indicate seminar announcements that are not included herein.

TITLE	TYPE	DATE
1984-85		
A LOSP SAMPLER	corporate	8411
A LOSP SAMPLER -- LONG SEMINAR ANNOUNCEMENT	corporate	8505
1986		
PARALLEL DEDUCTION	corporate	8603
A SIMPLE SPACE	Sign-in-Space	8608
SELF-ORGANIZING DISTINCTION NETWORKS	Sign-in-Space	8608
PURE LISP AS A NETWORK OF SYSTEMS	Knowledge Software	8610
LOSP	Hackers (invited)	8610
BOUNDARY OPERATORS AND FUNCTIONAL SPACES	Stanford Math	8611
1987		
PARALLEL DEDUCTION USING BOUNDARY OPERATORS	Stanford Math	8701
ANALYZING ERRORS IN ELEMENTARY MATHEMATICS	Stanford Ed (diss)	8702
THE POTENTIAL OF ICAI	IEEE CBE	8704
BOOLEAN FORMAL SYSTEMS	university	8705
BOUNDARY NUMBERS	corporate	8706
VISUAL PROGRAMMING	corporate	8707
DISTINCTION NETWORKS	A-LIFE	8709
GRAPHICAL FOUNDATION OF COMPUTATIONAL MATHEMATICS	corporate	8710
1988		
CONMAN: THE CONSTRAINT MANAGEMENT SYSTEM	corporate	8801
INTRODUCTION TO BOUNDARY MATHEMATICS	corporate	8802
BOUNDARY LOGIC	corporate	8806
DRAWING AS COMPUTATION	Stanford CIFE	8808
MATHEMATICA EXPOSED	corporate	8809
MODEL INTERFACE MODEL	Hackers (invited)	8810
LOSP	UniOps tutorial	8811
1989		
FIRST ANNUAL STATE OF THE LAB REPORT	corporate	8903
FRACTURTLES: USING PICTURES AS OPERATORS	corporate	8903
CYBERSPACE, THE DEMO	corporate	8906
EXPLORING VIRTUAL REALITY IN CYBERSPACE	AutoExpo	8907

1990

DESIGN FRAGMENTS*	university	9002
COGNITIVE MODELS*	university	9002
A VISION OF VIRTUAL REALITY	NASA RIACS (keynote)	9003
VR IS INHABITED	NCGA (invited)	9003
VR AS INTERFACE*	CHI	9004
LEARNING IN VR	university	9005
MINDWARE	Microsoft	9005
VISUAL COMPUTATION	university	9005
BOUNDARY LOGIC, BOUNDARY IMPLEMENTATIONS	university	9006
AN INTRODUCTION TO VIRTUAL REALITY	university	9007
SOCIAL IMPLICATIONS OF CYBERSPACE	DIAC (invited)	9007
THE SOFTWARE ARCHITECTURE OF VIRTUAL REALITY	DEC, SUN, Stanford	9007
TECHNICAL ISSUES FOR VIRTUAL WORLD TECHNOLOGY	SIGGRAPH (tutorial)	9008
DIALOG SYSTEMS*	university	9010
VR APPLICATIONS	Cyberthon	9010

1991

VIRTUAL WORLD SOFTWARE	VW Symposium	9102
EDUCATIONAL APPLICATIONS OF VR	London VR	9106
VIRTUAL INTERFACE TECHNOLOGY	SIGGRAPH (tutorial)	9108
THE MATHEMATICAL BASIS OF CYBERSPACE	Conf on VR (plenary)	9109
VIRTUAL WORLDS	Hackers (invited)	9110
VIRTUAL INTERFACE TECHNOLOGY	UCLA 3-day Course	9110
BOUNDARY NUMBERS	university	9112

1992

PROGRESS IN VIRTUAL REALITY	Imagina	9201
VIRTUAL REALITY IN THE CLASSROOM	NCCE (invited)	9202
VIRTUAL INTERFACE TECHNOLOGY	UCLA 3-day Course	9204
VEOS DESIGN GOALS	SIGGRAPH (tutorial)	9206
AT THE BOUNDARY OF REALITY	Compute&Phil (keynote)	9208
VIRTUAL REALITY AND EXPERIENTIAL COMPUTATION	ACM CSC	9209
SPATIAL REPRESENTATION of ELEMENTARY ALGEBRA	IEEE VisLang	9209
VIRTUAL REALITY AND EXPERIENTIAL COMPUTATION	VisLang (tutorial)	9209
MULTIPLE CONCURRENT REALITIES	VR Symposium	9210
VRt	CyberArts	9211

1993

EXPERIENTIAL MATHEMATICS	NASA ICAT/VET	9303
VIRTUAL INTERFACE TECHNOLOGY	UCLA 3-day Course	9303
BOUNDARY MATHEMATICS IN NON-LINEAR SPACES	university	9305
VR IS NOT A SIMULATION OF PHYSICAL REALITY	SIGGRAPH (panel)	9307
STUDENT ERRORS ARE UNIQUE	university	9310

1994

ENTITY-BASED MODELING IN VIRTUAL ENVIRONMENTS	ComplexAgents (invited)		9406
HOW USEFUL ARE VIRTUAL WORLDS	IFIP (panel)		9406
INCLUSIVE COMPUTING	IFIP (plenary)		9407
AT THE FOUNDATIONS OF VIRTUAL REALITY	German AI (keynote)		9407

1995

DISTINCTION NETWORKS	German CS (keynote)		9504

1996

COMPUTER HUMANITIES	university		9603

2000

SET ASIDE A SPACE	university		0003

2001*

THE CIRCUIT DESIGN GENERATOR	investor	various	0111
CM85A	corporate		0112

2002*

TECHNICAL VALIDATION PROJECT: SUMMARY REPORT	corporate		0203
BTC TECHNICAL DESIGN REVIEW	corporate		0205
BTC PRODUCT DESIGN	corporate		0205
BTC SOFTWARE TOOLS AND CAPABILITIES	investor	various	0205
COMESH PROGRESS REPORT	corporate		0205
CHANGING THE RULES OF DIGITAL DESIGN	investor	various	0208
COMESH TECHNICAL REVIEW	corporate		0209
BTC MARKETING PRESENTATION	investor	various	0210
BTC INVESTOR PRESENTATION	investor	various	0208
BTC COMPANY OVERVIEW	investor	various	0203
BTC TECHNICAL CAPABILITIES	investor		0210
EMBEDDED COMESH	investor		0211

2003*

COMESH BUSINESS PLAN	investor	various	0301
ILOC RESULTS	corporate		0309

2004*

BTC BOARD OF DIRECTORS TECHNICAL REVIEW	corporate		0403
ILOC REPORT	corporate		0404
SYNTHESIS APPLICATIONS OF BOUNDARY LOGIC	investor	various	0411

A LOSP SAMPLER
William Bricken
November 1984

Presented at the ADS Technology Transfer Seminar Series

ANNOUNCEMENT

Losp is a LISP compatible programming language and/or data structure consisting only of variable names and parentheses. All control structures, logical connectives and quantifiers are implicitly represented either by shared list membership or by containment within parentheses.

For example, the LISP to Losp conversion of IF-THEN-ELSE:

```
(IF a b c)   ==>   ( ((a) b) (a c) )
```

The formalism is a single operator logic, based on Spencer-Brown's Laws of Form. An extension due to Varela permits representation of non-terminating, unknowable and paradoxical programming constructs (autonomous structures) without degrading the functionality of the logic.

The focus of the seminar will be on Losp applications rather than on formal proofs. Topics will include:

1. Constructive and intuitive understanding of The Laws of Form

2. Predicate calculus as an interpretation of Losp

3. Fast resolution theorem proving

4. Losp program representation

5. The unity of declarative and functional programming

6. Program compilation, optimization, and Let-by-need

7. Circuitry, networks, and cons cells structures

8. LOSP distributed processing

9. Program Representation Language in Losp

10. Imaginary Boolean values, self-reference, and the clarification of paradox.

A LOSP SAMPLER -- LONG SEMINAR ANNOUNCEMENT

William Bricken
May 1985

Presented at the ADS Technology Transfer Seminar Series

This seminar will provide an overview of Spencer-Brown's LAWS OF FORM, and its interpretations as logic and as a programming language.

The LAWS OF FORM is a new mathematics that is simpler than (and foundational to) formal logic, set theory, and the arithmetic of numbers.

LOSP is a LISP compatible language with these characteristics:

a. The nesting of variables within parentheses replaces all control structures. For example, the LISP if-then-else, (IF a b c), is represented as (((a) b) (a c)) in LOSP.

b. Evaluation of expressions takes place by one operation: erasure. Memory is released, not consumed, by this process. Expressions that evaluate to FALSE disappear from the representation.

c. Inference (proof, deduction) takes place via erasure of irrelevant structures. The direction of inference (forward, backward) is an irrelevant concept.

d. The representation of non-terminating, unknowable, and paradoxical structures (autonomous values) does not degrade the functionality of the proof procedure.

e. Formally, LOSP is fully general, complete and consistent.

SEMINAR CONTENTS

PART I: THE MATHEMATICS OF INDICATION

1. A Constructive and Intuitive Understanding
2. The Arithmetic of Indication and the Origin of Distinction
3. The Algebra of Indication and the Origin of Computation
4. The Calculus of Indication and the Origin of Time

PART II: THE INTERPRETATION AS LOGIC

1. Mappings
2. Proof Techniques
3. The Clarification of Paradox

PART III: LOSP

1. Program Representation
2. Program Optimization
3. Circuitry, Networks, and Cons-cell Structures

DON'T MISS THE BEGINNING.

PARALLEL DEDUCTION
William Bricken
March 1986

Presented at the ADS Technology Transfer Seminar Series

ANNOUNCEMENT

I will present a network representation (for Lisp, Prolog, and Predicate Calculus) that permits parallel control and deduction. Nodes in the representation evaluate their local neighbors in parallel, erasing irrelevant structure. What remains is the simplest form of the input: the global solution emerges from local activity. The representational innovations that convert global, sequential notations to their distributed, parallel form are:

> Distinction nodes replace logical and control structures; each distinction node is a local control expert,
>
> A single point of access to the representation replaces the input/output duality,
>
> A goal gradient resolves communication conflicts,
>
> A single operator logic converts the semantics of TRUTH to that of GOAL. Falsity does not exist.

Examples include the transitivity of implication, expert system inference, LISP's COND, and the recursive function FACTORIAL.

SELF-ORGANIZING DISTINCTION NETWORKS

William Bricken
August 1986

Presented at the Sign-and-Space Conference

SEMINAR ANNOUNCEMENT

Problems in propositional calculus are transcribed into a linear version of the indicational calculus. Then the linearity of Spencer-Brown's notation is removed by transcription into a Distinction Network, in which the nodes represent either distinctions or individuals, and the arcs represent the containment relation. Each distinction node can apply the simplification theorems of the indicational calculus independently and in parallel to reduce the network from a representation of the problem to a representation of the solution. The changes in structure of the network represent a parallel proof. Since each distinction node acts only on its immediate locality, the entire network can be seen as exhibiting emergent self-organization around its logically optimal structure. Distinction networks are implemented in LISP and used as an inference engine for AI expert systems.

A SIMPLE SPACE
William Bricken
August 1986

Presented at the Sign-and-Space Conference

SEMINAR ANNOUNCEMENT

The lines upon which we write mathematical symbols impose constraints upon mathematical thinking. A simple space of representation is proposed that does not enforce the linear concepts of associativity, commutativity, duplicity of representation, and binary arity. The properties of this simple space are discussed in the degenerate case when the space is empty and in the self-referential case when the space contains only representations of itself. The implications of a representational space without linear properties is then explored for propositional calculus.

LOSP
William Bricken
October 1986

Presented at Hackers3.0 by invitation

Losp is an implementation of a relatively unknown deductive technique, Spencer-Brown's LAWS OF FORM. The Losp Deductive Engine accepts any set of well-formed sentences in predicate logic. If one sentence is specified as a goal, Losp deduces whether or not that goal is consistent with the other sentences.

The core of Losp's representational power resides in the redefinition of the concept of nothing, as expressed in LISP by the token NIL. LISP overlays another meaning upon the concept of nothing: that of the EMPTY LIST.

I will show you how *nothing* can be used in computation.

PURE LISP AS A NETWORK OF SYSTEMS

William Bricken
August 1986

Presented at the Second Kansas Conference on
Knowledge-based Software Development

The Intelligent Program Editor project has concentrated on a distributed representation of LISP code that is able to respond to queries about its own structure and function. We are currently able to automatically parse Pure LISP into a functionally invariant representation called Losp. In Losp, each token of the input code is a system, with an i/o buffer, an agenda and prioritization mechanism, and handlers that permit response to queries about the token's state. Systems are composed into data-flow networks, in which nodes represent objects and arcs represent function-argument dependencies. A novel, single-operator logic permits control structure to be expressed as a network of arbitrary arity NOR nodes. Queries to the functional network can locate unbound variables, determine usage of bound variables, and identify contingent function invocations.

PARALLEL DEDUCTION USING BOUNDARY OPERATORS I

William Bricken
November 1986

Presented at the Stanford Mathematical Methods Seminar Series

Portions of three reports are combined into a state-of-the-art presentation on the use of Laws of Form for parallel deduction. The talk is intended to be entertaining and provocative.

INTRODUCTION TO BOUNDARY OPERATORS AND FUNCTIONAL SPACES

Two innovative mathematical concepts are introduced and applied to the representation of logic. Boundary operators are a two dimensional representation of operators that are traditionally expressed by a single word. They contain their arguments. Functional spaces operate on all tokens that exist within that space. In effect, they make some operators implicit in the representation of objects. An empty functional space is an implicit constant. Not writing some operators yields a computational speed-up without loss of expressability.

A SIMPLE SPACE

A simple space of representation is proposed that does not enforce the linear concepts of associativity, commutativity, duplicity of object representations, and binary arity. The properties of this simple space are discussed in the foundational case when the space is empty and in the self-referential case when the space contains only representations of itself. These properties define boundary operators and functional spaces. The implications of a space of representation without linear properties are explored for propositional calculus. A single-operator graph notation is proposed as a simplification of traditional linear notations.

PARALLEL DEDUCTION USING DISTINCTION NETWORKS

Distinction networks represent boundary operations as graph operations. Simplification is strongly parallel: the global minimum is generated by nodes evaluating, in parallel, their local connectivity. Rather than goal-directed simplification, distinction networks can simplify to a representation of all achievable goals that use the initial forms. An example of parallel deduction is provided.

PARALLEL DEDUCTION USING BOUNDARY OPERATORS II

William Bricken

January 1987

Presented at the Stanford Mathematical Methods Seminar Series

Delimiting tokens such as parentheses and brackets are traditionally used in mathematics in indicate the precedence of operations. I will illustrate the use of delimiters as operators by assigning them a functional semantics, in the domains of integer arithmetic and logic. Since delimiters are inherently two dimensional tokens, they contain their arguments, permitting a representation that de-emphasizes the linear concepts of associativity, commutativity, duplicity of object tokens, and binary arity. Distinction networks represent boundary operators as graph operations. The implications of representations without linear properties are explored for propositional calculus. An example of strongly parallel deduction is provided.

ANALYZING ERRORS IN ELEMENTARY MATHEMATICS

William Bricken
February 1987

Presented as my Stanford School of Education dissertation oral defense.

An accurate model of the processes by which novices make errors in elementary mathematics would contribute to our understanding of learning and to our style of teaching. Recently, research in Intelligent Tutoring Systems has focused on the automated description and classification of errors in order to enhance the tutor's student model.

An error is defined by an *error schema*, a transformation pattern whose left-hand-side matches the problem before transformation, and whose right-hand-side matches the incorrect result. This research characterizes errors as *unique*, since most error schema occur only once for problem sets with similar problems. Errors that are stable over time for one student are rare, and confined to specific problems rather than to classes of problems. Common errors that occur for many students are common only to the extent that they describe course-grained features of the subject domain. On close inspection, students do not share common error processes, even when they get the same wrong answer. These conclusions are supported by two studies, and are consistent with those error studies incorporating detailed individual process data.

In a study of a minimal formal system which permitted accurate tracking of errors, each subject exhibited an idiosyncratic set of errors. Statistical, protocol, and clinical analysis of longitudinal data from an eighth grade pre-algebra class provided no support for the existence of algorithmic mal-rules. Students did exhibit

- *demand-driven errors* when expected to respond to problems for which they lacked necessary skills
- *goal-driven errors* when the expected form of the answer guided the response
- *compensatory errors* when a transformation that resulted in a recognizably absurd form was corrected
- *extra-domain errors* which introduced notation external to the lexicon of the domain

- *multiple solution techniques* which were selected from based on personal rather than formal criteria, and

- *catalytic clarification* when a single problem triggered complete understanding of a transformation rule without feedback.

The conclusion is that error schemata do not adequately describe error processes, and are highly sensitive to the degree of abstraction of the data they do model. The future of automated tutoring may be to model correct procedures and to identify common syntactic pattern errors, freeing teachers for individual remediation of semantic misunderstandings.

THE POTENTIAL OF ICAI

William Bricken
April 1987

Presented at the IEEE Symposium for Computer-based Education

Intelligent Computer-aided Instruction constructs models of student behavior in order to remediate errors. I show that errors cannot be modeled by ICAI, and therefore we should use computer-based instruction for teaching correct models only.

BOOLEAN FORMAL SYSTEMS
William Bricken
May 1987

Presented at the ADS Technology Transfer Seminar Series

ANNOUNCEMENT

A tour through typographical and pictorial Boolean representations. Beginning with the intuitive English concepts of elementary logic, we progress through increasingly more powerful and relevant representations of the foundations of rational thought. Typographic representations progress through implicational to algebraic to non-redundant to complex object systems, while embedding the weaknesses of lack of visual obviousness, static form, and representational explosion. Diagrammatic systems provide visual models, but introduce non-implementability and search. A void-based pictorial system (distinction networks) is visually friendly, implementable, efficient, and suggestive. The elimination of repeated objects is achieved by several two-dimensional techniques. Containment maps provide a cognitive model that returns to the intuitive understanding of English. The keys to representational elegance and efficiency include non-representation, unique objects, diagrammatic formalisms, set functions, and unity of object and process.

PREAMBLE

In celebration of the 133 anniversary what Russell calls the first book ever written on mathematics, George Boole's "An Investigation of the Laws of Thought, on Which Are Founded the Mathematical Theories of Logic and Probabilities" (1854), I'm giving a memorial technical exchange.

Boole invented formal mathematical symbolism by suggesting that each separate token could represent a relatively simple and distinct idea. What a charming and quaint notion! We can then be free of the irrelevancies of what a symbol means in different contexts. We no longer need to rely on all those words of description and explanation and qualification. We save on paper. We move toward ideographs, toward pictures of abstraction. We can judge, blindly, by appearance. We can divorce understanding from computation. The pen becomes mightier than the thought!

By a strange twist of fate, this idea is was just what was needed for *automated symbolic computation*. By reducing intelligence to tokenism, we can make machines smart. We can call this new found intelligence *artificial*. We can even pretend that we think by symbol processing!

Boole originated the Godelian concept that number and symbol are identical concepts. He quantified reality: 1 is the Universe, 0 is Nothing. Genius is identified by a characteristic equation. Boole's fundamental equation,

$$x\ x = x$$

honors Occam. See through duplicity, cyberpunks! Divine the roots!

$$\begin{aligned} X\ X &= X \\ X\ X - X &= 0 \\ X\,(X - 1) &= 0 \end{aligned}$$

what is never double?

The roots are $X = 0$ and $X = 1$.

Nothing is not double: $0\,(0 - 1) = 0$

Everything is not double: $1\,(1 - 1) = 0$

There is only one One, there is only one None. Same is same.

BOUNDARY NUMBERS

William Bricken
June 1987

Presented at the ADS Technology Transfer Seminar Series

SEMINAR ANNNOUNCEMENT

Boundary mathematics is a two dimensional redefinition of the traditional one dimensional mathematics we have grown to know and love. Boundary notation embodies an inherently parallel model of computation and of thought. The axioms for computation with boundary numbers are identical to the axioms of deduction for boundary logic.

Traditionally, numerical objects are trivial to specify while numerical computation requires effort. Boundary numbers are active objects which require computational effort to read. In exchange, they are trivial to compute with. Computation is independent of the magnitude of a number, and requires only pointer switching. Reading a boundary number is a once-only operation, after computation is completed. As well, reading is an asynchronous parallel process. I will show you how to add, subtract, multiply, divide, and solve equations using boundary numbers.

Yes, the implication of boundary innovation is that we can use a Connection Machine for computation that is faster than a crunching bit, more powerful than a super-computer, and able to reap tall generalizations in a single bound.

TECH SEMINAR REMINDER

BOUNDARY NUMBERS

Every now and then, someone invents a new kind of number with handy properties. *Negative* numbers permit the concept of debt. *Irrational* numbers permit the measurement of diagonals. *Imaginary* numbers permit objects to be analyzed as waves.

NOW! COMING TO ADS FOR THE FIRST TIME EVER!

Boundary numbers permit parallel computation by pointer switching. Tired of juggling bits when adding? Sick of borrowing when subtracting? Frustrated with multiplication tables and esoteric algorithms for long division?

STEP RIGHT UP FOLKS!

Boundary numbers make arduous computation obsolete. To add or to multiply, just point boundary numbers at each other and its done. Compute by naming alone, avoid the ugly mess of figuring.

DON'T MISS THIS GOLDEN OPPORTUNITY!

Find out why everything you learned about arithmetic is baroque. Discover the true tyranny of the place-value notation. Escape from the anachronistic assumptions of sequential processing.

LOOK, UP IN THE SKY! IT'S A BSURD! IT'S A PLAIN! IT'S BOUNDARY NUMBERS!

(This offer is void where prohibited. Adults over 17 must be accompanied by an open mind.)

VISUAL PROGRAMMING

William Bricken

July 1987

Presented at the ADS Technology Transfer Seminar Series

SEMINAR ANNOUNCEMENT

We proudly present a picture show accompanied by some words.

Abstract

To be computationally useful, a visual language must have mathematical characteristics such as semantics, composability, and deductive transformation. I'll show you some recent advances in *diagrammatic formal systems*, and comment on the structure and feel of pictorial proof. Examples will include boundary logic, boundary integers, Boolean lattices, pictorial programs, distinction networks, and computational maps. The utility of a mathematical visual language includes program animation, pictorial metaphor, and parallelism.

DISTINCTION NETWORKS

William Bricken

September 1987

Presented at the First Artificial Life Conference

Boundary mathematics provides techniques for computing with tokens that represent boundaries. The formal system created by assigning the semantics of elementary mathematics to boundary tokens (such as parentheses) rather than to label tokens (such as words) is both representationally elegant and computationally efficient. In boundary logic, for example, a single boundary token, called a distinction, replaces all the constants and connectives of elementary logic. The rules of deduction erase and create structure, rather than rearrange tokens. These rules are inherently parallel. The global representation of a problem is a dynamic network; local, asynchronous application of rules settle the network into a representation of the solution. The global structure can be viewed as a symbolic organism with a metabolism (or value structure) driven by the need to minimize structural redundancy. Thus, distinction networks are self-determining. Each node in the network is an independent process. If the computational resource that is host to a distinction network is seen as an environment providing raw material, the minimization process of the network becomes analogous to organic self-organization. Thus, distinction networks are a symbolic analog of life. As a side-effect, they perform parallel computation.

THE GRAPHICAL FOUNDATION OF COMPUTATIONAL MATHEMATICS

William M. Bricken
October 1987

Presented at the ADS Technology Transfer Seminar Series

ANNOUNCEMENT

All computation can be automated as transformation on diagrams rather than as rearrangement of tokens. *Boundary mathematics* is a diagrammatic formal system that achieves computation by permitting boundaries to be both descriptive and operational. In contrast to geometric figures and Venn diagrams, boundary operators act locally on the spaces they enclose to transform map representations from problems to solutions. Currently we know how to express the following token-based theories in boundary mathematics: Boolean algebra, propositional logic, equational predicate logic, rational arithmetic, set theory, recursive function theory, and functional LISP and Prolog.

The diagrammatic languages are more concise (a many-to-one map from tokens), more expressive (accommodating contradiction), more efficient in computational time and space (erasure as the primary operation), strongly parallel (no global clock, locks, semaphores or blocking), and pictorial (trace is an animation). The diagrammatic techniques are both powerful and visual. Hill-climbing to locate global maxima, for example, is accomplished by rearrangement of terrain rather than by search of the surface points.

Applications include verification, compilation, optimization, and partitioning of rulebases and programming languages, parallel computational and deductive engines, non-iconic visual programming languages, and non-symbolic cognitive models.

CONMAN: THE CONSTRAINT MANAGEMENT SYSTEM

William Bricken and Eric Gullichsen

January 1988

Presented at the ADS Technology Transfer Seminar Series

Constraint-based programming techniques compute over sets of possibilities, rather than over instances. Traditional programs require a variable to be *bound* whenever it is used. In contrast, constraint programs use *free* variables, treating them as algebraic unknowns. Rather than searching instance-space, constraint programs specify possibility-space. Problems of bindings, search, multiple hypotheses, and sequential determinism are not relevant to constraint programming. Similar to other declarative approaches, constraint programs emphasize domain engineering rather than writing code.

We will describe two versions of ConMan, our constraint management system.

Crypto-ConMan solves cryptarithmetic problems (SEND + MORE = MONEY, find the unique numerical solution). It exemplifies our approaches to domain specification, minimization of search, opportunistic representations, intelligent data structures, type demons, conditional demons, and algebraic narrowing.

Graphic-ConMan is a "spread-sheet" for diagrams. It illustrates our approaches to functional equations, inequalities, and interactive control and display.

Y'all come.

"Mathematicians stand on each other shoulders while computer scientists stand on each others toes."
--Richard Hamming

INTRODUCTION TO BOUNDARY MATHEMATICS

William Bricken
February 1988

Presented as the inaugural talk at the Autodesk Technical Forum

MAIN IDEA

Boundary Mathematics is a new foundation for mathematics. It is constructive, intuitive, and exceedingly simple. Boundary Mathematics places meaning on edges rather than words. The fundamental concept is the distinction rather than the label. A distinction bounds and constrains, providing reference from a higher dimension. On a flat page, a closed loop, such as a circle, makes a distinction.

SYNOPSIS

We begin with the story of creation. The initial distinction embodies the unity of observer&observed, process&object, sentience&symbol. From the singular act of existence, we construct the rules of creation: indistinguishability and self-reference. We concretize, forming cardinality&recursion, ego&introspection, form&function. We identify logic and category and arithmetic in the form of distinction.

Having reached the two-year-old mind, we individuate. The audience relaxes. The speaker shares his adventures in mapping. Metaphysical, physical, psychological, and mathematical anchors. Computational examples. Fast deduction. Representation hacking. Visual programming. Asynchronous parallelism. Contradiction maintenance.

The speaker reviews the synopsis. The audience concludes.

DON'T MISS THE BEGINNING!

Mention the VOID without refuting it and your admission will be cheerfully refunded.

BOUNDARY LOGIC
William Bricken
June 1988

Presented at the Autodesk Technology Forum

Elementary logic is considered to be the basis of rational thinking and reasoning. It forms the core of mathematical proof, computer programming, legal argument, and responsible discussion. And it is needlessly complex, cluttered with redundancy, and difficult to follow.

We will take a brief tour through the evolution of logic: Aristotle's discovery of reason, Boole's invention of symbolism, Russell's construction of the logical foundations. Our destination is a clear view of the stuff of thinking.

Boundary logic demonstrates that traditional logic is made of one thing: the *distinction*. We can indicate a distinction by placing a boundary between ourselves and the object we consider to be distinct. I'll show how making boundaries is the same as being rational, how boundaries can replace the logic of true, false, and, or, not, if, every, some, etc.

Simple examples will illustrate that the single concept in boundary logic is easier to use and to understand. Deducing conclusions from complex premises becomes simply the erasure of unnecessary parts. Implementations of boundary logic are efficient, elegant, and inherently parallel.

DRAWING AS COMPUTATION

William Bricken
August 1988

Presented at the Stanford Center for Integrated Facilities Quarterly Meeting

The Autodesk Research Lab is developing and exploring graphic languages, with the goal of using drawing and drafting as computational languages. This research takes several directions:

1) **Intelligent Drafting Tools**: By combining the drafting capabilities of AutoCAD with symbolic computation engines, we are building constraint-based design tools. The dream is to be able to specify building codes symbolically, and have the CAD system automatically disallow plans that do not meet code.

2) **Boundary Mathematics**: This innovative formalism uses graphic structures for both description and execution of computation. The user will be able to draw programs, debug visually, and watch execution animations. Boundary programs are inherently parallel. Boundary computation provides natural algorithm animation. The operation of the CAD system would be visually obvious to the user.

3) **Interactive Simulated Realities**: We are designing systems in which components of pictures are computational entities. The drawings on a screen are capable of autonomous activity, including self-configuration. The user participates within the drawing interactively, obeying the rules of the simulated reality. As well as drawing a building, the architect would be able to simulate dwelling in that building.

Visual computation can be both easy to understand and formally rigorous. It provides immediate feedback and supports visual parallelism. And intelligent drawings can store and compute large amounts of information efficiently. We envision a future in which every thousand words is replaced by a single picture.

MATHEMATICA EXPOSED

William Bricken

September 1988

Presented at an Autodesk Technical Forum

Wolfram Research's MATHEMATICA is an integrated numerical and symbolic computational system which includes graphics output, modern programming languages, a developing environment, and full integration with UNIX, C, TEX, and POSTSCRIPT. Internally, MATHEMATICA has one representation for all objects (labeled boundaries), and one mechanism for symbolic computation (substitution of equals for equals).

I'll describe the computational philosophy of MATHEMATICA, some of its more powerful features, MATHEMATICA programming, and our ideas for connecting MATHEMATICA to AutoCAD.

Naked symbols! Narcotic transformations! Modern programming frenzy! Numbers without shame! MATHEMATICA exposed! At the next scandalous Autodesk Forum.

MODEL INTERFACE MODEL

William Bricken
October 1988

Presented at Hackers4.0 by invitation

The dominant model of communication is a broadcast model which involves two end-points and a transmission line. I'll show how to convert the transmission model into a boundary model by removing the conceptual flaws of the transmission model applied to human communication.

FRACTURTLES: Using Pictures as Operators

William Bricken
March 1989

Presented at an Autodesk Technical Forum

Using an intrinsic perspective, data that represent drawings can be interpreted as programs that construct drawings. By adopting a formal model of pictures based on the constructive commands FORWARD and TURN, we gain the ability to automate drawing.

In Fracturtles, drawings are expressed as short strings of instructions. Every drawing can be used as an operator to process other drawings. {tree, leaf, fish}, for instance, generates a tree with fish as leaves. The formal string language provides the mathematical structure for an interpreter (from drawings to strings), for an optimizing compiler (image compression), for a visual programming language (draw and point), and for a picture generator (strings to drawings). The implementation is verifiable, robust, extendable, modular (abstract) and very fast.

Fracturtles is a hierarchy of graphics languages, each providing a unique and rich domain for the construction and exploration of automated drawings:

simple drawings	(figures)
stack drawings	(graftals)
reentrant drawings	(fractals)
recursive drawings	(recurtals, new)
parametric drawings	(constraint classes)
iterative drawings	(turtle graphics) and
branching drawings	(L-systems and natural models)

Lots of examples, surprising characteristics, and implementable generalizations.

FIRST ANNUAL STATE OF THE LAB REPORT

William Bricken
March 1989

Presented at an Autodesk Technical Forum

An informal report on what the Autodesk Research Lab has been doing. Software prototypes (Rube, Lispkit, Fracturtles) and long-term projects (Forum, Boundary Mathematics, Cyberspace). The role of software research in business. What we think the next generation CAD systems will look like. The key ideas of formal modeling and visual programming. Knowledge, language, and meaning, and the advantages of formalism. Dimension, space, and interface, and the advantages of visualism. Modern science and the construction of reality.

CYBERSPACE, THE DEMO

William Bricken and the Autodesk Research Lab
June 1989

Presented at the Autodesk Technology Forum

Next week the Autodesk Research Lab's Cyberspace team will present the Cyberspace demo to the public, but first we want to show it to YOU, our fellow employees, whose hard work and continuing success makes this research project possible.

Company Demo Day

Here's how the demos are set up:

There will be a 10-minute time slot for each individual demo, scheduled from 10:00 A.M. to 10:00 P.M., with a break from 4:00 - 6:00 for the Forum.

If you want a demo, please sign up for one of these slots.

The demos will be videotaped, and participants will be asked to fill out a short questionnaire before leaving.

Weird Science Forum

A panel discussion will be presented by the Cyberspace team. William Bricken will talk about the purpose of the demo, and serve as moderator. Eric Gullichsen will discuss systems architecture; Randy Walser will talk about the software architecture; Gary Wells will present some interface ideas and devices; Meredith Bricken will discuss movement and perception in Cyberspace.

EXPLORING VIRTUAL REALITY IN CYBERSPACE

William Bricken
July 1989

Presented at AutoExpo'89 and at
Multimedia and Hypermedia Expo'89

The Autodesk Research Lab is exploring a new interface and modeling technology for CAD called Cyberspace. The idea is to create a CAD working environment that simulates reality, and then to immerse the user in that virtual reality. Cyberspace hardware is intended to provide the illusion of direct experience with computer-generated models. Cyberspace software provides the tools for constructing and interacting with these models as if they were real.

Eventually, Cyberspace will provide tools that will redefine the way we use computers. Imagine a 3-D image of a building, for example. Cyberspace lets you step into the image and walk around inside it to explore it. Imagine looking out of a virtual window, and seeing the scene that would be there in reality. In Cyberspace, you could change the view by grasping the window frame and pulling it to a new location. The underlying database would be updated automatically.

Today, Cyberspace is a research project, a primitive prototype of future capabilities. Tomorrow, Cyberspace could be yours.

A VISION OF VIRTUAL REALITY

William Bricken
March 1990

Presented as the keynote address for
NASA Research Institute for Advanced Computer Science Conference'90

Computers are not only symbol processors, they are reality generators. Until recently, computers have generated only one dimensional symbolic strings. Text and numbers. Text is a code which, when read, generates images of reality in our minds. During the 80s, we enhanced the expressability of computation by adding space and time dimensions to the realities being generated. Two dimensional windows, 2D animation, solid modeling, simulation. Now, in the 1990s, computer systems can generate virtual environments, entire multisensory worlds which include us as interactive participants. Digital information can seem as-if-real, changing our notions of computation, symbolism, meaning, metaphysics, self, and culture.

The potential for VR to contribute to societal infrastructures such as manufacturing, marketing, telecommunications, science, entertainment, art, education, medicine, and media, suggests an economic impact that rivals the Gross National Product. We live in two superimposed worlds, the one of mass and the one of information. The huge accumulation of difficult to access words on paper indicates that the world of mass is not particularly well-suited for dealing with information. As our culture matures into an information society, we are now discovering the virtual world, an ideal place for interacting with information.

VR IS INHABITED

William Bricken

March 1990

Presented at the Annual National Computer Graphics Association Conference

The computer is first a symbol processor. Although decades have barely passed, it is transforming our concepts of information and information processing. But the computer has yet to be understood for what it is of itself, we still view it from the impoverished model of what it replaces. McLuhan said that computers extend our central nervous system. But our CNS is not a symbol processor, it is a reality generator. The essence of the computer revolution is yet to come, computers are essentially generators of realities. Cyberspace, virtual reality, embodies the fundamental nature of computers, the creation of a diversity of realities.

BOUNDARY LOGIC
William Bricken
May 1990

Presented to the UW Industrial Engineering Graduate Seminar, and to the UW Intelligent Systems Lab in Electrical Engineering

Elementary logic is the basis of rational thinking, mathematical proof, computer programming, legal argument and responsible discussion. It is also needlessly complex, cluttered with redundancy, and difficult to follow. Boundary logic, an extension of Spencer-Brown's Laws of Form, demonstrates that propositional logic is made of one thing: the *distinction*. I'll show how making boundaries is the same as being rational, how boundaries can replace logic.

Boundaries delineate a space, boundary transformations form a graphical calculus. The map from logical tokens to boundaries is many-to-one, boundary logic therefore simplifies as well as illustrates. Graphical proofs trivialize most problems in logical deduction. The clarification of logic is attributable to assigning semantics to the Void. Graphical (as opposed to typographical) computation is more efficient, visually comprehensible and inherently parallel. I will demonstrate several other visual languages by mapping containment into networks, maps, paths, rock walls and toy blocks. These formalisms permit concrete visualization and manipulation of abstract mathematics.

LEARNING IN VR

William Bricken

May 1990

Presented to the UW Educational Technology Seminar

Rather than teaching a structure of symbols (such as algebra) and then teaching the meaning of that structure, in VR we will first teach meaning through experience, then (if necessary) teach the symbolic abstraction of our experiences. Virtual environments are not constrained to only viewing. The student can interact with objects and spaces in VR. The student can use tools to create new environments, to modify old ones, to take simulation exams, to fix errors, to play.

VR teaches active construction of the environment. Data is not an abstract list of numerals, data is what we perceive in our environment. Learning is not an abstract list of textbook words, it is what we do in our environment. The hidden curriculum of VR is: make your world and take care of it. Try experiments, safely. Experience consequences, then choose from knowledge.

VISUAL COMPUTATION

William Bricken
May 1990

Presented to the UW Industrial Engineering Graduate Seminar

By representing mathematical concepts with tokens strung together in lines, we place a symbolic filter on mathematical understanding. Associativity, commutativity, and duplication of variables are artifacts of linear notation which place unnecessary computational burdens on proof systems. Most pictorial approaches to the representation of mathematics (graphs for example) are descriptive, they do not provide axioms for computational transformation.

Focusing on propositional calculus, I will demonstrate a many-to-one transformation from tokens to containers that maintains logical semantics. This boundary logic therefore simplifies as well as illustrates. Pictorial proofs trivialize most problems in logical deduction. Graphical (as opposed to typographical) computation is more efficient, visually comprehensible and inherently parallel. I will then generate several other visual languages by transforming containers into networks, maps, paths, rock walls and toy blocks. These formalisms permit manipulation and interaction with abstract mathematics in virtual reality.

MINDWARE
William Bricken
May 1990

Presented at the Microsoft Technology Exchange

The Virtual Environment Operating Shell is a software suite currently written in C that wraps around the UNIX operating system. VEOS provides resource and communication management for coordination of the modules which make a VR system:

i/o hardware, behavior transducing input and display devices
world construction kits, CAD packages
dynamic simulation kits, for interaction and animation
virtual world tools
computational and display processors

The behavior and sensory transducing subsystem implements the fundamental interface paradigm shift of VR, from user actions that accommodate the needs of symbolic computation to natural participant actions which are interpreted by the computational system.

The VEOS coordinates display and computational hardware, software tools and resources, and world models. The VEOS provides a wide range of software tools for construction of and interaction with models, including editors of objects, spaces, and abstractions; movement and viewpoint control; object inhabitation; boundary integrity; display, resource and time management; multiple concurrent participants; programmable internal processes within models; and history and statistics accumulation. Some potential user interface tools include the Wand, for identifying objects, connecting, moving, jacking, grasping, and drawing; and the Virtual Body for attaching arbitrary hardware sensing devices to arbitrary representations of body components, for collecting physiological measurements of behavior, and for maintaining coherence between a participant's model of physical activity and the virtual representation.

BOUNDARY LOGIC, BOUNDARY IMPLEMENTATIONS

William Bricken
May 1990

Presented to the UW Computer Science Graduate Colloquium

Boundary Mathematics (BM) is

> a theory of representation in which higher-dimensional structures (containers, blocks, networks,...) carry mathematical and functional semantics, and
>
> a theory of computation in which non-representation (void-based concepts) has functional utility.

These characteristics make BM tricky to implement on register-based computer architectures. But the following performance features make BM particularly desirable:

Computational efficiency: use of the void makes representation more efficient and converts most computational steps from rearrangements to erasures.

Visualization: use of space makes representation visual and redefines many computational steps from rearrangement to direct observation.

Parallelism: containers naturally partition independent and dependent processes, providing a parallel control strategy.

SOCIAL IMPLICATIONS OF CYBERSPACE

William Bricken
July 1990

Presented at CPSR Directions and Implications of Advanced Computing '90

VR is coming, inevitably and rapidly. It has captured the public imagination, it taps a positive emotional core. Consider the comments: empowering, the feeling of freedom, dreaming while awake, vast potential in every direction. The characteristics of VR include inclusion (encapsulating the participant), pluralism (complete individual customization), cognitive integration, immaterial realism, cornucopia, and maintainable paradox. Life in cyberspace poses fundamental social and ethical questions. What is the role of physical reality in the information age? Is Psychology the Physics of VR? What are the cognitive effects of programmable bodies, of synesthetic sensation, of masslessness, of complete empowerment? Who has the right to limit the immaterial? How will folks respond to explicitly interpenetrating world views? What are the rights of autonomous programs? Are there sensory channels to ecstasy? Is living inside VR necessarily pathological?

THE SOFTWARE ARCHITECTURE OF VIRTUAL REALITY

William Bricken
July 1990

Presented the research groups at Digital Equipment Corporation,
SUN Microsystems, and
Stanford Center for Design Research

Interface is a boundary between two systems. VR is a new paradigm which eliminates interface by including the participant (formerly the excluded user) in a surrounding, multisensory, digital environment. The interface is the body; interaction is defined by physiology and by natural behavior. The Virtual Environment Operating System is software for construction of, maintenance of, and interaction with virtual realities. I'll describe the design of systems which mediate between graphics tools, simulation engines, behavior transducers, and display hardware. I'll also describe software tools which empower participants in a virtual environment (Virtual Body, Virtual Community, The Wand, Abstraction Editors, Inhabitation, Autonomous Entities). Computers are no longer symbol processors, they are reality generators.

AN INTRODUCTION TO VIRTUAL REALITY

William Bricken

August 1990

Presented to Pre-Engineering students at UW (January 1991)

Virtual Reality is a direct experience digital environment, it is inclusive computing. The fundamental shift of perspective is from the outside looking in to the inside looking about. VR redefines Computer Science. The hardware used to access VR senses the natural behavior of the participant and displays computational results from the subjective viewpoint of the participant. VR software permits construction of, maintenance of, and interaction with inclusive digital environments. Interface design endeavors to adapt computation to humans, rather than contorting humans to the needs of digital representations. Semantics is embedded in VR experience; symbolics, in sensation. This profound shift of perspective initiates the end of the childhood of computers.

TECHNICAL ISSUES FOR VIRTUAL WORLD TECHNOLOGY

William Bricken
August 1990

Presented at the *Hip, Hype, and Hope Panel*, SIGGGRAPH'90.

Innovations are classically first described in terms of what they replace. The automobile was first a horseless carriage. And it was built as though it were a carriage and it moved at the same speed as a carriage. It was fifty years later that we found that it transformed culture and our notion of space and our notion of travel. Television was first radio with pictures. Computers are replacing calculators and typewriters and desktops and filing cabinets. I'll suggest that VR marks the end of the infancy of computation. Computers are not symbol processors, they are reality generators.

VR APPLICATIONS
William Bricken
October 1990

Presented at Cyberthon

Extensions and applications of the cyberspace concept are being considered in diverse fields such as engineering, medicine, space exploration, education and entertainment.

VIRTUAL WORLD SOFTWARE

William Bricken

February 1991

Presented to the Virtual Worlds Symposium

Virtual environment software tools coordinate display and computational hardware, software functions and resources, and world models. Software tools for construction of and interaction with digital environments include

-- movement and viewpoint control
-- object inhabitation
-- boundary integrity
-- editors of objects, spaces, and abstractions
-- display, resource and time management
-- coordination of multiple concurrent participants, and
-- history and statistics accumulation.

EDUCATIONAL APPLICATIONS OF VR

William Bricken
June 1991

Presented as the keynote at the London Virtual Reality'91 Conference

VR interface technology couples human intention with increasingly powerful information systems. VR offers intuitive, experiential access to the potentially overwhelming complexity of fiber-optic telecommunications systems that will link individuals, groups, and vast amounts of information into the global matrix referred to as cyberspace. The multi-participant system we propose to develop links workstations using VR interface technology with fiber-optic telecommunications. This system can be extended to provide participants with synchronous 3-D multi-sensory access to virtual worlds.

VR's augmentation of behavior for amplified perception provides new modes of interaction in a diversity of fields, including education and collaborative work. However, education is not simply another application area for VR. The principles of learning, perception, and performance have shaped the design and evolution of this interface technology since its inception. The purpose of VR is to provide a more natural way for us to access and to understand complex information. It allows us to use the increasing power of technology in ways that are compatible with how we have learned to do things since infancy. Decreased learning time and increased performance are demonstrated advantages of this interface technology.

THE MATHEMATICAL BASIS OF CYBERSPACE

William Bricken

September 1991

Presented as a plenary talk at the Second Annual Conference on Virtual Reality, Artificial Reality, and Cyberspace

In the beginning is the mind. To formalize VR, we must shift from a symbolic mathematics to a spatial mathematics and we must build the participant into the axioms of space. I will discuss a theory of representation and transformation which permits us to describe and maintain VRs mathematically. The central concept in this theory is the Void.

Let ( ) be a distinction between realities. For a familiar interpretation, let ( ) be the boundary between mind and body. In another interpretation, ( ) is the boundary between reality and cyberspace. Physical reality on the outside, participant on the inside. Let i be the participant. Spencer Brown's Laws of Form provide an axiomatic basis for VR:

Observe: **i ( ) = ( )**

Participate: **(i) =**

The left-hand-side of each equation is descriptive (objective), explicitly mentioning the participant. The right-hand-side is experiential (participatory), implicitly using the participant's perspective. Architectural design has a sensual, experiential semantics. It is but a quirk of typography that we have ignored the experiential semantics of mathematical languages.

I will show how these axioms generate logic, numbers, and sets, the basis of traditional mathematics. I'll also show how to build visual, non-symbolic mathematical forms (networks, maps, paths, rockwalls, toy blocks, and rivers) in cyberspace. These forms permit direct manipulation and interaction with abstract mathematics and with the programs running VR from within VR. Inclusion is essentially self-referential. In the end is the paradox.

VIRTUAL WORLDS

William Bricken

October 1991

Presented as an invited speaker at Hackers7.0

I will show you our work in designing and exploring virtual worlds. These worlds include:

from Autodesk:

```
OpenPlan    -- interaction with kitchen objects
CyberCity   -- using a physical bicycle to tour a virtual world
FishWorld   -- swimming with sharks
Racketball  -- hitting a ball using simulated physics
```

from University of Washington:

```
Virtual Seattle   -- just that
VSX         -- an Osprey to fly around
Metro       -- a train ride around Seattle
Kid' Worlds -- constructed during a one week summer program
```

BOUNDARY NUMBERS

William Bricken
December 1991

Presented as a graduate seminar at UW

The final examination is our last glance at how we have changed over the course. This talk attempts to lift us from blind confidence on learned knowledge, assuring that the experience of "finals" is indeed a glance at the past.

Boundary Numbers

We will redefine the representation of integers so that they are trivial to compute with but difficult to read. We'll step back to kindergarten, learning how to count with number networks rather than with number lines. We'll redefine group theory so that traditional axioms are embedded in the experiential properties of space. We'll explore dynamic representations and functional spaces which provide autonomous simplification of form in parallel. We'll see how to add and to multiply numbers in constant time, independent of magnitude. We'll examine spatial inverses, distribution via idempotency, and boundary solutions to simple equations. And we'll learn that algebraic substitution is sufficient mechanism for numerical computation.

Find out why everything you learned about arithmetic is baroque. Discover the true tyranny of the place-value notation. Escape from the anachronistic assumptions of sequential processing.

LOOK, UP IN THE SKY! IT'S A BSURD! IT'S A PLAIN! IT'S BOUNDARY NUMBERS!

A curious asymmetry in our perception of reality is that there are always more questions than answers. Have we confused representation with computation, must we abandon the data/process dichotomy? Has the binary heritage that supports mind/body duality finally collapsed into unary symbolism? Has the Arabic trick of stringing orders of magnitude in lines finally flowered into space? Does non-existence have meaning? What is the prettiest number that you ever knew? Is Ivan Sutherland actually at Sun working on gate circuitry that operates asynchronously using difference as the only state? Are even the foundations of mathematics but fads passing in the night? How come we can space out but not space in?

PROGRESS IN VIRTUAL REALITY
William Bricken
January 1992

Presented as the keynote at Imagina'92

After a few years experience with virtual reality systems, years we have learned some lessons:

> Psychology is the Physics of VR.
> Our body is our interface.
> Knowledge is in experience.
> Data is in the environment.
> Scale and time are explorable dimensions.
> One experience is worth a trillion bits.
> Realism is not necessary.

A major theme of VR research is that Psychology, in the broad sense of behavior, perception, cognition and intention, provides the rules and the constraints of virtual worlds. Psychology is the Physics of VR.

Our body is our interface. Interface is not something that is out there, in some machine. Interface is a boundary which both connects and separates, interface takes place at the surface of our skin. From the perspective of VR, interface is Physiology, interaction is natural behavior.

VIRTUAL REALITY IN THE CLASSROOM

William Bricken
February 1992

Presented as the keynote at the Northwest Council for Computer Education together with a class on the subject

The central issue for Education in virtual reality is transfer of learning. There are two kinds of transfer of interest: generalization of experience in VR to later experiences in VR, and transfer of experience in VR to behavior in physical reality. The generalization question is relevant to interaction with information in VR, the transfer question is relevant to simulation of physical interactions.

As well, VR poses deeper, more philosophical issues for Education. With good teaching, attention comes first, learning comes after attention is focused. And learning is primarily action. VR provides an empowering context for focus of attention, and learning through action. For curricula, VR provides a substrate for construction of arbitrary learning environments. The idea is simple, everything we do to educate with words and with pictures can be provided as virtual experience. We can vary location, scale, density of information, interactivity and responsiveness, time, degree of participation. VR provides the opportunity for individualized instruction and personalized learning environments. VR provides an automated, responsive learning context, autonomous entities that can track behavior, guide interaction, and remediate errors. The issues will be identify tools and sequences of behavior that provide long-term learning. These are the same issues of curriculum development in general.

AT THE BOUNDARY OF REALITY

William Bricken

August 1992

Presented as the keynote at Computing and Philosophy'92

The Copernican revolution introduced a physics that differed fundamentally from appearance. VR introduces a metaphysics that differs fundamentally from the material. At the foundation of Objectivism is an attempt to be realistic about the material world. VR calls for immaterial realism, for being realistic about information. The currency of VR is organization, not possession, not accumulation, not territory. All laws are transmutable, we can satisfy fantasy rather than fact. It is science itself that is redefined. In VR, we can choose to be reductionalist, but at the bottom of it all, there is not Mass or Nature, there is the Void. VR is representational, but not a priori rational, empirical, or verifiable. VR is illogical positivism: if you can specify it, it is meaningful. All empirical hypotheses are true.

SPATIAL REPRESENTATION of ELEMENTARY ALGEBRA

William Bricken

September 1992

Presented at the IEEE Workshop on Visual Languages'92

Our understanding of a concept is tightly connected to the way we represent that concept. Traditionally, mathematics is presented textually. As a consequence novice errors, in elementary algebra for example, are due as much to misunderstandings of the nature of tokens as they are to miscomprehensions of the mathematical ideas represented by the tokens. This paper outlines a *spatial algebra* by mapping the structure of commutative groups onto the structure of space. We interact with spatial representations through natural behavior in an inclusive environment. When the environment enforces the transformational invariants of algebra, the spatial representation affords experiential learning. *Experiential algebra* permits algebraic proof through direct manipulation and can be readily implemented in virtual reality. The techniques used to create spatial algebra lay a foundation for the exploration of experiential learning of mathematics in virtual environments.

MULTIPLE CONCURRENT REALITIES

William Bricken
October 1992

Presented at the Second Annual VR Symposium

Virtual Reality is the first scientific instrument of metaphysics, permitting validation of experience across alternative realities. I will describe mathematical tools which coordinate actions of multiple participants in virtual worlds, and software architectures which permit inhabitation of arbitrary virtual bodies.

Experience is dual, both objective and subjective, both exclusive and inclusive. The two corollary forms of information exchange are broadcast,

[A]<-->[B]

and direct

[A][B]

where the boundary,][, represents direct contact. Virtual worlds permit arbitrary direct communication between entities. Virtual worlds also permit mutually inconsistent models across multiple participants. Each participant can maintain a separate personal reality concurrently in the same virtual space. Communality of mutually shared perspectives is negotiated rather than assumed. Inconsistency (both A and not A) is formalized using imaginary Boolean operators.

VRt
William Bricken
November 1992

Presented at CyberArts'92

We describe innovations in terms of what they replace. Only after decades do we come to understand the pervasive impact of new technologies on our culture. The automobile was first the horseless carriage. It replaced the carriage, looked like a carriage, and moved at the speed of a horse. Decades later, the automobile has transformed our landscapes, the pace of our travels, and our concept of space. The television replaced the radio. Television programs were first radio programs with pictures. Decades later, the television has transformed our evenings, the pace of our senses, and our concepts of news and entertainment.

The computer is first a symbol processor. Although decades have barely passed, it is transforming our concepts of information and information processing. But the computer has yet to be understood for what it is of itself, we still view it from the impoverished model of what it replaces. McLuhan said that computers extend our central nervous system. But our CNS is not a symbol processor, it is a reality generator. The essence of the computer revolution is yet to come, computers are essentially generators of realities. Cyberspace, virtual reality, embodies the fundamental nature of computers, the creation of a diversity of realities.

EXPERIENTIAL MATHEMATICS

William Bricken

March 1993

Presented at the 1993 NASA Conference on
Intelligent Computer-Aided Training and Virtual Environment Technology

We have been able to demonstrate that mathematics itself (in particular logic, integers, and sets) can be expressed concretely, using 3D arrangements of physical things, such as blocks on a table, doors open or shut, rock walls that respond to gravity, the things of everyday life. String-based symbolic representations of mathematical concepts are typographically convenient, but tokens are not at all essential to mathematical expression. VR makes it convenient to express abstract ideas using spatial configurations of familiar objects. One benefit of this approach is that we can build visual programs, set them on a virtual table, and watch them work. We can experience programs as other entities rather than as dumps of text. Bugs would manifest as structural anomalies, as visual irregularities. Architectural design has a sensual, experiential semantics. It is but a quirk of typography that we have ignored the experiential semantics of computational languages. More fundamentally, experiential computing unites our spatial and our symbolic cognitive skills, permitting mathematical visualization, analytic gestalt, whole brain processing.

BOUNDARY MATHEMATICS: BASIC MATHEMATICS IN NON-LINEAR SPACES

William Bricken
May 1993

Presented at UW Computer Science Seminar

Typography has lead us to confuse linearity with rationality. I will describe a mathematics based on spatial containers, and illustrate its interpretations for algebra, logic, and numbers. Containers, or boundaries, can refer to spaces of arbitrary dimension, increasing the dimensionality of possible notations. Boundary based mathematical representations are more concise (a many-to-one map from tokens), more expressive (accommodating contradiction), more efficient in computational time and space (erasure is the primary operation), strongly parallel (no global clock, locks, semaphores or blocking), and experiential (interactive in three dimensional space).

Focusing on propositional logic (the simplest interesting formal system), I will demonstrate how conventional logic is needlessly complex, how proof can be achieved by examination, and how space both simplifies and illustrates. I will then generate several visual languages by transforming containers into networks, maps, paths, rock walls and toy blocks. These experiential formal systems permit manipulation and interaction with computer programs and with abstract mathematics in virtual reality.

VIRTUAL REALITY IS NOT A SIMULATION OF PHYSICAL REALITY

William Bricken
July 1993

One of the weakest aspects of current software tools for VR is that designers are bringing the assumptive baggage of the world of mass into the digital world, undermining the essential qualities of the virtual. Information is not mass; meaning is *constructed* in the cognitive domain. Psychology is the Physics of VR. In building virtual worlds, we are continually discovering that they are strongly counter-intuitive, that our training as physical beings obstructs our use of the imaginary realm.

The greatest design challenge for VR tools is mediating between physical sensation and cognitive construction. VR software must directly resolve the mind-body duality which plagues both Western philosophy and computer languages. VR calls for a philosophy of *immaterial realism*. VR doesn't matter, it informs.

I will briefly describe a new generation of software tools which emphasize virtual rather than physical modeling concepts. VR tools are situated, pluralistic, synesthetic, paradoxical, and most importantly, autonomous. Their programming techniques include behavioral programming (entity-based models), inconsistency maintenance (imaginary Booleans), possibility calculi (set functions), relaxation (satisficing solutions), experiential mathematics (spatial computation), participatory programming (inclusive local parallelism) and emergence (non-linear dynamics).

STUDENT ERRORS ARE UNIQUE

William Bricken

October 1993

Presented at the UW Graduate Student Education Colloquium

Recently, research in Intelligent Tutoring Systems has focused on the automated description and classification of errors in order to enhance the tutor's student model. This research characterizes errors as *unique*, since most error schema occur only once for problem sets with similar problems. Errors that are stable over time for one student are rare, and confined to specific problems rather than to classes of problems. Common errors that occur for many students are common only to the extent that they describe course-grained features of the subject domain. On close inspection, students do not share common error processes, even when they get the same wrong answer. These conclusions are supported by two studies, and are consistent with those error studies incorporating detailed individual process data.

ENTITY-BASED MODELING IN VIRTUAL ENVIRONMENTS

William Bricken

July 1994

Presented at the Complex Agent Architectures'94, by invitation

Immersive virtual environments provide a natural testbed for complex interaction between humans and computational agents. In VR, every object should be able to exhibit both reactive and autonomous behavior; every participant should be free to interact arbitrarily with any object. The demands of immersive interaction have lead us to a particular type of agent architecture: entity-based modeling. Entity-based modeling extends object-oriented programming to systems-oriented programming by creating agents that act as independent operating systems, controlling their own process resources, memory resources, and interprocess communication. An entity can be conceptualized as an organizationally closed quasi-biological system with control functions that define perception, action, and motivation. An essential component of this model is that every entity serves both as an object interacting within an external context and as an environment providing global context for its internal content. For VR applications, entities also serve as virtual bodies which are controlled by the dynamic activity of human participants.

AT THE FOUNDATIONS OF VIRTUAL REALITY

William Bricken
July 1994

Presented as a keynote at the German Artificial Intelligence Conference

VR has spawned two cultures, technical and societal. To date, technical VR has focused on hardware capabilities and on integration of technical subcultures (simulation, telepresence, networking, operating systems, graphics, etc.). Societal VR has focused on the exciting potential and the exhilarating freedom of information space. Both cultures struggle to comprehend the domain they have stumbled upon.

I will present some initial explorations into grounding cyberspace. We can make VR simulate physical reality only by severely constraining its potential, by eliminating its desirable features. Bits and mass are incommensurable, but a semantic double-cross permits us to confound meaning and representation. VR incorporates the participant intimately, making an objective perspective unavailable. It places the programmer and the participant inside information, in a subordinate yet humane role within a larger environment. I'll describe the properties of the information environment and their implementation, starting with the mathematical foundations (void, distinction, perception) and ending with the virtualization of experience.

INCLUSIVE COMPUTING

William Bricken
July 1994

Presented in the plenary session of the World Computer Conference

Computer technology has only recently become advanced enough to solve the problems it creates with its own interface. One solution, *virtual reality* (VR), immediately raises fundamental issues in both semantics and epistemology. *Inclusive symbolic environments* effectively redefine the relationship between experience and representation, rendering the syntax-semantics barrier transparent. Reading, writing, and arithmetic are hidden from the computer interface, replaced by direct, non-symbolic environmental experience. *Inclusion*, the defining characteristic of virtual environments, is achieved through the integration of four component technologies: behavior transducers, inclusive computation, intentional psychology, and experiential design. The structure and function of VR systems reflect the mathematical necessity of *pervasion* of worlds, physical pervading digital pervading virtual. Pervasion permits novel semantic mappings which challenge the dominance of physical reality. *Physical semantics* is defined by the map between behavior and digital representation. *Virtual semantics* is defined by the map between digital representation and perceived behavior in the virtual environment. *Natural semantics* is achieved by eliminating our interaction with the intermediate digital syntax.

DISTINCTION NETWORKS

William Bricken

September 1995

Presented as keynote at the 19th Annual Conference of the AI Division of the German National Computer Science Society

Intelligent models rest on a foundation of elementary logic. Intelligent systems call for organizationally closed networks of interacting processes. An interesting step in the evolution from intelligent models to intelligent systems is to approach logic itself as a system of autonomous elementary processes, called distinctions.

Distinction networks (dnets) represent logical expressions as directed graphs of nodes and links. Distinction nodes act as fine-grain logical agents with knowledge only of their local connectivity, behaving as a generalized NOR operator. Links represent logical dependency. Dnets differ from circuits in that logical values are mapped onto the existence of links between nodes rather than onto signals propagating over links. Nodes can then enact Boolean reduction independently and asynchronously based on their local context of connectivity. The entire dnet operates in parallel to arrive at valid logical conclusions and reduced Boolean functions. Dnet deduction is equifinal but non-deterministic.

The map from a conventional basis for logic {if, true} onto distinctions is many-to-one, demonstrating that dnets are conceptually simpler than propositional logic. This simplicity is a consequence of confounding the existence of a communication channel with the message on that channel. The structure of the network embodies logical semantics, while the organization of the network is closed and homogeneous.

The computational complexity of the Tautology problem for propositional calculus occurs in dnets as the necessity to identify duplicated graph structures. Even in this simple domain, nodes must cooperate with neighbors in order to identify common networks of connectivity. Cyclic loops in dnets introduce primitive models of time and recursion, generating a Boolean waveform calculus and complex Boolean values with strong similarities to complex numbers.

COMPUTER HUMANITIES
William Bricken
March 1996

Presented at the Seattle University Graduate Seminar in Computer Science

Our interaction with computers and with computing is evolving. We have moved from an emphasis on humans learning the symbol systems of the computer (command lines and programming languages) to an emphasis on the computational generation of environments familiar and comfortable to humans (multimedia and virtual environments).

Computer Humanities addresses the forth-coming humanization of the computer interface. The multimedia/virtual reality paradigm shift is a renegotiation of the boundary between human friendliness and computer formalism. This shift introduces new theories of representation and of meaning. New software interaction techniques focus on the entire body as interface, involving the whole person in a digital interaction. New programming techniques are used to construct computational entities and personal agents. Deeply involving the computer participant in a digital environment requires ecological and introspective software tools. The technical content of this course would cover multimedia computer architecture, interactive programming methodology, distributive agent modeling, computer graphics, hypertext, virtual reality, human physiology, cognitive psychology, and collaboration theory.