PROJECT NARRATIVE

CURRICULUM SOFTWARE FOR COMPARING
SYMBOLIC AND MANIPULATIVE FORMAL SYSTEMS

A Proposal to the Department of Education Institute of Education Sciences
CFDA Number: 84.305 Topic 3 Goal 2

PI: William Bricken, Ph.D.
Lake Washington Technical College

1. PROPOSAL

We propose to design, develop, test in classrooms, and formatively evaluate curriculum materials that compare mathematics errors across symbolic and spatial modes of presentation. We wish to address what Lee Shulman calls the *pedagogical content knowledge* of mathematics, that particular form of mathematics that is most germane to its learnability. What makes mathematics comprehensible to a novice? Which mathematical forms afford the fewest errors in understanding both structure and meaning? Which transformation rules and axioms are more error-prone? When should symbolic models of mathematical concepts be augmented with manipulatives? And in particular, does intervention using additive systems enhance concept learning in mathematics classrooms?

The Proposal outlines a development plan for software that displays multiple representations of mathematical problems, permits manipulative interaction with any display, and facilitates fine- and course-grain measurement of violations of transformation rules used to solve the problem. The multiple representations will include *symbolic forms* in textual notation and *manipulable additive forms* in spatial notation. Thus the software will serve as a testbed for empirical comparison of learning facilitated by both symbolic and manipulative mathematical forms. The Proposal describes three manipulative formal systems to be compared to conventional symbolic representations: *unit arithmetic*, *depth-value notation*, and *spatial algebra*.

1.1 ADDITIVE SYSTEMS

Additive numeric systems have been use since the dawn of history. An additive system is one in which the representation of a sum is the same as the representation of the parts. When we place coins in our pocket, we use an additive system; the value of the coins is equivalently represented by the coins individually, and by the collection in our pocket. Additive systems usually have a physical interpretation, however they can also be abstract -- an example is the tally system, which uses identical unitary marks to indicate cardinality. In contrast, symbolic systems require rote memorization of both representations and algorithms to determine a sum, since by design the representation of concepts is independent of their meaning. Additive systems are graphic and intuitive, while symbolic systems are typographic and formal. While the intent of a symbolic system is to completely separate semantics from syntax, the intent of an additive system is to maintain a close connection between visceral understanding and representation.

INCREASING CORRELATION BETWEEN REPRESENTATION AND REALITY

In mathematics education, older students *learn* symbolic rules to manipulate numerals and algebraic expressions, while younger students *use* manipulatives to understand how numbers and abstractions work. Concrete and virtual manipulatives rely upon *spatial models* and transformations. Symbolic calculation and modern algebra rely upon textual, *string-based models* and transformations. This difference leads to a significant discontinuity in the teaching of mathematics, the change in perspective that students face when moving from an elementary curriculum grounded in concrete manipulatives to a secondary curriculum grounded in abstract symbols. The discontinuity is deeply connected to how mathematical concepts are represented. The above figure illustrates how representation approaches reality as we move from symbolic to spatial models. Even within symbolic syntax, some representations intentionally carry a visual meaning within their structure, while others intentionally do not. For example, some syntactic variants of the concept three include:

••• , Ⅲ , 3 , 𝒷 , 三 , ٣ , ㄹ , 11 , (•)• , three , **drei** , *trois* , 1+1+1 , 6/2 , {{},{{}},{{},{{}}}}

Differences between the rules and axioms of systems incorporating additive manipulatives and systems incorporating symbolic abstractions may manifest in a variety of student errors, including failure to connect the two systems, slips in symbol manipulation, misapplication of tools and algorithms, partial and fragmented integration of concepts, and confusion and muddling of cognitive models. More visual, semantics-laden notations may aid mathematics understanding for concrete, visual, tactile and experiential learners, and for those who have disabilities, disadvantages, or cognitive and behavioral resistance to the conventional symbolic structure of mathematical expressions.

1.2 MANIPULATIVES

Conventional manipulatives are analogical, relying on an underlying geometry or metric to construct a mapping between spatial form and mathematical concept. One problem with assessing their effectiveness is that each manipulative relies upon a different analogy, and thus affords different understandings and errors. We propose to standardize measurement of the potential benefit of manipulatives by introducing a formal basis for spatial representation of mathematical concepts, providing axioms that rigorously define meaning and transformation of spatial forms. Thus we will be able to isolate manipulative errors associated with the violation of specific structural axioms in both symbolic systems (the axioms of group theory) and spatial systems (the axioms of additive mathematics). Not only should we be able to identify which rules and axioms are associated with learning difficulties, but we should also be able to localize the specific transformation rules for which one approach (textual/symbolic or spatial/

manipulative) might improve upon the other. The proposed research will use *comparative axiomatics* to identify differential performance qualities associated with string-based and spatial approaches to mathematics teaching. In particular, we will develop software tools that permit side-by-side comparison of performance using spatial and symbolic representations, either alone or in combination. We wish to look deeper into how each mode uniquely contributes to mathematics understanding. This in turn may shed some light on when and how to combine manipulative and symbolic systems for effective teaching.

1.3 CURRICULUM DEVELOPMENT SOFTWARE

The feasibility of incorporating *manipulative formal systems* into the curriculum depends upon computer-based display and animation. Only recently, with the advent of web-based virtual manipulatives, have display and interaction technologies become available to promote diagrammatic and spatial mathematical systems from a second-class role as informal aids to a first-class role as rigorous formal tools. Virtual interaction and manipulation provides flexibility, extensibility, goal-directed guided interactivity, and tractable cost. We propose to develop *virtual manipulatives for additive systems*, prototype software that provides support for:

- ✦ comparison of token-string and spatial formal systems (comparative axiomatics),
- ✦ orthogonal variation of content and structure for multivariate research,
- ✦ development of supplementary and comparative curriculum units, and
- ✦ comparative evaluation of learning under different syntactic and semantic approaches.

We will implement generic virtual manipulative animation software that incorporates a core engine sensitive to the axiomatic system being presented. The software will input and output both spatial and symbolic notation and display a dynamic representation of ongoing computations in both systems. Students will be able to directly manipulate the symbolic and the spatial forms to effect their own computational sequences. The modular software architecture will permit syntax, semantics, and interactivity to be decomposed into orthogonal components, in support of multivariate evaluation of performance and error behavior over representational dimensions, axiomatic bases, display conventions, and interaction styles. We will use a diversity of measurement techniques, including informal interview and discussion, protocol analysis of error behavior, formative performance evaluation, and structured pilot studies to assess both the value of the interactive software tool and the adequacy of the software to support factored experiments.

Software design, iteration and refinement will include continuous significant classroom usage by math challenged students at LWTC and at the Lake Washington Technical Academy high school located on-campus. During the second year of the Project we will evaluate the software tools in elementary classrooms.

1.4 PILOT STUDY

The pilot study has a 2x3x5x3 incomplete block factored design, with main effects of types of system (additive and symbolic), dimensions of representation (1D, 2D, and 3D), transformation axioms (associativity, commutativity, identity, inverse, and distributive rules), and course level (remedial arithmetic, arithmetic, and elementary algebra). The dependent variables will be a diversity of measures of error behavior on problems from standardized arithmetic and algebra tests used in Lake Washington Technical College (LWTC) mathematics courses, and on highly structured problem sets that systematically vary the transformations needed to reach a solution. This analytic structure provides informative results in most measurement cases. A key aspect of the design is to put both manipulation and symbolic transformation on equal formal footing, thus removing the effect of particular physical types of manipulative. A main effect for systems would indicate that the formal structure of one particular system engenders fewer errors. A main effect for dimension of representation would clearly indicate that the *formal structure* of either 1D symbol manipulation, 2D diagrammatic visualization, or 3D physical interaction is more effective. Main effects for transformation rules would point to specific sets of transformations associated with errors. Least interesting would be a main effect for content, indicating that some content is more difficult. As with any multifactor design, it is the presence of interactions that would provide the most information. Interaction between systems and transformations would indicate specific techniques that are preferable for specific performance difficulties. Similarly, interactions between representation and transformation would indicate that particular mathematical models have inherent learning difficulties for particular tasks. Interaction between course level and structure would confirm current practices of using different models at different levels of student maturity. Finally, with no effects or interactions, we would know that the representation of mathematical concepts does not effect performance.

1.5 POTENTIAL BENEFITS AND DISADVANTAGES

We see these potential benefits in an empirical comparison of the pedagogical qualities of symbolic and manipulative systems:

- ✦ Comparison of errors associated with various systems of representation can guide research insight into the syntactic and semantic sources of mathematics miscomprehension.
- ✦ An alternative spatial notation may assist teaching and learning of some concepts by some students. In particular, we expect to identify different types of learners who benefit from different types of instruction.
- ✦ Multiple representations can broaden mathematics understanding. A comparative axiomatics provides students with multiple perspectives on mathematical concepts and may enrich our teaching of mathematics.
- ✦ A capability to directly compare spatial and symbolic forms may help to bridge the gap between learning mathematical concepts through manipulation of objects,

and learning mathematical concepts through substitution and rearrangement of symbolic tokens.

Some potential disadvantages of the proposed intervention include:

- Non-standard axioms, even if they incorporate significant pedagogical advantages, do not conform with the prevailing group theoretic basis for modern algebra. This may create problems for communication of knowledge.
- Students may become confused when presented alternative perspectives on the foundations of mathematics. Teachers will not be familiar with the availability of alternative foundations.
- Computer-based virtual manipulatives might blur the distinction between concrete and abstract.
- The community of scholars knowledgeable about mathematical foundations is small. The axiomatic approach may be inappropriate.
- The software systems used to develop the prototype curriculum materials may not be available to all classrooms.

We believe that each of these concerns can be adequately addressed.

- *Communication of knowledge:* Additive systems are intuitive, are taught in early grade school mathematics, and will be presented as supplemental to understanding the symbolic mathematics of middle school and high school. The non-standard basis is widely taught, it is just not associated with axiomatic rigor. The proposal adds formal models of already existing practices.
- *Alternative perspectives:* Students are currently exposed to both manipulative and symbolic systems, and already exhibit confusion. Part of the motivation of the proposal is to develop software tools that provide interactive comparison of models of mathematical concepts, so that the effectiveness of the models can be evaluated.
- *Abstract manipulatives:* Virtual manipulatives do create an interaction between physical and symbolic mathematical models, we would like to characterize this interaction empirically. Virtual manipulatives have been widely funded by the NSF; this proposal contributes to their understanding.
- *Community:* There is little research or pedagogical interest in mathematical foundations, although the Rules of Algebra are in every textbook. The axiomatic perspective is necessary for measuring and evaluating performance, but it should be overtly in the curriculum only if measured to be beneficial.
- *Software availability:* We intend to build software systems in languages that have free downloadable players available over the Internet.

Therefore we expect a reasonable chance that the prototype curriculum software and materials could provide a potentially positively impact on mathematics education, since

- Identification of the relative strengths and weaknesses of symbolic and spatial approaches may guide better structuring of classroom curricula and materials.
- Spatial models are rigorous yet both simple and familiar, possibly affording less opportunity for student error [1].
- Representational variety strengthens both mathematical content and reasoning [2].
- Learning through concrete interaction with spatial and visual forms can enhance mathematics understanding, particularly when the structure of the manipulative closely aligns with abstract concepts [3].
- Providing comparative models of mathematical concepts side-by-side may help students who are failing to comprehend symbolic form, and may suggest curriculum approaches that smooth the transition from concrete to abstract mathematics.

2. RELEVANCE

2.1 EDUCATIONAL MOTIVATION

It is well known that America's students are underperforming in mathematics education. On the Washington [State] Assessment of Student Learning (WASL) in 2006, for example, half of the students in Grades 6 through 10 failed to meet grade-level standards of performance for mathematics [4]. The 2006 overall current failure rate of 50% incorporates a 75% failure rate for Washington State minorities and a 70% failure rate for students living at the poverty level, a group composed primarily of white and Asian students [5]. The WASL is based on the Principles and Standards for School Mathematics developed by the National Council of Teachers of Mathematics [6]. After a decade of extensive effort, the 2006 results represent a significant improvement, having increased from a 33% pass rate in 1999.

The LWTC Academy High School has a 30% pass rate for basic math courses. Students enrolled in adult education and job training math courses at LWTC are an average age of 33, and have a much higher pass rate. However these adult students come to LWTC with very little prior math training and/or comprehension. They are 60% female, 30% of color, 5% disabled, 30% educationally disadvantaged, and 16% economically disadvantaged. Thus the Project will be able to work with minority and math challenged students throughout the development of the curriculum materials. Although multivariate analyses over sub-populations is not proposed during the Project, some demographic data will be gathered for informal analysis. This information will be used primarily to guide curriculum development that addresses the mathematics learning needs of at-risk groups.

2.1.1 The Additive Principle

Early primary school, as well as early recorded history, identifies addition by the Additive Principle: the representation of a sum is identical to the representation of its parts [7]. This principle is exemplified by *unit-arithmetic*, unary stroke or tally arithmetics within which all units are represented by individual unit marks or tokens [8]. Addition in unit-arithmetic is achieved by placing representations of units together in the same space. The Additive Principle might be considered to be the *definition* of the additive operation.

Addition is introduced in early elementary school as the physical act of joining together collections. In contrast, symbolic addition, also introduced in early elementary school, does not follow the Additive Principle; the symbols being added do not possess the structural properties of the cardinality they represent. Instead, students memorize number facts and algorithms to determine a sum. Of course, symbolic numerals can be decomposed into units, reverting their symbolic behavior back to that of additive unit-arithmetic. Teacher training texts recognize the importance of additive systems throughout lower elementary mathematics. However, these texts explain the meaning of addition in terms of symbol manipulation, not in terms of the spatial intuitions of the Additive Principle. For example, commutativity of unit-arithmetic is achieved by fiat:

> "We may associate 3+5 with putting a set of 3 members in a dish, and then putting a set of 5 members in a dish to form the union of the sets. We associate 5+3 with putting the 5 set in a dish and then putting in the 3 set." [9, p 121]

Here, the Additive Principle is used as a physical analog for the Commutativity of Addition. Unit-arithmetic addition, however, does not incorporate an external dish or a temporal ordering of actions. When two or more collections of units share the same table, children can add all of them by pushing the piles together concurrently.

2.1.2 Virtual Manipulatives

There is abundant evidence that interactivity can assist students who are having difficulty learning abstract material [10][11]. Students find it easier to learn math if it is made concrete through the use of manipulatives [12]. Use of manipulatives produces greater gains in achievement [13]. Spatial representations enhance understanding, since expressing ideas spatially allows information to be analyzed more effectively by parallel perceptual processes than by linear cognitive processes [14][15]. Many different ways of making mathematical concepts more concrete have been shown to be effective in learning algebra [16][17].

Teachers use manipulative and diagrammatic techniques widely [18]. Due to the pragmatic limitations of physical objects, physical manipulatives are constrained to "concrete examples", just as a triangle drawn in a geometry class is only a *particular* triangle. However, with new software tools such as Geometer's Sketchpad [19], a student can "draw" a generic triangle, and use the diagram itself as an object of computation. Digital technology has expanded the domain of representation of mathematical concepts from typographical strings to spatial forms and to virtual manipulatives.

A *virtual manipulative* is "an interactive, Web-based visual representation of a dynamic object that presents opportunities for constructing mathematical knowledge" [20, p373]. Reimer [21] and Clements [17] discuss potential benefits of virtual manipulatives for learning. Today, web-based applets that provide virtual models of mathematical concepts and computations are burgeoning. Extensive collections of free virtual manipulatives software for mathematics are maintained by the math archives at Drexel University [22], by the National Library of Virtual Manipulatives at Utah State [23], by the National Council of Teachers of Mathematics Electronic Examples [24], and by public [25], commercial [26], and home-schooling [27] interest groups.

Virtual manipulatives provide iconic models that simulate concrete manipulation [28]. They currently bear a strong resemblance to concrete manipulatives such as Cuisenaire rods, base-ten blocks, pattern blocks, rulers, number lines, logic blocks, fraction pieces, and geoboards. *Spatial analogs* are lines, scales, dials, etc. that support the concept of number within their geometry. Visual and spatial analogs and models are elevated to a curriculum design principle in *How Students Learn: Mathematics in the Classroom*:

> "**Design Principle 3:** Providing Visual and Spatial Analogs of Number Representations That Children Can Actively Explore in a Hands-On Fashion" [29, p292]

Often though, representations based on spatial analogs fail to adequately model the concept of multiplication. The additive spaces of dials and scales do not support multiplicative concepts. Spatial analogs provide a visual model but not an algorithmic understanding. A primary difficulty with spatial analogs is that we do not know how the structure of the analog will interact with the abstract concepts being taught. Uttal [30] for example, has observed that students identify with familiar but incidental attributes of a manipulative, and thus make the wrong generalizations, connecting incidental characteristics of the teaching aid to the mathematical concepts being taught.

Virtual manipulatives are decidedly constructivist. Students construct meaning by using computer input devices to control apparently physical actions of virtual objects through translation, rotation, flipping and other spatial transformations. For example, graphing linear equations can be made virtual both by generating a graph given textual input of a linear equation, and by generating the linear equation that corresponds to dragging and rotating the line graph itself [31].

> "Students liked the immediate feedback they received from the applets, the virtual manipulatives were easier and faster to use than paper-and-pencil, and they provided enjoyment for learning mathematics. Their use enabled all students, from those with lesser ability to those of greatest ability, to remain engaged with the content, thus providing for differentiated instruction." [32]

Today the libraries of virtual manipulatives are organized around specific content and grade levels. We know that they sometimes enhance learning, but we do not yet know which particular manipulative techniques will address which particular difficulties, how a particular technique might correct a difficulty, or when a teacher should elect to change from a symbolic to a physical model. Web-based virtual manipulatives generally confound the distinction between concrete application and symbolic abstraction. Alternatively, they can be seen as a unifying force, integrating haptic, visual, conceptual and behavioral interpretations [33]. Manipulatives must be related explicitly to relevant concepts, otherwise students learn different but unconnected techniques [34].

Multiple perspectives and representations are known to improve concept learning [35]. "The usefulness of numerical ideas is enhanced when students encounter and use multiple representations for the same concept."[3, p2]

> "Mathematics programs in the early grades should make extensive use of appropriate objects, diagrams, and other aids...Different ways of representing numbers, when to use a specific representation, and how to translate from one representation to another should be included in the curriculum."[29, p292]

Bruner [36] distinguishes three types of representation for mathematical operations. Each plays an essential role in mathematics understanding. *Enactive* addition, for example, is concrete; collections of objects are placed together physically. *Symbolic* addition abstracts the cardinality of a set of objects into a symbolic name such as "3" or "7". Rules for combining symbolic names guide the determination of their sum. *Iconic* addition is most commonly presented as pictures of groups of objects with specific cardinality. The value of multiple representations suggests the possibility of concurrent use of concrete, virtual and symbolic forms. This in turn calls for a deeper understanding of the relative and contextual pedagogical merits of each form of representation.

Algorithms are known to depend upon representation. In digital computation, for example, decimal numerals are abandoned entirely in favor of binary numerals. In math education, the algorithms for manipulation of fractions differ from those of decimals, although fractions and decimals can express the same abstract concept of ratio of magnitudes. Operations on pie diagrams representing fractional quantities are again of a fundamentally different, concrete nature than the symbolic algorithms for both fractions and decimals.

> "Physical representations serve as tools for mathematical communication, thought, and calculation, allowing personal mathematical ideas to be externalized, shared, and preserved...mathematical ideas are enhanced through multiple representations, which serve not merely as illustrations or pedagogical tricks but form a significant part of the mathematical content and serve as a source of mathematical reasoning." [3, p 94-95]

Not only do we need to know more about the comparative value of representations, we need to understand the strengths and weaknesses of the transformation sequences engendered by each representational form.

2.2 HILBERT'S PROGRAM

A century ago, well prior to the advent of digital display technologies, David Hilbert used his considerable influence to exclude diagrams and spatial intuition from formal mathematics and consequently from the mathematics curriculum [37]. Hilbert's agenda has been rigorously pursued, so that today "Mathematics is a game played according to simple rules with meaningless marks on paper." *(David Hilbert, c. 1900)*

> "... numbers have neither substance, nor meaning, nor qualities. They are nothing but marks...[38]
>
> "...proof is a syntactic object consisting only of sentences arranged in a finite and inspectable array." [39]
>
> "...despite the obvious importance of visual images in human cognitive activities, visual representation remains a second-class citizen in both the theory and practice of mathematics." [40, p3]

The fundamental structures of formal mathematics do not necessarily align with the needs of novice learners or with the goals of mathematics education. Higher mathematics is expressed almost exclusively in symbolic string languages designed by experts for experts, not for novices who may have never considered that a letter could stand in place of a set of numbers. The symbolic approach is excellent for digital computers but possibly quite inappropriate for students who must master not only mathematical concepts, but also their notations which are, by design, independent of the concepts.

Much of elementary mathematics dwells on specific algorithms for a singular specific system of representation (the decimal place-value system). Kaput [41] is directly critical of the emphasis of form over content in elementary mathematics, agreeing that the predominance of math education addresses only a particular set of representations and algorithms. Place-value notation is universal. Place-value algorithms mix arithmetic *operations* with the maintenance of the place-value *notation*, and thus confound semantics (mathematical concepts) with syntax (the representation of mathematical conceptxs). Symbolic addition is purely syntactic. Addition is defined by the three mechanisms that maintain the place-value representation: digit combination facts, alignment of place-values, and carrying. *Adding It Up: Helping Children Learn Mathematics* points to an inherent tension between the abstract and concrete aspects of mathematics. "This tension is a fundamental and unavoidable challenge for school mathematics." [3, p74]. The pedagogical shift from early childhood manipulatives to symbolic computation is a shift from semantics to syntax, from construction of meaning to management of a particular representation.

What is missing to date are rigorous diagrammatic and spatial systems for numerics, systems that are inherently interactive and manipulable while at the same time meeting all criteria of symbolic formality. The additive systems presented in this proposal meet these criteria. They are based on visual/interactive axioms, and provide what Philip Davis calls visual theorems:

> "...a visual theorem is the graphical or visual output from a computer program -- usually one of a family of such outputs -- which the eye organizes into a coherent, identifiable whole and which is able to inspire mathematical questions of a traditional nature or which contributes in some way to our understanding or enrichment of some mathematical or real world situation." [42, p 30]

We share Kempe's goal: "...to separate the necessary matter of exact or mathematical thought from the accidental clothing -- geometrical, algebraical, logical, etc." [43]. However, we do not believe that the pedagogical qualities that make mathematical formalism comprehensible and learnable are separate from mathematical thought. We hope to clarify the strengths and weaknesses of spatial and symbolic approaches to teaching through comparative axiomatics.

3. THREE ADDITIVE SYSTEMS

Additive models have recently been developed for several foundational systems, including predicate logic and the algebra and arithmetic of numbers [44]. *Boundary logic* is the most studied, beginning with Peirce [45][46][47][48], and followed by Spencer-Brown [49], Kauffman [50][51][52], Varela [53], Bricken [54][55], and Shoup [56]. Kauffman has made original contributions to additive models of numerics [57][58], as have Bricken [59], James [60] [61], and Winn [62].

This section presents three numeric systems based on spatial representations and axioms that maintain the Additive Principle. The first system is *unit-arithmetic*, a simple case of base-1, unary arithmetic (tally systems) that clearly contrasts differences between string-based and spatial conceptualizations. The second is a *depth-value notation* for unit-arithmetic that provides the benefits of place-value notation without the cost of a total ordering of digits. The third is a *spatial algebra* that models addition as sharing a common space, and multiplication as touching in space. Illustrations of the static and dynamic representation of these systems are located in Appendix A.

3.1 UNIT-ARITHMETIC

Unit-arithmetic is the arithmetic of strokes and tallies that has been in use for over 30,000 years. A unit is a mark, stroke, notch, pebble, shell, or other discrete singular distinction. Unit marks may be replicated, providing a supply of indistinguishable replicas. Replicate units are intended to be indistinguishable in order to reduce the idea of counting to a foundation of one-to-one correspondence between marks and objects. An integer is an *ensemble* of identical marks sharing a space. Units occupy space in two distinct ways, by sharing a common space (and thus forming an ensemble) and by being partitioned into different spaces (that is, by not sharing a common space). As illustrated in Figure 1, addition of unit-ensembles consists of placing them into the same space. Multiplication consists of replacing every unit of one ensemble with a replicate of another ensemble. Multiplication is thus substitution of ensembles for units. These representations and conceptualizations of elementary arithmetic maintain connection with a student's intuition; the abstract mathematical operations of sharing space and of substitution remain visual, visceral, and physically grounded.

One familiar example of unit arithmetic is the American flag. The fifty stars represent the fifty states, but no particular star maps to any particular state. Stars are added together to construct a sum by the shared space of the blue background. The geometric arrangement of the stars has no particular meaning. The stars form a unit ensemble. Similarly, the thirteen stripes represent the thirteen original colonies, but no stripe is associated with any particular colony, and the colors of the stripes have no interpretation.

Unit-arithmetic is the archetypical additive system that illustrates axiomatic differences between spatial and symbolic computation, and thus between embodied and rote mathematics education. Figure 2 presents the axioms of arithmetic expressed in the languages of both group theory and unit arithmetic. The four axioms of unit-arithmetic define addition as being placed in the same space (mereological fusion [63]), subtraction as annihilation of polar units, and multiplication as substitution. Distribution gives permission to apply substitution over entire spaces or to each member of a space individually. The textual notation of unit-arithmetic used in Figure 2 can be interpreted as follows:

```
         a + b + c:   a|b|c  =  a b c
     • = 1, ◊ = -1:   • ◊ = <void>
             a x b:   [substitute b for • in a]  =  [b • a] = [a • b]
             a ÷ b:   [substitute • for b in a]  =  [• b a] = [a b •]
   a(b+c) = ab + ac:   [a • b|c] = [a • b]|[a • c]
```

The notations for fusion and substitution stand in place of physical (or virtual) operations, not symbolic structures. The four axioms of unit arithmetic specify the Additive Principle, the Additive Inverse, Commutativity of Multiplication, and Distribution.

3.2 DEPTH-VALUE NOTATION

Unit-ensembles can be rewritten into an efficient depth-value notation by a standardization process that results in a spatial form with minimal structure. Herein, depth-value notation is illustrated using three varieties of spatial syntax:

- Parentheses used as 1D textual delimiters of contained spaces.
- Ovals used visually as 2D spatial containers.
- Stacked blocks used as 3D spatial manipulates, with stacking as containment.

Unit arithmetic was abandoned because of the huge inconvenience of having to count the result of every sum. Depth-value notation is proof-in-principle that additive systems can have efficient and concise notations. Its intended use is as a supplementary tool to help students who do not understand symbolic approaches. We will evaluate its efficacy as part of the proposed research.

The formalism constructed by expressing unit arithmetic in depth-value notation is worth studying for this reason. Place-value notation requires significant effort to master the algorithms for addition and multiplication. Many students have difficulty achieving this mastery. Manipulatives have been proposed as an intervention, however these manipulative techniques are analogical and lack a formal (i.e. mathematical) foundation. To improve mathematics teaching, we would like to know exactly what the difficulties of place-value notation are. Depth-value notation provides a formal alternative that can serve both as a control and as a route for identifying the barriers a student is encountering with symbolic techniques. The primary sources of error in elementary arithmetic and algebra are negative numbers, fractions, and the distributive rule. Additive systems provide similar unique yet formal alternatives for these three problem areas. Therefore, these systems provide potentially useful curriculum tools for specifically addressing and correcting many of the prevalent errors made by students of elementary mathematics. Further, by embedding comparison of symbolic and manipulative systems into the curriculum, students will be exposed to alternative models of mathematical concepts, and thus facilitated in overcoming difficulties and confusions associated with any one particular model.

Depth-value notation replaces the linear typographic sequence of place-value notation by nesting in space. To highlight depth-value mechanisms, we first illustrate it in base-2. The base-2 depth-value numerals for 0 to 16 are presented using textual delimiters in Figure 3, and using oval enclosures in Figure 4. A representation of zero is no longer necessary, since the absence of contents within an *empty* space provides a contextual, structural void in place of the symbolic zero. Figure 5 shows the steps for reading base-2 depth-value numerals as conventional decimal integers. The numerals for 8 and 12 are shown using oval notation, together with their decimal transcription. *Reading* a base-2 depth-value numeral begins with the innermost unit, and successively doubles the accumulating value as each boundary is crossed outward. Should two or more form share the same space, their values are added together.

3.2.1 Base-10 Depth-Value Notation

Figures 6 and 7 present base-10 depth-value numerals in textual notation. Figure 6 shows the numerals from 0 to 1000 without the use of decimal digits. Figure 7 replaces unit ensembles with digit symbols. Just as in place-value notation, digit symbols are abbreviations that stand in place of a collection of units.

Integers can be structurally partitioned in a variety of ways. Unit partitioning corresponds to unit-arithmetic; base-10 place-value partitioning is conventional world-wide; base-2 place-value partitioning is dominant in computer implementations. These partitioning strategies are at the foundation of algorithmic complexity for operations and for reading integers. Place-value represents magnitude in polynomial form, a form with maximal reference to the base. Depth-value represents magnitude in maximally factored form, making minimal reference to the base. Figure 8 illustrates this difference for the numeral 3258.

3.2.2 Structural Standardization

Modeling addition as sharing space and multiplication as substitution removes the algorithmic complexity of arithmetic operations. The cost is that there are several representations for each numerical quantity. For example, 2 can be represented as •• and as (•). This is analogous to the several operational ways to express a conventional number. 2 is also 1+1 and 2x1.

There are three structural standardization rules for depth-value numerals, first expressed in base-2 typographical notation for convenience:

UNIT JOIN (base 2):	• • = (•)	1+1 = 2x1
BOUNDARY JOIN:	(•)(•) = (• •)	2x1 + 2x1 = 2(1+1)
ANNIHILATION:	• ◊ =	1 + −1 = 0

Unit Join relates addition to multiplication, creating a base system. Boundary Join is distribution. In contrast to conventional number systems, subtraction is achieved using a negative unit, ◊, that annihilates the positive unit, •. The depth-value standardization rules, as well as the structure of counting, are presented in Figure 9 for base-2 numerals and in Figure 10 for base-10 numerals. The analogous Unit and Boundary Join rules to standardize base-10 forms are:

UNIT JOIN (base 10):	•••••••••• = (•)	1+1+1+1+1+1+1+1+1+1 = 10x1
BOUNDARY JOIN:	(a)(b) = (a b)	10a + 10b = 10(a+b)

Figure 10 shows the place-value numeral 3258 represented as (((3)2)5)8 in base-10 depth-value form augmented with digit symbols. Alternatively, nesting depth can be inverted so that 3258 would appear as 3(2(5(8))). Figure 10 also shows one notational technique for recording depth-value decimal numerals: a new boundary, [], is constructed to express negative powers. 3258.46 would be represented as 3(2(5(8[4[6]]))).

Unit arithmetic with depth-value notation explicitly separates abstract operations from notational standardization. Addition and multiplication operations maintain the semantic connection to unit-arithmetic. The result of each operation directly represents the sum or product. The three standardization rules act independently to convert a result to its minimal syntactic form.

3.2.3 Animated Operations in Oval Notation

The operations of addition and multiplication transform oval forms smoothly in two-dimensional space, in contrast to the step-wise, discrete transformations characteristic of symbolic notations. Still frame sequences of prototype animations of the operations of arithmetic for base-2 depth-value numerals are presented in two-dimensional oval notation in Figures 11 and 12. Figure 11 follows the addition: 5+7 = 12. The representation of the form of 5 and the form of 7 in the same space in Frame 1 is also a representation of the form of 12. The dynamic standardization process follows. Figure 12 follows the multiplication: 5x7=35. The result of the substitution of the form of 5 into the form of 7 in Frame 4 is a representation of 35 (28+6+1). The depth-sensitive mechanism of multiplication resembles function composition more than additive composition in space. This makes multiplication of depth-value numerals mathematically more abstract while at the same time broadening the constructive definition of multiplication beyond the recursive application of addition as taught conventionally.

3.2.4 Depth-Value Notation Using Stacks of Blocks

The blocks representation of depth-value notation presented in Figures 13 through 17 illustrates a physically manipulable, yet fully general form of arithmetic. In block notation, numerals are represented by three-dimensional stacks of blocks. The unit is a single block; depth-value is expressed by the height of a stack. The block form emphasizes visualization and tactile manipulation, while achieving the same level of abstraction as is incorporated in symbolic forms. Addition is stacking blocks side-to-side; multiplication is stacking blocks on top of one another. The standardization rules for depth-value notation become the physical actions of constructing stacks (doubling in base-2) and pushing stacks together (distribution).

Figure 13 shows counting and standardization rules in block notation; this figure contains the same structural information and rules as does Figure 9 for one-dimensional textual delimiters. Figure 14 shows 5+7=12 in block notation (compare to oval animation Figure 11); Figure 15 shows 5x7=35 (compare to Figure 12); and Figure 16 shows 7x5=35. Figure 17 shows multi-digit decimal multiplication 319x548=174812 in block notation, and illustrates the parallelism of joining operations during standardization.

3.2.5 Multiple Representations

Spatial forms support a diversity of notations, each derived from the others via spatial transformation [55]. These notations include the 1D textual delimiters, 2D containers, trees, and paths, and 3D blocks, maps, oriented maps, graphs, and rooms. Figure 18 presents some of these spatial notations, together with the transformation paths between them. These notations are purely syntactic varieties derived from geometric and topological transformation of depth-value forms.

3.3 SPATIAL ALGEBRA

The algebraic structure of addition and multiplication can be represented as properties of space (as well as properties of containment), to construct an additive system not based on unit-arithmetic. Addition is represented as numerals sharing the same space without touching, while multiplication is represented by numerals in physical contact. Figure 19 illustrates instances of objects and operations within this spatial algebra. Constants and variables are represented by labeled blocks. Characteristic of additive systems, zero has no representation, it is empty space. Like the previous depth-value systems, spatial algebra incorporates the Additive Principle, as shown in Figure 20. Since blocks can exist anywhere within a space, spatial algebra does not include either associativity or commutativity of addition. That is, commutativity and associativity would be metric distinctions in a non-metric space.

Figure 21 shows the spatial algebra multiplication operation. Unlike depth-value notation, multiplication is by touching rather than by substitution. By electing to represent objects as blocks, however, touching is afforded by flat faces, imposing the unintentional appearance of associative ordering. More blobby, clay-like objects rendered as virtual manipulatives can avoid this unintentional structure. The pedagogical strength of spatial algebra is in its representation of distribution, as shown in Figure 22. Forms are factored by joining identically labeled blocks, making two or more blocks into one. Conversely, factored forms are multiplied into polynomial terms simply by cutting blocks into spatially separate pieces. There are many open design questions about how spatial algebra might be presented. Fortunately, when spatial blocks are constructed as virtual manipulatives, exploring different representations is relatively easy.

Figure 23 shows the multiplication of two polynomials. Each term within one polynomial is in contact with each term within the other polynomial. The multiplication itself requires only cutting all blocks into multiple pieces. To factor trinomials, identically labeled blocks for each polynomial degree are joined into a single block, while constants are sliced into additive and multiplicative components. Spatial algebra blocks may also aid understanding of the structure of nested factored forms. In Figure 24, a somewhat daunting conventional algebraic expression with apparently complex nesting of forms can be converted into the equivalent polynomial form by a single parallel act of slicing stacks of spatial objects. The rules of arithmetic applied within each stack, together with spatial distribution complete the simplification of the original expression.

4. METHODS

The two major goals of this Project proposal are:

- ✦ To develop prototype but high-quality curriculum materials that permit fine-grain evaluation of performance on problems expressed in both symbolic and additive systems, and
- ✦ To implement modular presentation software that facilities multivariate research over mathematical representations in one-, two- and three-dimensions and over specific transformation axioms and rules,

We intend to address the basic arithmetic of numbers, including integers, decimals, and fractions in Year I; and elementary algebra in Year II. Naturally, should any particular system, or any particular notational variety, result in highly negative formative evaluation comments or performance measures, we would abandon it in favor of other alternatives. Similarly, it is possible that a particular system or notation may engender highly positive formative measures. In such cases we would concentrate efforts on developing more interactive curriculum units that utilize it.

4.1 PLAN OF WORK

All systems will be prototyped within the *Mathematica 6.0* computational and programming environment. The use of Mathematica for rapid prototyping avoids many difficulties that plague other software development projects, including extensive low-level debugging of code, effort spent coordinating and interfacing different programming languages and techniques, development of low-level interaction and display capabilities, and conformance to various communication and operating system protocols. The developed prototype will support stepwise student manipulation of both computational problems and spatial proofs, and will store performance statistics and raw data for any input problem. All data will be collected in the Mathematica environment, so that statistical analysis tools would be immediately available. The *Mathematica Player* is a free downloadable application that permits use of developed software on any platform, and that is maintained and updated to provide a high-quality presentation tool free of maintenance and portability overhead.

4.1.1 Research Approach

Since the work is exploratory, this Proposal does not commit to an in-depth examination of any one particular semantic or syntactic variant. Since we are intimately familiar with the initial overhead required to produce appropriate interactive software, the Proposal also does not include a summative research design. We believe that it is of primary importance to first provide research and curriculum software tools that will facilitate rapid and rigorous empirical evaluation of axiomatic domains. This Proposal is an initial, facilitating step.

Performance data is to be gathered in two phases. The taxonomic, or classification, phase incorporates protocol analysis to identify particular areas for later focus. The analytic, or experimental, phase incorporates structured tasks with closer control over exposure, ordering, problem complexity, and ambiguity. We intend to avoid weaknesses in other studies of error behavior by broadening the analysis beyond number of correct responses, data collected only from testing situations, and irregular problem difficulty.

The multivariate research to be facilitated incorporates a partial 2x3x5x3 factored design, with main effects of axiomatic approach (additive with symbolic serving as the control), syntactic variety (1D textual strings, 2D containers and 3D stacks of blocks), transformation rule (presence or absence of commutativity, associativity, identity, inverse, distribution) and pedagogical content level (primary school arithmetic, secondary school algebra, and remedial topics within tertiary coursework). The axiomatic approach and the transformation rules would be repeated measures, syntactic variety and content level would be randomized. Dependent variables will include error type (incorrect answer, process, or understanding) and subjective ratings of confidence and of comprehension. Appropriate experimental design may also call for control of order and time of exposure, stratification of sampling, enforced domain constraints [64], pre-training, classification and control of problem difficulty, and control groups that anchor specific hypotheses. Given sufficient sample sizes, covariate analysis of learning styles and demographic groups would be of interest, as would longitudinal analysis of retention and skill application. Follow-up work may include evaluation of presentation and interaction styles.

4.1.2 Software Development

The iterative software design strategy, quality control, debugging, user interaction and codesign, documentation, and other software engineering aspects of the Project will follow conventional rapid prototyping software engineering practices. The proposed generic animation engine accepts as input conventional mathematical forms such as multi-digit operations on base-10 numerals and linear algebraic equations. Students will be able to interact with and steer each fine-grain computational step using the dynamic display as a virtual manipulative. Figure 24 provides an architectural schematic of the animation tool.

The first software iteration should take about three months. This would leave six to nine months to develop and refine the display and interaction of the tool. Development will include three cycles of rapid prototype implementation, each with maximal exposure to student feedback. The LWTC Math Department is currently installing a computer-based Math Lab which will support lab-oriented math classes at LWTC, and incidentally support formative evaluation of the software. For Year I, we expect to constrain use of the developmental software to students within the multimedia and animation programs, and to volunteer students from math classes. By the end of Year I, we should be able to present the software prototype at relevant conferences, and release a beta version to selected colleagues.

Refinement and debugging can be concluded during first few months of Year II, leaving six months to develop structured curricula materials. These materials will be tested in pilot studies in both high school and elementary school classes. The goal will be improve test performance on standardized tests, with a particular focus on very low achievers. By the end of Year II, we should be ready to distribute fully documented prototype systems. We will disseminate documentation and other information on the availability and utility of the tools during the last two months of the funded Project. Details of personnel and FTEs are in the cost-justification section.

4.1.3 Curriculum Design

Curriculum design includes identification of intended learning outcomes; structuring of units and modules within a course, including sizing, clustering, timing and sequencing of content components; identification of skills being taught; development of teaching strategies; and planning and implementation of evaluation techniques and instruments. A rough outline of the intended coverage of curriculum units follows.

- Units for addition, subtraction, multiplication, and division of
  - whole numbers
  - signed numbers
  - fractions
  - exponents
  - some aspects of simple algebra
- Both decimal and binary versions
- Teaching units to show:
  - structural relationship between forms of representation
    - identity, zeros, commutativity, associativity, distribution
  - how to read and write in each system
  - operations using three varieties of additive systems
  - the relationship between depth-value and place-value notation
  - at least three different depth-value notations
    - textual, diagrammatic, manipulative
- Interactive display to include
  - type-in conventional problems
  - watch spatial computation unfold
  - drag-and-drop manipulation of spatial forms
- Problem presentation tool
  - pretest and posttest for all unit topics
  - a substantive body of practice problems
    - parallel to conventional text
  - show mapping between conventional and spatial solutions

4.2 FORMATIVE RESEARCH EVALUATION

Software iteration and refinement will include continuous significant classroom usage by math challenged students. The interaction design process will include subjective feedback, codesign with users, protocol analysis of error behavior, comparative evaluation of performance, and formative experimental evaluation of potential improvement in mathematics understanding and performance. Evaluation will focus on fine-grain protocol analysis of solution steps, rather than on more course-grain measures such as number of items correct. We hope to develop a mapping between specific axioms, specific notations, and error behavior. We intend to develop limited curriculum units guided by preliminary experimental evidence of appropriate teaching strategies that combine and sequence spatial/manipulative and textual/symbolic materials. Several conventional research techniques will be incorporated to refine the usage of the proposed tools, particularly during the second year of the Project. These include:

1) *Protocol analysis of the types and frequency of student errors.* This information is critical for formative evaluation of the benefits of additive techniques, and for identifying and isolating errors generated by semantic (understanding), syntactic (reading and manipulating representations), interactivity (tactical and strategic computational choices), and display (software ease of use) sources. Error types and frequency will be compared both between and within the various axiom systems. Protocol analysis contributes to the identification of the structural source of errors and to the assignment of blame to structural elements, while managing the huge diversity across error behaviors. It is more sensitive to the context of an error than test measures.

2) *Performance using spatial and symbolic axioms and transformation strategies.* We will examine error types and rates for structured problem sets containing rules known to cause difficulties for students, with particular focus on the distributive rule and on structural manipulations involving negative numbers and fractions. As well as error rates, we will examine student steps and jottings (i.e. work done while solving problems), and other informal steps using detailed observation protocols. We hope to identify the uniqueness, persistence, and consistency of errors associated with particular mathematical structures and transformations. Errors, for example, occur much more frequently during constructive transformations (such as applying an operation to both sides of an equation), than during reductive transformations (such as adding two numbers together)[1]. Also of importance is error specificity, whether or not a particular error or miscomprehension is common, or associated with one particular problem, or with a particular person, or is a singular occurrence.

3) *Direct evaluation of performance improvement on conventional mathematics tests.* Equivalent pre- and post-tests on conventional mathematics problems will used to measure performance improvement both with and without exposure to software models and representations. We will use the Basic Mathematics performance tests administered to all LWTC students, for which we have several years of results. This approach will provide comparable performance measures for long-term controls, short-term control and experimental groups, and

pre- and post- performance for experimental groups. Once curriculum units have been developed and sufficiently refined, controlled experimentation using classes that have not been exposed to the software development process can progress rapidly.

4.3 DISSEMINATION PLAN

Project website: We will design and construct an interactive web-site as a repository for Project materials. This web-site will also serve as a virtual development environment during the Project, and for coordination, communication, and collaboration of Project contributors. Such a capability facilitates team interaction across geographically distributed members. It also provides a virtual organizational structure for project management, and for coordination of meetings, discussion, analysis, and data collection.

We would like to develop the capability for remote students and teachers to use the Project curriculum materials through a web browser. This facility might include automated gathering of performance data and a capacity for direct user feedback. Monitored interaction can be at almost any level of detail, from specific mouse-clicks to passing or failing an exercise. A lower-effort alternative is to provide downloadable software, together with a Player, so that Project software would be physically distributed in a compiled version. The site can also serve for electronic publishing, and for distributing Project results and reports directly to those with interest. Interactive refinement of tools and research techniques with "virtual Project members" would then lead to high-quality submission to professional journals. We envision two papers covering Project research results, and several others in support of documentation of distributed software.

We intend to present papers at national conferences during the second year of the Project.

5. PERSONNEL AND RESOURCES

The planned Project Team consists of four part-time task coordinators (overall management, and development of curriculum, programming, and interactive animations), six consultants, 5-6 half-time graduate and undergraduate students, and one capstone student team consisting of five students. The total FTE effort is approximately 11.7 over two years. The FTE breakdown for all Project personnel is included in the budget justification.

Principle Investigator: Dr. William Bricken has a Diploma of Education in Mathematics, an M.S. in Statistics, and a Ph.D. in Mathematical Methods of Educational Research from the Stanford University School of Education. He has served as a high school mathematics teacher, as the principal of a community-based elementary school, as an Assistant Professor of Social Psychology and Education, as a Research Associate Professor of both Education and Industrial Engineering, as an Assistant Professor of Software Engineering and Computer Science, and currently as a member of the Mathematics Faculty of Lake Washington Technical College. He has taught a wide diversity of graduate-level courses, including Statistical Analysis in

Educational Research, Interactive Educational Technology, Intelligent Tutoring Systems, Applied Formal Methods, Artificial Intelligence, Computer Graphics, Human-Computer Interaction, and Computer Ethics. He has also conducted short courses in and given keynote lectures to dozens of national and international professional organizations.

Dr. Bricken has over twenty-five years of experience working with the concepts and structures of additive mathematics, with an emphasis on innovative computational techniques in logic. In his dissertation, Dr. Bricken empirically validated the unique nature of errors made by students learning algebra. Using a range of experimental techniques (multivariate experiment, exploratory factor analysis, in-depth protocol analysis, clinical case study, historical review, ontological deconstruction, and direct remediation), he demonstrated that symbolic errors made by novices are neither random nor predictable, rather they are context sensitive, situated, and unique. In industry, Dr. Bricken has extensive experience leading high-profile software projects. He has designed and implemented innovative software systems addressing human-computer interface, complex adaptive agent architecture, parallel processing algorithms, operating systems for virtual environments, and high performance algorithms for optimization of silicon circuitry area and timing. Dr. Bricken was one of the first developers for Mathematica, writing programs for the 1988 beta pre-release version of Wolfram's software.

Dr Bricken is uniquely qualified to direct the proposed Project, as a leader in the field of spatial formal systems, and as an experienced educator, graphics programmer, systems designer, interface designer, and software project administrator.

5.1 CONSULTANTS

The Project incorporates several strategic consultants to advise and evaluate Project designs, implementations, curricula, and methodologies. Since the majority of the Project effort addresses software and curriculum implementation details, the consultants will provide primary guidance for the high-level design of all aspects of the Project, including project planning and the design of software, curriculum materials, experiments, usability, and data analysis. Each member of the consulting team will contribute 80-100 hours per year to the Project. The team includes researchers with a deep knowledge and experience in mathematics teaching and curriculum design (Moyer-Packenham, Hamilton, Kauffman), in learning theory (Moyer-Packenham, Shapiro), in spatial mathematics (Kauffman, Shoup, Shapiro), in the skills and techniques of computer graphics necessary to design compelling interactive curriculum materials (Shoup, Frankel, Shapiro), and in the use of visual and manipulative tools in the classroom (Moyer-Packenham, Frankel, Hamilton, Kauffman). All team members have experience in the design and dissemination of mathematics and mathematics education research.

Dr. Louis Kauffman has a Ph.D. in Mathematics from Princeton University, and is a Full Professor of Mathematics at the University of Illinois at Chicago, specializing in knot theory, quantum algebras, and innovative mathematical systems. He has written several books and articles on the spatial mathematics presented herein, and is the originator of depth-value notation.

Dr. Kauffman will be responsible for the mathematical integrity of the Project and will contribute to the design of mathematical teaching styles and curriculum.

Dr. Patricia Moyer-Packenham has a Ph.D. in Curriculum and Instruction from the University of North Carolina at Chapel Hill, specializing in mathematics manipulatives. She is currently an Associate Professor at George Mason University, coordinating the Mathematics Education Leadership Program and directing the Mathematics Education Center. She specializes in mathematics representations and teacher quality in math education. Dr. Moyer-Packenham is an experienced high-school and grade school mathematics teacher, and is a project investigator in the five-year NSF project *Math and Science Partnership Program Evaluation*. She has also contributed to the development of the observation protocols for a national assessment of mathematics teaching. Her latest book is *What Principals Need to Know about Teaching Mathematics*. Dr. Moyer-Packenham will contribute to both curriculum and experimental design. Two graduate students working with Dr. Moyer-Packenham will assist with Project organization, with development of curriculum materials, and with experimental design, data collection and analysis.

Dr. Richard Shoup has a Ph.D. in Computer Science from Carnegie-Mellon University. He specializes in reconfigurable hardware architectures, visual languages, and innovative mathematical systems. Mr. Shoup has been awarded a Technical Oscar, a Technical Emmy, and a SIGGRAPH Computer Graphics Lifetime Achievement Award for his pioneering work in creative computer videographics systems. He is the President of the *Boundary Institute* for studying the foundations of mathematics, physics and computing, and currently divides his time between private industry and Cargnie-Mellon West. Dr. Shoup will contribute to the design of software, user-interface, and graphic display. He will also contribute substantively to the visual and computational design of spatial representations of arithmetic and algebra.

Dr. Daniel Shapiro has a Ph.D. in Management Science and Engineering from Stanford University. He specializes in artificial intelligence programming to model value-driven learning, and has contributed to diagnostic reasoning, hierarchical skill development and reinforcement learning, transfer of knowledge, cognitive architectures, and machine learning programming. Dr. Shapiro is the Assistant Director of the *Institute for the Study of Learning and Expertise* at Stanford. He will contribute to software design and interactivity, to user interaction capture and modeling, and to the structure of curriculum and user experiences. He will also contribute substantively to the design of learning experiences.

Felice Frankel is a Senior Research Fellow in the faculty of Arts and Sciences at Harvard University, where she leads the Envisioning Science Program, with a concurrent appointment at the Massachusetts Institute of Technology. Her graphics images have been published in over 300 scientific journal articles, including several journal covers. She is a Fellow of the American Association for the Advancement of Science, and has been supported by several NSF grants. Her book *On the Surface of Things: Images of the Extraordinary in Science* is considered a classic in its field. Her current NSF Project, *Picturing to Learn*, studies how representations made by students enhance teaching and learning. Ms. Frankel will contribute to the aesthetic design of the user experience, to the appearance and manipulability of mathematical form, and to student-centered curriculum design.

Mr. Earl Hamilton has an M.A. in Mathematics from the University of Washington, and has been an math instructor at North Seattle Community College since 1970, during which time he has taught every math course offered by the College. He also teaches applied engineering mathematics as an Engineering instructor. Mr. Hamilton was supported in team teaching Physics and Engineering Calculus by the NSF. Mr. Hamilton will apply his deep experience in mathematics education to the structure and design of curriculum elements, and to the classroom evaluation of developed software.

Mr. John Gabriel has been teaching Animation and User-Interaction at LWTC for several years. He has extensive experience is the software game industry and as an illustrator of children's books. Mr. Gabriel is not a consultant, but is on the Project development team at .3 FTE. He will supervise graphics design and animation production. Mr. Gabriel's students will have responsibility for implementing engine software and the user interface.

5.2 RESOURCES

The LWTC student body is extremely diverse, providing a unique environment within which to conduct studies of mathematics education. LWTC is decidedly a teaching institution, its faculty are dedicated full-time instructors grounded solidly in practical, job-related skills. A primary advantage is that LWTC students are preparing directly for entering the job market and for job-related promotions, rather than seeking a general academic education. Many are enrolled in ESL courses. LWTC also administers the Lake Washington Technical Academy, a junior/senior year high school with an enrollment of 450 students. The Academy is housed on the LWTC main campus, and provides high school students with an opportunity to earn concurrently a high school diploma, a vocational certificate, and a two-year college degree. Although Lake Washington students are adults, many have underdeveloped math skills, so that the majority of our math courses are designed to cover middle school subjects such as basic arithmetic skills and elementary algebra. This provides a secure environment for exploratory mathematics curriculum development, with many students who have a long-term aversion to symbolic mathematics, but who can communicate clearly about their learning needs.

The LWTC Mathematics Department services concrete and abstract math needs for all technical departments of the college, including information technology, manufacturing and transportation technologies, medical and dental technicians, accounting and business services, and biological sciences. The Math Department also provides math education for Academy high school students, adult education students, those completing a certificate of high school graduation, and other community members seeking to improve their understanding of basic and applied mathematics. The Math Department participates in a College in the High School Program, which conducts math and other advanced classes for college credit in local high schools.

5.3 ADDITIONAL SUPPORT

Wolfram Research has agreed to donate long-term licenses for Mathematica 6.0 in support of this Project. Project software and tools will be disseminated through the Wolfram Research MathWorld and Mathematica 6.0 Demonstration Project web-sites, as well as through the Project web-site, conference presentations, and conventional channels of academic publication.

The faculty at LWTC is in contact with numerous elementary and secondary school teachers who would be receptive to the introduction of innovative approaches to mathematics education. The idea of visual and spatial mathematics has received enthusiastic responses. Thus we intend to demonstrate the Project software and curriculum materials to local parent groups and high schools toward the end of the second Project year.

PROJECT NARRATIVE
APPENDIX A: SUPPORTING FIGURES

CURRICULUM SOFTWARE FOR COMPARING
SYMBOLIC AND MANIPULATIVE FORMAL SYSTEMS
A Proposal to the Department of Education Institute of Education Sciences
CFDA Number: 84.305 Topic 3 Goal 2
PI: William Bricken, Ph.D.
Lake Washington Technical College

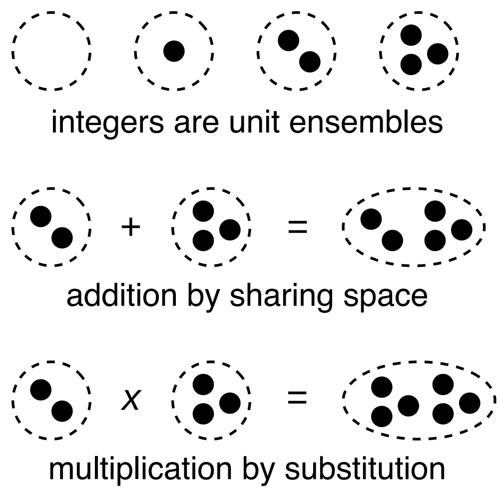

Figure 1: Unit-Ensemble Operations

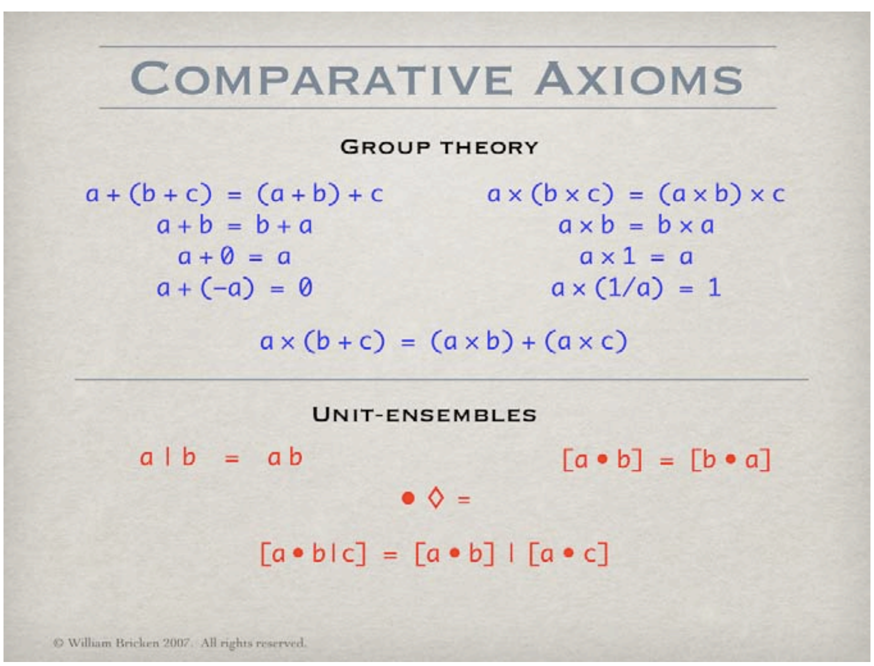

Figure 2: Group Theoretic and Unit-Ensemble Axioms

0	<void>		
1	•	9	(((•)))•
2	(•)	10	(((•))•)
3	(•)•	11	(((•))•)•
4	((•))	12	(((•)•))
5	((•))•	13	(((•)•))•
6	((•)•)	14	(((•)•)•)
7	((•)•)•	15	(((•)•)•)•
8	(((•)))	16	((((•))))

Figure 3: Base-2 Depth-value Numerals for 0 to 16, Textual Notation

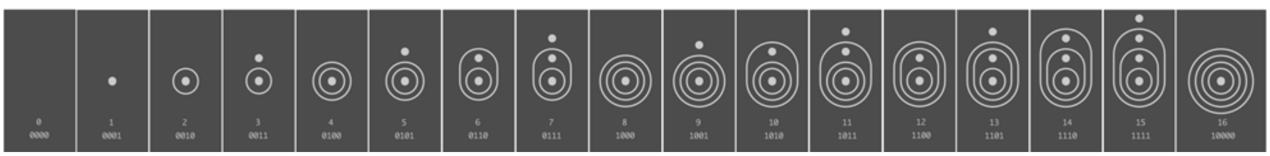

Figure 4: Base-2 Depth-value Numerals for 0 to 16, Oval Notation

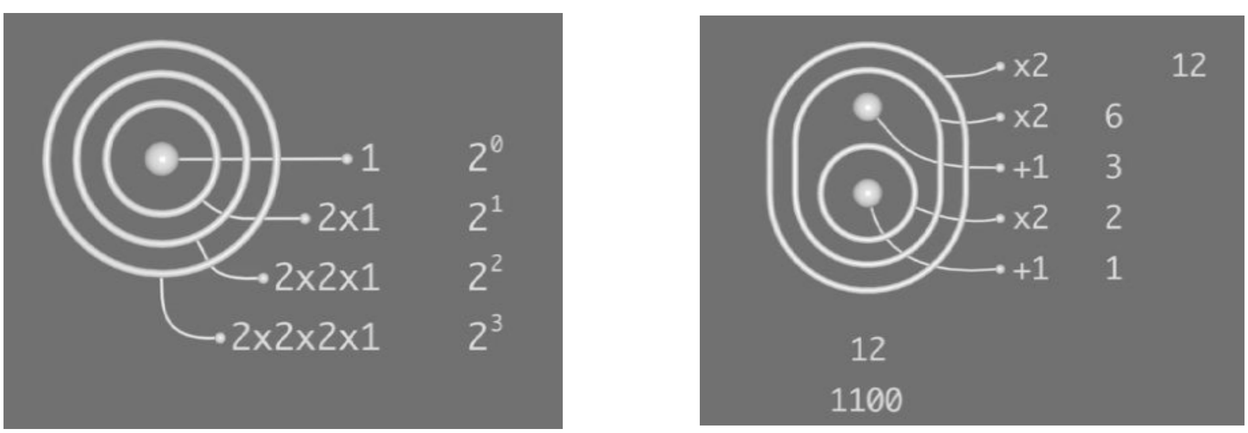

Figure 5: Reading Base-2 Depth-value Numerals, Oval Notation

0	<void>				
1	•	10	(•)	100	((•))
2	••	20	(••)	200	((••))
3	•••	30	(•••)	300	((•••))
4	••••	40	(••••)	400	((••••))
...					
9	•••••••••	90	(•••••••••)	900	((•••••••••))
				1000	(((•)))

Figure 6: Base-10 Depth-value Numerals for 0 to 1000, Textual Notation

0	<void>				
1	1	10	(1)	100	((1))
2	2	20	(2)	200	((2))
3	3	30	(3)	300	((3))
4	4	40	(4)	400	((4))
...					
9	9	90	(9)	900	((9))
				1000	(((1)))

Figure 7: Base-10 Depth-value Numerals for 0 to 1000, with Digit Abbreviations

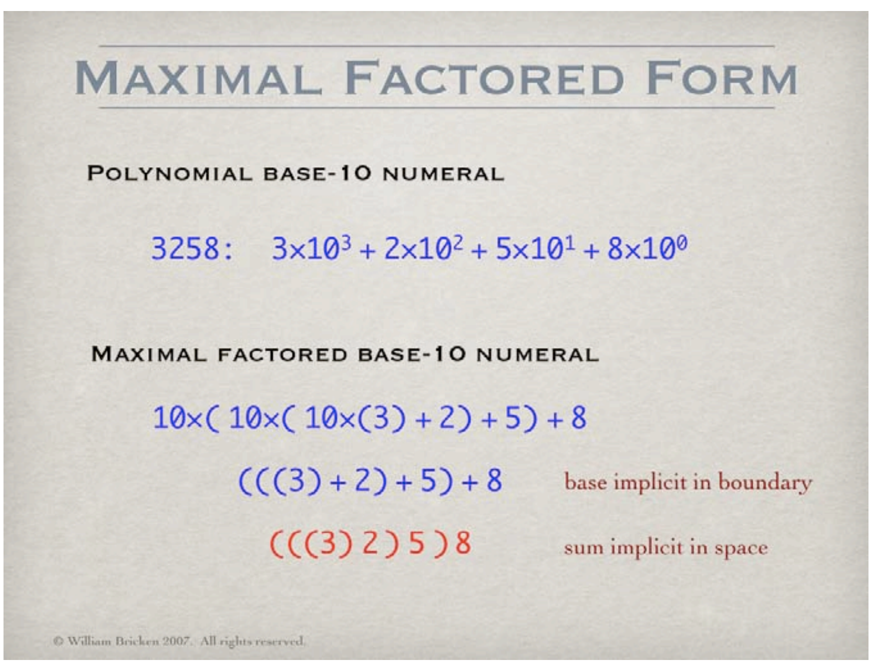

Figure 8: Polynomial and Maximally Factored Forms of the Numeral 3258

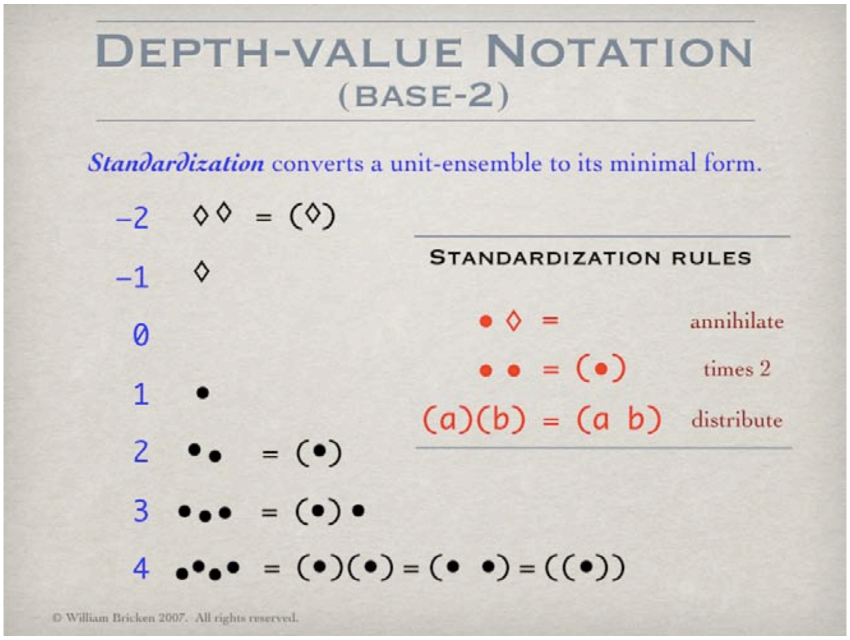

Figure 9: Base-2 Counting and Standardization, Textual Notation

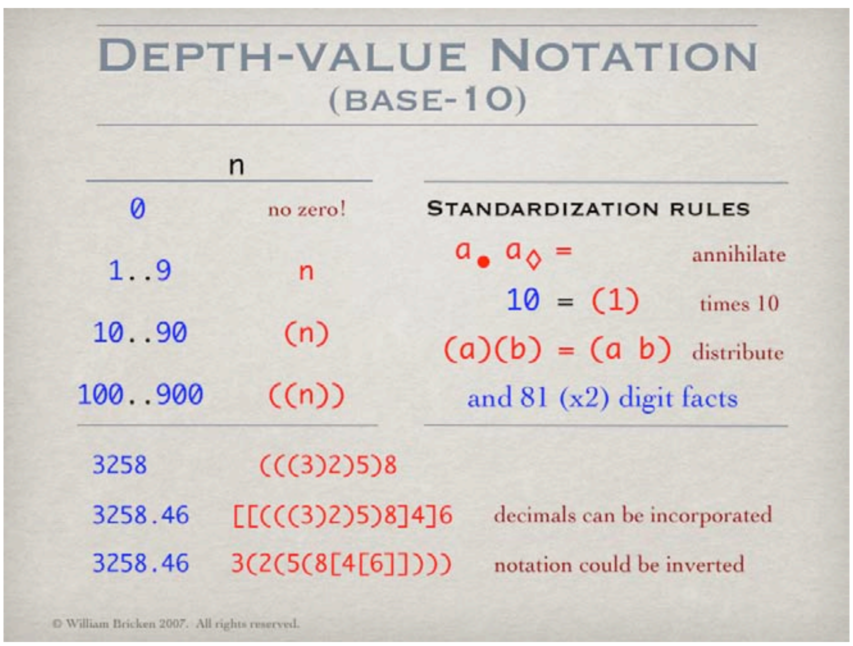

Figure 10: Base-10 Counting and Standardization, Textual Notation

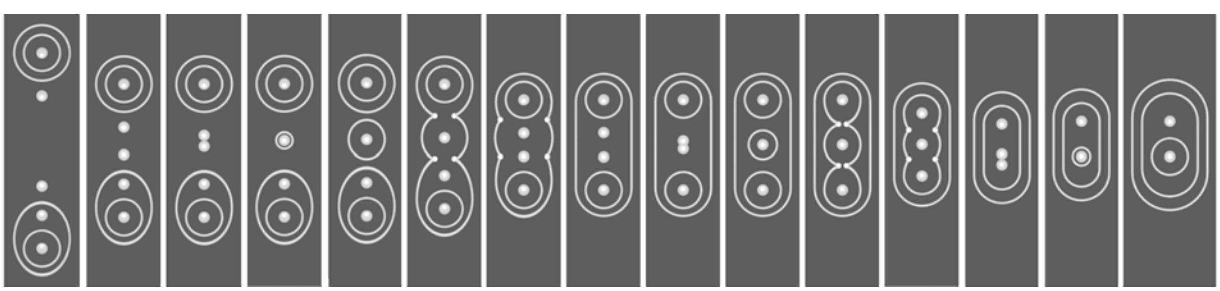

Figure 11: Animation Stills for 5 + 7 = 12, Oval Notation

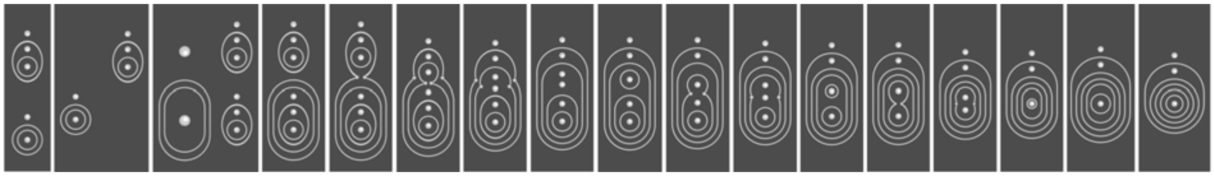

Figure 12: Animation Stills for 5 x 7 = 35, Oval Notation

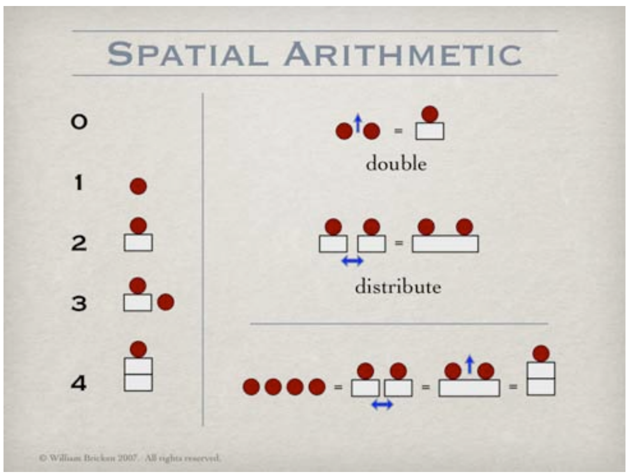

Figure 13: Standardization

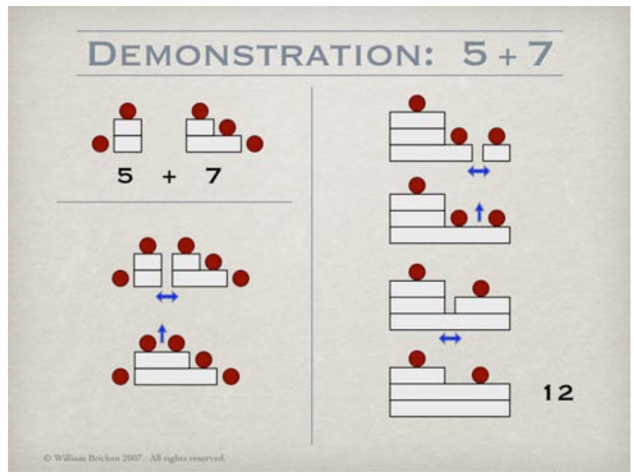

Figure 14: 5 + 7 = 12, Blocks

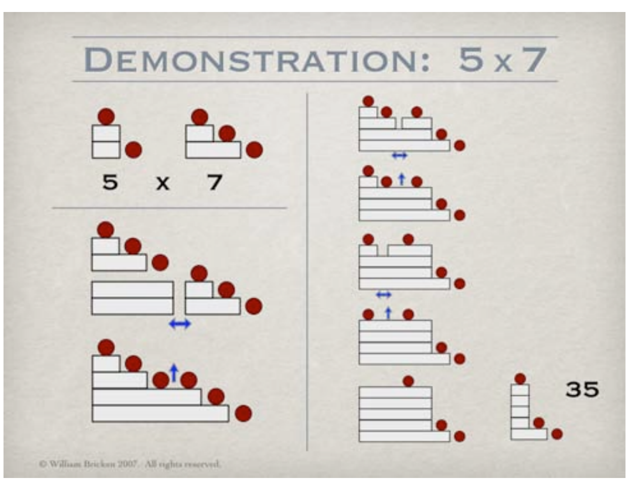

Figure 15: 5 x 7 = 35, Blocks

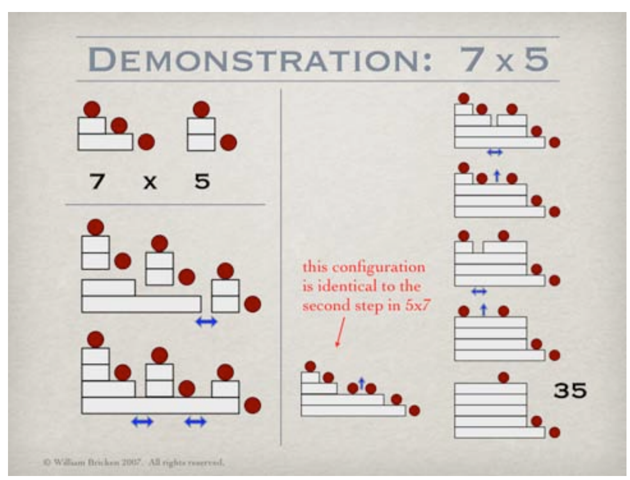

Figure 16: 7 x 5 = 35, Blocks

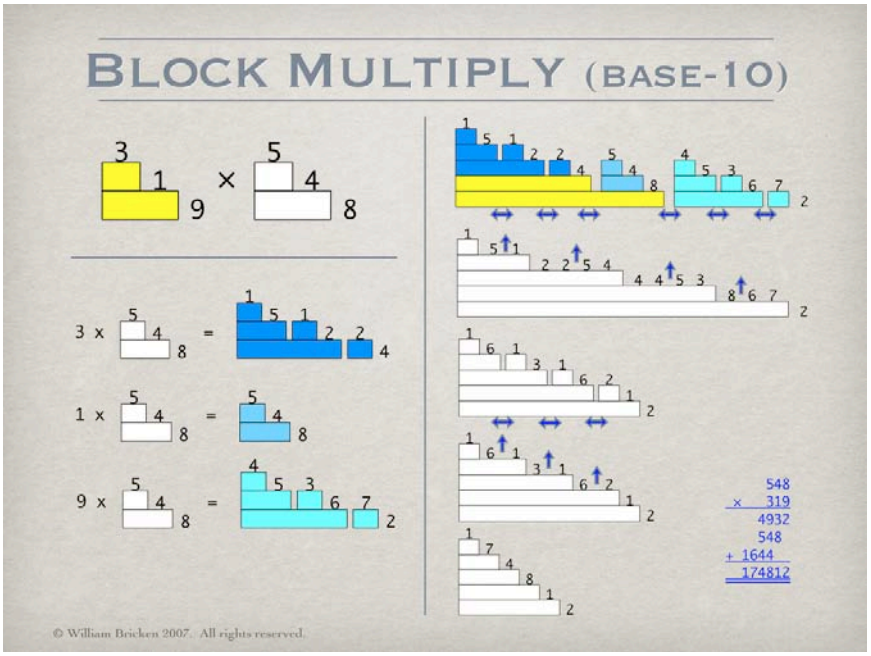

Figure 17: 548 x 319 = 174812, Base-10 Block Notation

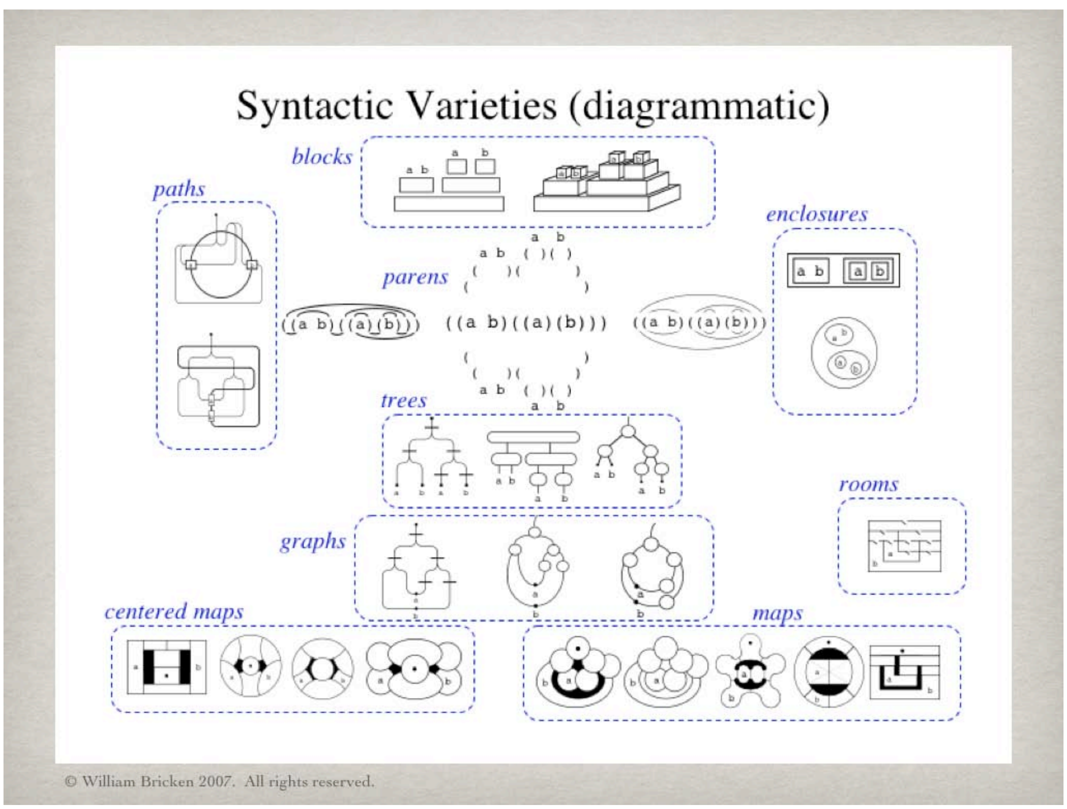

Figure 18: Syntactic Varieties of Depth-value Notation
(Textual Delimiters, Blocks, Containers, Trees, Graphs, Rooms, Maps, Paths)

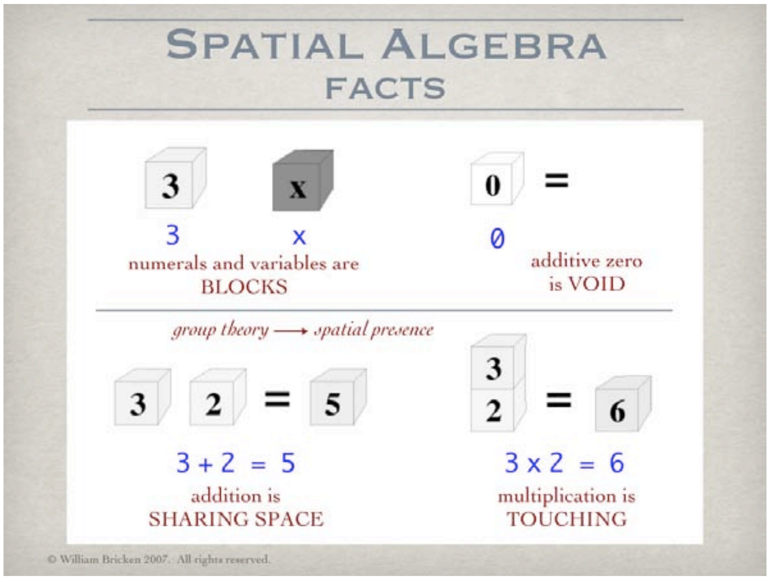

Figure 19: Spatial Algebra Objects, Zero, Addition, Multiplication

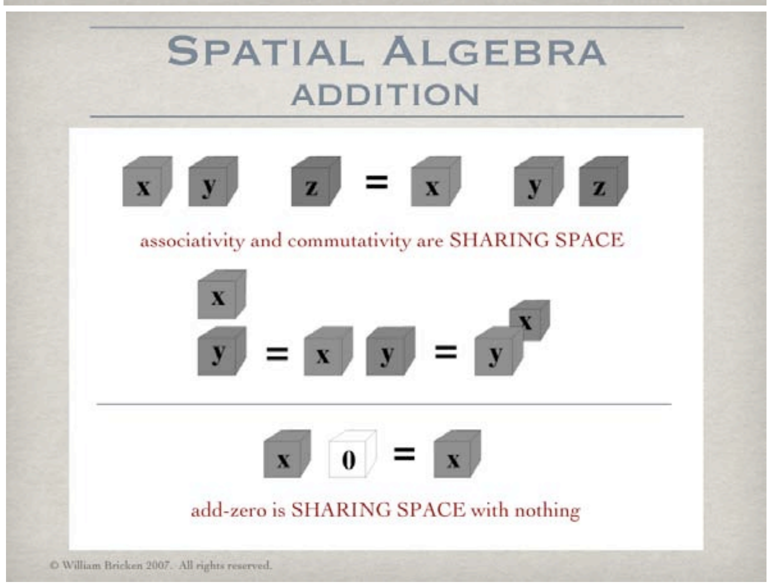

Figure 20: Spatial Algebra Sharing Space as Addition

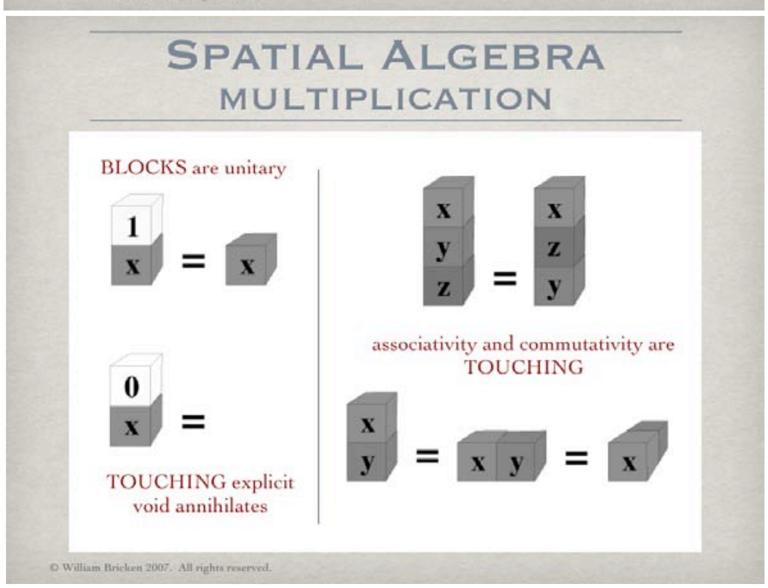

Figure 21: Spatial Algebra Touching as Multiplication

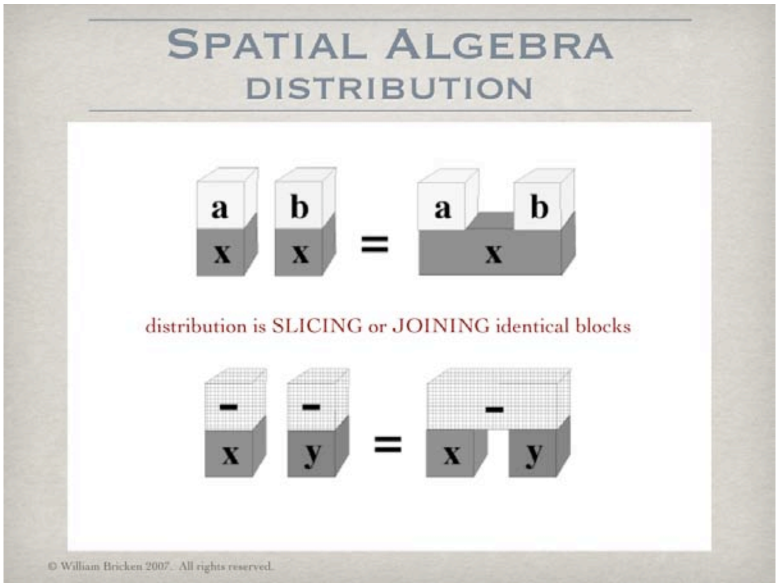

**Figure 22:
Spatial Algebra
Distribution via Joining
and Cutting,**

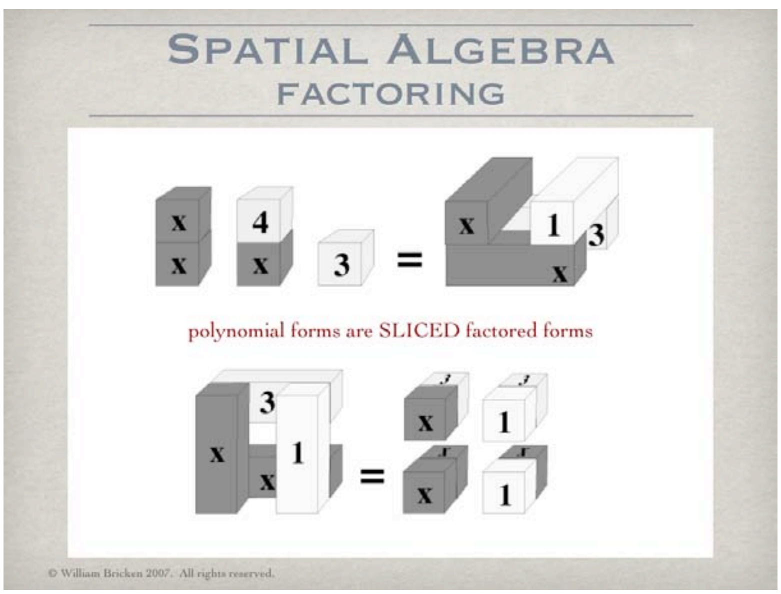

**Figure 23:
Spatial Algebra
Factoring via Joining
and Cutting,**

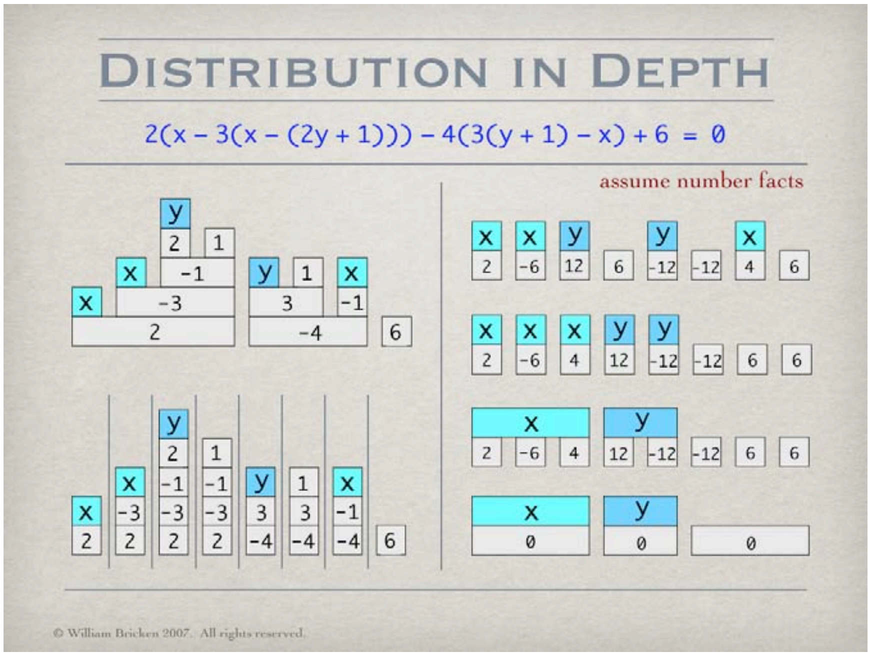

Figure 23: Spatial Algebra Distribution of Nested Forms via Cutting

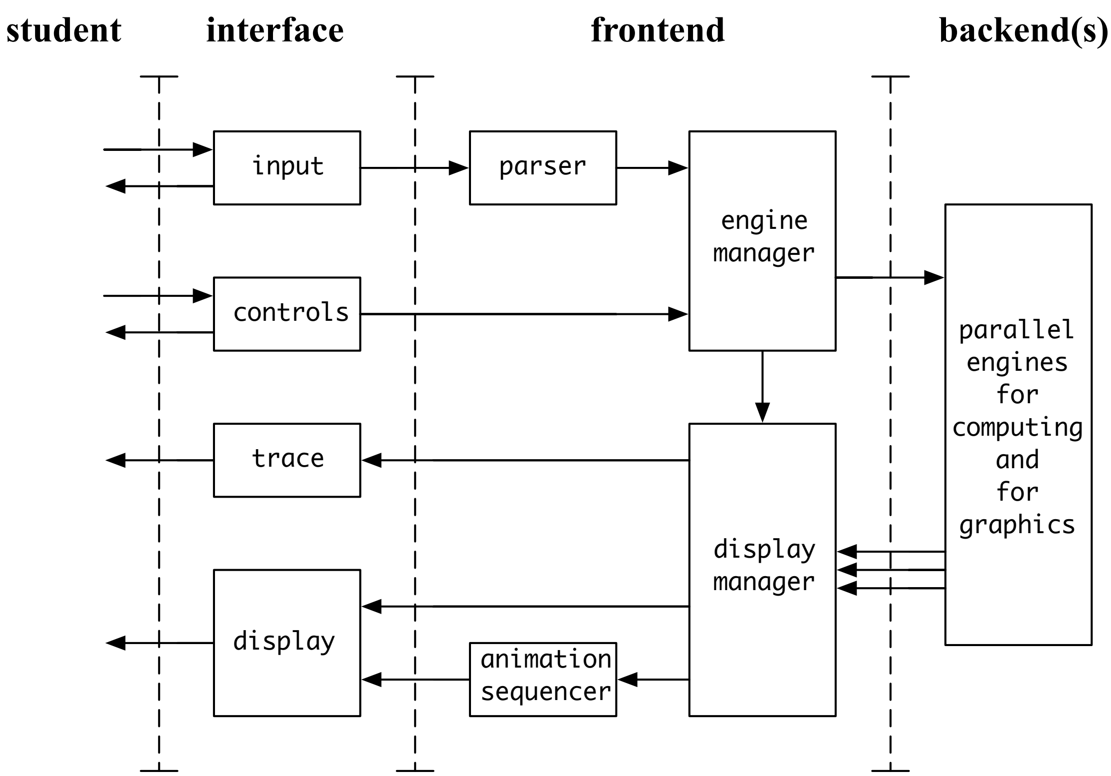

Figure 24: Block Architecture for the Boundary Animation Tool