Geometric Algebra with Conzilla

Building a Conceptual Web of Mathematics

Master Thesis Report in Mathematics

Mikael Nilsson
Center for User-Oriented IT-design,
Royal Institute of Technology, Stockholm

22nd January 2002

Abstract

In this paper, the technique of *conceptual modeling* using the Unified Modeling Language (UML), is applied to the mathematical field known as Geometric Algebra for probably the first time. Geometric Algebra is a unified language for analyzing the full range of geometric concepts in mathematics and physics, first developed by H. Grassmann and W. K. Clifford, and later revitalized by D. Hestenes. A thorough introduction to the field of Geometric Algebra is given, with accompanying conceptual models. Examples of the technique of *multiple narration*, where the same model is used to tell different stories, are analyzed. The specific problems and advantages of using UML as a visual language for a conceptually rich mathematical communication are discussed. Using the conceptual browser CONZILLA, the conceptual models have been made available for online browsing within a virtual mathematics exploratorium being developed by the research group led by Ambjörn Naeve, mathematician and senior researcher at CID, Center for user-oriented IT-design at the Royal Institute of Technology in Stockholm. The specific advantages of using interactive models are discussed.

keywords: geometric algebra, UML, mathematics, concept maps, visual modeling, Conzilla

Contents

Chapter 1

Introduction

1.1 The Essence of Mathematical Content

Mathematics has been described as "the study of structures that the human brain is able to perceive"[1]. It is not unreasonable to add *"and communicate"*. Mathematics characterizes itself not only by the large body of constructs and theorems it contains, but also by its rich language for communicating those ideas. What can not be expressed in this language is not part of mathematics.

This language contains, as we know, a large set of symbols traditionally used to encode mathematical content, and it contains a number of diagrammatic techniques. But it also contains important conceptual structures that are introduced in order to better communicate abstract and symbolic content in a conceptually clear way. For example, to the human reader, mathematical proofs are not merely a sequence of symbols, but a rather complicated mixture of lemmas, constructions, analogies and so on.

Most mathematical texts rely on informal analogies, simplifications and intuition to convince the reader, even in proofs and supposedly formal contexts. The majority of mathematical texts cannot be said to be fully stringent, something which should not be taken as an indication of these texts being incorrect or incomplete. In almost all cases, the analogies are clearly correct and the intuition can easily be translated to formal proofs. However, from the time of Russell and Whitehead's *Principia Mathematica* it has been clear that the human brain is not capable of processing the often enormous amounts of symbols needed to make a formally complete mathematical proof. Simplifications are a necessary part of the language of mathematics.

Communicating mathematics thus involves communicating conceptual structures in addition to symbols. Traditionally, those conceptual structures are communicated implicitly, using ordinary text.

This thesis discusses a new language for communicating conceptual structures in mathematics: *conceptual modeling*, using the Unified Modeling Language, UML. This diagrammatic language is used to make explicit the conceptual structures and relations contained in the material, giving a clear overview of the conceptual context and content.

[1]See [27, p. 36]

1.2 How to Read This Thesis

In section 2, UML and conceptual modeling is introduced and compared with other approaches for visual modeling, such as mind-maps and concept maps.

In section 3, the field Geometric algebra is introduced. Geometric algebra is a universal mathematical language for expressing geometry analytically. It unifies seemingly different branches of mathematics such as vector calculus, complex numbers, and tensor algebra in a powerful common framework.

In section 4, the concept browser Conzilla is presented. A short introduction is given to the underlying philosophy of Knowledge Manifolds and the Conceptual Web, and Conzilla is compared with other concept mapping tools.

In section 5, the conceptual web that is being built for Geometric algebra is presented. The application of UML to mathematics in general is discussed. Examples of multiple narration and other features of a concept browser is illustrated, and the advantages to traditional ways of communicating mathematics is discussed.

1.3 The ILE Work at CID

This thesis has been written in cooperation with the Interactive Learning Environments group at CID, whose general goals include the development of methods and tools for interactive forms of explorative individual learning.

This thesis is part of a larger effort within that group to improve the state of the current mathematics education at all levels. There is an emerging awareness that mathematics education is entering a profound crisis, where students at all levels are having greater and greater trouble grasping the nature of the subject. Critical thinking and self-confidence in mathematics among students is reaching frighteningly low levels, leading to the reduction of the subject to mere training in the use of algorithms. The severity of the problem was succinctly summarized in a recent study [2]:

> ...the students seem to deliver *better* mathematical solutions in a *non*-mathematical context and if the task has *not* been part of the mathematics curriculum.

[my emphasis and translation].

Within the field of interactive learning environments in mathematics, we are working towards the vision of designing a learning environment that is

- explorative and without features that unnecessarily limit the user. The users are not supposed to be locked into "their level" of knowledge.

- multi-faceted but conceptually clear. It should be able to integrate (existing and future) historical accounts, visualizations, demonstrations, definitions, clarifications, etc. into a coherent and conceptually clear whole (in sharp contrast to the present-day World Wide Web structure).

- individualized. The system should be adjustable to and useful for a wide spectrum of users: teachers, mathematicians, students at all levels and complete beginners. Accessibility for humans and devices is also an important factor.
- distributed and dynamic. The system should allow the reuse of globally distributed resources (digital content as well as individuals).
- technically sound. The system should follow the latest technical standards within the field, as well as being portable to a wide range of platforms and environments.
- free software. The system should be free for everyone to use, develop and distribute.

The Garden of Knowledge project, described in [25], ended in 1997 and was the first attempt at an interactive learning environment at CID, where some of these thoughts were first formulated. The concept of a Knowledge manifold as a framework for the design of interactive learning environments was introduced in [26]. At the same time, around 1998, the idea of a concept browser began to take concrete form. The techniques of conceptual browsing and multiple narration in a knowledge manifold were first described in [27]. The first prototype version of Conzilla was developed in 1999, and is documented in [31] and [32]. A more detailed account on the history of mathematical learning environments at KTH from the perspective of the group supervisor and main architect Ambjörn Naeve is given in [29]. A tutorial on modeling mathematics using UML is available in [28].

Chapter 2

Conceptual Modeling with UML

Throughout this report, the technique of *conceptual modeling* is used to convey ideas and give overviews. This chapter introduces the technique, comparing it with other, more well-known techniques such as mind-maps and concept maps. The purpose is not to enter a detailed analysis, but only to highlight the important differences.

2.1 Mind-Maps and Their Siblings

2.1.1 Mind-Maps

The techniques for using mind-maps and conceptual maps have been around since the 1960s, when Tony Buzan developed and patented the mind-mapping technique as a graphic way of conveying information (see [3]). An example mind-map is given in figure 2.1, which illustrates the main characterizing features of mind-maps:

- Mind-maps are tree-like, always starting out from a central concept. Objects in the map are connected with lines.
- They are often very colorful, including images and creative layout.
- They are highly personalized, utilizing any mnemonic technique the creator can think of.

Mind-maps are being widely used in activities such as brainstorming, note taking and creation of overviews, and are generally acknowledged to be a very useful mnemonic technique.

2.1.2 Concept Maps

The concept mapping technique was developed by Prof. Joseph D. Novak at Cornell University in the 1960s (see, e. g., [33] and [34]). Concept maps are like mind-maps in that they are “nodes and arcs” diagrams. But concept maps differ from mind-maps in several important aspects:

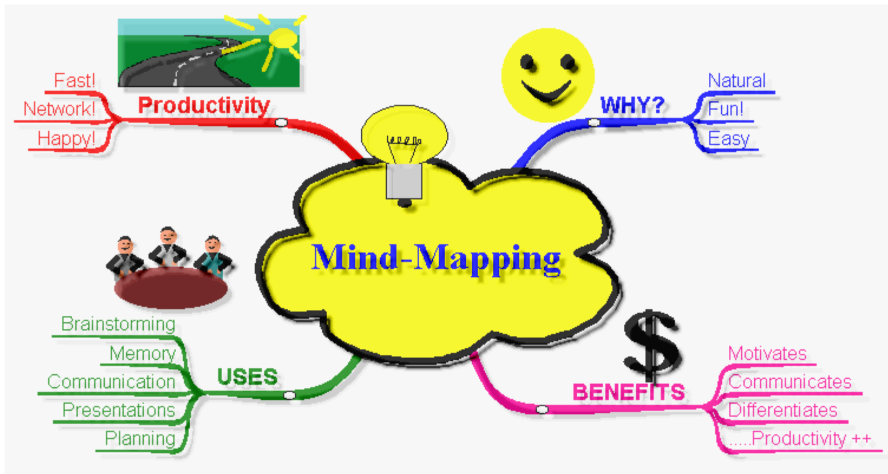

Figure 2.1: Example mind-map (from `http://www.nigeltemple.com/`).

- There is in general no central concept, and concept maps are seldom hierarchical. Instead, they are often general graphs.
- The use of images and colors is very limited.
- The objects in the map are concepts, linked together with directed arrows, containing verbs. This way, a concept maps can be verbalized by forming sentences from the concepts and relations.

An example concept map is given in figure 2.2.

Concept maps have been extensively studied and their benefits in learning and assessment of learning are well documented (see, for example, [35] and [4]). They provide a technique for capturing the internal knowledge of learners and experts, and making it explicit in a visual, graphical form that can be easily examined and shared. They have also been used in making expert systems [7] and in cognitive science/artificial intelligence [22].

2.1.3 Other Diagrams

There are many other forms of visual diagrams resembling the above, such as cognitive mapping, semantic nets, topic maps and others. They all share many of the features of concept maps and mind-maps, and there is no need to discuss them separately.

2.2 Visual Modeling with UML

The Unified Modeling Language, or UML, is an industry standard for object-oriented modeling from OMG, the Object Management Group.

The OMG specification of UML [37] states the purpose of UML as follows:

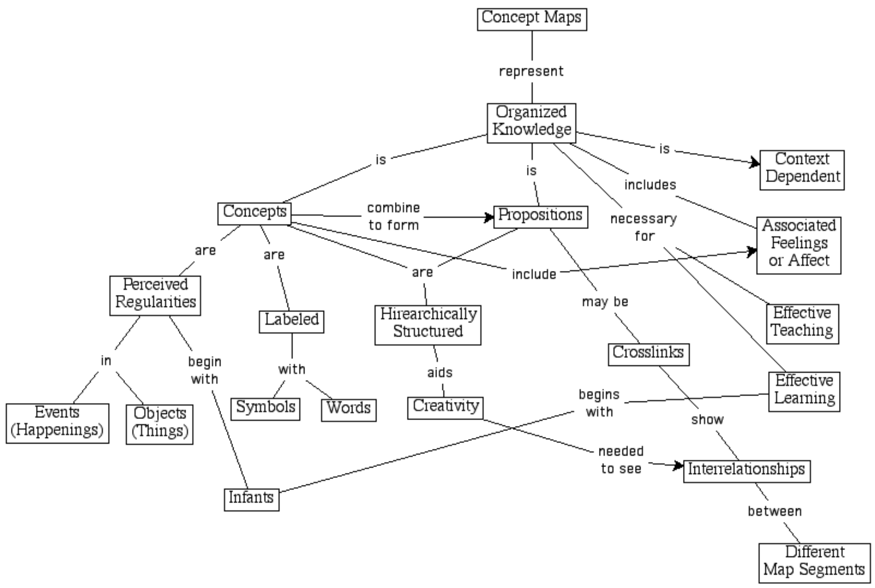

Figure 2.2: Example concept map, taken from [36].

> The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML offers a standard way to write a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components.

UML represents the unification of several hundred modeling languages that were in use before UML (hence the name). UML is used virtually everywhere where software is developed, as a language for communicating structures and relations in a graphical way.

In later years, the use of UML has broadened to include modeling in many different areas, that have nothing *à priori* to do with software systems. Examples include business process modeling, organizational charts etc. It is in this wider applicable form that UML is used here.

2.2.1 UML Basics

UML is a diagrammatic language with its own grammar. For a more complete description of UML, see [39]. For a general introduction to object-oriented modeling and design, see [38]. The most fundamental kind of UML diagram is the *class diagram*. A class diagram depicts the relations between terms using the following fundamental kinds of relations:

- *specialization/generalization* relation*s.* This corresponds to *narrower--broader* terms, such as "car" and "vehicle*".* The concept "car" is a spe-

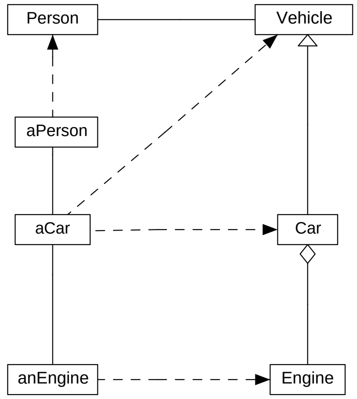

Figure 2.3: Simple class diagram example: Cars, vehicles and motors.

cialization of the concept "vecihle", while the concept "vehicle" is an abstraction of the concept "car"[1]. The relation is always drawn as
, pointing from the more special to the more general.

- *exemplification/classification* relations. This expresses *instance/type* relations between, for example, "my car" and the concept "car". That is, "my car" is an *example/instance* of the concept "car", while the concept "car" is the *class/type* for "my car". The relation is always drawn using the arrow , pointing from the instance to the type.

- *associations*, that are any kind of relations between concepts. An association is in general directionless, and is drawn as a plain line: .

- *part/whole* associations, also called *aggregations,* are an important kind of association, that relates concepts where one in some sense is a part of the other. We say that engines are *parts of* cars, or that cars *contain* engines. This is modeled by creating an *aggregation* relationship between the concept "car" (being the *whole*), and the concept "engine" (being the *part*). The relation is always drawn as , pointing from the part to the whole.

The diagrammatic representations are exemplified in figure 2.3. Figure 2.4 contains a verbalization of the relations. The class diagrams in UML contain many more constructs, but the four given above are the most important for conceptual modeling.

Another kind of UML diagram is the so-called *activity diagram.* An activity diagram encodes the logical structure of an activity in a graphical way. An example activity diagram is given in figure 2.5. An activity diagram encodes

[1]This construct has many names in different communities. In software contexts, it is called a *sub-class/super-class* relation; in the linguistic community the terms *hyponym/hyperonym* are often used.

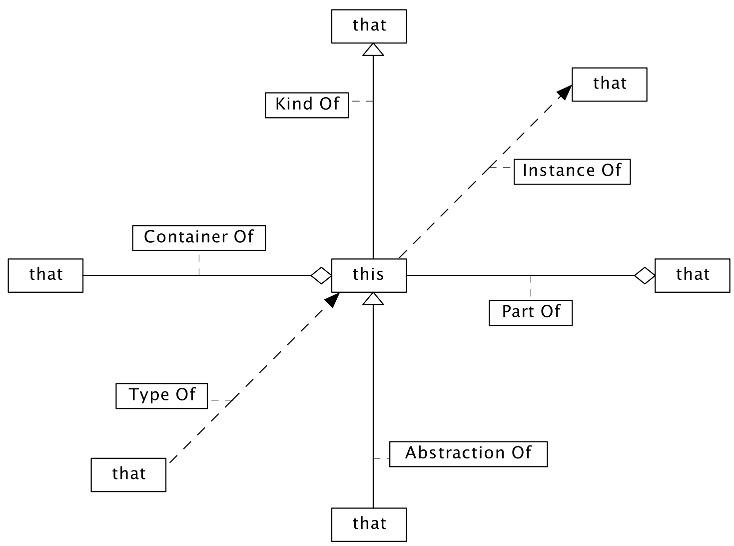

Figure 2.4: The three main relations in UML. Read the figure from *this* to *that*, e.g., *this* is an instance of *that*, *this* is an abstraction of *that*, etc. Note that the explanatory boxes are normally not present. Figure taken from [27].

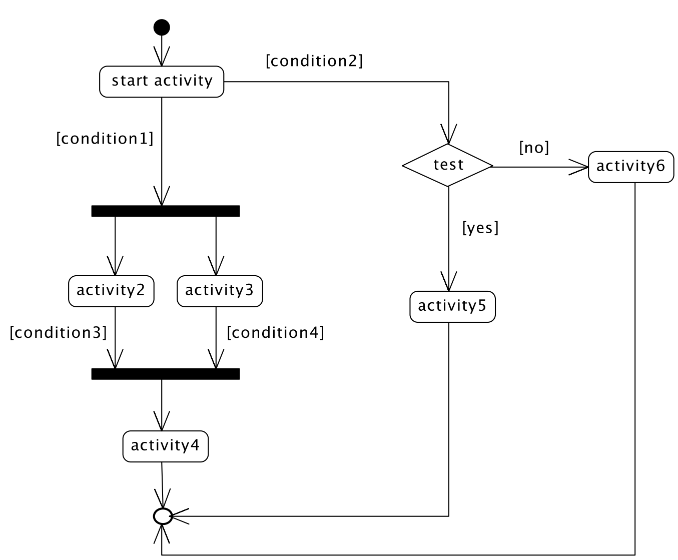

Figure 2.5: An example activity diagram

such things as

- Conditions that have to be fulfilled before continuing to the next activity. In figure 2.5, we see that conditions are drawn within square brackets: [**condition**].
- Activities that can be performed in parallel. The horizontal bar is called a *synchronization bar*. In the example figure 2.5, **activity2** and **activity3** can be performed independently of each other, while they both must be performed before **activity4** is started.

Again, there are many more constructs available in activity diagrams, but we will not need them. There are also other kinds of UML diagrams, such as, e.g., state diagrams, use case diagrams, sequence diagrams, collaboration diagrams and object diagrams, none of which will be used here.

2.3 A Comparison

The above introduction to UML should be enough to convince the reader that UML diagrams are, in essence, very similar to mind-maps and concept maps. There are, however, some important differences that need to be emphasized.

Structure Mind-maps are always association trees starting from a central concept. Concept maps and UML diagrams usually treat some specific topic, but are in general structured as graphs.

Syntax Mind-maps can often be understood by anyone comparatively fast, as they use the perceptual apparatus to trigger concepts. They use no special syntax of their own, but are very "anarchistically" and intuitively constructed. On the other hand, concept maps have a very strict syntax that builds directly on ordinary language. UML diagrams also use a strict syntax to express commonly used concept relationships. Thus, both UML diagrams and concept maps have much more restricted syntax than mind-maps.

Semantics The relations in mind-maps are given no explicit meaning, and the guiding organisational principles are implicit. Trying to express relations in a more precise way using mind-maps is therefore difficult. In contrast to mind-maps, all relations in concept maps are given verbal form and thus an explicit meaning. In UML diagrams, the relations also have well-defined semantics, but they are given a visual form. Thus, both concept maps and UML diagrams can express structures and complex relationsships in a much more precise way than mind-maps.

Verbosity Thanks to the mnemonic techniques used in mind-maps, they often manage to convey many ideas quickly, without using too many words. By contrast, the relations in a concept map must be read as a proposition before it can be understood. Concept maps can easily be perceived as being cluttered, being too verbose. UML diagrams, on the other hand, need no such verbalization to be understood, once their grammar has been learned. Similar relationships are immediately perceived as such and UML-diagrams are, in general, much less cluttered than concept maps.

In short, UML diagrams combine some of the the visual clarity of mind-maps with the expressiveness of concept maps, at the cost of needing a syntax of their own.

2.4 Conceptual Modeling as a Human Language

The term *object-oriented modeling* is often used to describe the process of constructing UML diagrams. However, this term is too heavily associated with software systems. We therefore use the term *conceptual modeling*, introduced in [28], to describe UML-based modeling when applied in non-software scenarios. The term accurately describes the activity as *modeling*, i.e., trying to describe a phenomenon in a simplified and idealized form.

In fact, UML models can be used in much the same way that concept maps are used, that is, both for creating personal overviews, for communicating expert knowledge, for student assessment, etc. Because of this similarity, the kind of UML diagrams used in this report will be called concept maps, although they do not follow the concept map syntax as described above.

It is important to stress that UML models in general do not represent a *formalization* of information or knowledge. Instead, they are just as subjective as ordinary language. They represent attempts at *externalizing* and *communicating* knowledge, just the way ordinary language does. Thus, just like ordinary text, such models must be accompanied with information of *who* made them, *when* they were made, etc. Seen this way, conceptual modeling can be regarded as a form of human language.

As a language, UML models have several benefits over plain text. The main benefits are

- A visual overview, where structures are immediately evident and easy to remember.

- Compactification of language. In graph form, many redundancies that are inevitable in linear text disappear.

- Structure is separated from verbal expression. An UML diagram can quickly be translated to another language.

- Multiple narration. The same diagram tells many stories, depending on how it is read.

For an example of this, consider the concept map of the previous discussion that is given in figure 2.6. Note that there is a certain kind of grammar in the UML relations:

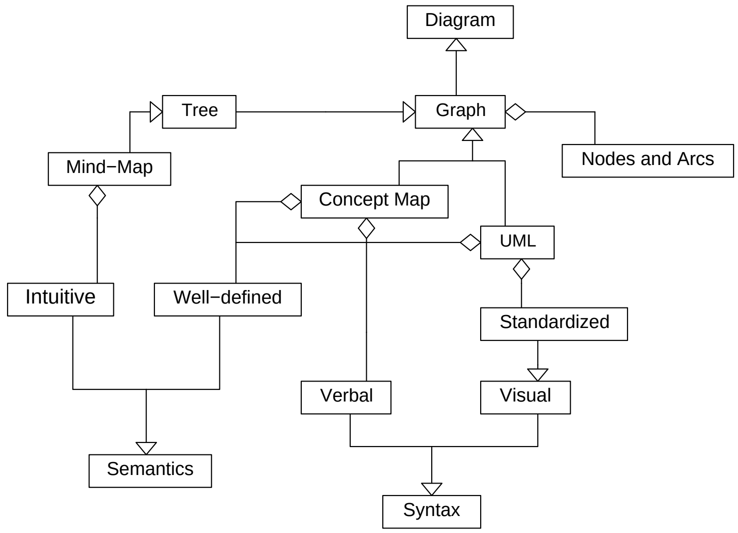

Figure 2.6: Overview of section 2.3.

- Generalizations can often be read together as one word: In the figure, the concept *Visual syntax,* being a kind of *Syntax*, is written only as *Visual*. Reading along the generalization we can recreate *Visual Syntax*. In the same way, *UML Diagrams* are kinds of *Diagrams*, etc.

- Aggregations form words the other way. *Concept maps* contain a *Verbal Syntax*, the *Concept map Syntax*. Or in figure 2.3, *cars* contain *car motors*.

This grammar can be used to remove many redundancies and clarify strucures.

The concept map in figure 2.6 can be verbalized in many ways: *UML Diagrams use nodes and arcs that have a standardized visual syntax. Mind-maps are tree diagrams with an intuitive semantics.* Thus a single map contains large amounts of information in a structured way.

Chapter 3

Geometric Algebra

This chapter contains a small introduction to the basic concepts in geometric algebra following loosely the treatment by David Hestenes in [14].

Geometric algebra is the name of an algebraic system designed to express geometric relations and structures in a natural way. It is extremely flexible, as demonstrated by the fact that the following mathematical systems all can be described in a unified and intuitive way using geometric algebra:

- Complex numbers
- Matrix algebra
- Quaternions
- Vector algebra
- Tensor algebra
- Spinor algebra
- Differential forms

and all this in a coordinate-free theory. There is not enough room in this paper to explore the consequences within mathematics and physics of having such a unified mathematical system. Fortunately, much work (although sketchy in places, due to the enourmous scope of such a task) has already been carried out by David Hestenes at the Department of Physics at Arizona State University. For the interested reader, I refer to the bibliography.

I specifically want to mention [14], which contains the mathematical foundations of geometric algebra and geometric calculus, and [11], which contains a very large number of reformulations of results from classical mechanics using geometric algebra. An overview of geometric algebra is found in figures 3.1 and 3.2.

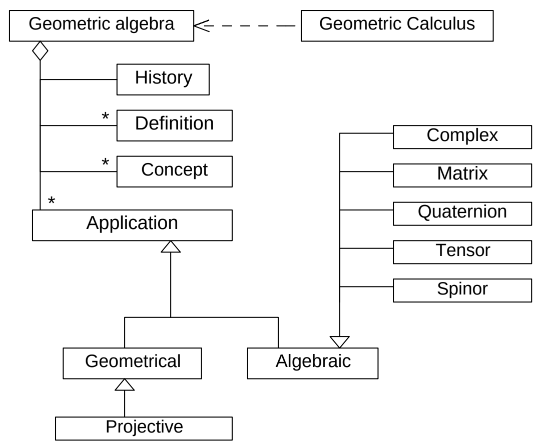

Figure 3.1: Overview of geometric algebra.

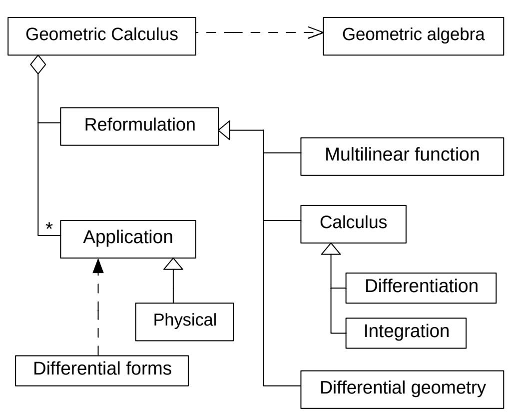

Figure 3.2: Overview of geometric calculus.

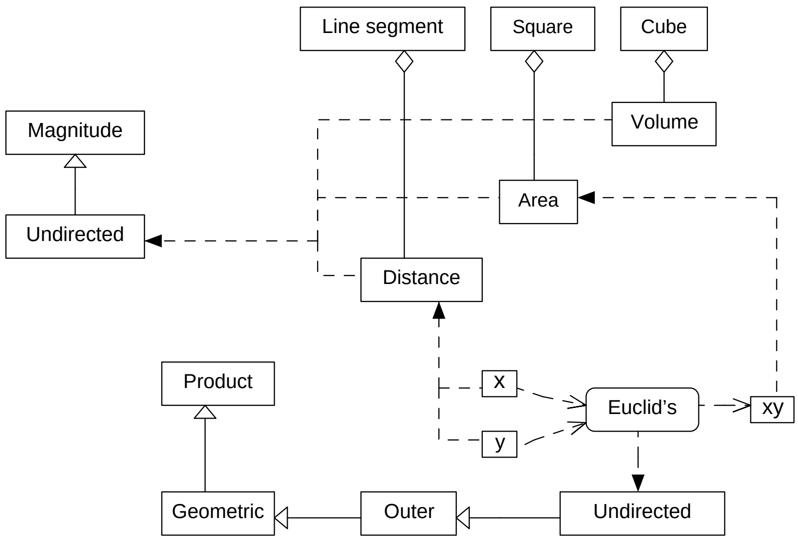

Figure 3.3: The geometric product of Euclid. This map uses a new contruct from UML, the *object flow*. We see the objects x and y flowing into the product, producing the object xy. This will be used more below.

3.1 A Short History of Geometric Algebra

3.1.1 Euclid and Descartes

The history of geometric algebra is the history of directed numbers and the geometric product. Already in Euclid's *Elements*, a clear distinction is made between what he calls *numbers*, which are integers[1], and *magnitudes*, which are measures of geometrical quantities such as length and area. Magnitudes had the peculiar property of not always being comparable as a ratio with another magnitude, and thus the concept of real number to the greeks was a genuinely geometrical quantity.

The product of two length magnitudes was an area, represented by a rectangle. Similarly, the product of a length magnitude and an area magnitude was a volume. But higher-dimensional measures were unknown to the greeks, so length magnitudes and volume magnitudes could not be multiplied. By contrast, multiplying two numbers resulted in a new number, a non-geometric entity. A number could be represented as a magnitude of the corresponding length, while the reverse mapping was impossible as not all magnitudes could be represented as numbers[2]. This is illustrated in figure 3.3.

By the time of René Descartes in the seventeenth century, the concepts of

[1]He used the concept of ratio, but did not conceive of it as a number.

[2]even for a suitable choice of unit length, as proven by the fact that the diagonal of a square is incommensurable with its sides.

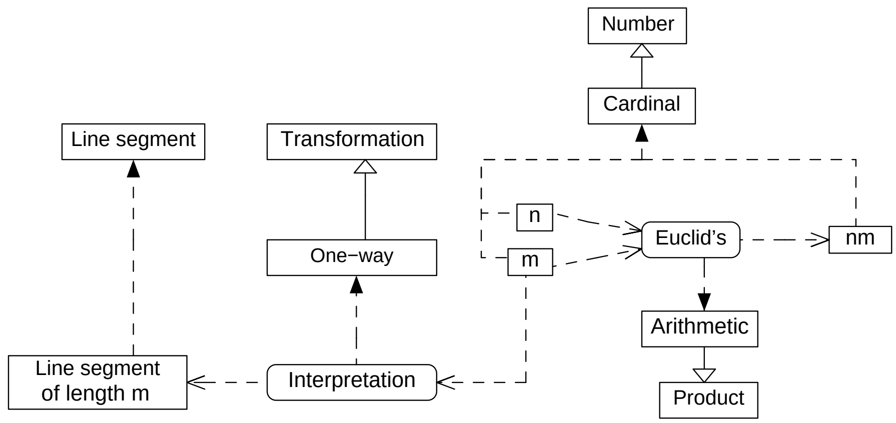

Figure 3.4: The number concept of Euclid.

number and magnitude had changed. No longer was it felt that numbers and magnitudes were fundamentally different – in fact, Descartes without hesitation labelled line segments with symbols representing their numeric lengths. The history of this development of the number concept is too complicated to go into here, but it is interesting to see what happened to the concept of multiplication of numbers/magnitudes. As seen in figure 3.5, Descartes made no distinction between multiplying numbers and line segments. The product of two line segments is a new line segment with a length equal to the product of their lengths, for which he found a geometric construction. Thus the same multiplication was used for both line segments and numbers. This identification of numbers with lengths marked the beginning of analytic geometry, which has made possible the great advancements in science and mathematics to this very day.

One important observation is that Descartes' definition of the geometric product as well as Euclid's number product are *binary compositions*, returning a new object of the same type. Euclid's geometric product, on the other hand, was a product that returned a higher-dimensional object. This kind of product will be called *outer product.*

3.1.2 The Birth of the Vector Concept: Grassmann, Hamilton etc.

Descartes made no distinction between line segments of the same length but different directions when multiplying or adding segments. The idea of using *directed* line segments in analytic geometry is due to Hermann Grassmann and William R. Hamilton in the nineteenth century. Hamilton invented *quaternions*, an algebraic construct which was intended as a generalization of complex num-

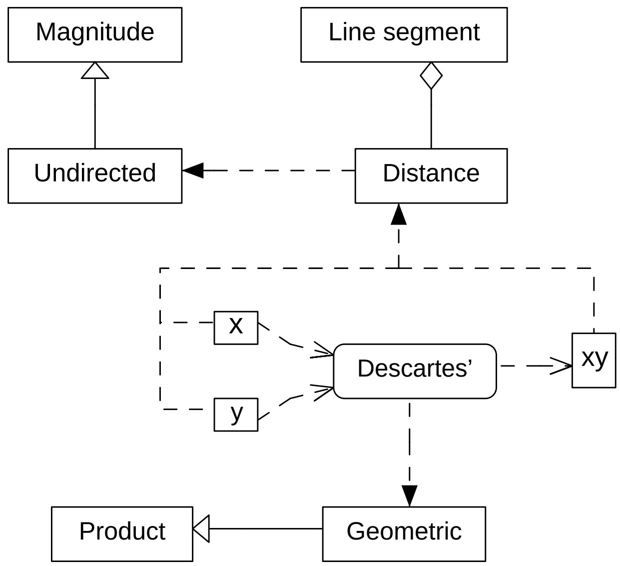

Figure 3.5: The number concept of Descartes

bers to three dimensions. They have vectorial qualities and were (and still are) used to describe rotations in three-dimensional spaces.

However, it was Grassmann who in his *Lineale Ausdehnungslehre [8]* first developed a vector concept like the one we use today, where a vector is seen as representing all line segments of equal length and direction. He also introduced the classical operations on vectors:

- Multiplication by a scalar (a directionless real number), which corresponds to Descartes' multiplication in that it multiplies the length of the vector by a real number. In contrast to Descartes, it keeps the direction of the vector. See figure 3.6.

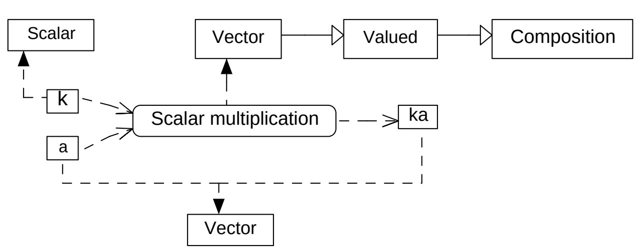

Figure 3.6: Multiplication of vector and a scalar.

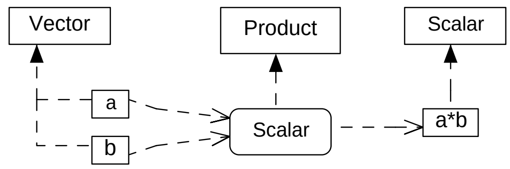

Figure 3.7: The inner product of Grassmann

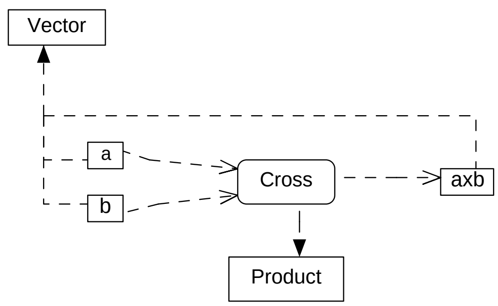

Figure 3.8: The vector product of Gibbs.

- Addition of vectors. This was done, as by Descartes, by placing the line segments end-to-end. But Grassmann *kept their directions*, while Descartes lined them up. The effect was, as we know, a very useful vector addition.[3]

- Inner multiplication of vectors (see figure 3.7). This was a new concept, for the first time taking the directions of line segments into consideration when multiplying them, something neither Descartes nor Euclid had done. This geometric product is an *inner product* in the sense that it returns not a vector, but a scalar, a lower-dimensional object.

Using these operations, the theory of vector algebra was developed. Later, Gibbs added to the list of geometric products the one we know today as the *vector product*, which is only applicable in three dimensions, and which gives a vector orthogonal to each of the vectors in the product. It is an important product, as it complements the information given by the inner product regarding the relative directions of two vectors. More significantly, it is actually the first of the geometric products since Euclid to result in a geometric object! This is one of the more fundamental reasons why it plays such an important role in vector analysis. See figure 3.8. It is also the first binary composition of vectors.

[3] Which, by the way, was used already by Galilei when adding forces.

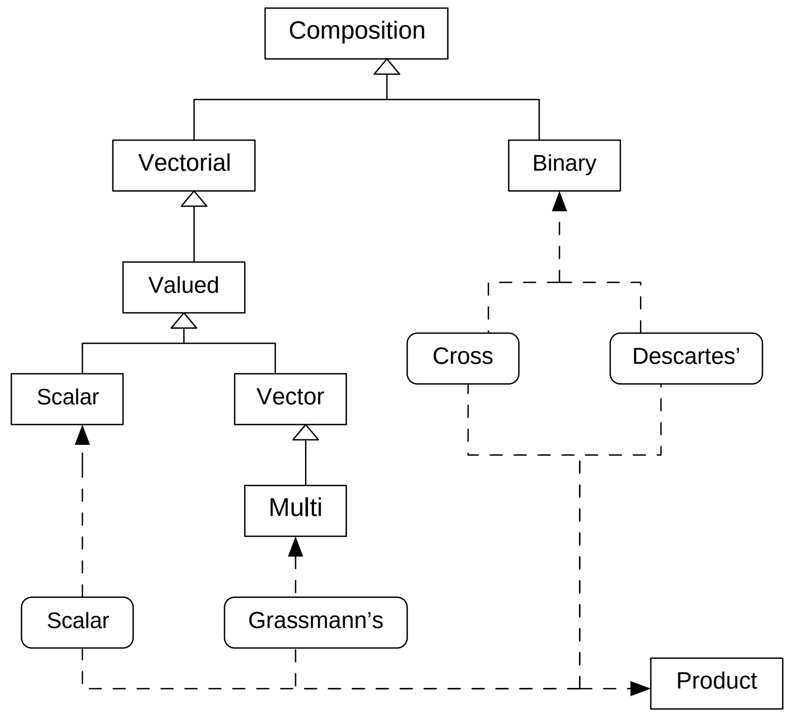

Figure 3.9: Compositions.

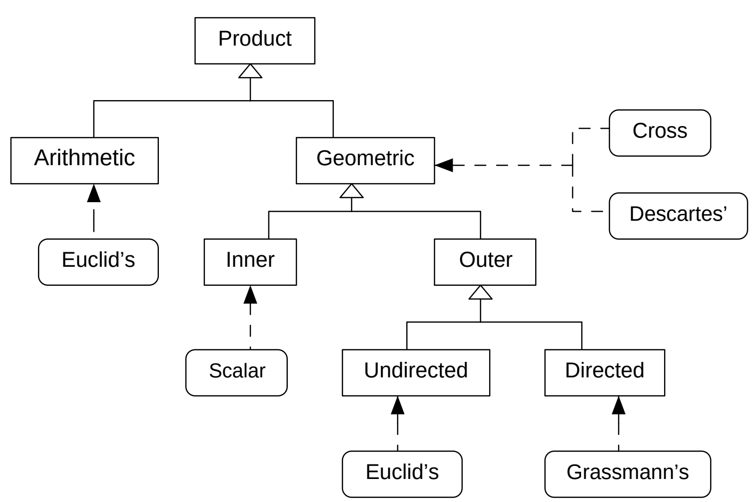

Figure 3.10: Products.

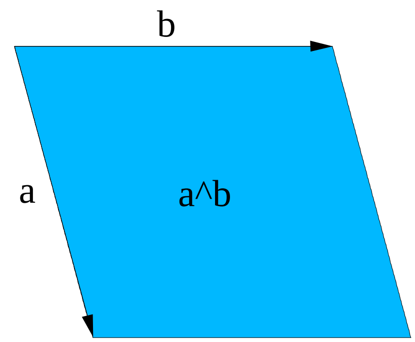

Figure 3.11: The 2-blade $a \wedge b$.

3.1.3 The Outer Product

But the vector product leaves many questions unanswered. What is the generalization to more than three dimensions? Why does it not work even in two dimensions, unlike the other geometric operations? And in three dimensions, what is the significance of the fact that the length of the resulting vector equals the area of the parallelogram spanned by the two vectors in the product?

It is stunning to realize that Grassmann had the answer to those questions, an answer which has not been widely recognized even today. The problem of developing a true geometric product that is capable of providing a full expression of geometrical ideas has recurred in mathematics since Euclid, and many systems have been developed that try to address it, as seen in the introduction to this chapter. However, none of them has succeeded in providing a generally applicable language for geometry that can be express all the ideas contained in those systems. Luckily, Grassmann provided, in the 1840s, the necessary constructs to create such a language.

Grassmann introduced a new product, called the *outer product*, into his vector algebra. He defines the outer product $a \wedge b$ of two vectors a and b to be a new kind of object: a *2-blade*. In effect, just as a vector has a length and a direction, a 2-blade is a two-dimensional object which has an area and a direction. In effect, a 2-blade is a *two-dimensional directed magnitude*. The direction of a bivector is simply a plane in which it is located.

The outer product of two vectors can be represented by the parallelogram spanned by them, as in figure 3.11. This parallelogram is the same as used when constructing the vector product, except that we now represent the product with the parallelogram itself! And just as the vector product, the outer product is anti-commutative: $a \wedge b = -b \wedge a$, which corresponds to a change in orientation of the parallelogram. Note that the parallelogram is only one of many possible representations of a 2-blade. In fact, all plane two-dimensional figures lying in the same plane and with the same numeric area can be used to represent the same 2-blade.

Note that, of all the products seen so far, this product most closely resembles Euclid's line segment product (see figure 3.12). It really generates geometric quantities of higher dimension. Note that in contrast to the vector product,

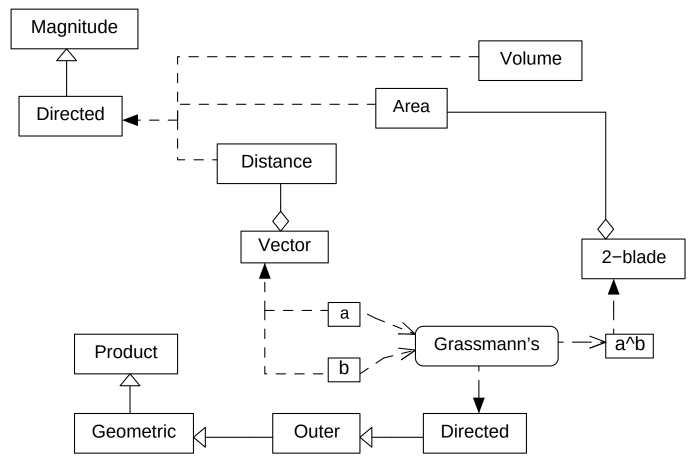

Figure 3.12: The outer product of vectors.

this product works without trouble in two, three, four etc. dimensions, as the resulting object lies in the plane spanned by the two vectors.

Now this definition of the outer product readily lends itself to generalization: the outer product of three vectors, or equivalently, of a vector and a bivector, is defined as a new kind of object, a *3-blade.* A 3-blade is a three-dimensional object which has a volume and a direction. It can be represented as the parallelepiped swept out by the three vectors. This concept is readily generalized to higher dimensions, and the corresponding entities are called k-blades. The next chapter contains a formal definition of the outer product.

3.1.4 The Geometric Product

Grassmann spent most of his efforts on developing the properties of the inner and outer product. In the later stages of his life, he realized that there was one more step of generalizations that could be done, one that unified the products and finally created a true geometric product. But he never had the time to draw the conclusions. Instead, it was Clifford that formalized the ideas into a coherent algebraic systems that also bears his name: *Clifford Algebra*. It is not widely recognized that this algebra is the natural continuation of ideas from Euclid, Descartes and Grassmann on how to create a truly universal geometric algebra.

The fundamental step that was made by Grassmann and perfected by Clifford, was recognizing that the inner and outer product together fully describe the relative directionality of two vectors – the inner product measures the amount of their orthogonality, and the outer product measures the direction of their orthogonality. What they did was that they introduced a new product, which we will call the *geometric product*, defined as

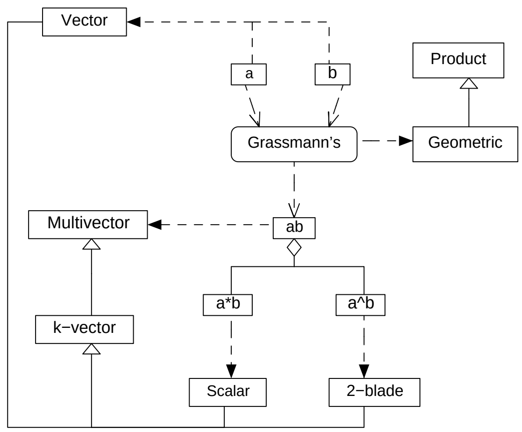

Figure 3.13: The geometric product.

$$ab = a \cdot b + a \wedge b \tag{3.1}$$

This step is highly nontrivial. As one can see, it adds two altogether different things: a scalar and a 2-blade. The question arises: what kind of entity results from this addition? The answer can be understood by an analogy to the complex numbers: adding a real number and an imaginary number creates a new number, a *complex number*, that can be decomposed into the two parts. The same way, the above sum results in a new object consisting of a scalar and a 2-blade! This sum is a so-called *formal* or *direct* sum, which does not mix the objects, but just assembles them. We call the resulting entity a *multivector* consisting of a scalar and a 2-blade part. This is illustrated in figure 3.13. The geometric product can easily be used to retrieve the inner and outer product – we just note that, as the inner product is commutative and the outer product anti-commutative, we have

$$\begin{aligned}
\frac{1}{2}(ab + ba) &= \frac{1}{2}(a \cdot b + a \wedge b + b \cdot a + b \wedge a) \\
&= \frac{1}{2}(a \cdot b + a \cdot b + a \wedge b - a \wedge b) \\
&= a \cdot b
\end{aligned}$$

$$\begin{aligned}
\frac{1}{2}(ab - ba) &= \frac{1}{2}(a \cdot b + a \wedge b - b \cdot a - b \wedge a) \\
&= \frac{1}{2}(a \cdot b - a \cdot b + a \wedge b + a \wedge b) \\
&= a \wedge b
\end{aligned}$$

In general, the elements in a geometric algebra can be of different kinds. Bivectors, being two-dimensional entities, are said to have *grade* 2. Similarly,

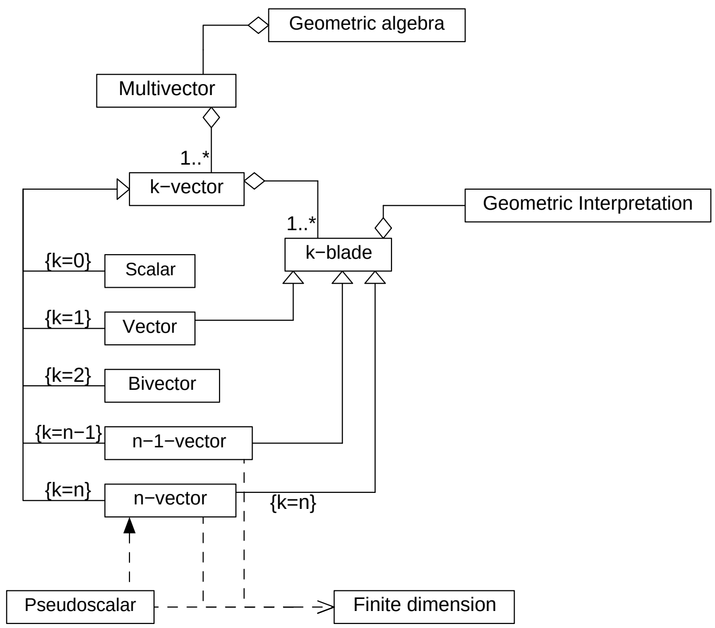

Figure 3.14: The vector concepts in geometric algebra.

we say vectors have grade 1, and 3-vectors grade 3. Scalars are given grade 0, as they have no dimensional properties. In general, k-vectors have grade k. Multivectors thus combine elements of different grade. This is illustrated in figure 3.14. The geometric product creates a powerful and beautiful algebra of multivectors which is defined formally in the next section.

This Clifford algebra has not received much attention as an algebra for geometry, and most mathematicians and physicists are not aware of its algebraic and geometrical power. It is only during the last couple of decades that Grassmann's universal geometric algebra has seen a revival. The physicist David Hestenes at Arizona State University has launched a comprehensive program for the revival of *Geometric Algebra* as the algebraic language of choice for geometry in mathematics and physics, and has contributed reformulations of many mathematical and physical theories in terms of geometric algebra. This includes complex numbers, quaternions, spinor theory, linear algebra, differential geometry, projective geometry as well as large part of classical mechanics.

3.2 Definition

We now turn to a short formal introduction to geometric algebra. In contrast to what Grassmann did, but in line with Clifford, we start with the definition of the geometric product, something that leads to a much cleaner algebraic system.

Definition: A *geometric algebra* is a set G together with two binary compositions: multiplication ($a, b \mapsto ab$) and addition ($a, b \mapsto a + b$), and for each $i \in \mathbf{N}$ an operator $< \cdot >_i : G \to G$ (the grade operator), for which the axioms below are met. The elements of G are called *multivectors*:

Ring Axioms: These axioms make the set into a non-commutative ring. For each element A, B and C from G, the following must hold

1. Commutativity of addition:
$$A + B = B + A$$
2. Associativity of addition:
$$(A + B) + C = A + (B + C)$$
3. Associtativity of multiplication:
$$(AB)C = A(BC)$$
4. Distributivity of multiplication:
$$\begin{aligned} A(B + C) &= AB + AC \\ (A + B)C &= AC + BC \end{aligned}$$
5. There is an element 0, such that $A + 0 = A$
6. There is an element 1, such that $1A = A1 = A$
7. There is an element $-A$ such that $A + (-A) = 0$

Grade Axioms: These axioms ensure the existence of the different sorts of multivectors. A multivector A is said to be a *k-vector*, or to be of grade k, if $A =< A >_k$.

1. The grade operator is a projection, i.e., for all A in G, $<< A >_k>_k =< A >_k$. Thus, $< A >_k$ is always a k-vector, and we can call $< A >_k$ the *k-vector component* of A. Thus the grade operator $< \cdot >_i : G \to G$ returns, for each i, the i-vector component of a multivector.
2. Each element A is the sum of its k-vector components:
$$A =< A >_0 + < A >_1 + < A >_2 \dots$$
This creates a unique decomposition of elements in G into k-vectors. This is the reason for calling the elements of G multivectors.
3. The elements of grade 0 are the real numbers, and are also called *scalars*. Addition and multiplication of scalars are the ordinary addition and multiplication, and the 0- and 1-elements of the geometric algebra are 0 and 1, respectively. Additionally, scalars commute with all multivectors.

4. The grade operator is linear over elements of grade 0. That is, for all elements λ of grade 0 (i.e. real numbers), and all multivectors A and B,
$$< \lambda A + B >_k = \lambda < A >_k + < B >_k$$
Before formulating the next axiom, we need a definition: A k-*blade* is any multivector of the form $a_1 a_2 a_3 \dots a_k$, where all a_i are non-zero vectors (elements of grade 1) that all anticommute, i.e. $a_i a_j = -a_j a_i$ for $i \neq j$. We can now formulate the blade axiom:
5. A k-blade is a non-zero k-vector, and any k-vector can be written as a sum of k-blades. This creates a grounding for multivectors in vectors, as any multivector can be described as a sum of products of vectors.

Euclidean Sign Axiom:

1. For each vector (i.e. 1-vector) a, its square a^2 is a positive real number. This axiom gives us a Euclidean structure on the algebra.

An equivalent, shorter definition is:

> A *geometric algebra* is a graded, non-commutative algebra over the real numbers satisfying Grade Axioms 4 and 5, and the Euclidean Sign Axiom.

The axiom structure is illustrated in figure 3.15. In this figure, we also see that the euclidean sign axiom is a special case of a more general sign axiom that only requires a^2 to be a real number. The non-degenerate sign axiom requires that there is a basis $\{e_i\}$ with $e_i e_j = -e_j e_i$ (i.e., it is orthogonal) and $a_i^2 \neq 0$, and is used in non-euclidean geometry such as four-dimensional Minkowski space.

Another thing we see is that we may assume things about the dimensionality of the space. We will study some properties of finite spaces below, but most properties of geometric algebra are independent of dimension.

3.3 Intuition

This list of axioms seems complicated, and indeed it is rather extensive. It is also not very clear how to go about calculating with them. So, now that we have an axiomatic grounding for geometric algebra, let us try to give an idea what the algebra looks like.

3.3.1 The Linear Space of 1-vectors

First, let us note that the 1-vectors form a linear space over the 0-vectors. This is easily seen directly from the axioms. There is a twist, however: the 0-vector and the number 0 must be identical! The reason for this difference is that the real numbers are embedded into a larger algebra, that by definition must have only one zero element. The resulting space of 1-vectors is denoted by G^1. In

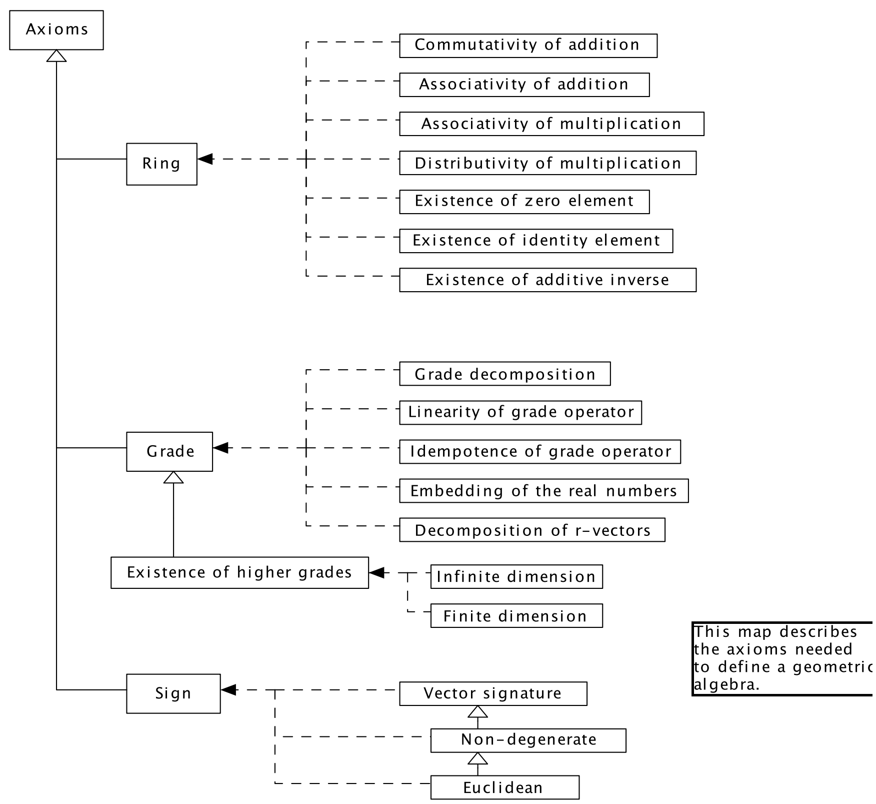

Figure 3.15: The axioms of geometric algebra

the examples that follow, we assume that G^1 has the finite dimension n, even though most of the results are more general.

3.3.2 The Inner Product of Vectors

Taking two vectors a and b, we can now define the *inner product* of them as we did in section 3.1.4:

$$a \cdot b = \frac{1}{2}(ab + ba).$$

This is always a scalar, as $(a+b)^2$, a^2 and b^2 are scalars, and

$$(a+b)^2 = a^2 + ab + ba + b^2 = a^2 + b^2 + 2a \cdot b.$$

Now we can readily verify that this defines an inner product with the usual properties: bilinear, symmetric and positive definite. We thus get a Euclidean space, with the usual definition of *orthogonality*: $a \cdot b = 0$. In the case of finite dimension, let us introduce an orthonormal basis $\{e_1, e_2, e_3, \ldots, e_n\}$ in this space, which we will use from now on to illustrate the algebra. We note some of the properties of the inner product:

1. a and b are orthogonal if and only if $ab = -ba$.

2. a and b are *parallel*, i.e., $a = \lambda b$ for some scalar λ, if and only if $a \cdot b = ab$, or equivalently, $ab = ba$.

   Note first that if $a = \lambda b$,

   $$\begin{aligned} a \cdot b &= (\lambda b) \cdot b = \frac{1}{2}(\lambda bb + b\lambda b) \\ &= \lambda b^2 = (\lambda b)b = ab \end{aligned}$$

   The other way around, let $b = b_\| + b_\perp$, where $b_\|$ is parallel to a, and $b_\perp$ is orthogonal to a. If $ab = ba$, then $0 = ab - ba = 2ab_\perp$. But Grade Axiom 5 then says that either $b_\perp = 0$ or $a = 0$, both of which indicate $a \parallel b$.

3. If the vector b is written as $b_\| + b_\perp$, we have $a \cdot b = ab_\|$

4. We define the *length* of a vector the usual way: $|a| = \sqrt{a \cdot a} = \sqrt{a^2}$. A more general *norm* on the whole of G will be defined later.

3.3.3 The Outer Product of Vectors

Similarly, we introduce the *outer product* of two vectors as earlier:

$$a \wedge b = \frac{1}{2}(ab - ba)$$

and note some of its properties which follow immediately:

1. It is bilinear and anti-commutative: $a \wedge b = -\frac{1}{2}(ba - ab) = -b \wedge a$.

2. a and b are orthogonal if and only if $a \wedge b = ab$.

3. a and b are parallel if and only if $a \wedge b = 0$. In particular, $a \wedge a = 0$.

4. If the vector b is written as $b_{\parallel} + b_{\perp}$, where $b_{\parallel}$ is parallel to a, and $b_{\perp}$ is orthogonal to a, we have $a \wedge b = ab_{\perp}$. Hence, according to Grade Axiom 5, when $a \neq 0$ and $b \nparallel a$, $a \wedge b$ is always a 2-blade and thus a bivector.

5. We always have
$$ab = a \cdot b + a \wedge b$$
So the geometric product of two vectors has a scalar part and a bivector part. We also note that $a \wedge b =< ab >_2$, the highest-grade part of the geometric product of a and b, which we will use below to generalize the outer product.

This concludes the formalization of the informal discussion in the previous section.

3.3.4 Calculating with the Geometric Product

We have now seen what happens when taking the geometric product of vectors, but the geometric product in general remains obscure. But it turns out that using the basis defined in 3.3.2, we can make the geometric product explicit – we just need to see what it does to basis vectors. And this is an easy task – we need only remember the following two rules, which are direct consequences of the ortonormality of the basis:

$$\begin{aligned} e_i e_j &= -e_j e_i \text{ if } i \neq j \\ e_i^2 &= 1 \end{aligned}$$

In words: different base vectors anticommute, while the square of a base vector is 1. For example, the geometric product of $a = a_1 e_1 + a_2 e_2$ and $b = b_1 e_1 + b_2 e_2$ is

$$\begin{aligned} ab &= (a_1 e_1 + a_2 e_2)(b_1 e_1 + b_2 e_2) \\ &= a_1 b_1 + a_2 b_2 + (a_1 b_2 - a_2 b_1) e_1 e_2 \end{aligned} \tag{3.2}$$

which makes it perfectly clear that

$$\begin{aligned} a \cdot b &= a_1 b_1 + a_2 b_2 \\ a \wedge b &= (a_1 b_2 - a_2 b_1) e_1 e_2. \end{aligned}$$

Note that $e_1 e_2$ is a bivector by definition, and that $a \wedge b$ has a natural measure $a_1 b_2 - a_2 b_1$, which is exactly the directed area of the parallelogram spanned by a and b. This way, products of any multivectors expressed in this basis can easily be calculated. But we still only have a basis for G^1, so we turn to the problem of finding a basis for the whole of G.

3.3.5 Bivectors Explained

With the aid of this simple formulation we can now analyze the space of 2-vectors (or *bivectors*), which we call G^2. Remember that all bivectors can be described as a sum of 2-blades. 2-blades are of the form a_1a_2, where $a_1a_2 = -a_2a_1$, which we now know to mean that a_1 and a_2 are orthogonal. But if $a_1 = \sum_1^n \lambda_i e_i$ and $a_2 = \sum_1^n \mu_i e_i$, then we can expand and rearrange the product a_1a_2, collecting terms containing the same pair e_ie_j:

$$a_1a_2 = \sum_{i \neq j} \lambda_i \mu_j e_i e_j = \sum_{i<j} (\lambda_i \mu_j - \lambda_j \mu_i) e_i e_j$$

The terms where $i = j$ all add to zero, as $\sum \lambda_i \mu_i e_i e_i = a_1 \cdot a_2 = 0$. The remaining terms are all multiples of e_ie_j, and thus are all bivectors. This shows that all bivectors can be expressed as a sum of terms of the form $\lambda_{ij} e_i e_j$. We now only need to verify that the bivectors $\{e_ie_j \mid i, j = 1 \ldots n,\ i < j\}$ are linearly independent to prove that they form a basis for the space of bivectors. Let

$$\sum_{i<j} \lambda_{i,j} e_i e_j = 0.$$

Multiplying this from the left with any bivector e_se_r, where $1 \leq r < s \leq n$ gives the multivector

$$\lambda_{r,s} + [\text{terms of grade 2 and 4}] = 0$$

which shows $\lambda_{r,s}$ must be zero. The same argument applies to any r and s, so all $\lambda_{i,j} = 0$. Thus we know that $\{e_ie_j \mid i, j = 1 \ldots n,\ i < j\}$ is a basis for the bivectors.

3.3.6 k-vectors Explained

An analogous argument can be applied to the space of k-vectors, which is a linear subspace G^k of G, showing that terms of the form

$$e_{i_1} e_{i_2} \ldots e_{i_k}$$

with strictly increasing index i_r, form a basis for G^k. Thus, a little combinatorics tells us that G^k has the dimension $\binom{n}{k}$.

As we only have n independent base vectors, there are no elements of grade higher than n. Hence, by combining the bases for all G^k, we get a basis for the whole of G. For example, when the vector space G^1 has dimension three, a basis for G is:

$$\{1,\ e_1,\ e_2,\ e_3,\ e_1e_2,\ e_2e_3,\ e_1e_3,\ e_1e_2e_3\}$$

If we regard the whole geometric algebra G as a linear space over the real numbers, this linear space has the dimension

$$\sum_{k=1}^{n} \binom{n}{k} = 2^n$$

which is a beautiful result. Since we now have a basis for G, we have sufficient information to carry out all sorts of calculations, and any definitions we make can easily be formulated in terms of this basis.

3.3.7 k-blades Explained

The one remaining unexplained multivector concept seems to be k-blades. We will now show:

1. that a k-blade $A_k = a_1 \dots a_k$ uniquely determines a linear subspace $\overline{A_k}$ of the vector space G^1, namely the subspace spanned by the vectors $a_1, \dots, a_k$.
2. that a given subspace V of G^1 defines a unique k-blade A_k, except for a scalar factor.

When we have proven this, we will adopt the terminology that a vector b is said to be *parallel* to A_k if it lies in $\overline{A_k}$, and *orthogonal* to A_k if it lies in $\overline{A_k}^{\perp}$.

Step 1:

How can we, given a k-blade A_k, be sure that $\overline{A_k}$ does not depend on the choice of orthogonal vectors $a_1, \dots, a_k$? We must find a definition of $\overline{A_k}$ that is independent of the choice of $a_1 \dots a_k$.

Let us analyze the product of A_k with a vector $b = b_{\|} + b_{\perp}$, where $b_{\|}$ is the component of b parallel to $a_1, \dots, a_k$, and $b_{\perp}$ the component orthogonal to $a_1, \dots, a_k$. We first note that $A_k b_{\perp}$ is the $(k+1)$-blade $a_1 \dots a_k b_{\perp}$.

On the other hand, $b_{\|} = \sum \lambda_i a_i$, and the product $A_k b_{\|}$ therefore is a sum of terms of the form $a_1 a_2 \dots a_k a_i$. Such a term can be rearranged using the rules $a_i a_j = -a_j a_i$ and $a_i^2 =$ [scalar]. The result is always a $(k-1)$-blade, consisting of a multiple of the vectors a_j, $j \neq i$. For this reason, $A_k b_{\|}$ gives a $(k-1)$-vector – the sum of those $(k-1)$-blades.

Finally, we have:

$$A_k b = \underbrace{A_k b_{\|}}_{\text{grade } k-1} + \underbrace{A_k b_{\perp}}_{\text{grade } k+1} \tag{3.3}$$

that is, multiplying a k-blade with a vector has a grade-lowering and a grade-raising function, played by the components of the vector parallel and orthogonal to $a_1, \dots, a_k$, respectively. The above decomposition actually implies that $A_k b$ has components of grade $k-1$ and $k+1$ only, so that

$$A_k b = < A_k b >_{k-1} + < A_k b >_{k+1} \tag{3.4}$$

This means that b is parallel to the vectors $a_1, \dots, a_k$ precisely when $< A_k b >_{k+1} = 0$, and orthogonal to them precisely when $< A_k b >_{k-1} = 0$. These expressions are independent of the choice of vectors $a_1, \dots, a_k$, and we can thus define $\overline{A_k}$ to be the space of vectors b such that $< A_k b >_{k+1} = 0$.

Step 2:

On the other hand, given a k-dimensional subspace V of G^1, and two k-blades A_k and B_k with $\overline{A_k} = \overline{B_k} = V$, i.e., both defining the same unique subspace of G^1, we can show that $B_k = \lambda A_k$. In other words, a subspace of G^1 uniquely defines a k-blade (up to multiplication by a scalar).

To show this, let $A_k = a_1 \ldots a_k$ with orthogonal a_i, and $B_k = b_1 \ldots b_k$ with orthogonal b_j. Both sets of vectors $\{a_i\}$ and $\{b_j\}$ are by definition bases for V. Let us express B_k in terms of the a_i. We have

$$b_j = \sum_{i=1}^{k} \lambda_{ij} a_i \quad \text{for some } \lambda_{ij}.$$

Using this to expand $B_k = b_1 \ldots b_k$, we get a sum of terms of the form $\lambda a_{i_1} a_{i_2} \ldots a_{i_k}$. Such a term is a k-blade if all a_{i_l} are different. If the same a_i occurs twice, they will produce a scalar, and the term will be a blade of grade at most $k-2$. However, we know that B_k is of grade k, so the only terms that do not cancel are those where no a_i occurs twice, i.e., where each a_i occurs exactly once. But those terms can be rearranged to be of the form $\lambda a_1 a_2 \ldots a_k$, and thereafter added. This shows that, in fact, $B_k = \lambda A_k$ for some λ.

We have now established the geometrical interpretation of k-blades: they lie in a k-dimensional subspace of G^1, and within such a subspace, they only differ by a scalar constant. This constant can be interpreted as measuring the relative k-volume of the blades. This justifies the interpretation of k-blades as directed areas. For example, the 2-blade e_1e_2 defines a unit area in the e_1, e_2-plane. The 2-blade e_2e_1 lies in the same plane, but has the opposite area: $e_2e_1 = -e_1e_2$. As the value of the constant is proportional to the length of each vector a_i, the k-blade concept really seems to capture the idea of a directed measure of the area. The absolute value of the area of a k-blade $A_k = a_1 \ldots a_k$ is simply $\sqrt{a_1^2 \ldots a_k^2}$. A more general *norm* of multivectors will be defined below.

3.3.8 Conclusions

This ends our short introduction to the basic concepts of geometric algebra. We have managed to:

- Construct an explicit basis for the whole algebra in the finite case, together with simple algebraic rules for calculation in that basis.
- Give a geometric interpretation of the k-blades that corresponds to the intuitive picture of directed planes and volumes.
- Analyze the basic features of the different products in the geometric algebra.

The reader is referred to figure 3.14 for an overview of the multivector concept.

3.4 Constructions

As seen in figure 3.16, a geometric algebra can also be explicitly constructed in several ways. This is important, as this enables us to establish the consistency of the axioms[4], and also aids us in understanding the algebra. However, there

[4] relative to what was used in the construction, of course, as absolute consistency cannot be proven.

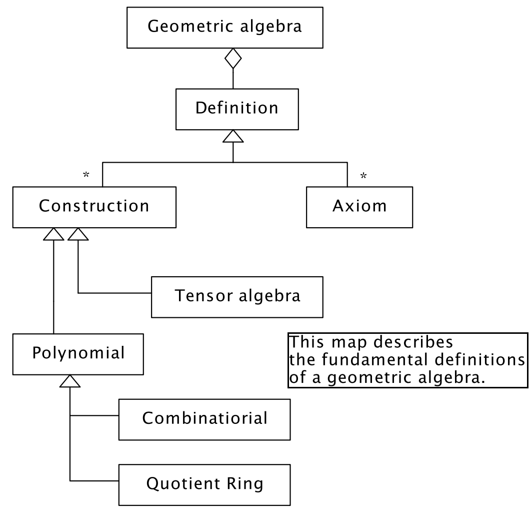

Figure 3.16: Definitions of geometric algebra

is no room here to go into the details of the constructions. But in figures 3.17 and 3.18, three constructions are described as activity diagrams.

3.5 Algebraic Concepts

We are now ready to formally define some algebraic concepts in geometric algebra. This section is far from exhaustive; for further reference, see [14] and [11]. An overview of the operators in geometric algebra is given in figure 3.19, while figure 3.20 shows the interdependencies between the definitions of the operators.

3.5.1 Generalizing the Outer Product

Our first mission is to generalize the outer product. We want to generalize it to the whole of G as Grassmann did. The intuition was that the outer product of two vectors was to represent the parallelogram spanned by the two vectors, and in the same way, we want $a \wedge b \wedge c$ to be the parallelepiped spanned by the three vectors, with the appropriate volume. In general, for the outer product of k vectors $a_1, \ldots, a_k$, we want $a_1 \wedge \ldots \wedge a_k$ to be a k-blade of volume resulting from taking the product of the orthogonal lengths of the a_i (this is precisely the generalized parallelepiped volume), lying in the space spanned by $a_1, \ldots, a_k$. In fact, if $a_1, \ldots, a_k$ are orthogonal, we want $a_1 \wedge \ldots \wedge a_k = a_1 \ldots a_k$.

From equation 3.4 above we know that

$$A_k b = < A_k b >_{k-1} + < A_k b >_{k+1}$$

for k-blades A_k. We also know that the second term is the $k+1$-blade $A_k b_\perp$, which corresponds precisely to what we would want $A_k \wedge b$ to mean: the $(k+1)$-

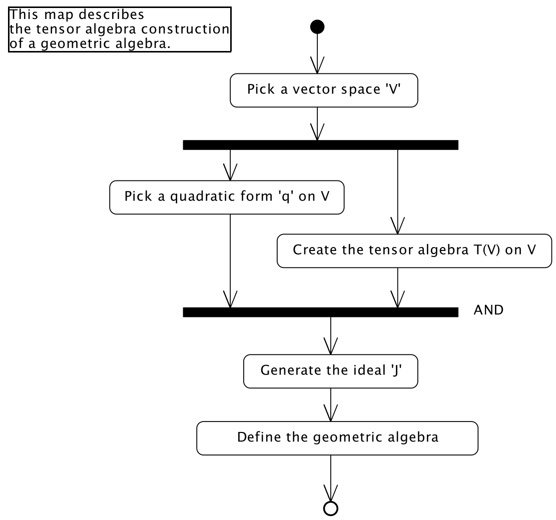

Figure 3.17: Tensor algebra construction

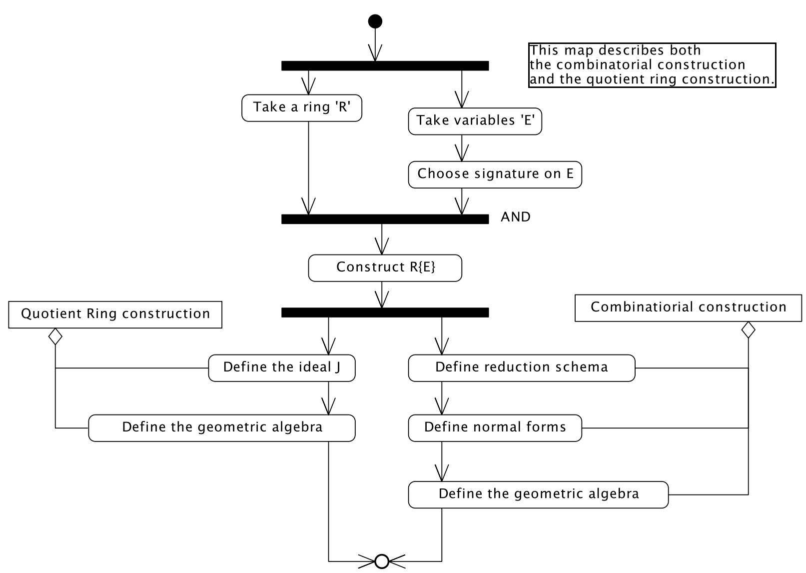

Figure 3.18: Quotient ring/Combinatorial construction

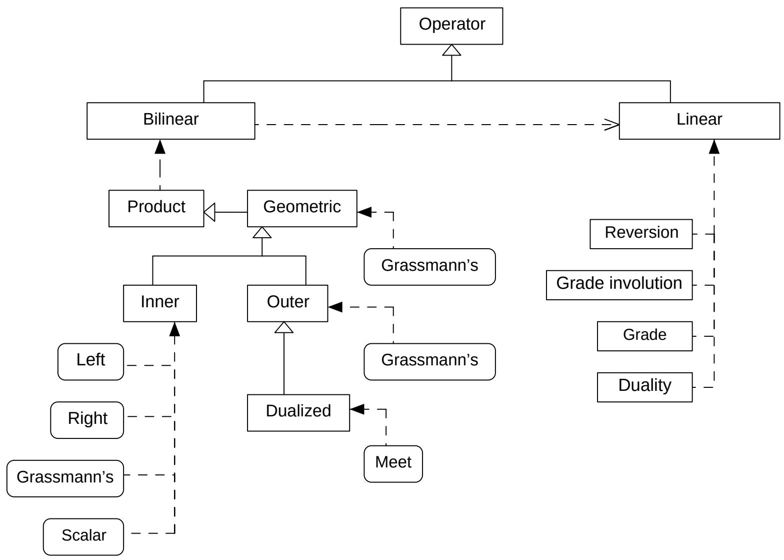

Figure 3.19: The operators of a geometric algebra

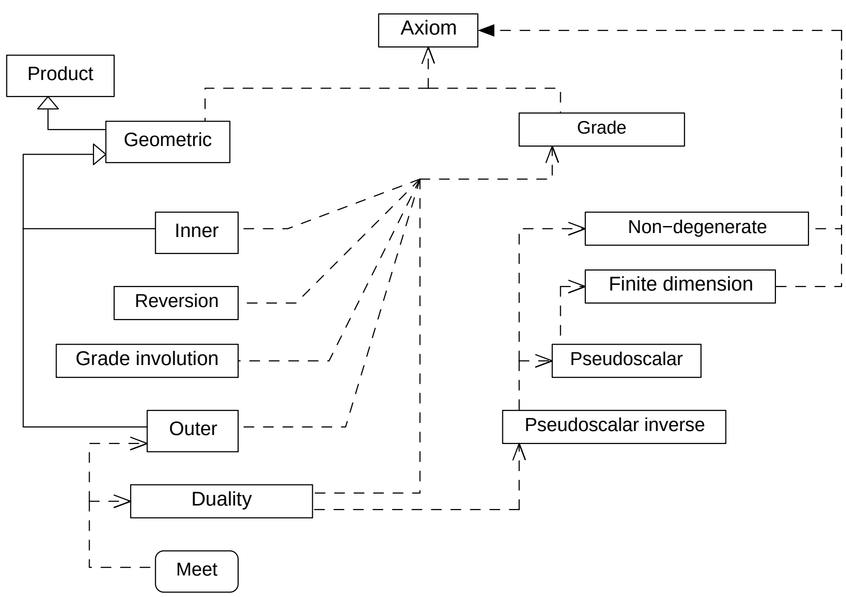

Figure 3.20: The dependencies between the constructs in geometric algebra. Note that this map contains a new kind of UML relation: the *dependency*. For example, the meet product depends on the outer product and the duality operator for its definition.

blade swept out by the orthogonal parts of the vectors $a_1, \ldots, a_k, b$. So we define

$$A_k \wedge b =< A_k b >_{k+1} \tag{3.5}$$

This definition has an natural generalization to k-blades: If A_r is an r-blade, and B_s an s-blade, we define:

$$A_r \wedge B_s =< A_r B_s >_{r+s} \tag{3.6}$$

Now that we know how this works for any two blades, it can be linearly extended to the whole space. This product can easily be shown to be associative.

But how to calculate with the outer product? We need to make sure we know how to calculate with the base elements. If we want to take the outer product of two base elements, $e_{i_1} \ldots e_{i_k}$ and $e_{j_1} \ldots e_{j_m}$, we only need to check if they contain a common base vector. If they have, say, the vector e_{j_l} in common, it will act grade-lowering, and there will be no $k+m$-vector component of the outer product, and so the outer product is 0. Otherwise, all vectors are orthogonal, and the outer product is just $e_{i_1} \ldots e_{i_k} e_{j_1} \ldots e_{j_m}$. For example,

$$\begin{aligned}(e_1 + e_3) \wedge (e_3 + e_2 e_3) &= e_1 \wedge e_3 + e_1 \wedge (e_2 e_3) + e_3 \wedge e_3 + e_3 \wedge (e_2 e_3) \\ &= e_1 e_3 + e_1 e_2 e_3\end{aligned}$$

We now state some consequences of the definition of the outer product, in addition to those in section 3.3.3:

1. $A_k \wedge A_k = 0$ for all k-blades A_k, $k > 0$, as $A_k^2 = a_1 \ldots a_k a_1 \ldots a_k$ is a scalar.

2. If a is a vector and A_k a k-blade, it follows from 3.3 and 3.5 that
$$A_k \wedge a = A_k \wedge a_\perp = A_k a_\perp \tag{3.7}$$

3. On the other hand, from the same equations it follows that a is orthogonal to A_k precisely when
$$A_k \wedge a = A_k a \tag{3.8}$$
while a is parallel to A_k precisely when
$$A_k \wedge a = 0 \tag{3.9}$$

4. If $a_1, \ldots, a_k$ are any linearly independent vectors, and $b_1, \ldots, b_k$ are the vectors that result from carrying out Gram-Schmidt orthogonalization on $a_1, \ldots, a_k$, then
$$a_1 \wedge \ldots \wedge a_k = b_1 \wedge \ldots \wedge b_k = b_1 \ldots b_k$$
This is clear, since only the successive orthogonal components remain in each step in both cases. If the vectors $a_1, \ldots, a_k$ are linearly dependent, the expressions are all equal to zero.

5. Thus, if $a_1, \ldots, a_k$ are linearly independent, then $a_1 \wedge a_2 \wedge \ldots \wedge a_k$ is a k-blade with volume $\sqrt{b_1^2 \ldots b_k^2}$.

6. If $a_1, \ldots, a_k$ are linearly dependent, then $a_1 \wedge a_2 \wedge \ldots \wedge a_k = 0$ and vice versa. This is because being linearly dependent is equivalent to not spanning a k-volume.

3.5.2 The Inner Product

We now generalize the definition of the inner product. The *inner product* of two blades A_r and B_s is defined as

$$A_r \cdot B_s =< A_r B_s >_{|r-s|} \tag{3.10}$$

which is linearly extended to the whole of G, with the additional requirement $\lambda \cdot A_k = A_k \cdot \lambda = 0$ for scalars λ.

Let a be a vector, and A_k a k-blade. Recall the formula 3.4:

$$A_k a =< A_k a >_{k-1} + < A_k a >_{k+1}$$

With the above definition of the inner and outer products, we have

$$A_k a = A_k \cdot a + A_k \wedge a \tag{3.11}$$

so we see that the inner product is the grade-lowering part of the geometric product with a vector, and the outer product the grade-raising part.

Calculating with the inner product is as simple as for the outer product, if we only note that we only need to keep those products of base elements, where one of the sets of base vectors e_i is completely contained in the other (so that they maximally cancel, and the grade of their product is $|r-s|$):

$$\begin{aligned}(e_1 + e_2e_3) \cdot (e_2e_3 + e_1e_2) &= e_1 \cdot (e_2e_3) + e_1 \cdot (e_1e_2) + (e_2e_3) \cdot (e_2e_3) + (e_2e_3) \cdot (e_1e_2) \\ &= e_1e_1e_2 + e_2e_3e_2e_3 \\ &= e_2 - 1\end{aligned}$$

Let us enumerate some properties of the inner product:

1. If a is a vector and A_k a k-blade, it follows from 3.11 and 3.10 that
$$A_k \cdot a = A_k \cdot a_{\|} = A_k a_{\|} \tag{3.12}$$

2. On the other hand, from the same equations it follows that a is parallel to A_k precisely when
$$A_k \cdot a = A_k a \tag{3.13}$$
while a is orthogonal to A_k precisely when
$$A_k \cdot a = 0 \tag{3.14}$$

3. The inner product $A_k \cdot a$ of a vector a and a k-blade $A_k = a_1 \dots a_k$ gives a $(k-1)$-vector which is inside A_k (in the sense of being composed of the a_i), but orthogonal to the projection $a_{\|}$ on A_k.

   That it is a combination of the vectors in A_k was seen in section 3.3.7. That it is orthogonal to $a_{\|}$ is seen by proving $(A_k \cdot a) \cdot a_{\|} = 0$:
$$\begin{aligned}(A_k \cdot a) \cdot a_{\|} &= (A_k a_{\|}) \cdot a_{\|} = [\text{definition}] \\ &= < A_k a_{\|} a_{\|} >_{k-2} \\ &= < A_k a_{\|}^2 >_{k-2} = 0\end{aligned}$$

which gives some intuition of the inner product between a vector and a k-blade. Similar relations hold between two blades: the inner product is always orthogonal to the projection.

4. Two distinct base k-blades $e_I = e_{i_1} \dots e_{i_k}$ and $e_J = e_{j_1} \dots e_{j_k}$ are "orthogonal" in the sense that $e_I \cdot e_J = 0$.

3.5.3 The Geometric Product Revisited

The equation 3.4

$$A_k a =< A_k a >_{k-1} + < A_k a >_{k+1}$$

is linear in A_k, so it is still valid when A_k is a k-vector. Multiplying this formula with a vector b that is orthogonal to a, and re-applying the formula gives:

$$\begin{aligned} A_k ab &= (< A_k a >_{k-1} + < A_k a >_{k+1})b \\ &= (B_{k-1} + B_{k+1})b \\ &= < B_{k-1}b >_{k-2} + < B_{k-1}b >_k + < B_{k+1}b >_k + < B_{k+1}b >_{k+2} \end{aligned}$$

Thus, $A_k ab$ has components of grade $k-2$, k and $k+2$:

$$A_k ab =< A_k ab >_{k-2} + < A_k ab >_k + < A_k ab >_{k+2}$$

Re-applying this argument with any blade instead of ab to the right, we see that we always get terms differing in grade by two. The maximal grade is $k+$[number of vectors to the right], and the minimal $k-$[number of vectors to the right]. Thus, for any r-blade A_r and any s-blade B_s, with $r \geq s$,

$$A_r B_s =< A_r B_s >_{r-s} + < A_r B_s >_{r-s+2} + \ldots + < A_r B_s >_{r+s}$$

However, the same argument applies when multiplying from the left instead, with $r \leq s$, so in general:

$$A_r B_s =< A_r B_s >_{|r-s|} + < A_r B_s >_{|r-s|+2} + \ldots + < A_r B_s >_{r+s}$$

This expression is bilinear, so it applies just as well when A_r and B_s are r- and s-vectors, respectively. We now see that the highest-grade component is the outer product: $< A_r B_s >_{r+s} = A_r \wedge B_s$, while the lowest-grade component is the inner product: $< A_r B_s >_{|r-s|} = A_r \cdot B_s$.

Of course, when calculating using the basis vectors, the above formulas are simple to understand: the grade is lowered by one when a base vector is multiplied by itself, otherwise the grade is raised. For example:

$$\begin{aligned} (e_1 + e_3)(e_2 e_3 + e_3 e_4) &= e_1 e_2 e_3 + e_1 e_3 e_4 + e_3 e_2 e_3 + e_3 e_3 e_4 \\ &= e_1 e_2 e_3 + e_1 e_3 e_4 - e_2 + e_4 \end{aligned}$$

3.5.4 The Scalar Product

The scalar product is of interest for multivectors in general. It is defined as

$$A * B = < AB >_0$$

i.e., the scalar part of the geometric product of A and B. Some properties of the scalar product:

1. If A_k and B_k are k-vectors, we have

$$A_k * B_k = A_k \cdot B_k$$

2. If A_r and B_s are r- and s-vectors, respectively, with $r \neq s$, we have

$$A_r * B_s = 0$$

3.5.5 Reversion

We define the *reverse* $A_k^\dagger$ of a k-vector A_k as

$$A_k^\dagger = (-1)^{\frac{k(k-1)}{2}} A_k = (-1)^{\binom{k}{2}} A_k$$

This is then extended linearly to the whole algebra.

For k-blades $A_k = a_1 \dots a_k$, this is precisely what happens when reversing the order of multiplication: $a_k \dots a_1 = (-1)^{\frac{k(k-1)}{2}} a_1 \dots a_k = (a_1 \dots a_k)^\dagger$, as this involves $\binom{k}{2} = \frac{k(k-1)}{2}$ swaps of adjacent elements. For any vectors a and b, we have $(ab)^\dagger = a \cdot b + (a \wedge b)^\dagger = a \cdot b - a \wedge b = ba$, as $a \wedge b$ is a 2-blade. In fact, it can be proven that for any multivectors A and B, $(AB)^\dagger = B^\dagger A^\dagger$, justifying the name "reversion". As a special case, for any vectors $a_1, \dots, a_k$, we have $(a_1 \dots a_k)^\dagger = a_k \dots a_1$.

3.5.6 Multivector Norm

We define the *norm* $|A|$ of a multivector A as

$$|A| = \sqrt{A^\dagger * A}$$

This is shown to be well-defined by first noting that $A^\dagger * A = \sum_r < A^\dagger >_r * < A >_r$. Letting $< A >_r = \sum_i \lambda_i e_{i_1} \dots e_{i_r} = \sum_i A_r^i$ be the decomposition of $< A >_r$ into a linear combination of base r-blades $e_{i_1} \dots e_{i_r}$, we see that $< A^\dagger >_r * < A >_r = \sum_i (A_r^i)^\dagger A_r^i$, as all mixed terms cancel. We now note that for a base r-blade $e_I = e_{i_1} \dots e_{i_r}$, we have $e_I^\dagger e_I = e_{i_r} \dots e_{i_1} e_{i_1} \dots e_{i_r} = 1$, so $< A^\dagger >_r * < A >_r = \sum_i \lambda_i^2 \geq 0$, and so $A^\dagger * A \geq 0$, and the norm is well-defined. Note that the definition is not dependent on a choice of basis.

3.5.7 Inverse of k-blades

Every non-zero k-blade $A_k = a_1 \dots a_k$ has a multiplicative inverse $A_k^{-1} = \frac{A_k^\dagger}{|A_k|^2}$, as $A_k A_k^\dagger = A_k^\dagger A_k = a_1^2 \dots a_k^2 = |A_k|^2 > 0$.

3.5.8 The Pseudoscalar

In a geometric algebra where the vector space is of dimension n, the space of n-vectors is one-dimensional, with the single base element $e_1 \dots e_n$. This element is called the *pseudoscalar* of G, and is denoted by I. As $I^\dagger = e_n \dots e_1$, we see that $I^\dagger$ is the inverse of I. We note that any product $a_1 \dots a_n$ of n orthogonal vectors must be a multiple of I, as the space of n-vectors is one-dimensional. This multiple is $\lambda = (a_1 \dots a_n)I^{-1}$.

More generally, for any vectors $a_1, \dots, a_n$ (orthogonal or not), the outer product $a_1 \wedge \dots \wedge a_n$ is an n-vector (which is zero when they are linearly dependent). If we define

$$det(a_1, \dots, a_n) = (a_1 \wedge \dots \wedge a_n)I^{-1}$$

this defines the usual determinant in a coordinate-free way, as is easily verified. Using this definition, the whole theory of determinants can be developed (which is done in [10]).

3.5.9 Dualization

In a geometric algebra where the vector space is of dimension n, we can define the *dual* of a multivector A as

$$\widetilde{A} = AI^{-1}$$

For k-blades A_k, we will show that $\widetilde{A_k}$ is a blade which is completely orthogonal to A_k, and that in fact, $\overline{\widetilde{A_k}} = \overline{A_k}^\perp$, i.e. that being orthogonal to A_k is the same as being parallel to $\widetilde{A_k}$. This is why $\widetilde{A_k}$ is called the dual of A_k.

By normalization, A_k can always be written as $A_k = \lambda a_1 \dots a_k$, where the a_i are orthogonal unit vectors. Extending this set to an orthonormal set $a_1, \dots, a_n$ creates an orthonormal basis for G^1. We see that $a_1 \dots a_n = I$, and that $I^{-1} = a_n \dots a_1$. Thus

$$\widetilde{A_k} = A_k I^{-1} = \pm\lambda a_{k+1} \dots a_n$$

But $\overline{A_k}^\perp = a_{k+1} \dots a_n$, so it is clear that $\overline{\widetilde{A_k}} = \overline{A_k}^\perp$.

3.5.10 The Meet Product and the Subspace Algebra on G

The *meet* product, or *dual outer product*, is defined so that

$$\begin{aligned} \widetilde{A \vee B} &= \widetilde{A} \wedge \widetilde{B} \\ A \vee B &= (\widetilde{A} \wedge \widetilde{B})I \end{aligned}$$

In words, the meet of A and B is calculated by taking the outer product of their duals, and then taking not the dual, but the inverse dual of the result.

The duality operator is fundamental to geometric algebra, in that it relates the subspace $\overline{A_k}$ defined by A_k to its orthogonal complement $\overline{A_k}^\perp$ in an algebraic way, namely as

$$\overline{A_k}^\perp = \overline{\widetilde{A_k}}$$

The outer product, in this view, acts as the subspace sum, for if $A_r = a_1 \ldots a_r$ and $B_s = b_1 \ldots b_s$ are blades for which $A_r \wedge B_s \neq 0$, we have that the subspace spanned by $A_r \wedge B_s$ is the sum of the subspaces spanned by A_r and B_s:

$$\overline{A_r \wedge B_s} = \overline{A_r} + \overline{B_s}$$

This is clear, as $A_r \wedge B_s \neq 0$ means that all the vectors $a_1, \ldots, a_r, b_1, \ldots, b_s$ are linearly independent.

The meet product, on the other hand, formalizes subspace intersection. Thus, if A_r and B_s are blades that together span the space (this means that $\widetilde{A_r} \wedge \widetilde{B_s} \neq 0$), the meet $A_r \vee B_s$ spans precisely the intersection of the spans of A_r and B_s:

$$\overline{A_r \vee B_s} = \overline{A_r} \cap \overline{B_s}$$

This is shown using the previous equalities as follows (note that $\overline{AI} = \overline{AI^{-1}}$ and that $\widetilde{A_r} \wedge \widetilde{B_s} \neq 0$):

$$\begin{aligned}
\overline{A_r \vee B_s} &\equiv \overline{(\widetilde{A_r} \wedge \widetilde{B_s})I} = \overline{\widetilde{A_r} \wedge \widetilde{B_s}}^{\perp} = \left(\overline{\widetilde{A_r}} + \overline{\widetilde{B_s}}\right)^{\perp} = \left(\overline{A_r}^{\perp} + \overline{B_s}^{\perp}\right)^{\perp} \\
&= \overline{A_r} \cap \overline{B_s}
\end{aligned}$$

3.6 Applications

We now turn to some applications of geometric algebra.

3.6.1 Projections

Let A_k be a k-blade and b be a vector with the decomposition $b = b_\perp + b_\parallel$ with respect to A_k. Recalling the equations 3.7 and 3.12, we see that

$$\begin{aligned}
b_\parallel &= b_\parallel A_k A_k^{-1} \\
&= (b_\parallel \cdot A_k) A_k^{-1} \\
&= (b \cdot A_k) A_k^{-1}
\end{aligned}$$

and

$$\begin{aligned}
b_\perp &= b_\perp A_k A_k^{-1} \\
&= (b_\perp \wedge A_k) A_k^{-1} \\
&= (b \wedge A_k) A_k^{-1}
\end{aligned}$$

so that $b_\parallel$ and $b_\perp$ can be easily calculated using b and A_k. In fact, the above motivates the definition of *projection on* A_k as:

$$P_{A_k}(b) = (b \cdot A_k) A_k^{-1}$$

and the *rejection from* A_k as

$$P^{\perp}_{A_k}(b) = (b \wedge A_k) A_k^{-1}$$

so that $b = P_{A_k}(b) + P^{\perp}_{A_k}(b)$ is the decomposition of b into components parallel and orthogonal to A_k.

3.6.2 Planes and Lines

From the properties of the outer product, we know that for any k-blade A_k, $a \wedge A_k = 0$ precisely when a lies in the k-volume defined by A_k. For a vector b, we see that the equation

$$x \wedge b = 0$$

defines the line spanned by b. The equation $(x - a) \wedge b = 0$ defines the line parallel to b, going through a. And the equation for a plane is

$$(x - a) \wedge A_2 = 0$$

Note that those equations are independent of the dimension of the space.

The shortest distance from the origin to a plane on the point a defined in this way is described by the vector $a_\perp = P^\perp_{A_2}(a)$, which is that part of a which is orthogonal to A_2. Thus the vector d describing the distance from an arbirtary point p to the plane is

$$d = P^\perp_{A_2}(a - p) = ((a - p) \wedge A_2)A_2^{-1}$$

the magnitude of which is the distance between the plane and the point.

3.6.3 Complex Numbers

Let $A = a_1a_2$ be any 2-blade of unit length, i.e. $|A|^2 = a_1^2a_2^2 = 1$. Then note that

$$A^2 = a_1a_2a_1a_2 = -a_1^2a_2^2 = -1$$

and, for scalars x_1, x_2, y_1, y_2

$$(x_1 + y_1A)(x_2 + y_2A) = x_1x_2 - y_1y_2 + (x_1y_2 + y_1x_2)A$$

Thus, the subalgebra generated by $\{1, A\}$ is isomorphic to the complex numbers, with A playing the role of the imaginary unit i. This way, all "imaginary" qualities of i suddenly disappear, and the complex numbers are given a firm place in geometry.

This subalgebra naturally operates on the vector space. Let v be any vector in $\overline{A}$, say $v = xa_1 + ya_2$. Then

$$Av = -ya_1 + xa_2$$

effectively rotating v by 90°. Thus, multiplication with the multivector $z = \cos\alpha + A\sin\alpha$ gives

$$(\cos\alpha + A\sin\alpha)v = (x\cos\alpha - y\sin\alpha)a_1 + (x\sin\alpha + y\cos\alpha)a_2$$

which can be verified to mean rotation of v by the angle $\alpha = arg(z)$. Thus all rotations in a plane can be described using multivectors of the above form.

3.6.4 Reflection

Picking a unit vector u, the operator

$$U_u(x) = -uxu$$

reflects vectors in the plane orthogonal to u. To see this, decompose $x = x_\perp + x_\parallel$, orthogonal and parallel to u. We note that $ux_\parallel = u \cdot x_\parallel = x_\parallel \cdot u = x_\parallel u$, and $ux_\perp = u \wedge x_\perp = -x_\perp \wedge u = -x_\perp u$. We get:

$$\begin{aligned} U_u(x_\parallel + x_\perp) &= -ux_\parallel u - ux_\perp u \\ &= x_\perp - x_\parallel \end{aligned}$$

Thus, U reflects the component of x parallel to u.

3.6.5 Rotation in Three Dimensions

We know that a rotation in three dimensions is the composition of two reflections[5]. Thus every reflection can be written in the form $U_u \circ U_v$ for some vectors u and v. We note that

$$U_u \circ U_v(x) = uvxvu$$

Letting $A = uv = u \cdot v + u \wedge v$, we see that any rotation can be described by R_A, where

$$R_A(x) = A^\dagger x A$$

The multivector A is a combination of a scalar and a bivector. It is easy to show that any such multivector describes a rotation. The space of all such multivectors has the basis

$$\{1,\, e_1e_2,\, e_2e_3,\, e_3e_1\}$$

Setting $i = e_1e_2$, $j = e_2e_3$, and $k = e_3e_1$, we note that

$$\begin{aligned} i^2 = j^2 = k^2 &= -1 \\ ij = -ji &= k \\ jk = -kj &= i \\ ki = -ik &= j \\ ijk &= -1 \end{aligned}$$

and we see that we have resurrected the quaternion theory of rotation, developed by Hamilton in the 19th century. This discussion generalizes to *spinor theory*, which is widely used in quantum physics.

[5] This can of course be proven in geometric algebra as well. See [11], where the theory of roations is more fully developed.

3.6.6 The Vector Cross Product

In an algebra where the vector space has dimension three, we can introduce the following vector operator:

$$a \times b = \widetilde{a \wedge b}$$

The result is a vector since the dual of a bivector in three dimensions is a vector. In fact, we know it is a vector orthogonal to the plane of the bivector. Noting that $aI = Ia$ for any vector a, and so $A_2 I = a_1 a_2 I = I a_1 a_2 = I A_2$ for any 2-blade A_2 , we get the length of the vector $a \times b$ by

$$|a \times b|^2 = |(a \times b)^2| = |(a \wedge b)I(a \wedge b)I| = |(a \wedge b)^2 I^2| = |-(a \wedge b)^2| = |a \wedge b|^2$$

Thus, the length of this vector is the area of the parallelogram spanned by a and b. The orientation of $a \times b$ is clear by noting

$$e_1 \times e_2 = e_1 e_2 e_3 e_2 e_1 = e_3$$

The conclusion is that the cross product defined here is identical to the ordinary cross product.

3.6.7 Projective Geometry

Having explored some of the uses of geometric algebra within analytical geometry, we now turn to a different geometric interpretation of geometric algebra: *projective geometry.* We assume the vector space G^1 of G has dimension n, and make the following definitions:

1. A *point* is a one-dimensional subspace of G^1.
2. A *line* is a two-dimensional subspace of G^1.
3. A *k-plane* is a $(k+1)$-dimensional subspace of G^1.

This amounts to the classical subspace interpretation of projective geometry. But there is a twist: just as a vector defines a point (the subspace which is the span of the vector), a 2-blade A_2 defines a line, namely $\overline{A_2}$, the two-dimensional subspace defined by A_2. In general, a k-blade A_k defines a $(k-1)$-plane $\overline{A_k}$.

The advantage of the k-blade representation is clear when we list some algebraic properties interpreted as projective geometric relations:

1. Two points p, q are the same if $p \cdot q = pq$. Two k-planes A_k and B_k are the same if $A_k \cdot B_k = A_k B_k$.
2. Two points p, q define a unique line $p \wedge q$. A point p is on a line A if and only if $p \wedge A = 0$.
3. Two lines A and B intersect in a point if and only if $A \wedge B = 0$. In the case of a three-dimesional algebra ($\mathbf{P}^2$), their intersection is simply $A \vee B$.
4. Three points are collinear if and only if $p \wedge q \wedge r = 0$.

5. In $\mathbf{P}^2$, three lines are concurrent (meet in a single point) if and only if $(P \vee Q) \wedge R = 0$.

Continuing in this fashion, all concepts, theorems and proofs of projective geometry can be formulated in geometric algebra, enabling algebraic proofs and constructions without reference to coordinates. In this way, the gap between the algebraic and synthetic approaches to projective geometry is finally closed. A detailed treatment is given in [18]. An algebraic unification of projective, affine and metric geometry using geometric algebra is given in [30].

3.6.8 Other Applications

Other applications of geometric algebra to mathematical systems include a coordinate-free reformulation of linear algebra and and the theory of multilinear functions. The benefits of this approach are detailed in [13] and [14].

The theory of Lie groups, spinor algebra and tensor algebra can also be formulated in geometric algebra. This promises to have important applications in physics.

3.6.9 Conclusions

It should be obvious from this quick overview that geometric algebra provides a powerful and versatile framework for most of analytic geometry, and that it provides a much more consistent and intuitive theory than what is usually presented to the mathematics students. Using geometric algebra, seemingly different theories can be described within a single mathematical system. The implications for the application and teaching of mathematics are far reaching.

3.7 Geometric Calculus

We have thus far only treated the purely algebraic features of geometric algebra. But there is much more to the theory. David Hestenes has pioneered the development of a *Geometric Calculus*, which is the application of geometric algebra to the theories of differentiation, integration and differential geometry. There is no room to examine geometric calculus here (see, for example, [14], [11] and [20]), but we note some of the main features:

- A coordinate-free theory of directed integration with all the benefits of differential forms, but fully integrated into the vector algebra of the space, including a coordinate-free directed form of the coordinate-change theorem.

- A coordinate-free formulation of differential geometry.

- A generalization of the Cauchy integral theorem to n dimensions.

- A more general and complete definition of the derivative, finally fully integrating the *curl* and the *divergence,* making possible a formulation of Maxwell's four laws as $\partial F = J$.

Chapter 4

Conzilla

Conzilla is a powerful interactive tool for browsing concept maps, which has been developed at CID over the past couple of years. To understand the difference between the printed versions of the map and the online form, this chapter presents an overview of the most important aspects of conceptual browsing with Conzilla. A more complete introduction is given in [27].

4.1 Browsing, Viewing and Informing

From a very concrete point of view, there are three main modes of interaction with Conzilla: *browsing, viewing* and *informing.*

Browsing the context means examining the concepts in a map and their relationships, and switching to view other maps. Each concept is annotated with textual descriptions that pop up when hovering over it. This is illustrated in figure 4.1. This way, the user can understand things that may not be immediately evident from the visual information. However, once the intention of the concepts in the diagram has been made clear, the user is not distracted by this information, but can rely on the visual information. Associations are also annotated – an important feature to explain conceptual relationships.

Changing maps is done in one of two major ways. Firstly, each concept can be hyper-linked to a specific diagram describing it in more detail. So, for example, when clicking on the concept *Definition* in the overview map in figure 3.1, you are taken to the definition map in figure 3.16. Secondly, you can jump to any map where a certain concept occurs. This is illustrated in figure 4.2. Using these two navigation features, one can explore the space of maps in a convenient manner.

The concepts are associated with *content* in the form of images, web pages and other digital material, such as interactive demonstrations. *Viewing the content* of each concept is done by bringing up the popup-menu over a concept. The different forms of material can then be viewed in an ordinary web browser, and in fact, any material that can be made available on the web is a candidate for inclusion as content for a concept. This content can be sorted and filtered into different aspects. An example is given in figure 4.3.

Each map and each concept can also be annotated with *metadata*, describing

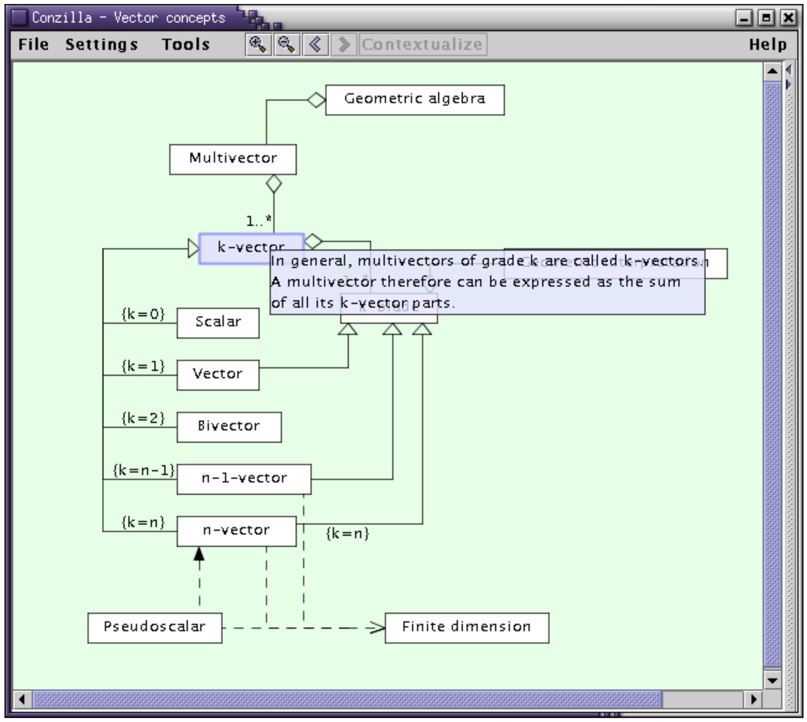

Figure 4.1: Pop-up descriptions

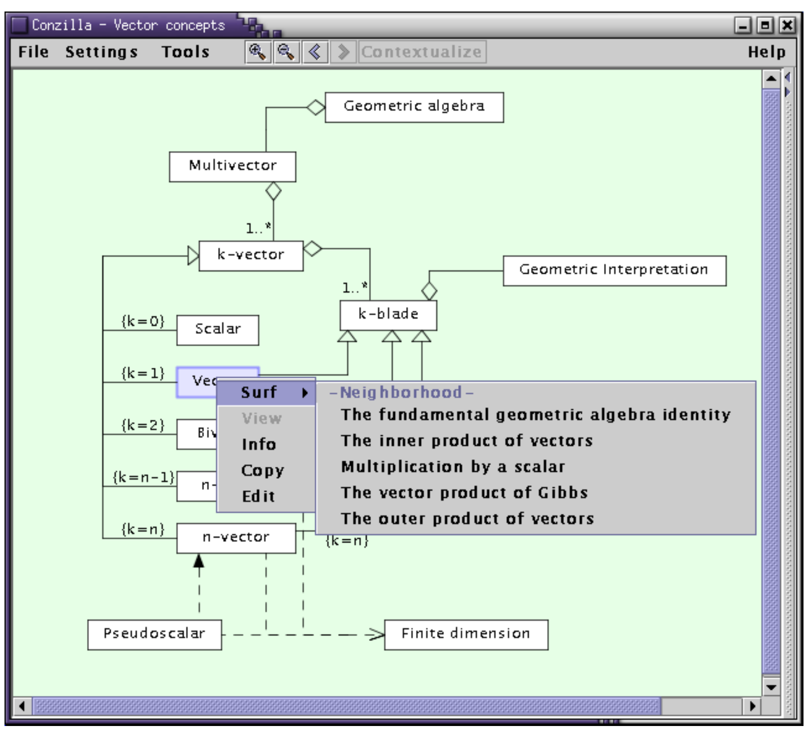

Figure 4.2: Surfing to other maps containing the *Vector* concept.

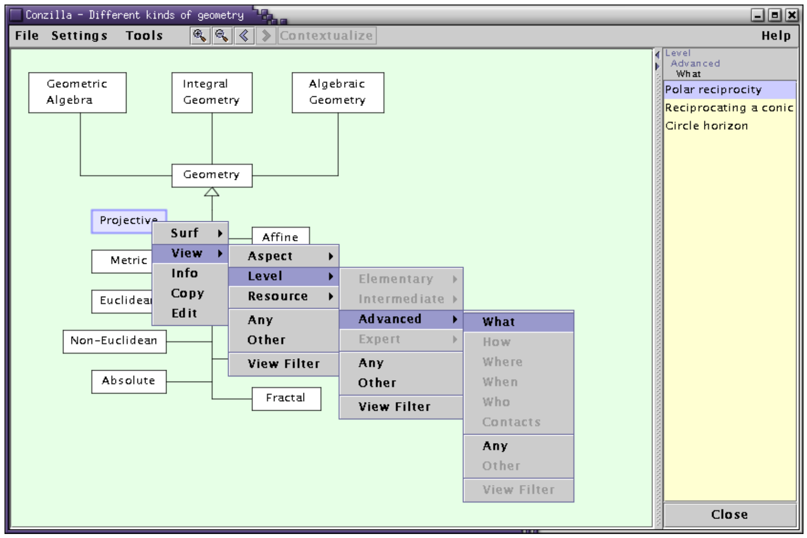

Figure 4.3: Sorting the content of *Projective Geometry,* looking at advanced-level content under the conceptual aspect (i.e. "what is projective geometry?"). Clicking on a link to the right, for example "Polar Reciprocity", opens up a video describing reciprocity.

such things as author, classification, keywords and other indexing and descriptive information according to the emerging *de facto* standard for markup of digital learning resources, IMS/IEEE LOM metadata[1]. This information can be seen by choosing the *Info* menu entry over an item. One biproduct of this markup is that all maps are easily translated to another language, as seen in figure 4.4 .

This summarizes the main ways of navigating amongst the concept maps.

4.2 Knowledge Manifolds

The underlying framework behind conceptual browsing is the idea of a *knowledge manifold*, which we will now present. First introduced by Naeve in [26], the knowledge manifold is a model for the construction, management and use of digital learning material that supports inquiry-based explorative learning.

A knowledge manifold consists of *knowledge patches*, each of which is maintained by an individual or group called the *knowledge gardener*. The patch contains reusable digital content in the form of *knowledge components*, such as multimedial or interactive content, structured and presented from the gardener's personal view on the world. The aim of a knowledge patch is to communicate this personal view in a digital form.

Different patches maintained by different gardeners often overlap, in the sense that they present personal views of the same concepts or reuse the same digital material. Together, the patches form a knowledge patchwork that constitute the knowledge manifold. This is analogous to the mathematical concept

[1]See `http://www.imsglobal.org`.

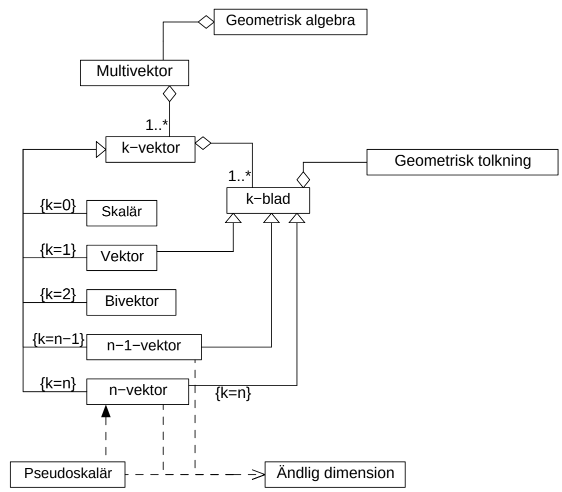

Figure 4.4: The vector map in Swedish.

of a manifold, which is a geometrical space described by overlapping maps in the same way as an atlas of the Earth does.

In contrast to the mathematical manifold, the patches in a knowledge manifold are usually not coherent. Different gardeners with differing views of the world may produce incompatible patches. It is one of the principles of the knowledge manifold design that such incompatibilities should be made explicit.

The concepts and the conceptual relationships in a knowledge patch are made explicit using UML diagrams like the ones used in this report. But the knowledge manifold framwork contains more principles, such as support for conceptual calibration between gardeners, competence profiling, component archives, personalization, multiple narration and more. See [26, 27].

The Conceptual Web is our name for the technical realization of the knowledge manifold framework which is taking form at CID. It contains a number of tools, the most important of which is the *concept browser*. The next section will present the philosophical background behind conceptual browsing.

4.3 The Concept Browser

As we have seen, conceptual browsing relies on metaphors from the present-day World Wide Web technologies for navigation. However, the Web in its current form is not well suited to the task of presenting a conceptually clear view of a knowledge manifold. The hyper-linked structure of the web presents the user with a totally fluid and dynamic relationship between context and content, which makes it hard to get an overview of the conceptual context within which the information is presented. As soon as you click on a hyperlink, you are helplessly transferred to a new and often unfamiliar context. This results in the all too well-known “surfing-sickness” on the web, which could be characterized as “Within what context am I viewing this, and how did I get here?”

In a learning context, the conceptual structure of the content is an essential part of the learning material. Losing the contextual information of the content means more than just "surfing-sickness". It means that you will not be able to contextually integrate the concepts that you are trying to learn, which is vitally important in order to achieve an understanding of any specific subject area. In traditional books, this is achieved by carefully constructing a linear sequence of material, which is often organized after some taxonomical scheme (which becomes chapters and sections). This creates a clear sense of conceptual context of the material in each section, at the expense of making it very difficult to reuse the material in new contexts, i.e., the context is clear but static. This conceptual clearness is sadly lacking in the Web of today, which can be characterized as having a dynamic but unclear context.

A concept browser is a navigational tool that presents a dynamic, hyperlinked view of a knowledge manifold, but tries to avoid the pitfalls of the current Web. The fundamental requirements on a concept browser, as described in [27] are:

1. Separate content from context.
2. Describe each separate context in terms of a concept map.
3. Assign an appropriate set of components as the content of a concept and/or conceptual relationship.
4. Filter the components through different aspects.
5. Label the components with a standardized data description (=metadata) scheme.
6. When a content component is a concept map, allow transforming it into a context by *contextualizing* it.

Conzilla now conforms to all of these requirements. The hyperlinked, conceptually oriented structure you build and browse using Conzilla is what we call the Conceptual Web. It is a form of hypertext, and lives on top of the current web, but with a fundamentally different topological metaphor, based on the framework of a knowledge manifold.

Chapter 5

A Conceptual Web of Mathematics

The maps presented in this report are part of a larger conceptual web of geometric algebra, which has been developed in order to experiment with UML as a conceptual language for mathematics. This chapter reports some of the lessons learned.

5.1 The Virtual Mathematics Exploratorium

One of the long-term goals of the ongoing work on UML and mathematics at CID is to build a large conceptual web of mathematics, intended as a virtual mathematics exploratorium containing pictures, animations, videos, presentations, interactive material, etc., all organized as a knowledge manifold, and thus containing a multitude of concept maps describing the subject and the material. More information can be found at `http://cid.nada.kth.se/il` and `http://amt.kth.se/matemagi`, and the background is given in [29].

5.2 UML as a Mathematical Language

It is now time to analyze the maps that were made of Geometric algebra. We start with listing the various kinds of diagrams seen so far, along with a discussion of what the corresponding structure would be in a mathematical text. Thus far, we have seen the following kinds of diagrams:

- *Class diagrams* such as the overview in figure 3.1, describing the relationships between concepts in terms of generalization hierarchies and containment. These diagrams correspond to two structures in mathematical text. In part, they play a *taxonomical* role, creating a structural overview. But they are also an important part of the *definition* of concepts, which are often defined in terms of specializations of other concepts, or containment of them. This is clearer in figure 3.14, where most multivector concepts in geometric algebra are actually defined.

- *Dependency diagrams*, which are a kind of UML object diagrams, show the logical dependencies between concepts. An example is figure 3.20. This kind of diagram shows dependencies in a way they are seldom, if ever, described in mathematical literature, but that is very helpful for anyone trying to digest the material.

- *Activity diagrams*, such as figure 3.17, can be used to describe *constructions*, which are very common in mathematical text (signaled by the use of imperative verb forms: "Take a vector a..." etc.). The diagrammatic form is very pleasing as it allows the suppression of uninteresting details (hiding them within pop-up descriptions or in maps that are linked to) without loosing contact with them when necessary. The construction can be delinearized and split up visually in separate parts. It can even be reused in other, larger constructions by simple hyperlinking.

- *Object-flow diagrams*, such as figure 3.3, can be used to describe *functions* and *mappings* of different kinds.

It should be noted that all the above diagrams contain only UML constructs. No extra terminology has been introduced. It is, however, very likely that some additional syntax would need to be introduced.

The kinds of mathematical structures not fully covered by the above diagrammatic techniques include

- Proofs. Though most proofs mainly rely on construction, and thus would be described by activity diagrams, there is definitely a need for expressing relations such as prerequisite/conclusion, logical equivalence, etc., as such relations are of very high conceptual value. Most non-trivial proofs would most certainly benefit from a diagrammatic representation.

- Commutativity diagrams. This is a kind of diagram which is already present in diagrammatic form in mathematical texts. Finding a way of presenting them in a UML-compatible way is still an open problem.

The possible added value of a UML-based approach to communicate mathematics, compared to using static text, include:

- Visual overviews, making understanding the context easier.

- Focus on explicit and clear conceptual relations rather than formulas or other, more low-level relations.

- Compactification of language, summarizing many complex relations in simple diagrams.

It should be clear from this summary and the examples that UML is a good candidate for a visual mathematical language. However, the precise advantages and shortcomings remain to be studied and analyzed. This problem is being examined in a project in Kista [insert description here].

It is also clear, however, that the features of conceptual browsing provide important additional value to non-interactive maps. These advantages of conceptual browsing will now be discussed.

5.3 Using Conceptual Browsing to Communicate Mathematics

What benefits can you expect from a knowledge manifold-based approach to communicate mathematics? There are several important properties of concept browsers that enhance the expressiveness of UML diagrams:

- A clear separation of the conceptual context and the conceptual content. Examples and explanations are given and suppressed at the users request, providing a much more flexible environment.
- The many forms of content. The need for a multitude of different kinds of examples of the same concept, focused on different aspects and on different levels of knowledge, can be satisfied.
- The high degree of connectivity. All occurences of a given concept are immediately available, providing many different views of the same concept. Concepts can link directly to more detailed explanations, providing quick connections in any context to the details of the concept, without having to clutter the presentation with those details.
- The feeling of exploration. As all concepts have natural and interesting links to related material, that is not necessarily directly related to the current curriculum, the user has the freedom to explore the surroundings of a concept without either getting lost or getting hindered by the system.
- The creation, reuse and integration of personal material in the form of maps and content. Creating personalized maps describing concepts is an important step in the understanding of a subject. Making it possible to reuse existing material, linking to existing content etc. would greatly enhance the value of this process.

As this summary hopefully indicates, conceptual browsing is a potentially central element in the design of an interactive learning environment with the features described in the introduction.

Chapter 6

Conclusions

The purpose of this thesis has been twofold: to show how geometric algebra can be used as a universal language for geometry and physics, and to show how conceptual modeling with Conzilla can be used to present a mathematical subject.

Geometric algebra still has ways to go before it is accepted as a fundamental part of any mathematician's or physicist's set of tools. It is clear that geometric algebra has the potential to revitalize analytic geometry not only for practitioners, but for students of the subject as well. With the help of geometric algebra, analytic geometry can be introduced gradually in a geometric fashion, without the need for matrix algebra to do calculations. The strong connection with geometric intuition, for both euclidean, non-euclidean and projective geometry, is perhaps the most important advantage of geometric algebra. over traditional approaches to analytic geometry.

Although presented in a fairly sketchy way, it should be clear that conceptual modeling and conceptual browsing offer interesting possiblities for the presentation of mathematics. It offers a conceptually oriented, non-formal entry point into the subject *at any level*, something that is hard to reproduce in another medium. We are taking conceptual modeling as a starting point for the design of the learning environments of the future. Many of the possibilities created by the conceptual modeling techniques remain to be explored.

Chapter 7

Acknowledgements

Many thanks to my girlfriend Anna, who supported me in the sometimes painful process of producing this thesis. I would also like to thank Ambjörn Naeve, my tutor and source of inspiration for geometric algebra and conceptual modeling, whose ideas are essential for the material presented here. Thanks also to Lasse Svensson, geometric algebra wizard, who has really helped me to understand some of the subtleties of the subject. My fellow student, Matthias Palmér has also helped strengthening the presentation of the subject.

Bibliography

[1] Biezunski, M., Bryan, M., Newcomb, S. R. (eds.), (1999), *ISO/IEC 13250:2000 Topic Maps: Information Technology – Document Description and Markup Languages*.

[2] Bråting, K., Mattsson J-E. (2001), *Problemlösning i olika lärandemiljöer*, Internal Report GyA0172, Uppsala: ILU.

[3] Buzan, T. (1995), *The MindMap book*. (2 ed.). London: BBC Books.

[4] Grevholm, B. (2001), *Teacher education in transition – the case in Sweden*, Department of Mathematics and Science, Kristianstad university.

[5] Doran, C., Hestenes, D., Sommen, F., Van Acker, N. (1993), *Lie groups as spin groups*, in *J. Math. Phys.* **34** (8) August 1993, pp. 3642-3669.

[6] Fisher, K. M., Wandersee, J. H., Wideman, G. (2000), *Enhancing cognitive skills for meaningful understanding of domain specific knowledge,* American Association for the Advancement of Science, Annual Meeting, Washington, DC, February 17-22, 2000. Online: `http://www.sci.sdsu.edu/CRMSE/Fisher_aaas2000.html`.

[7] Ford, K. M., Cañas, A. J., & Adams-Webber, J. (1992), *Participatory explanation*: *A new paradigm*? In *Proceedings of the Tenth European Conference on Artificial Intelligence Workshop on Expert Judgement, Human Error, and Intelligent Systems*, pp. 146-155.

[8] Grassmann, H. (1844), *Die Lineale Ausdehnungslehre* (Linear extension theory).

[9] Hestenes, D. (1987), *Spacetime algebra*, New York, Gordon and Breach.

[10] Hestenes, D. (1986), *A unified language for mathematics and physics*, in J. S. R. Chisholm and A. K. Common (eds.), *Clifford algebras and their Applications in Mathematical Physics*, pp.1-23, Dordrecht: D. Reidel.

[11] Hestenes, D. (1986), *New foundations for Classical Mechanics*, Dordrecht: D. Reidel.

[12] Hestenes, D. (1988), *Universal geometric algebra*, in *Simon Stevin*, **63**, pp. 253-274.

[13] Hestenes, D. (1991), *The design of linear algebra and geometry*, in *Acta Appl. Math.* **23**, pp. 65-93.

[14] Hestenes, D., and Sobczyk, G. (1984), *Clifford Algebra to Geometric Calculus*, Dordrecht: D. Reidel.

[15] Hestenes, D. (1968), *Multivector Functions*, in *J. Math. Anal. and Appl.* **24**, pp. 313-325.

[16] Hestenes, D. (1968), *Multivector Calculus*, in *J. Math. Anal. and Appl.* **24**, pp. 467-473.

[17] Hestenes, D. (1991), *Mathematical Viruses*, in A. Micali, R. Boudet, J Helmstetter (eds.)*, Clifford Algebras and their Applications in Mathematical Physics*, pp. 3-16, Dordrecht/Boston: Kluwer.

[18] Hestenes, D. (1991), *Projective Geometry with Clifford algebra*, in *Acta Appl. Math.* **23**, pp. 25-63.

[19] Hestenes, D. (1993), *Differential forms in Geometric Calculus*, in F.Brackx *et al* (eds.)*, Clifford Algebras and their Applications in Mathematical Physics*, pp 269-285, Dordrecht/Boston: Kluwer.

[20] Hestenes, D. (1993), *Hamiltonian Mechanics with Geometric Calculus*, in Z. Oziewicz *et al* (eds.), *Spinors, Twistors, Clifford Algebras and Quantum Deformations*, pp. 203-214, Dordrecht/Boston: Kluwer.

[21] Hestenes, D. (1996), *Grassmanns vision*, in Gert Schubring (ed.), *Hermann Gunther Grassmann (1809-1877): Visionary Mathematician, Scientist and Neohumanist Scholar*, pp. 191-201, Dordrecht/Boston: Kluwer.

[22] Hofstadter D. & The Fluid Analogies Research Group (1995), *Fluid Concepts & Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, New York: Basic Books.

[23] Jonassen, D.H., Beissner, K., & Yacci, M.A. (1993), *Structural knowledge*: *Techniques for conveying*, *assessing*, *and acquiring structural knowledge*. Hillsdale, NJ: Lawrence Erlbaum Associates.

[24] Lawson, M. J. (1994), *Concept Mapping*. In T. Husén & T. N. Postlethwaite (eds.), *The international encyclopedia of education* (2nd ed., Vol. 2, pp. 1026-1031). Oxford: Elsevier Science.

[25] Linde, R., Naeve, A., Skantz, K., Westerlund, B., Winberg, F., Åsvärn, K. and Olausson, K. (1997), *Kunskapens Trädgård* (CID-18), Technical report TRITA-NA-D9709, Stockholm: KTH. `http://cid.nada.kth.se/pdf/cid_18.pdf`

[26] Naeve, A. (1997), *The Garden of Knowledge as a Knowledge Manifold – a conceptual framework for computer supported subjective education*, Centre for user-oriented Information technology Design (CID-17), TRITA-NA-D9708, Stockholm: KTH. `http://cid.nada.kth.se/pdf/cid_17.pdf`

[27] Naeve, A. (1999), *Conceptual navigation and multiple narration in a knowledge manifold*, Centre for user-oriented Information technology Design (CID-52), TRITA-NA-D9910, Stockholm: KTH. `http://cid.nada.kth.se/pdf/cid_52.pdf`

[28] Naeve, A. (2001), *Begreppsmodellering och matematik*, Centre for user-oriented Information technology Design (CID-109), TRITA-NA-D0103, Stockholm: KTH. `http://cid.nada.kth.se/pdf/cid_109.pdf`

[29] Naeve, A. (2001), *The Work of Ambjörn Naeve within the Field of Mathematical Education Reform*, Centre for user-oriented Information technology Design (CID-110), TRITA-NA-D0104, Stockholm: KTH. `http://www.amt.kth.se/projekt/matemagi/mathemathics_educational_reform.doc`

[30] Naeve, A. & Svensson, L. (2001), *Geo-Metric-Affine-Projective Unification*, in Sommer (ed.), *Geometric Computing with Clifford Algebras*, Ch. 5, pp.105-126, Springer.

[31] Nilsson, M. and Palmér, M. (1999), *Conzilla Towards a Concept Browser*. Technical report TRITA-NA-D9911, Stockholm: KTH. `http://cid.nada.kth.se/pdf/cid_51.pdf`

[32] Nilsson, M (2000), *The Conzilla Design – The definitive reference*, `http://conzilla.sourceforge.net/doc/conzilla-design/conzilla-design.html`

[33] Novak, J. D. (1991), *Clarify with concept maps*: *A tool for students and teachers alike*. The Science Teacher, 58(7), pp. 45-49.

[34] Novak, J. D. (1993), *How do we learn our lesson*? : *Taking students through the process*. The Science Teacher, 60(3), pp. 50-55.

[35] Novak, J. D. (1985), *Metalearning and metaknowledge strategies to help students learn how to learn.* In West, L., Pines, A. (eds.) *Cognitive structure and conceptual change*, pp. 189-207, New York: Academic Press.

[36] Novak, J. D., *The Theory Underlying Concept Maps and How To Construct Them.* Available online: `http://cmap.coginst.uwf.edu/info/`.

[37] OMG, The Object Management Group (2000), *Unified Modeling Language* (*UML*) 1.3 *specification*, OMG specification **formal/00-03-01**, `http://www.omg.org/`.

[38] Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W. (1991), *Object-Oriented Modeling and Design*, New Jersey: Prentice Hall.

[39] Rumbaugh, J., Jacobson, I., Booch, G. (1999), *The Unified Modeling Language Reference Manual*, Addison Wesley Longman Inc.