What is a Concept?

Joseph Goguen
University of California at San Diego, Dept. Computer Science & Engineering
9500 Gilman Drive, La Jolla CA 92093–0114 USA

Abstract. The lattice of theories of Sowa and the formal concept analysis of Wille each address certain formal aspects of concepts, though for different purposes and with different technical apparatus. Each is successful in part because it abstracts away from many difficulties of living human concepts. Among these difficulties are vagueness, ambiguity, flexibility, context dependence, and evolution. The purpose of this paper is first, to explore the nature of these difficulties, by drawing on ideas from contemporary cognitive science, sociology, computer science, and logic. Secondly, the paper suggests approaches for dealing with these difficulties, again drawing on diverse literatures, particularly ideas of Peirce and Latour. The main technical contribution is a unification of several formal theories of concepts, including the geometrical conceptual spaces of Gärdenfors, the symbolic conceptual spaces of Fauconnier, the information flow of Barwise and Seligman, the formal concept analysis of Wille, the lattice of theories of Sowa, and the conceptual integration of Fauconnier and Turner; this unification works over any formal logic at all, or even multiple logics. A number of examples are given illustrating the main new ideas. A final section draws implications for future research. One motivation is that better ways for computers to integrate and process concepts under various forms of heterogeneity, would help with many important applications, including database systems, search engines, ontologies, and making the web more semantic.

1 Introduction

This paper develops a theory of concepts, called the Unified Concept Theory (**UCT**), that integrates several approaches, including John Sowa's lattice of theories [61] (abbreviated **LOT**), the formal concept analysis (**FCA**) of Rudolf Wille [15], the information flow (**IF**) of Jon Barwise and Jerry Seligman [3], Gilles Fauconnier's logic-based mental spaces [13], Peter Gärdenfors geometry-based conceptual spaces [16], the conceptual integration (**CI**, also called blending) of Fauconnier and Turner, and some database and ontology approaches, in a way that respects their cognitive and social aspects, as well as their formal aspects. UCT does not claim to be an empirical theory of concepts, but rather a mathematical formalism that has applications to formalizing, generalizing, and unifying theories

in cognitive linguistics, psychology, and other areas, as well as to various kinds of engineering and art; it has implementations and is sound engineering as well as sound mathematics.

Although the question in the title does not get a final answer, I hope that readers will enjoy the trip through some perhaps exotic seeming countries that lie on the borders between the sciences and the humanities, and return to their home disciplines with useful insights, such as a sense of the limitations of disciplinary boundaries, as well as with some new formal tools.

Section 2 illustrates several of the difficulties with concepts mentioned above, through a case study of the concept "category" in the sense of contemporary mathematics; Section 2.1 outlines some sociology of this concept, and Section 2.2 discusses some methodological issues, drawing particularly on theories of Bruno Latour. Section 3 reviews research on concepts from several areas of cognitive semantics, especially cognitive psychology and cognitive linguistics. This section provides a technical reconciliation of two different notions of "conceptual space" in cognitive semantics, Fauconnier's symbolic notion and Gärdenfors' geometric notion; it also gives an extended illustration of the use of this new theory, and an example using fuzzy convex sets to represent concepts. In addition, it suggests an approach to the symbol grounding problem, which asks how abstract symbols can refer to real entities. Section 4 reviews research from sociology, concerning how concepts are used in conversation and other forms of interaction; values are also discussed. Section 5 explains the UCT approach to semantic integration, using tools from category theory to unify LOT, FCA, IF and CI, and generalize them to arbitrary logics, based on the theory of institutions; an ontology alignment example is given using these methods. Finally, Section 6 considers ways to reconcile the cognitive, social and technical, and draws some conclusions for research on formal aspects of concepts and their computer implementations. Unfortunately, each section is rather condensed, but examples illustrate several main ideas, and a substantial bibliography is provided.

Acknowledgements I thank John Sowa, Mary Keeler, Rod Burstall, Erick von Schweber, Linda von Schweber, and Lianne Gabora for inspiration, constructive suggestions, references, and/or helpful remarks, though of course any remaining errors are my own. This paper is based on work supported by the National Science Foundation under Grant No. 9901002, SEEK, Science Environment for Ecological Knowledge.

2 A Case Study: "Category"

Category theory has become important in many areas of modern mathematics and computer science. This section presents a case study, exploring the mathematicians' concept of category and its applications, in order to help us understand how concepts work in practice, and in particular, how they can evolve, and how they can interact with other concepts within their broader contexts. This socio-historical situation is made more complex and interesting by the fact that the mathematical notion of category formalizes a certain kind of mathematical concept; Section 5 makes considerable use of categories in this sense.

The word "category" has a complex history, in which it manifested as many different concepts, several of which are of interest for this paper. Aristotle introduced the term into philosophy, for the highest level kinds of being, based on basic grammatical categories of classical Greek. The idea was later taken up and modified by others, including the medieval scholastics, and perhaps most famously, by Immanual Kant. Later Friedrich Hegel and Charles Sanders Peirce criticized Kant, and put forth their own well known triads, the three elements of which are commonly called "categories."

In the early 1940s, Samuel Eilenberg and Saunders Mac Lane developed their theory of categories [12] to solve certain then urgent problems in algebraic topology[1]. They borrowed the word from Kant, but their concept is very different from anything in Kant's philosophy. They gave a semantic formalization of the notion of a mathematical structure: the category for a structure contains all mathematical objects having that structure; but the real innovation of category theory is that the category also contains all structure preserving "morphisms" between its objects, along with an operation of composition for these morphisms. This concept, which is made explicit in Definition 1 below, allows an amazing amount of mathematics to be done in a very abstract way, e.g., see [51, 57]. For example, one can define notions of "isomorphism," "product," "quotient," and "subobject," at the abstract categorical level, and then see what form they take (if they exist) in particular categories; many general theorems can also be proved about such notions, which then apply to an enormous range of concrete situations. Even more can be done

[1] There were several different homology theories, some of which seemed equivalent to others, but it was very unclear how to define an appropriate notion of equivalence. Eilenberg and Mac Lane characterized homology theories as functors from the category of topological spaces to the category of groups, and defined two such theories to be equivalent if there was a natural equivalence between their functors.

once functors are introduced, and that is not the end of the story by any means. Thus "category" is not an isolated concept, but part of a large network of inter-related concepts, called "category theory."

The phrase "all mathematical objects having that structure" raises foundational issues, because Zermelo-Fraenkel (**ZF**) set theory, the most commonly used foundation for mathematics, does not allow such huge collections. However, Gödel-Bernays set theory does allow them under its notion of "classes," which it distinguishes from the small "sets." Further foundational issues arise when one considers the category $\mathbb{C}at$ of all categories. This has also been solved, by so-called universes of sets, along lines developed by either Alexander Grothendieck or Solomon Feferman. Some of these issues are discussed in [51]. This illustrates how a concept can generate difficulties when pushed to its limits, and then stimulate further research to resolve those difficulties[2].

Such seemingly esoteric issues can have significant social consequences. For example, in the 1970s, I had several perfectly good papers rejected, due to referee reports claiming that the mathematics used, namely category theory, lacked adequate foundations! Such incidents no longer occur, but there is lingering disquiet due to the fact that foundations more sophisticated than standard ZF set theory are required.

Definition 1: A **category** $\mathbb{C}$ consists of: a class, denoted $|\mathbb{C}|$, of **objects**; for each pair A, B of objects a set $\mathbb{C}(A, B)$ of **morphisms** from A to B; for each object A, a morphism 1_A in $\mathbb{C}(A, A)$, called the **identity** at A; and for each three objects A, B, C, an operation called **composition**, $\mathbb{C}(A, B) \times \mathbb{C}(B, C) \to \mathbb{C}(A, C)$, denoted "**;**" such that $f; (g; h) = (f; g); h$ and $f; 1_A = f$ and $1_A; g = g$ whenever these compositions are defined. Write $f\colon A \to B$ when $f \in \mathbb{C}(A, B)$ and call A the **source** and B the **target** of f. □

Example 2: Perhaps the easiest example to explain is the category $\mathbb{S}et$ of sets. Its objects are sets, its morphisms $f\colon A \to B$ are functions, 1_A is the identity function on the set A, and composition is as usual for functions. Another familiar example is the category of Euclidean spaces, with $\mathbb{R}^n$ for each natural number n as objects, with morphisms $f\colon \mathbb{R}^m \to \mathbb{R}^n$ the $n \times m$ real matrices, with identity morphisms the $n \times n$ diagonal matrices with all 1s on the diagonal, and with matrix multiplication as composition. □

[2] An example that followed a somewhat different trajectory is Gottlob Frege's formalization of Cantor's informal set theory, which was shown inconsistent by Russell's paradox and then supplanted by theories with a more restricted notion of set; see [39] for details.

There are also other ways of defining the category concept. For example, one of these involves only morphisms, in such a way that objects can be recovered from the identity morphisms. The precise senses in which the various definitions are equivalent can be a bit subtle. Moreover, a number of weakenings of the category concept are sometimes useful, such as semi-categories and sesqui-categories.

A powerful approach to understanding the category concept is to look at how it is used in practice. It turns out that different communities use it in different ways. Following the pioneering work of Grothendieck, category theory has become the language of modern algebraic geometry, and it is also much used in modern algebraic topology and differential geometry, as well as other areas, including abstract algebra. Professional category theorists generally do rather esoteric research within category theory itself, not all of which is likely to be useful elsewhere in the short term. Many mathematicians look down on category theory, seeing it more as a language or a tool for doing mathematics, rather than an established area of mathematics. Within theoretical computer science, category theory has been used to construct and study models of the lambda calculus and of type theories, to unify the study of various abstract machines, and in theories of concurrency, among many other places. It is also often used heuristically, to help find good definitions, theorems, and research directions, and to test the adequacy of existing definitions, theorems and directions; such uses have an aesthetic character. I tried to formulate general principles for using category theory this way in "A Categorical Manifesto" [21].

Let us consider two results of this approach in my own experience: initial algebra semantics [34] and the theory of institutions [28]. As discussed in [21], initial algebras, and more generally initial models, formalize, generalize and simplify the classical Herbrand Universe construction, and special cases include abstract syntax, induction, abstract data types, domain equations, and model theoretic semantics for functional, logic and constraint programming, as well as all of their combinations. The first application was an algebraic theory of abstract data types; it spawned a large literature, including several textbooks. But its applications to logic programming and other areas attracted much less attention, probably because these areas already had well established semantic frameworks, whereas abstract data types did not; moreover, resistance to high levels of abstraction is rather common in computer science.

Institutions [28, 53] are an abstraction of the notion of "a logic," originally developed to support powerful modularization facilities over any

logic; see Section 5 for more detail. The modularization ideas were developed for the Clear specification language [7] before institutions, but institutions allow an elegant and very general semantics to be given for them [28, 35]; these facilities include parameterized modules that can be instantiated with other modules. This work influenced the module systems of the programming languages Ada, C++, ML, LOTOS, and a large number of lesser known specification languages[3]. Among the programming languages, the Standard ML (or SML) variant of ML comes closest to implementing the full power of these modularization facilities, while the specification languages mentioned in footnote 3 all fully implement it. ML is a functional programming language with powerful imperative features (including assignment and procedures)[4].

2.1 Sociology of Concepts

The evolution of concepts called "category" from Aristotle's ontological interpretation of syntax, through a multitude of later philosophies, into mathematics and then computer science, has been long and complex, with many surprising twists, and it is clear that the mathematical concept differs from Aristotle's original, as well as from the intermediate philosophical concepts. This evolution has been shaped by the particular goals and values of the communities involved, and views of what might be important gaps in then current philosophical, mathematical, or technical systems, have been especially important; despite the differences, there is a common conceptual theme of capturing essential similarities at a very abstract level.

The heuristic use of the category theoretic concepts for guiding research in computer science suggested in [21] is not so different from that of Aristotle and later ordinary language philosophers, who sought to improve how we think by clarifying how we use language, and in particular by preventing "category errors," in which an item is used in an inappropriate context. However, the approach of [21] is much more pragmatic than the essentially normative approach of Aristotle and the ordinary language philosophers. Many theoretical computer scientists have reported finding [21] helpful, and it has a fairly high citation index.

[3] These include CafeOBJ [11], Maude [8], OBJ3 [36], BOBJ [31], and CASL [54].

[4] Rod Burstall reports that much of the design work for the module system of the SML version of ML was done in discussions he had with David MacQueen. Perhaps the most potentially useful ideas still not in SML are the renamings of Clear [28] and the default views of OBJ3 [36].

On the other hand, beyond its initial success with abstract data types, the initial model approach, despite its elegance and generality, has not been taken up to any great extent. In particular, the programming languages community continues to use more concrete approaches, such as fixed points and abstract operational semantics, presumably because they are closer to implementation issues.

The extent to which a concept fits the current paradigm (in the general sense of [44]) of a community is crucial to its success. For example, Rod Burstall and I expected to see applications of institutions and parameterized theories to knowledge representation (**KR**), but that did not happen. Perhaps the KR community was not prepared to deal with the abstractness of category theory, due to the difficulty of learning the concepts and how to apply them effectively. Although we still believe this is a promising area, we were much more successful with the specification of software systems, and the most notable impact of our work has been on programming languages.

Let us now summarize some of what we have seen. Concepts can evolve over very long periods of time, but can also change rather quickly, and the results can be surprising, e.g., the "categories" of Aristotle, Kant, Hegel, and Peirce. Multiple inconsistent versions of a concept can flourish at the same time, and controversies can rage about which is correct. Concepts can become problematic when pushed further than originally intended, requiring the invention of further concepts to maintain their life, as when category theory required new foundations for mathematics. Otherwise, the extended concept may be modified or even abandoned, as with Frege's logic. New concepts can also fail to be taken up, e.g., initial model semantics and the security model mentioned in footnote 5.

The same concept can be used very differently in different communities: the uses of categories by working mathematicians, category theorists, and computer scientists are very different, so much so that, despite the mathematical definitions being identical, one might question whether the concepts should be regarded as identical. These differences include being embedded in different networks of other concepts, other values, and other patterns of use; to a large extent, the values of the communities determine the rest. Moreover, these differences can lead to serious mutual misunderstandings between communities. For example, mathematicians value "deep results," which in practice means hard proofs of theorems that fit well within established areas, whereas computer scientists must be concerned with practical issues such as efficiency, cost, and user satisfaction.

These differences make it difficult for category theorists and computer scientists to communicate, as I have often found in my own experience.

In retrospect, it should not be so surprising that our research using category theory had its biggest impact in programming languages rather than formal logic or mathematics, because of the great practical demand for powerful modularization in contemporary programming practice. To a great extent, science today is "industrial science" or "entrepreneurial science," driven by the goals of large organizations and the pressures of economics, rather than a desire for simple general abstract concepts that unify diverse areas.

It seems likely that the above observations about the "category" concept also hold for many other concepts used in scientific and technical research, but further case studies and comparative work would be needed to establish the scope and limitations of these observations. There is also much more in the category theory literature that could be considered, such as the theory of topoi [4].

2.2 Methodology

Because much of the intended audience for this paper is unlikely to be very interested in social science methodology, this subsection is deliberately quite brief. Science Studies is an eclectic new field that studies science and technology in its social context, drawing on history, philosophy and anthropology, as well as sociology; its emphasis is qualitative understanding rather than quantitative reduction, although quantitative methods are not excluded. The "strong program" of David Bloor and others [5] calls for a "symmetric" approach, which disallows explaining "true" facts rationally and "false" facts socially; it is constructionist but not anti-realist. Donald McKenzie has a fascinating study of the sociology of mechanized proof [52][5]. The actor network theory of Bruno Latour [49, 48] and others, emphasizing the interconnections of human and non-human "actants" required to produce scientific knowledge and technology, grew out of the strong program. Symbolic interactionism (e.g., [6]) is concerned with how meaning arises out of interaction through interpretation;

[5] While reading this book, I was startled to encounter a sociology of the concept "security" which (along with many other things) explained that the early 1980s Goguen-Meseguer security model was largely ignored at the time, due to pressure from the National Security Agency, which was promoting a different (and inferior) model. This illustrates how social issues can interact with formal concepts.

it was strongly influenced by the pragmatism of Peirce[6]. Ethnomethodology [17], a more radical outgrowth of symbolic interactionism, provides useful guidelines when the analyst is a member of the group being studied; Eric Livingston has done an important study of mathematics in this tradition [50]. Among more conventional approaches are the well known historical theories of Thomas Kuhn [44] and the grounded theory of Anselm Strauss and others [19]. Some related issues are discussed in Section 4.

3 Cognitive Science

This section is a short survey of work in cognitive semantics, focused on the notion of "concept." In a series of papers that are a foundation for contemporary cognitive semantics, Eleanor Rosch designed, performed, and carefully analyzed innovative experiments, resulting in a theory of human concepts that differs greatly from the Aristotelian tradition of giving necessary and sufficient conditions, based on properties. Rosch showed that concepts exhibit *prototype effects*, e.g., degrees of membership that correlate with similarity to a central member. Moreover, she found that there are what she called *basic level concepts*, which tend to occur in the middle of concept hierarchies, to be perceived as gestalts, to have the most associated knowledge, the shortest names, and to be the easiest to learn. Expositions in [46, 47, 37] give a concise summary of research of Rosch and others on conceptual categories. This work served as a foundation for later work on metaphor by George Lakoff and others. One significant finding is that many metaphors come in families, called *basic image schemas*, that share a common sensory-motor pattern. For example, MORE IS UP is grounded in our everyday experience that higher piles contain more dirt, or more books, etc. Metaphors based on this image schema are very common, e.g., "That raised his prestige." or "This is a high stakes game."

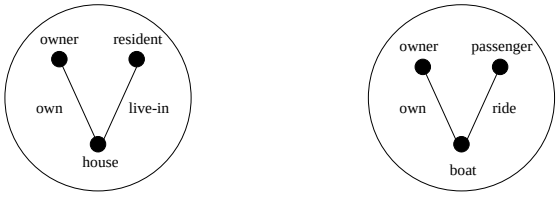

Fig. 1. Two Simple Conceptual Spaces

[6] Interpretation in this tradition is understood in essentially the sense of Peirce's thirdness [56].

Fauconnier's *mental spaces* [13] (called *conceptual spaces* in [14]) do not attempt to formalize concepts, but instead formalize the important idea that concepts are used in clusters of related concepts. This formalization uses a very simple special case of logic, consisting of individual constants, and assertions that certain relations (mostly binary) hold among certain of those individuals; it is remarkable how much natural language semantics can be encoded with this framework (see [13, 14]). Figure 1 shows two simple conceptual spaces, the first for "house" and the second for "boat." These do not give all possible information about these concepts, but only the minimal amount needed for a particular application, which is discussed in Section 5. The "dots" represent the individual constants, and the lines represent true instances of relations among those individuals. Thus, the leftmost line asserts `own(owner, house)`, which means that the relation `own` holds between these two constants.

Goguen [23] proposed *algebraic theories* to handle additional features that are important for user interface design. In particular, many signs are complex, i.e., they have parts, and these parts can only be put together in certain ways, e.g., consider the words in a sentence, or the visual constituents of the diagram in Figure 1. Algebraic theories have *constructor functions* build complex signs from simpler signs; for example, a window constructor could have arguments for a scrollbar, label, and content. Then one can write $W1 = window(SB1, L1, C1)$; there could also be additional arguments for color, position, and other details of how these parts are put together to constitute a particular window. This approach conveys information about the relations between parts and wholes in a much more explicit and useful way than just saying `has-a(window, scrollbar)`, and it also seems to avoid many problems that plague the `has-a` relation and its axiomatizations in formal mereology[7] (see [58] for some discussion of these problems).

Algebraic theories also have *sorts* (often called "types"), which serve to restrict the structure of signs: each individual has a sort, and each relation and function has restrictions on the sorts that its arguments may take. For example, `owner` and `resident` might have sort `Person` while `house` has sort `Object` and `own` takes arguments only of sorts `Person, Object`. Allowing sorts to have *subsorts* provides a more effective way to support inheritance than the traditional `is-a` relation. For example, `Person` might have a subsort `Adult`. *Order sorted algebra* [32] provides a mathematical foundation for this approach, integrating inheritance with whole/part structure (using constructor functions instead of the `has-a`

[7] Mereology is the study of whole/part relations.

relation) in an elegant and computationally tractable algebraic formalism that also captures some subtle relations between the two[8].

Algebraic theories may have conditional equations as axioms (though arbitrary first order sentences could be used if needed) to further constrain the space of possible signs; for example, certain houses might restrict their residents to be adults. Fauconnier's mental spaces are the special case of order sorted algebraic theories with no functions, no sorts or subsorts, and with only atomic relation instances as axioms. There is extensive experience applying algebraic theories to the specification and verification of computer-based systems (e.g., [36, 31, 11]). Hidden algebra [31] provides additional features that handle dynamic systems with states, which are central to computer-based systems. *Semiotic spaces* extend algebraic theories by adding priority relations on sorts and constructors, which express semiotic information that is vital for applications to user interface design [23, 24]. Semiotic spaces are also called *sign systems*, because they define systems of signs, not just single signs, e.g., all possible displays on a particular digital clock, or a particular cell phone.

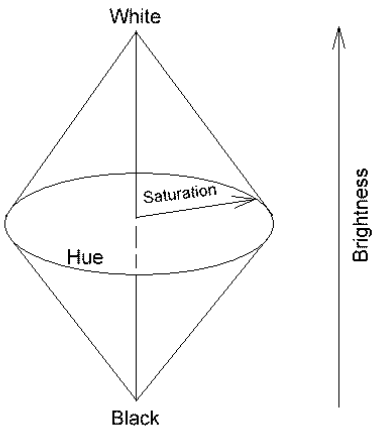

Fig. 2. Human Color Manifold

Gärdenfors [16] proposes a notion of "conceptual space" that is very different from that of Fauconnier, since it is based on geometry rather than logic. One of the most intriguing ideas in [16] is that all conceptual spaces are *convex*[9]; there is also a nice application of Voronoi tessellation

[8] E.g., the monotonicity condition on overloaded operations with respect to subsorts of argument sorts [32].

[9] A subset of Euclidean space is *convex* if the straight line between any two points inside the subset also lies inside the subset; this generalizes to non-Euclidean man-

to a family of concepts defined by multiple prototypes[10]. Although [16] aims to reconcile its geometric conceptual spaces with symbolic representations like those of Fauconnier, it does not provide a unified framework. However, such a unification can be done in two relatively straightforward steps. The first step is to introduce *models* in addition to logical theories, where a model provides a set of instances for each sort, a function for each function symbol, and a relation for each relation symbol; since we are interested in the models that satisfy the axioms in the theory, an explicit notion of satisfaction is also needed[11]. The second step is to fix the interpretations in models of certain sorts to be particular geometrical spaces (the term "standard model" is often used in logic for such a fixed interpretation). Sorts with fixed interpretation give a partial solution to the *symbol grounding problem*[12] [38], while those without fixed interpretation are open to arbitrary interpretations. This way of combining logic with concrete models can be applied to nearly any logic[13].

For example, a sort `color` might be interpreted as the set of points in a fixed 3D manifold representing human color space, coordinatized by hue, saturation and brightness values, as in Figure 2, which is shaped like a "spindle," i.e., two cones with a common base, one upside down. This provides a precise framework within which one can reason about properties that involve colors, as in the following:

Example 3: A suggestive example in [16] concerns the (English) terms used to describe human skin tones (red, black, white, yellow, etc.), which have a very different meaning in that context than e.g., in a context of describing fabrics. Gärdenfors explains this shift of meaning by embedding the space of human skin tones within the larger color manifold, and showing that in this space, the standard regions for the given color names

ifolds by using geodesics instead of straight lines, but it is unclear what convexity means for arbitrary topological spaces.

[10] Given a set of n "prototypical points" in a convex space, the Voronoi tessellation divides the space into n convex regions, each consisting of all those points that are closest to one of the prototypes.

[11] These items are the usual ingredients of a logic, but Section 5 argues that the extra ingredients provided by institutions, e.g., variable context and context morphisms, are also needed.

[12] This is the problem in classic logic-based AI, of how abstract computational symbols can be made to refer to entities in the real world. The approach in this paragraph is partial because it only shows how abstract symbols can refer to geometrical models of real world entities.

[13] More precisely, to any *concrete institution*, which is a specialization of the notion of institution described in Section 5, in which symbols have sorts and models are many-sorted sets; see e.g. [53] for details.

are the closest fits to the corresponding skin tone names. Technically, it is actually better to view the spaces as related by a canonical projection from the spindle to the subspace, because Gärdenfors' assumption that all such spaces are convex guarantees that such a canonical projection exists. Gärdenfors does not give a formal treatment of the color terms themselves, but we can view them as belonging to a mental space in the sense of Fauconnier, and view their relationship as a classification between the space of skin color terms and the space of all colors; note that many colors will not have any corresponding skin tone name. □

Unified Concept Theory (see Section 5) uses the term **frame** for a combination of a context, a symbolic space, a geometrical space (or spaces), and a relation between them for the given context. Thus, there are two frames in the above example, and the embedding and projection of color spaces mentioned above can be seen as *frame morphisms*.

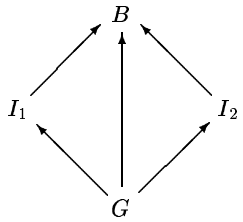

Fig. 3. Information Integration over a Shared Subobject

The most important recent development of ideas in the tradition of Rosch, Lakoff, and Fauconnier is *conceptual blending*, claimed in [14] to be a (previously unrecognized) fundamental cognitive operation, which combines different conceptual spaces into a unified whole. The simplest case is illustrated in Figure 3, where for example I_1, I_2 might be mental spaces for "house" and "boat" (as in Figure 1), with G containing so-called "generic" elements such that the maps $G \to I_i$ indicate which individuals should be identified. Some "optimality principles" are given in Chapter 16 of [14] for judging the quality of blends, and hence determining which blends are most suitable, although these distillations of common sense are far from being formal. In contrast to the categorical notion of colimit[14], blends are not determined uniquely up to isomorphism; for example, B could be "houseboat," or "boathouse," or some other combination of the two input spaces I_i (see [30] for a detailed discussion of this example).

[14] Colimits abstractly capture the notion of "putting together" objects to form larger objects, in a way that takes account of shared substructures; see [21] for an intuitive discussion.

Blending theory as in [14] also refines the metaphor theory of Lakoff, by proposing that a metaphorical mapping from I_1 to I_2 is really a kind of "side effect" of a blend B of I_2 and I_2, since a metaphor really constructs a new space in which only certain parts of I_1 and I_2 appear, and in which some new structure found in neither I_1 nor I_2 may also appear; the usual formulation of metaphor as a "cross space mapping" $m\colon\ I_1 \to I_2$ is the reflection of the identifications that are made in B, i.e., if i_1, i_2 are constants in I_1, I_2 respectively, that map to the same constant in B, then we set $m(i_1) = i_2$.

Example 4: In the metaphor "the sun is a king," the constants "sun" and "king" from the input spaces are identified in the blend, so "sun" maps to "king," but the hydrogen in the sun is (probably) not mapped up or across, nor is that fact that kings may collect taxes. But if we add the clause "the corona is his crown," then another element is added to the cross space map. □

Example 5: One of the most striking examples in [14], called "the Buddhist monk," is not a metaphor. It can be set up as follows: A Buddhist monk makes a pilgrimage to a sacred mountain, leaving at dawn, reaching the summit at dusk, spending the night there in meditation, then departing at dawn the next day, and arriving at the base at dusk. A question is then posed: is there a time such that the ascending monk and the descending monk are at the same place at that time? This question calls forth a blend in which the two days are merged into one, but the one monk is split into two! The reasoning needed to answer the question cannot be done in a logic-based blend space, because some geometrical structures are needed to model the path of the monk(s), in addition to the individuals and relations that are given logically. The table below shows the semiotic spaces for the first and second day in its first and second columns, respectively; notice the explicitly given types, which are needed to constrain possible interpretations of the declared elements.

$Time = [6, 18]$	$Time = [6, 18]$
$Loc = [0, 10]$	$Loc = [0, 10]$
$m\colon Time \to Loc$	$m\colon Time \to Loc$
$m(6) = 0$	$m(6) = 10$
$m(18) = 10$	$m(18) = 0$
$(\forall\ t, t'\colon Time)\ t > t' \Rightarrow$ $m(t) > m(t')$	$(\forall\ t, t'\colon Time)\ t > t' \Rightarrow$ $m(t) < m(t')$

The first two lines of each theory are type definitions. A model for the first day will interpret $Time$ as the fixed interval [6,18] (for dusk and dawn,

in hours); it will also interpret Loc as another fixed interval, [0,10] (for the base and summit locations, in miles). Then m is interpreted as some continuous function $[6, 18] \to [0, 10]$, giving the monk's distance along the path as a function of time. The key axiom is the last one, a monotonicity condition, which asserts that the monk always makes progress along the path, though without saying how quickly or slowly. Each such function m corresponds to a different model of the theory. The theory for the second day is the same except for the last three axioms, which assert that the monk starts at the top and always descends until reaching the bottom. Notice that the types $Time$ and Loc must be given exactly the same interpretations on the two days, but the possible paths are necessarily different. The blended theory is shown in the array below, in which m indicates the monk's locations on the first day and m' on the second day.

$Time = [6, 18]$	$Loc = [0, 10]$
$m, m', m^* : Time \to Loc$	
$t^* : Time$	
$m(6) = 0$	$m(18) = 10$
$m'(6) = 10$	$m'(18) = 0$
$(\forall\, t, t' : Time)\ t > t' \Rightarrow$	$m(t) > m(t')$
$(\forall\, t, t' : Time)\ t > t' \Rightarrow$	$m'(t) < m'(t')$
$m^*(t) = m'(t) - m(t)$	
$m^*(t^*) = 0$	

To answer the question, we have to solve the equation $m(t) = m'(t)$. If we let t^* denote a solution and let $m^* = m' - m$, then the key "emergent" structure added to the blend space is $m^*(t^*) = 0$, since this allows us to apply (the strict monotone version of) the Intermediate Value Theorem, which says that a strict monotone continuous function which takes values a and b with $a \neq b$ necessarily takes every value between a and b exactly once. In this case, m^* is strict monotone decreasing, $m^*(6) = 10$ and $m^*(18) = -10$, so there is a unique time t^* such that $m^*(t^*) = 0$.

It is interesting to notice that if we weaken the monotonicity axioms to become non-strict, so that the monk may stop and enjoy the view for a time, as formally expressed for the first day by the axiom

$(\forall\, t, t' : Time)\ t > t' \Rightarrow m(t) \geq m(t')$

then (by another version of the Intermediate Value Theorem) the monk can meet himself on the path for any fixed closed proper subinterval $[a, b]$ of [6,18] (i.e., with $6 \leq a \leq b \leq 18$ with either $a \neq 6$ or $b \neq 18$). Moreover, if we drop the monotonicity assumption completely but still assume continuity, then (by the most familiar version of the Intermediate Value

Theorem) there must still exist values t^* such that $m(t^*) = m'(t^*)$, but these t^* are no longer confined to a single interval, and can even consist of countably many isolated intervals. It seems safe to say that such observations would be very difficult to make without a precise mathematical analysis like that given above[15]; in fact, Fauconnier and Turner had not previously relized that the monk could meet himself more than once.

Finally, it is important to notice that the structures used in this example, consisting of a many-sorted logical theory and a class of models that satisfy that theory, such that some sorts have fixed interpretations in the models, can be seen as a classification in the sense of Barwise and Seligman [3]; however, it is better to consider these structures as frames in the sense of the theory of institutions (in Section 5), because in this example the geometrical structure of the models is important, and the different theories involve different vocabularies. Notice that it is not just the theories that are blended, but the frames, including the vocabularies over which they are defined. □

Concepts with prototype effects have a fuzzy logic, in that similarity to a prototype determines a grade of membership. A formal development of first order many sorted fuzzy logic is given in [20], where membership can be measured by values in the unit interval, or in more general ordered structures; it is not difficult to generalize semiotic spaces to allow fuzzy predicates, and it is also possible to develop a comprehensive theory of convex fuzzy sets[16]. The example below demonstrates how these ideas are useful in extending Gärdenfors' approach.

Example 6: The "pet fish" problem from cognitive semantics is, in brief, how can a guppy be a bad exemplar for "pet" and for "fish," but a very good exemplar for "pet fish"? Let A be a space for animals, coordinatized by k measurable (real valued) quantities, such as average weight at maturity, average length at maturity, number of limbs, etc. For each point a in A that represents an animal, let $p(a)$ denote the extent to which a is judged to be a "pet" (in some set of experiments), and similarly $f(a)$ for "fish"; for convenience, we can interpolate other values for p and f between those with given experimental values, so that they are continuous functions on A, which can be assumed to be a rectilinear subset of $\mathbb{R}^k$. In fuzzy logic, "pet fish" is the intersection of p and f, which

[15] Of course, some results of this analysis are unrealistic, due to assuming that the monk can move arbitrarily quickly; a velocity restriction be added, but at some extra cost in complexity.

[16] The author has done so in an unpublished manuscript from 1967. A fuzzy set $f\colon X \to L$ is *convex* iff for each $\ell \in L$, the level set $f^\ell = \{x \mid f(x) \geq \ell\}$ is convex, where L is any partially ordered set (usually at least a complete lattice).

we an write as $pf(a) = p(a) \wedge f(a)$, where $\wedge$ gives the minimum of two real values; see Figure 4. Then "guppy" can be have a maximal value for pf even though it is far from maximal for either p or f. (It makes sense that the maximum value of pf is smaller than those for p or f, because "pet fish" is a somewhat rare concept, but if desired, pf can be renormalized to have maximum 1.) Moreover, if Gärdenfors is right, then the level sets $p_\ell = \{a \in A \mid p(a) \geq \ell\}$ are convex for each $0 \leq \ell \leq 1$, and similarly for f_ℓ. It follows from this that pf is also convex, because $pf_\ell = \{a \in A \mid p(a) \geq \ell \text{ and } f(a) \geq \ell\} = p_\ell \cap f_\ell$ and the intersection of convex subsets of $\mathbb{R}^k$ is convex. □

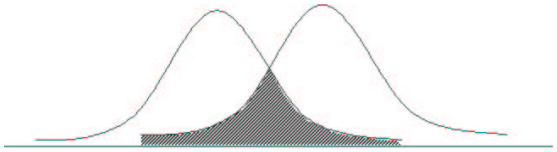

Fig. 4. A fuzzy intersection

As a final remark, our answer to the symbol grounding problem (in the sense of footnote 12) follows Peirce, who very clearly said that signs must be interpreted in order to refer, and interpretation only occurs in some pragmatic context of signs being actually used; this means that the symbol grounding problem is artificial, created by a desire for something that is not possible for purely symbolic systems, as in classic logic-based AI, but which is easy for interpreted systems[17].

4 Social Science

There is relatively little work in the social sciences addressing concepts in the sense of this paper. However, there are some bright spots. *Ethnomethodology* emphasizes the situatedness of social data, and closely examines how competent members of a group actually organize their interactions. A basic principle of *accountability* says that members may be required to account for certain actions by their social groups, and that exactly those actions are socially significant to those groups. From this follows a principle of *orderliness*, which says that social interaction can be understood, because the participants understand it due to accountability; therefore analysts can also understand it, once they discover how members make sense of their interactions.

[17] From an engineering perspective, one can say that sensors, effectors, and a world model are needed to ground the constants in conceptual spaces; this is of course what robots have, and our previously discussed partial solution can provide intermediate non-symbolic (geometric) representations for systems with such capabilities.

To understand interaction, ethnomethodology looks at the *categories* (i.e., concepts) and the *methods* that members use to render their actions intelligible to each other; this contrasts with presupposing that the categories and methods of the analyst are necessarily superior to those of members. Harvey Sacks' *category systems* [59] are collections of concepts that members distinguish and treat as naturally co-occurring. Some rules that govern the use of such systems are given in [59], which also demonstrates how category systems provide a rich resource for interpreting ordinary conversation. A similar approach is used in a study reported in [22] which aimed to understand the workflow and value systems of a corporate recruitment agency. Ethnomethodology has many commonalities with Peirce's pragmatism and semiotics that would be well worth further exploration.

The *activity theory* of Lev Vygotsky [64, 65] and others is an approach to psychology emphasizing human activity, its material mediation (taken to include language), and the cultural and historical aspects of communication and learning; Michael Cole's *cultural psychology* [9] builds on this, and is particular concerned with social aspects of learning. The *distributed cognition* of Edwin Hutchins [40] and others claims that cognition should be understood as distributed rather than purely individual, and studies "the propagation of representational states across media." Leigh Star's boundary objects [62], in the tradition of science studies, provide interesting examples in which different subgroups use the same material artifact in quite different ways. It might be objected that this discussion concerns how concepts are used in social groups, rather than what they "are." But the authors cited in this section would argue that concepts cannot be separated from the social groups that use them.

An important ingredient that seems to me under-represented in all these theories is the role of *values*, i.e., the motivations that people have for what they do. It is argued in [22] that these can be recovered using the principle of accountability, as well as through discourse analysis, in the socio-linguistic tradition of William Labov [45] and others; some examples are also given in [22].

5 Logical and Semantic Heterogeneity and Integration

This section explains Unified Concept Theory; although mainly mathematical, it is motivated by non-mathematical ideas from previous sections, as the final section makes more explicit.

We have already shown how frames unify the symbolic concept spaces of Fauconnier [13] with the geometrical conceptual spaces of Gärdenfors [16] (see Example 3). This section shows how LOT, FCA, IF and CI can be included, and also shows how these theories can make sense over any logic, although they are usually done over first order, or some closely related logic (such as conceptual graphs [61]). For example, Section 3 argued that order sorted algebra is an interesting alternative for applications to ontologies, databases and so on. One might think a lot of work would be required to transport these theories to such a different logic. But the theory of institutions [28, 53] actually makes this quite easy [26].

Institutions formalize the notion of "a logic", based on a triadic satisfaction relation, which like Peirce's signs, includes a context for interpretation. Intuitively, institutions capture the two main ingredients of a logic, its sentences and its models, as well as the important relation of satisfaction between them, with the novel ingredient of parameterization by the vocabulary used; this is important because different applications of the same logic in general use different symbols. The formalization is very general, allowing vocabularies to be objects in a category; these objects are called "signatures," and they serve as "contexts," or "interpretants" in the sense of Peirce. The possible sentences (which serve as "descriptions," or "representamen" in Peirce-speak) are given by a functor from signatures. The possible models ("tokens," or "objects" for Peirce) are given by a (contravariant) functor from the signature category. Finally, given a signature Σ, satisfaction is a binary relation between sentences and models, denoted $\models_\Sigma$, required to satisfy a condition asserting invariance of satisfaction (or "truth") under context morphisms. The following makes this precise:

Definition 7: An **institution** consists of an abstract category $\mathbb{S}\mathrm{ign}$, the objects of which are signatures, a functor $Sen\colon \mathbb{S}\mathrm{ign} \to \mathbb{S}\mathrm{et}$, and a functor $Mod\colon \mathbb{S}\mathrm{ign}^{op} \to \mathbb{S}\mathrm{et}$ (technically, we might uses classes instead of sets here). **Satisfaction** is then a parameterized relation $\models_\Sigma$ between $Mod(\Sigma)$ and $Sen(\Sigma)$, such that the following **Satisfaction Condition** holds, for any signature morphism $\varphi\colon \Sigma \to \Sigma'$, any Σ'-model M', and any Σ-sentence e,

$$M' \models_{\Sigma'} \varphi(e) \text{ iff } \varphi(M') \models_\Sigma e$$

where $\varphi(e)$ abbreviates $Sen(\varphi)(e)$ and $\varphi(M')$ abbreviates $Mod(\varphi)(e)$. □

Much usual notation of model theory generalizes, e.g., we say $M \models_\Sigma T$ where T is a set of Σ-sentences, if $M \models_\Sigma \varphi$ for all $\varphi \in T$, and we say $T \models_\Sigma \varphi$ if for all Σ-models M, $M \models_\Sigma T$ implies $M \models_\Sigma \varphi$. Moreover, if V is a class of models, let $V \models_\Sigma T$ if $M \models_\Sigma T$ for all models M in V. We

call the structure that results when a signature is a fixed frame[18]. It is now very interesting to notice that the Buddhist monk example is really a blending of two frames, not just of two conceptual spaces.

Example 8: First order logic is a typical example: its signatures are sets of relation names with their arities; its signature morphisms are arity preserving functions that change names; its sentences are the usual first order formulae; its models are first order structures; and satisfaction is the usual Tarskian relation. See [28] for details. It is straightforward to add function symbols and sorts. □

Other logics follow a similar pattern, including modal logics, temporal logics, many sorted logics, equational logics, order sorted logics, description logics, higher order logics, etc. Database systems of various kinds are also institutions, where database states are contexts, queries are sentences, and answers are models [25]. The theory of institutions allows one to explore formal consequences of Peirce's triadic semiotic relation, which are elided in the dyadic formalisms of IF and FCA; for example, one insight with significant consequences is that Peirce's "objects" (i.e., an institution's models) are contravariant with respect to "interpretants" (i.e., signatures).

In the special case where the signature category has just one object and one morphism (the identity on that object), institutions degenerate to the classifications of [3], which are the same as the formal contexts of [15] (as well as the frames of institutions). The translation is as follows (where we use the prefix "I-" to indicate institutions): I-models, IF-tokens, and FCA-objects correspond; I-sentences, IF-types, and FCA-attributes correspond; and I-satisfaction, IF-classification, and FCA formal concept correspond. Chu spaces (see the appendix in [2]), in the usual case where $Z = \{0, 1\}$, are also a notational variant of one signature institutions, and general Chu spaces (and their categories) are the one signature case of the "generalized institutions" of [27], where satisfaction takes values in an arbitrary category V (but [63] gives a better exposition).

Institution morphisms can be defined in various ways [33], each yielding a category, in which many important relations among institutions can be expressed, such as inclusion, quotient, product, and sum. This gives a rich language for comparing logics and for doing constructions on logics.

[18] In [27], the word "room" is used, and the more precise mathematical concept is called a "twisted relation" in [28]); in fact, institutions can be seen as functors from signatures to frames.

In the special case of institutions with just one signature, institution morphisms (actually "comorphisms") degenerate to the "infomorphisms" of [3], and the formal context morphisms of [15]. Categories of institutions support *logical heterogeneity*, i.e., working in several logics at the same time, as developed in some detail in [11, 26] and many other papers.

Given an institution $\mathbb{I}$, a *theory* is a pair (Σ, T), where T is a set of Σ-sentences. The collection of all Σ-theories can be given a lattice structure, under inclusion[19]: $(\Sigma, T) \leq (\Sigma', T')$ iff $\Sigma \subseteq \Sigma'$ and $T \subseteq T'$; this is the lattice of theories (LOT) of Sowa [61]. However, a richer structure is also available for this collection of theories, and it has some advantages: Let $Th(\mathbb{I})$ be the category with theories as objects, and with morphisms $(\Sigma, T) \to (\Sigma', T')$ all those signature morphisms $f\colon\ \Sigma \to \Sigma'$ such that $T' \models_{\Sigma'} f(\varphi)$, for all $\varphi \in T$.

The category $Th(\mathbb{I})$ was invented to support flexible modularity for knowledge representation, software specification, etc., as part of the semantics of the Clear specification language [28]. The most interesting constructions are for parameterized theories and their instantiation. An example from numerical software is `CPX[X :: RING]`, which constructs the complex integers, `CPX[INT]`, the ordinary complexes, `CPX[REAL]`, etc., where `RING` is the theory of rings, which serves as an *interface theory*, in effect a "type" for modules, declaring that any theory with a ring structure can serve as an argument to `CPX`. For another example, if `TRIV` is the trivial one sort theory, an interface that allows any sort of any theory as an argument, then `LIST[X :: TRIV]` denotes a parameterized theory of lists, which can be instantiated with a theory of natural numbers, `LIST[NAT]`, of Booleans, `LIST[BOOL]`, etc. Instantiation is given by the categorical pushout construction (a special case of colimits) in $Th(\mathbb{I})$, where $\mathbb{I}$ is an institution suitable for specifying theories, such as order sorted algebra.

Other operations on theories include renaming (e.g., renaming sorts, relations, functions) and sum; thus, compound "module expressions," such as `NAT + LIST[LIST[CPX[INT]]]` are also supported. This module system inspired those of C++, Ada, ML, and numerous specification languages (e.g., those in footnote 3). Colimits are suggested in [28] for evaluating module expressions such as $E =$ `REAL` $+$ `LIST[CPX[INT]]`. For example, we may instantiate Figure 3 with B the module expression E above, $I_1 =$ `REAL`, $I_2 =$ `LIST[CPX[INT]]`, and $G =$ `INT` as a common subobject, noting that `INT` is a subtheory of `REAL`.

[19] The converse ordering is used by Sowa, and seems to have more intuitive appeal, because a larger theory has a smaller collection of models.

For any signature Σ of an institution $\mathbb{I}$, there is a *Galois connection* between its Σ-theories and its sets of Σ-models: $(\Sigma, T)^\bullet = \{M \mid M \models_\Sigma T\}$, and if $\mathcal{M}$ is a collection of Σ-models, let $\mathcal{M}^\bullet = \{\varphi \mid \mathcal{M} \models_\Sigma \varphi\}$. A theory T is *closed* if $T^{\bullet\bullet} = T$, and it is natural to think of the closed theories as the "*concepts*" of $\mathbb{I}$, or following FCA [15], to define *formal concepts* to be pairs $(T, \mathcal{M})$ such that $T^\bullet = \mathcal{M}$ and $\mathcal{M}^\bullet = T$. Much of FCA carries over to arbitrary institutions. For example, the closed theories form a complete lattice under inclusion which is (anti-) isomorphic to the lattice of closed model sets; this lattice is called the **formal concept lattice** in [15]. The Galois connection also gives many identities connecting the various set theoretic operations with closure, such as $(T \cup T')^\bullet = T^\bullet \cap T'^\bullet$. Example 9 below applies these ideas to an ontology problem, and a more elegant, categorical version of these ideas is given just after that example.

Ontologies are a promising application area for ideas in this section. First, numerous logics are being proposed and used for expressing ontologies, among which description logics, such as OWL, are the most prominent, but by no means the only, examples. Second, a given semantic domain will often have a number of different ontologies. Thus both logical and semantic heterogeneity are quite common, and integration at both levels is an important challenge. Fortunately, category theory provides appropriate tools for this. The "IFF" approach of Robert Kent has pioneered work in this area, integrating FCA and IF in [42], and more recently, using institutions to unify the IEEE Standard Upper Ontology [43], which defines a set of very high level concepts for use in defining and structuring specific domain ontologies.

Example 9: A simple but suggestive example, originally from [61], but elaborated in [60] and further elaborated here, illustrates applications to the problems of "ontology alignment" and "ontology merging," where the former refers to how concepts relate, and the latter refers to creating a joint ontology. The informal semantics that underlies this example is explained in the following quote from [61]:

> In English, size is the feature that distinguishes "river" from "stream"; in French, a "fleuve" is a river that flows into the sea, and a "riviére" is a river or a stream that runs into another river.

We can now set up the problem as follows: there are two (linguistic) contexts, French and English, each of which has two concepts; let us also suppose that each context has three instances, as summarized in the two classification relations shown in the following tables.

English	river	stream
Mississippi	1	0
Ohio	1	0
Captina	0	1

French	fleuve	riviére
Rhône	1	0
Saône	0	1
Roubion	0	1

Although these tables are insufficient to recover the informal relations between concepts given in the quotation above, if we first align the instances, a surprising amount of insight can be obtained. So let us further suppose it is known that the corresponding rows of the two tables have the same conceptual properties, e.g., that the Mississippi is a fleuve but not a riviére, that the Saône is a river but not a stream, etc.

One might expect that this correspondence of rows should induce some kind of a blend, and indeed, we will formalize the example in such a way that a blend is obtained in our formal sense. We first describe the very simple logic involved. Its signatures are sets of unary predicates. Its sentences over a signature Σ are the nullary predicate *false*, and the unary predicates in Σ. Signature morphisms are maps of the unary predicate names. A model for Σ is a one point set, with some subset of predicates in Σ designated as being satisfied; the satisfaction relation for this model is then given by this designation. Note that for this logic, $Th(\Sigma)$ has as its objects the set of all sets of Σ-sentences.

The English signature Σ^E in this example is just $\{\text{river}, \text{stream}\}$ and the French signature is $\Sigma^F = \{\text{fleuve, riviére}\}$, while the blend signature Σ^B is their union. The two tables above can be seen as defining 6 models, and also as defining the two satisfaction relations, or better, two frames.

It is possible to recover some interesting relationships among the concepts represented by the predicates if we extend the satisfaction relation to sets of sentences and sets of models as indicated just after Definition 7. Since the entities corresponding to the three rows of the two tables have the same properties, we can merge them when we blend the signatures. If we denote these merged entities by MR, OS and CR, then the merged models and their satisfaction relation are described by the following:

	river	stream	fleuve	riviére
MR	1	0	1	0
OS	1	0	0	1
CR	0	1	0	1

Figure 5 shows the formal concept lattice of this merged context over the merged signature Σ^B, with its nodes labeled by the model sets and theories to which they correspond. It is interesting to notice that any minimal

set of generators for this lattice gives a canonical formal vocabulary for classifying models. Although in this case, there are $2^3 = 8$ possible sets of models, only 7 appear in the concept lattice and a subset of 3 are sufficient to generate the lattice. (The reason there are only 7 elements is that there is no model that both is small and flows into the sea.)

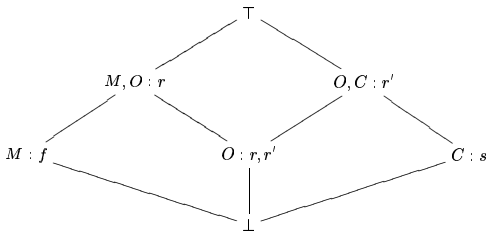

Fig. 5. A formal concept lattice

A more sophisticated institutional view is that we have blended the English and French contexts into the single context of the final table, by a pushout construction (as in Figure 3) in the category of frames, where the contravariant nature of the model part of the institution means that the frame pushout does a pullback on models, giving the identifications (or "alignment") that we made, whereas the covariant nature of the signature and sentence components gives a union. The Buddhist monk example can be understood in the same way. □

Given an institution $\mathbb{I}$, we define another institution $\mathbb{I}^G$, the **Galoisification** of $\mathbb{I}$, as follows: its signature category is the same as that of $\mathbb{I}$, its sentences are the closed theories of $\mathbb{I}$, its models are the closed model classes of $\mathbb{I}$, and its satisfaction relation is the natural extension of that of $\mathbb{I}$. We may call a frame of $\mathbb{I}^G$ a **Galois frame**, or **G-frame** for short. Then the satisfaction relation of each G-frame is a bijective function, the corresponding pairs of which are the formal concepts of [15] in the special case of their logic, though this construction applies to any institution.

The category $Th(\mathbb{I})$ is also an example of a very general construction, called *Grothendieck flattening*; see [26] for the definition and many examples, as well as many more details about information flow and channel algebra over an arbitrary institution. Grothendieck flattening supports *semantic heterogeneity*, i.e., working in several different contexts at the same time, by defining new morphisms, called *heteromorphisms*, between objects from different categories; this gives rise to the **Grothendieck category**, of which $Th(\mathbb{I})$ is an example. A number of useful general theorems are known about Grothendieck categories, for example, conditions for limits and colimits to exist.

Colimits are not an adequate formalization of blending in the sense of cognitive linguistics, because more than one blend is possible for two input spaces (e.g., "houseboat" and "boathouse"). This raises the mathematical challenge of weakening colimits so that non-multiple isomorphic solutions are allowed. Another challenge is to discover precise "optimality principles" that measure the quality of the blend space. These problems are addressed in *algebraic semiotics* [23, 24], which is a general theory of representation, based on the semiotic spaces discussed in Section 3) and *semiotic morphisms*, which (partially) preserve the structure of semiotic spaces, and which model representations, such as an index to a book, a graph of a dataset, or a GUI for UNIX. A basic principle in [23] is that the quality of a representation depends on how well its semiotic morphism preserves structure, and [23] proposes a number of quality measures on that basis; [23] also introduces 3/2-category theory, including 3/2-colimits as a model of blending, under the hypothesis that optimal blends are achieved by using semiotic morphisms that score well on the quality measures.

This theory is tested against some simple examples in [23], and has also been implemented in a *blending algorithm* called "Alloy" [30, 29], which generates novel metaphors on the fly as part of a poetry generation system (called "Griot") developed by Fox Harrell. The optimality principles used are purely formal (since they could not otherwise be implemented); they measure the degree to which the injection morphisms $I_i \to B$ in Figure 3 preserve structure, including constants, relation instances, and types[20].

6 Reconciliations and Conclusions

This paper considers cognitive, social, pragmatic, and mathematical perspectives on concepts. Despite this diversity, there has been a single main goal, which is to facilitate the design and implementation of systems to support information integration. Good mathematical foundations are of course a great help, in providing precise descriptions of what should be implemented. But without some understanding of broader issues, it is impossible to understand the potential limitations of such systems. In addition, models from cognitive psychology and sociology can inspire new

[20] An interesting sidelight is that recent poetry (e.g., Neruda and Rilke) often requires what we call "disoptimality principles" to explain certain especially striking metaphors [29]. For example, type coercions are needed when items of very different types are blended.

approaches to information integration. So the huge gaps that currently exist between disciplines are really counter-productive. This section is devoted to sketching various forms of reconciliation among them, along with some general conclusions.

It can be argued that some concepts (scientific concepts) have a hard physical reality, manifesting as perceivable regularities of behavior, or in a more sophisticated language, as invariants over perceptions, e.g., in James Gibson's "ecological" approach to perception [18]. It can also be argued that other concepts (mathematical) are formal transcendental, existing independently of humans and even physical reality. But as realized long ago by the Indian philosopher Nagarjuna [55], phenomenological human concepts are not like that: they are elastic, situated, evolving, relative, pragmatic, fuzzy, and strongly interconnected in domains with other concepts; the thoughts we actually have cannot be pinned down as scientific or mathematical concepts. Contemporary cognitive science has explored many reasons for this, and Section 4 suggests that there are also many social reasons.

There are several views on how phenomenological, social and scientific approaches to concepts can be reconciled; in general, they argue that the phenomenological and the social are interdependent, in that concepts necessarily exist at both levels, and that scientific and mathematical concepts are not essentially different from other concepts, though it is convenient to use these more precise languages to construct models, without needing to achieve perfect fidelity. For example, Gärdenfors [16] argues that formal entities, such as equations, "are not elements of the internal cognitive process, but function as *external devices* that are, literally, manipulated"; this is similar to the material anchors of Hutchins [40], which represent shared concepts in concrete form to facilitate cooperative work. Terrence Deacon [10] argues that concepts evolve in social environments in much the same way that organisms evolve in natural environments, as part of an ambitious program that (among other things) seeks a biological reconstruction of Richard Dawkins' implausible reification of concepts as "memes".

In a brilliant critique of modernism, Bruno Latour [48] proposes to reconcile society, science, and the myriad hybrid "quasi-objects" that have aspects of both (among which he apparently includes natural language) through a "symmetry principle" that refuses to recognize the "modern" distinctions among these areas, and that grounds explanations in the quasi-objects, which inherently mix cognitive, social, and formal aspects. Latour rightly criticizes the Saussurian dyadic semiotics that dominated

French cultural thought for a time, but seems unaware of Peirce's [56] triadic semiotics, which I believe offers a better solution, because its signs are already defined to be hybrid "quasi-objects"; note that one of Peirce's goals for his triadicity was to reconcile nominalism and realism [41], an enterprise with a flavor similar to that of Latour's. A Peircean perspective is also the basis of Deacon's theories [10] concerning how concepts evolve within social contexts.

Ideas in this paper have implications for the study of formal aspects of concepts, including knowledge representation, analysis and integration. Section 3 unified two notions of conceptual space, due to Fauconnier and to Gärdenfors using frames. The Unified Concept Theory of Section 5 generalized this, as well as LOT, FCA, IF and CI, to arbitrary logics by using institutions. Institutions can be considered to formalize Peirce's triadic semiotics, as well as Latour's quasi-objects. It could be interesting to apply UCT to description logics [1], which are commonly used for ontologies, or alternatively, as argued in Section 3, to an order sorted algebra approach to ontologies. I believe there are also fruitful applications to database systems, e.g., the problem of integrating information from multiple databases; see [25]. These examples illustrate how the exploration of ways to reconcile cognitive, social, pragmatic, and formal approaches to concepts can be useful in suggesting new research directions.

Perhaps the most important conclusion is that research on concepts should be thoroughly interdisciplinary, and in particular, should transcend the boundaries between sciences and humanities. Unfortunately, such efforts, including those of this paper, are likely to attract criticism for blurring distinctions between established disciplines, which indeed often operate under incompatible assumptions, using incomparable methods. It is my hope that the reconciliations and unifications sketched above may contribute to the demise of such obstructions, as well as to a better understanding of concepts and their applications.

References

1. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, editors. *Description Logic Handbook*. Cambridge, 2003.
2. Michael Barr. $\star$-autonomous categories and linear logic. *Mathematical Structures in Computer Science*, 1:159–178, 1991.
3. Jon Barwise and Jerry Seligman. *Information Flow: Logic of Distributed Systems*. Cambridge, 1997. Tracts in Theoretical Computer Science, vol. 44.
4. John Lane Bell. *Toposes and Local Set Theories: An Introduction*. Oxford, 1988. Oxford Logic Guides 14.
5. David Bloor. *Knowledge and Social Imagery*. Chicago, 1991. Second edition.

6. Herbert Blumer. *Symbolic Interactionism: Perspective and Method.* California, 1986.
7. Rod Burstall and Joseph Goguen. Putting theories together to make specifications. In Raj Reddy, editor, *Proceedings, Fifth International Joint Conference on Artificial Intelligence*, pages 1045–1058. Department of Computer Science, Carnegie-Mellon University, 1977.
8. Manuel Clavel, Steven Eker, Patrick Lincoln, and José Meseguer. Principles of Maude. In José Meseguer, editor, *Proceedings, First International Workshop on Rewriting Logic and its Applications.* Elsevier Science, 1996. Volume 4, *Electronic Notes in Theoretical Computer Science.*
9. Michael Cole. *Cultural Psychology: A once and future discipline.* Harvard, 1996.
10. Terrence Deacon. Memes as signs in the dynamic logic of semiosis: Beyond molecular science and computation theory. In Karl Wolff, Heather Pfeiffer, and Harry Delugach, editors, *Conceptual Structures at Work: 12th International Conference on Conceptual Structures*, pages 17–30. Springer, 2004. Lecture Notes in Computer Science, vol. 3127.
11. Răzvan Diaconescu and Kokichi Futatsugi. *CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification.* World Scientific, 1998. AMAST Series in Computing, Volume 6.
12. Samuel Eilenberg and Saunders Mac Lane. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58:231–294, 1945.
13. Gilles Fauconnier. *Mental Spaces: Aspects of Meaning Construction in Natural Language.* Bradford: MIT, 1985.
14. Gilles Fauconnier and Mark Turner. *The Way We Think.* Basic, 2002.
15. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations.* Springer, 1997.
16. Peter Gärdenfors. *Conceptual Spaces: The Geometry of Thought.* Bradford, 2000.
17. Harold Garfinkel. *Studies in Ethnomethodology.* Prentice-Hall, 1967.
18. James Gibson. *An Ecological Approach to Visual Perception.* Houghton Mifflin, 1979.
19. Barney Glaser and Ansolm Strauss. *The Discovery of Grounded Theory: Strategies for qualitative research.* Aldine de Gruyter, 1999.
20. Joseph Goguen. The logic of inexact concepts. *Synthese*, 19:325–373, 1969.
21. Joseph Goguen. A categorical manifesto. *Mathematical Structures in Computer Science*, 1(1):49–67, March 1991.
22. Joseph Goguen. Towards a social, ethical theory of information. In Geoffrey Bowker, Leigh Star, William Turner, and Les Gasser, editors, *Social Science, Technical Systems and Cooperative Work: Beyond the Great Divide*, pages 27–56. Erlbaum, 1997.
23. Joseph Goguen. An introduction to algebraic semiotics, with applications to user interface design. In Chrystopher Nehaniv, editor, *Computation for Metaphors, Analogy and Agents*, pages 242–291. Springer, 1999. Lecture Notes in Artificial Intelligence, Volume 1562.
24. Joseph Goguen. Semiotic morphisms, representations, and blending for interface design. In *Proceedings, AMAST Workshop on Algebraic Methods in Language Processing*, pages 1–15. AMAST Press, 2003.
25. Joseph Goguen. Data, schema and ontology integration. In *Proceedings, Workshop on Combinination of Logics*, pages 21–31. Center for Logic and Computation, Instituto Superior Tecnico, Lisbon, Portugal, 2004.

26. Joseph Goguen. Information integration in instutions. In Lawrence Moss, editor, *Memorial volume for Jon Barwise*. Indiana, to appear.
27. Joseph Goguen and Rod Burstall. A study in the foundations of programming methodology: Specifications, institutions, charters and parchments. In David Pitt, Samson Abramsky, Axel Poigné, and David Rydeheard, editors, *Proceedings, Conference on Category Theory and Computer Programming*, pages 313–333. Springer, 1986. Lecture Notes in Computer Science, Volume 240; also, Report CSLI-86-54, Center for the Study of Language and Information, Stanford University, June 1986.
28. Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, January 1992.
29. Joseph Goguen and Fox Harrell. Style as a choice of blending principles. In Shlomo Argamon, Shlomo Dubnov, and Julie Jupp, editors, *Style and Meaning in Language, Art Music and Design*, pages 49–56. AAAI Press, 2004.
30. Joseph Goguen and Fox Harrell. Foundations for active multimedia narrative: Semiotic spaces and structural blending, 2005. To appear in *Interaction Studies: Social Behaviour and Communication in Biological and Artificial Systems*.
31. Joseph Goguen and Kai Lin. Behavioral verification of distributed concurrent systems with BOBJ. In Hans-Dieter Ehrich and T.H. Tse, editors, *Proceedings, Conference on Quality Software*, pages 216–235. IEEE Press, 2003.
32. Joseph Goguen and José Meseguer. Order-sorted algebra I: Equational deduction for multiple inheritance, overloading, exceptions and partial operations. *Theoretical Computer Science*, 105(2):217–273, 1992. Drafts exist from as early as 1985.
33. Joseph Goguen and Grigore Roşu. Institution morphisms. *Formal Aspects of Computing*, 13:274–307, 2002.
34. Joseph Goguen, James Thatcher, Eric Wagner, and Jesse Wright. Initial algebra semantics and continuous algebras. *Journal of the Association for Computing Machinery*, 24(1):68–95, January 1977.
35. Joseph Goguen and William Tracz. An implementation-oriented semantics for module composition. In Gary Leavens and Murali Sitaraman, editors, *Foundations of Component-based Systems*, pages 231–263. Cambridge, 2000.
36. Joseph Goguen, Timothy Winkler, José Meseguer, Kokichi Futatsugi, and Jean-Pierre Jouannaud. Introducing OBJ. In Joseph Goguen and Grant Malcolm, editors, *Software Engineering with OBJ: Algebraic Specification in Action*, pages 3–167. Kluwer, 2000.
37. Rebecca Green. Internally-structured conceptual models in cognitive semantics. In Rebecca Green, Carol Bean, and Sung Hyon Myaeng, editors, *The Semantics of Relationships*, pages 73–90. Kluwer, 2002.
38. Stevan Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
39. William S. Hatcher. *Foundations of Mathematics*. W.B. Saunders, 1968.
40. Edwin Hutchins. *Cognition in the Wild*. MIT, 1995.
41. Mary Keeler. Hegel in a strange costume. In Aldo de Moor, Wilfried Lex, and Bernhard Ganter, editors, *Conceptual Structures for Knowledge Creation and Communication*, pages 37–53. Springer, 2003. Lecture Notes in Computer Science, vol. 2746.
42. Robert Kent. Distributed conceptual structures. In Harre de Swart, editor, *Sixth International Workshop on Relational Methods in Computer Science*, pages 104–123. Springer, 2002. Lecture Notes in Computer Science, volume 2561.
43. Robert Kent. Formal or axiomatic semantics in the IFF, 2003. Available at `suo.ieee.org/IFF/work-in-progress/`.

44. Thomas Kuhn. *The Structure of Scientific Revolutions*. Chicago, 1962.
45. William Labov. *Language in the Inner City*. University of Pennsylvania, 1972.
46. George Lakoff. *Women, Fire and Other Dangerous Things: What categories reveal about the mind*. Chicago, 1987.
47. George Lakoff and Mark Johnson. *Philosophy in the Flesh: The Embodied Mind and its Challenge to Western Thought*. Basic, 1999.
48. Bruno Latour. *We Have Never Been Modern*. Harvard, 1993. Translated by Catherine Porter.
49. Bruno Latour and Steve Woolgar. *Laboratory Life*. Sage, 1979.
50. Eric Livingston. *The Ethnomethodology of Mathematics*. Routledge & Kegan Paul, 1987.
51. Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, 1998. Second Edition.
52. Donald MacKenzie. *Mechanizing Proof*. MIT, 2001.
53. Till Mossakowski, Joseph Goguen, Razvan Diaconescu, and Andrzej Tarlecki. What is a logic? In Jean-Yves Beziau, editor, *Logica Universalis*, pages 113–133. Birkhauser, 2005. *Proceedings*, First World Conference on Universal Logic.
54. Peter Mosses, editor. *CASL Reference Manual*. Springer, 2004. Lecture Notes in Computer Science, Volume 2960.
55. Nagarjuna. *Mulamadhyamika Karaka*. Oxford, 1995. Translated by Jay Garfield.
56. Charles Saunders Peirce. *Collected Papers*. Harvard, 1965. In 6 volumes; see especially Volume 2: Elements of Logic.
57. Benjamin C. Pierce. *Basic Category Theory for Computer Scientists*. MIT, 1991.
58. Simone Pribbenow. Merenymic relationships: From classical mereology to complex part-whole relations. In Rebecca Green, Carol Bean, and Sung Hyon Myaeng, editors, *The Semantics of Relationships*, pages 35–50. Kluwer, 2002.
59. Harvey Sacks. On the analyzability of stories by children. In John Gumpertz and Del Hymes, editors, *Directions in Sociolinguistics*, pages 325–345. Holt, Rinehart and Winston, 1972.
60. Marco Schlorlemmer and Yannis Kalfoglou. A channel-theoretic foundation for ontology coordination. In *Proceedings, 18th European Workshop on Multi-Agent Systems*. 2004.
61. John Sowa. *Knowledge Representation: Logical, Philosophical and Computational Foundations*. Brooks/Coles, 2000.
62. Susan Leigh Star. The structure of ill-structured solutions: Boundary objects and heterogeneous problem-solving. In Les Gasser and Michael Huhns, editors, *Distributed Artificial Intelligence*, volume 2, pages 37–54. Pitman, 1989.
63. Andrzej Tarlecki, Rod Burstall, and Joseph Goguen. Some fundamental algebraic tools for the semantics of computation, part 3: Indexed categories. *Theoretical Computer Science*, 91:239–264, 1991. Also, Monograph PRG–77, August 1989, Programming Research Group, Oxford University.
64. Lev Vygotsky. *Thought and Language*. MIT, 1962.
65. Lev Vygotsky. *Mind in Society*. Harvard, 1985.