WHAT IS A CONCEPT?

Joseph A. Goguen

Dept. Computer Science & Engineering
University of California at San Diego, USA

Partial support from NSF ITR 0225676, Science Environment for Ecological Knowledge (SEEK). Thanks to John Sowa, Mary Keeler and Rod Burstall for suggestions and references, and to Jenny Wang, Kai Lin, Vitaliy Zavesov for ideas and implementation work.

1. INTRODUCTION

Are many theories of concepts, from:
psychology, sociology, philosophy, mathematics, computer science, linguistics, artificial intelligence, ...; also some hybrids.
Want to survey, critically evaluate, & integrate them, insofar as possible.

Many new ideas,
often using institution theory,
which uses category theory

Called **Unified Concept Theory (UCT)**:
lattice of theories, LOT (Sowa);
formal concept analysis, FCA (Wille);
information flow, IF (Barwise & Seligman);
conceptual spaces, CF (Fauconnier);
conceptual spaces, CG (Gärdenfors);
blending, BT (Fauconnier & Turner);
information flow framework, IFF (Kent);
database semantics, DB, and more ....

But this project is far from finished .....

2. Case Study: “Category”

Very complex history:
from Aristotle’s grammatical categories
through Kant, Hegel, Peirce, ...
– the latter two with triads.
Evolution by metonymy, analogy, abstraction,
specialization, ...; punctuated equilibrium.

Eilenberg & Mac Lane took term from Kant,
solved problem in algebraic topology,
equivalence of homology theories

Provides abstract notion of structure,
based on structure preserving morphisms.

Does amazing amount of math very abstractly:
isomorphism, product, quotient, sum,
subobject, ...
and their properties.

Is the language of much contemporary math,
esp. algebraic & differential geometry.

Also very nice for much computer science.

A **category** $\mathbb{C}$ has:

- class $|\mathbb{C}|$ of **objects**
- set $\mathbb{C}(A, B)$ of **morphisms** $A \to B$
- **identity** morphism 1_A in $\mathbb{C}(A, A)$
- **composition** $\mathbb{C}(A, B) \times \mathbb{C}(B, C) \to \mathbb{C}(A, C)$, denoted "**;**"

s.t. $f; 1_A = f$, $1_A; g = g$, $f; (g; h) = (f; g); h$ whenever make sense.

- category $\mathbb{S}et$ of sets & functions;
- Euclidean spaces $\mathbb{R}^n$ with $n \times m$ real matrices as morphisms $\mathbb{R}^n \to \mathbb{R}^m$, diagonal 1s as identity, matrix multiplication;
- topological spaces & continuous maps;
- groups & homomorphisms;
- small categories & functors;
- Everything you want in practice,

And can prove many general results.

2.1 Sociology of Concepts

Thinking in arrows
  differs from usual set theory & logic
  composition fundamental, not elements.

Many applications in CS:
  – abstract data type theory;
  – constraint logic programming semantics;
  – Knuthian attribute semantics;
All are initial algebra semantics.
  – institutions for logic;
  – module composition for ML, OBJ, ...
    (but not Prolog)

But there is foundational challenge:
  – ZF sets not big enough;
  – even GNB not always big enough;
  – Grothendieck universes work well.

Foundations may seem esoteric,
but serious social consequences:
several good CS papers rejected in 1970s.

Many mathematicians against category theory, because it has different values:

- breadth vs. depth;
- unification vs. specialization;
- inclusive vs. exclusive.

Social aspects of Concepts:
need strong social framing to survive
– fit a Kuhnian "paradigm";
punctuated evolution – quick, or slow;
forks can happen;
used differently in different groups.

Values generally determine such phenomena.

2.2 Methodology

Social Study of Science & Technology:
– seeks qualitative understanding.
Bloor, MacKenzie, Latour, Callon, Star, Law, Garfinkel, Sacks & many others

Strong program, actor-network theory, out of symbolic interactionism, grounded theory, ethnomethodology;
all influenced by Peirce's pragmatism.

Generally constructivist, but *not* anti-realist.

Recommended reading:
Mechanizing Proof by Donald MacKenzie
social history of computer-based proof.

3. Cognitive Science

Eleanor Rosch's prototype theory of concepts:
mid-level in hierarchy;
short name;
most information;
fastest recognition;
...

Very different from Aristitle –
led to revolution in cog sci.

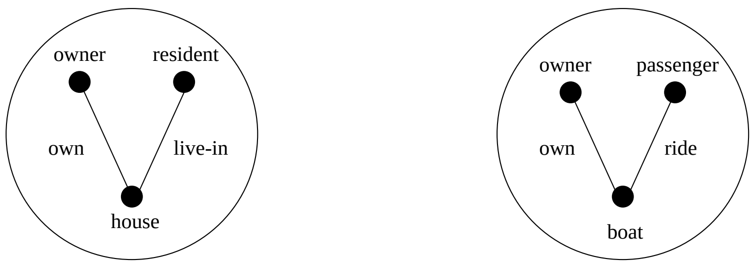
A major result is Lakoff's metaphor theory,
formulated as maps of conceptual spaces.

Important discovery: **basic schemas**:
embodied sensory-motor schemas
are basis of large metaphor families.

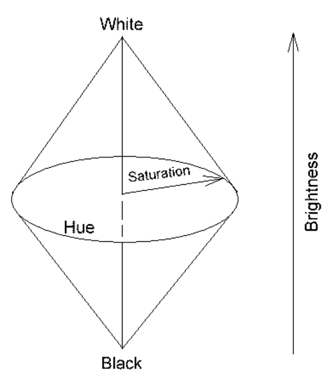
Example: UP-IS-MORE:
His salary went up.
The market is rising quickly.
Her expectations are high.

Fauconnier's conceptual spaces consist of:
constants; and
atomic relations among constants.
(Originally called "mental spaces".)

Example: "house" and "boat":



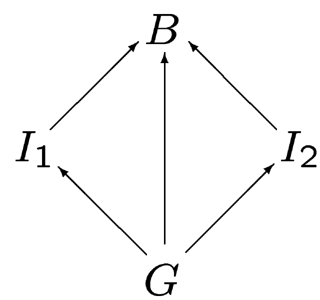
Gärdenfors' conceptual spaces geometrical:
Example: The color spindle:



Unification: Conceptual space C, geometrical space G, relation $\models$ of satisfaction between them, maybe fuzzy valued, called a **frame**.

3.1 Blending Theory

Blend diagram:
- two input spaces
- generic space (elts. common to inputs)
- put together in blend space.



Example: A metaphor:

"The sun is a king."

"The corona is his crown."

$Generic$	$Input1$	$Input2$	$Blend$
$obj1$	Sun	$King$	$Sun/King$
$obj2$	$corona$	$crown$	$corona/crown$
$has(obj1, obj2)$	$has(S, c)$	$has(K, c)$	$has(S/K, c/c)$

(abbreviate Sun to S, King to K, etc.)

Refines Lakoff metaphor theory:

identifications generate "cross-space map":

Sun to King, corona to crown, has to has.

Blend space has hybrid objects, emergent str.

Some Optimality Principles (F & T):

- Integration: Blend space is integrated scene
- Web: Tight connections blend & inputs
- Unpacking: Easy to reconstruct inputs
- Good Reason: Blend elements meaningful
- Topology: Blend relations similar to inputs
- Metonymy: Elts. same input close in blend

Example: Buddhist monk (F & T):
Monk climbs mountain at dawn,
arrives at dusk, meditates,
departs at dawn, arrives at dusk.
Question: Does he meet himself?
Blends two days to one & one monk to two!
Answer: Yes, by intermediate value theorem.
Details in paper: blending frames,
not just conceptual spaces.

Example: Pet fish problem:
How can guppy be bad example of fish & of pet, but good example of pet fish?
Use intersection of fuzzy sets.
See paper for details: is also blending frames.

4. Social Science

Ways to understand how concepts used:
ethnomethodology (Garfinkel, Sacks),
activity theory (Vygotsky),
distributed cognition (Hutchins, Star, ...),
cultural psychology (Cole, ...)
The "concept systems" of ethnomethodology are especially relevant.

My favorite neglected topic is *values*, as key to understanding much human behavior.

Semiotic spaces generalize conceptual spaces:
- functions, esp. constructors for structure;
- non-trivial axioms, not just relation instances;
- levels, priorities as value representations.

CSE 271, interface design,
uses **value-driven design**:
- semiotic spaces drive design
- values drive layout (font, color, size...)

5. Unified Concept Theory

Frame unifies logical, geometric concept spaces
– extends to other logics & spaces:
interpret relations, functions in geometric space.

Can also unify LOT, IF, FCA, BT,
over any logic at all.

Applications to:
ontologies, databases, semantic web, robotics,
integrate w. logical & semantic heterogeneity

Need to axiomatize notion of "a logic" –
not so difficult, uses category theory

Basic idea: parameterized family of frames:
Functor: $\mathbb{C}ontexts \to \mathbb{F}rames$
covariant in theories, contra in models,
where $\mathbb{C}ontexts$ are a category.

For usual logics, contexts are vocabularies,
called **signatures**, and
sentences are what models satisfy.

An **institution** $\mathcal{I}$ has:

- abstract signature category $\mathbb{S}ign$;
- sentence functor $Sen \colon \mathbb{S}ign \to \mathbb{S}et$;
- model functor $Mod \colon \mathbb{S}ign^{op} \to \mathbb{S}et$;
- satisfaction relation $\models_\Sigma$

s.t. for $f \colon \Sigma \to \Sigma'$ signature morphism,
$f(M) \models_{\Sigma'} e$ iff $M \models_\Sigma f(e)$.

Parameterization by signatures supports context dependence, thus formalizing Peirce's triadic semiosis.

It also provides a basis for complex structure in objects & attributes.

Are hundreds of papers with real applications & deep theory.

Examples of Institutions:

- first order logic;
- equational logic;
- first order logic with equality;
- Horn clause logic with equality;
- temporal, modal, epistemic logics;
- higher order logics;
- databases as signatures, with queries & answers;
- sheaves with temporal logic;
- in practice, everything you need.

Frame is institution with trivial sign category $\mathbb{1}$ (1 object, 1 morphism).

Formal contexts (FCS) & classifications (Barwise) special cases.

Institution is functor from signatures to frames.

5.1 Theories

Theory over $\mathcal{I}$ is (Σ, E), for E Σ-sentences.

M **satisfies** (Σ, E) iff $M \models_\Sigma e$, all $e \in E$.

Model class $(\Sigma, E)^*$ is all models satisfying it.

Theory $\mathcal{M}^*$ of models $\mathcal{M}$ is all sentences satisfied by all models in $\mathcal{M}$.

Galois connection between theories & models:
  closed theory has $(\Sigma, E)^{**} = (\Sigma, E)$;
  closed model class has $\mathcal{M}^{**} = \mathcal{M}$.
Many simple results, e.g., $(T \cup T')^* = T^* \cap T'^*$.

Define $(\Sigma, E) \leq (\Sigma', E')$ iff $\Sigma \subseteq \Sigma'$ & $E \subseteq E'$
  – gives lattices of theories
  – generalizes Sowa LOT to any logic.

Much of FCA also generalizes to any logic:
  closed theories form a lattice;
  closed model classes form a lattice;
  these lattices are anti-isomorphic;
Either one gives **formal concept lattice**.

Theory morphism $(\Sigma, E) \to (\Sigma', E')$ is signature morph $f\colon \Sigma \to \Sigma'$ s.t. $f(E) \subseteq E'^{**}$.

Theorem: Theories & theory morphisms are category $Th(\mathcal{I})$ with same (co)limits as $\mathbb{S}ign$.

Enables powerful operations to combine & manipulate theories or modules,
called **parameterized programming**.

Influenced ML, C++ templates, Ada, ...
Basis of algebraic specification languages
OBJ & BOBJ, Maude, CafeOBJ, CASL, ...
origin in Clear spec language

Powerful **module system** to combine ontologies over heterogeneous logics, provides:

- instantiation of parameterized theories,
- renaming of signature parts,
- importation & sharing of theories,
- join over shared subtheories, ...

Structures large and/or complex theories
to support reuse of specs, programs, proofs.

Example: Complex number package:
`CPX[X :: RING]`,where `RING` is theory of rings:
`CPX[INT]`, complex integers
`CPX[DINT]`, complex double precision ints
`CPX[REAL]`, usual complex numbers
Each composes very many functions.
Instantiation given by **pushout** in $Th(\mathcal{I})$.

Example: `LIST[X :: TRIV]`:
`LIST[INT]`
`LIST[BOOL]`

Also **module expressions** to compose systems:
`NAT + LIST[LIST[CPX[INT]]]`,
`NAT` is shared, since subtheory of `INT`.

Can also do renaming, e.g.,
`LIST[CHAR] * (List to String)`
Very convenient for composition, but
OBJ family default views help even more.

5.2 Information Flow

Ontology semantics using classifications, infomorphisms, local logics of Barwise & Seligman
— by Kent, Schorlemmer & others.

Classifications are simple frames:

- token sets are models,
- types are sentences,
- classification is satisfaction,
- infomorphisms are institution (co)morphisms.

See River/Riviére example in paper:
blend given by pushout in frame category.

Colimits give "information flow" in distributed systems, described as theories in any logic.

See 1971 categorical general systems theory (CGST): new insights for IF & IFF, e.g., limits for behavior & Interconnect Theorem.

5.3 Institution Morphisms

Institution morphism $\mathcal{I}$ to $\mathcal{I}'$ is

- functor $\Phi\colon \mathbb{S}ign \to \mathbb{S}ign'$,
- nat trans $\alpha_\Sigma\colon Sen'(\Phi(\Sigma)) \Rightarrow Sen(\Sigma)$,
- nat trans $\beta_\Sigma\colon Mod(\Sigma) \Rightarrow Mod'(\Phi(\Sigma))$,

s.t. for Σ in $\mathcal{I}$, M Σ-model in $\mathcal{I}$, e $\Phi(\Sigma)$-sentence in $\mathcal{I}'$,

$M \models_\Sigma \alpha_\Sigma(e')$ iff $\beta_\Sigma(M) \models_{\Phi(\Sigma)} (e')$.

Are truth preserving translations of logics.
Are many nice examples.

Institutions with them form category $\mathbb{I}ns$;
it has colimits & limits.

Also several variant notions of morphism:

- comorphisms;
- theoroidal morphisms;
- .....

Infomorphisms are comorphisms with $\mathbb{S}ign = \mathbb{1}$

5.4 Blending

Pushouts inadequate for blending:
pushouts unique up to isomorphism, but
blending is non-unique:
Example: house + boat = houseboat, boathouse.

Thus need to weaken colimits:
Algebraic semiotics uses **3/2-colimits**
in 3/2-categories = order enriched categories;
3/2-pushouts have multiple solutions.

Alloy blending algorithm for semiotic spaces,
is core of **Griot** system, for
interactive multimedia narrative,
implemented by Fox Harrell at UCSD.
interactive multimedia poems, stories, games
– featured in new CalIT2 building opening.

Blending also called conceptual integration,
but our blending is beyond conceptual spaces.

6. HETERO THEORY INTEGRATION

Many ways to combine logics, but
to integrate theories over different logics,
(modified) Diaconescu approach seems best.

Applies important construction of Grothendieck to institutions (in special case of strict indexed categories).

Diaconescu gets remarkable institution from indexed family of institutions,
called **Grothendieck institution**.

Logical properties tend to lift:

- Craig interpolation;
- colimits & limits;
- Beth definability;
- .....

Given category $\mathbb{O}$ of institutions, $\mathbb{GT}h(\mathbb{O})$ has:

- *objects* $(\mathcal{I}, \Sigma, E)$ where $\mathcal{I}$ in $\mathbb{O}$, and (Σ, E) theory of $\mathcal{I}$,
- *morphism* from $(\mathcal{I}, \Sigma, E)$ to $(\mathcal{I}', \Sigma', E')$ is institution morph $(\Phi, \alpha, \beta) \colon \mathcal{I} \to \mathcal{I}'$ in $\mathbb{O}$ sign morphism $f \colon \Sigma' \to \Phi(\Sigma)$ in $\mathcal{I}'$ s.t.
  $E \subseteq \alpha_\Sigma(f(E'))^{**}$,
- with "obvious" composition.

Is Grothendieck cat of functor $Th \colon \mathbb{O} \to \mathbb{C}at$.

Also theory cat of Grothendieck institution.

For reasonable $\mathbb{O}$, $\mathbb{GT}h(\mathbb{O})$ is cocomplete:
allows combining theories over logics in $\mathbb{O}$.

But $\mathbb{G}Th(\mathbb{O})$ has too many morphisms:
need to find good ones for given application;
for large theories, need tool to help.

Use of ontologies to integrate data unclear:
Key task is finding schema morphisms.

Schemas & ontologies are both theories,
usually over different logics.

Ontology for schema is hetero morphism,
from ontology to schema
$a_i\colon \Phi(O_i) \to T_i$.
In general, partial & non-injective.

Easy case:
Given two schemas with same ontology & an ontology morph, find schema morphism:

- If a_i partial, can enrich T_i.
- If a_i non-injective, can enrich O_i.

4. SCIA MAPPING TOOL

SCIA can integrate, query & translate DBs with DTDs or XML Schemas.

Most important task for integration is to:
find mapping from one schema to another.

Fully automatic schema mapping infeasible; SCIA minimizes user effort by identifying critical points, where small user input yields largest reduction of future matching effort.

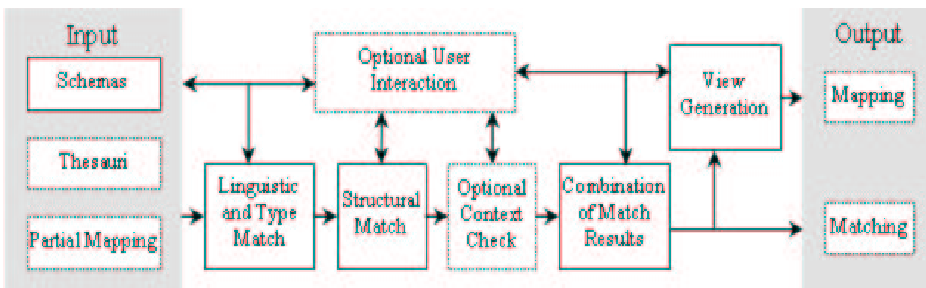
Critical point has no good match for core context, or > 1 good 1-to-1 match.

Core contexts typically have large subtrees.

Interactive mode: solicit user input at critical points, iterate until both user, tool satisfied.

Automatic mode: one pass, default strategies.

Each pass four steps: linguistic & data type matching; structural matching; context checking; combining match results.



Other tools only find easiest 1-to-1 matches,
difficult ones left for user to edit;
Also no semantic functions or conditions.

SCIA integrates all these.
Part of SEEK ecology toolset, called Kepler.

Results in significant reduction of total user effort (about 50%).

7. UNIFICATIONS & CONCLUSIONS

Considered cognitive, cultural-historical, social, pragmatic, philosophical & mathematical perspectives on concepts.

Very diverse, but one main goal:
systems for information integration.

Math foundation important, but so is
social context, to see limits & potential.

Want to reduce disciplinary boundaries:
Concepts can't be reified as science, math.

Example: Equations are:
cognitive, social & material;
material anchors (cf. Hutchins &c).
Of course, also seen as formal.
But valuable to see how used in practice,
e.g., by ecologists, engineers, architects.
See as examples of distributed cognition.

Latour **quasi-objects** & **symmetry**:
social & formal, material vs.
modernist dichotomy.
Similar to Peirce's triadic semiosis, which
sought to reconcile nominalism & realism:
Hybrids.

Main math contributions:
Galoisification, $\mathbb{G}Th(\mathbb{O})$ functors;
3/2-colimits (with Rosu);
institutions (with Burstall).

Main other contributions: unification of:
FCA, LOT, IF, IFF, CF, CG, BT, CGST,
over any logic;
views of Peirce, Latour, Hutchins, Deacon.

Main problem:
Interdisciplinarity vs. specialialization.