

PROOF_CombinatoriaIDM.n3 - Sientific Knowledge Explorer

File Edit View Tools Plugins Help

document

Entrez OMM

1.2951294 0.9539342
5.8535447 1.0050368

Query Tool

SPARQL Endpoint: Currently Loaded Databases

Graphical Query Text Search SPARQL

Run Query

Query Graph

Clear

Load

Save

Graph Query

Patient	Protein Value	Value	Value
AR1	1.2951294	5.56968135	4.109822228
AR2	1.0509057	5.449994069	4.195062123
AR5	1.2796977	7.046012152	4.362954269

Graph Results

Import to Memory

Export

3 results returned.

Ready

Back Forward

Status: Ready

Relation

- hasEC
- hasEntrez
- hasGene
- hasPI Acc
- hasPatient
- hasPGC
- hasProtein

Protein TotalRecords

Knowledge Representation

Lyda Walling

First Edition, 2012

ISBN 978-81-323-3482-8

© All rights reserved.

Published by:

Research World

4735/22 Prakashdeep Bldg,

Ansari Road, Darya Ganj,

Delhi - 110002

Email: info@wtbooks.com

Table of Contents

Chapter 1 - Knowledge Representation and Reasoning

Chapter 2 - Belief Revision

Chapter 3 - Colon Classification

Chapter 4 - Default Logic

Chapter 5 - E-Services

Chapter 6 - Event calculus and Categorization

Chapter 7 - Figurative System of Human Knowledge

Chapter 8 - General Architecture for Text Engineering

Chapter 9 - Geopolitical Ontology

Chapter 10 - Minimum Information Standards and Defeasible Reasoning

Chapter 11 - Attempto Controlled English

Chapter 12 - Upper Ontology (information science)

Chapter-1

Knowledge Representation and Reasoning

Knowledge representation (KR) and reasoning' is an area of artificial intelligence whose fundamental goal is to represent knowledge in a manner that facilitates inferencing (i.e. drawing conclusions) from knowledge. It analyzes how to formally think - how to use a symbol system to represent a domain of discourse (that which can be talked about), along with functions that allow inference (formalized reasoning) about the objects. Generally speaking, some kind of logic is used both to supply formal semantics of how reasoning functions apply to symbols in the domain of discourse, as well as to how to supply operators such as quantifiers, modal operators, etc. that, along with an interpretation theory, give meaning to the sentences in the logic.

When we design a knowledge representation (and a knowledge representation system to interpret sentences in the logic in order to derive inferences from them) we have to make choices across a number of design spaces. The single most important decision to be made, is the *expressivity* of the KR. The more expressive, the easier and more compact it is to "say something". However, more expressive languages are harder to automatically derive inferences from. An example of a less expressive KR would be propositional logic. An example of a more expressive KR would be autoepistemic temporal modal logic. Less expressive KR's may be both complete and consistent (formally less expressive than set theory). More expressive KR's may be neither complete nor consistent.

The key problem is to find a KR and a supporting reasoning system that can make the inferences your application needs within the resource constraints appropriate to the problem at hand. Recent developments in KR have been driven by the Semantic Web, and have included development of XML-based knowledge representation languages and standards, including Resource Description Framework (RDF), RDF Schema, Topic Maps, DARPA Agent Markup Language (DAML), Ontology Inference Layer (OIL), and Web Ontology Language (OWL).

Overview

In field there are a number of representation techniques such as frames, rules, tagging, and semantic networks which have originated from theories of human information processing. Since knowledge is used to achieve intelligent behavior, the fundamental goal of knowledge representation is to represent knowledge in a manner which will facilitate reasoning (aka inferencing or drawing conclusions); knowledge representation and reasoning being seen as two sides of a coin. A good knowledge representation must be both declarative and procedural knowledge. What is knowledge representation can best be understood in terms of five distinct roles it plays, each crucial to the task at hand :

- A knowledge representation (KR) is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.
- It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world?
- It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation sanctions; and (iii) the set of inferences it recommends.
- It is a medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences.
- It is a medium of human expression, i.e., a language in which we say things about the world."

Some issues that arise in knowledge representation from an AI perspective are:

- How do people represent knowledge?
- What is the nature of knowledge?
- Should a representation scheme deal with a particular domain or should it be general purpose?
- How expressive is a representation scheme or formal language?
- Should the scheme be declarative or procedural?

There has been very little top-down discussion of the knowledge representation (KR) issues and research in this area is a well aged quillwork. There are well known problems such as "spreading activation" (this is a problem in navigating a network of nodes), "subsumption" (this is concerned with selective inheritance; e.g. an ATV can be thought of as a specialization of a car but it inherits only particular characteristics) and "classification." For example a tomato could be classified both as a fruit and a vegetable.

In the field of artificial intelligence, problem solving can be simplified by an appropriate choice of *knowledge representation*. Representing knowledge in some ways makes

certain problems easier to solve. For example, it is easier to divide numbers represented in Hindu-Arabic numerals than numbers represented as Roman numerals.

Characteristics

A good knowledge representation covers six basic characteristics:

- Coverage, which means the KR covers a breadth and depth of information. Without a wide coverage, the KR cannot determine anything or resolve ambiguities.
- Understandable by humans. KR is viewed as a natural language, so the logic should flow freely. It should support modularity and hierarchies of classes (Polar bears are bears, which are animals). It should also have simple primitives that combine in complex forms.
- Consistency. If John closed the door, it can also be interpreted as the door was closed by John. By being consistent, the KR can eliminate redundant or conflicting knowledge.
- Efficient
- Easy to modify and update.
- Supports the intelligent activity which uses the knowledge base

To gain a better understanding of why these characteristics represent a good knowledge representation, think about how an encyclopedia is structured. There are millions of articles (coverage), and they are sorted into categories, content types, and similar topics (understandable). It redirects different titles but same content to the same article (consistency). It is efficient, easy to add new pages or update existing ones, and allows users on their mobile phones and desktops to view its knowledge base.

History of knowledge representation and reasoning

In computer science, particularly artificial intelligence, a number of representations have been devised to structure information.

KR is most commonly used to refer to representations intended for processing by modern computers, and in particular, for representations consisting of explicit objects (the class of all elephants, or Clyde a certain individual), and of assertions or claims about them ('Clyde is an elephant', or 'all elephants are grey'). Representing knowledge in such explicit form enables computers to draw conclusions from knowledge already stored ('Clyde is grey').

Many KR methods were tried in the 1970s and early 1980s, such as heuristic question-answering, neural networks, theorem proving, and expert systems, with varying success. Medical diagnosis (e.g., Mycin) was a major application area, as were games such as chess.

In the 1980s formal computer knowledge representation languages and systems arose. Major projects attempted to encode wide bodies of general knowledge; for example the

"Cyc" project (still ongoing) went through a large encyclopedia, encoding not the information itself, but the information a reader would need in order to understand the encyclopedia: naive physics; notions of time, causality, motivation; commonplace objects and classes of objects.

Through such work, the difficulty of KR came to be better appreciated. In computational linguistics, meanwhile, much larger databases of language information were being built, and these, along with great increases in computer speed and capacity, made deeper KR more feasible.

Several programming languages have been developed that are oriented to KR. Prolog developed in 1972, but popularized much later, represents propositions and basic logic, and can derive conclusions from known premises. KL-ONE (1980s) is more specifically aimed at knowledge representation itself. In 1995, the Dublin Core standard of metadata was conceived.

In the electronic document world, languages were being developed to represent the structure of documents, such as SGML (from which HTML descended) and later XML. These facilitated information retrieval and data mining efforts, which have in recent years begun to relate to knowledge representation.

Development of the Semantic Web, has included development of XML-based knowledge representation languages and standards, including RDF, RDF Schema, Topic Maps, DARPA Agent Markup Language (DAML), Ontology Inference Layer (OIL), and Web Ontology Language (OWL).

Topics in Knowledge representation and reasoning

Language and notation

Some think it is best to represent knowledge in the same way that it is represented in the human mind, or to represent knowledge in the form of human language.

Psycholinguistics investigates how the human mind stores and manipulates language. Other branches of cognitive science examine how human memory stores sounds, sights, smells, emotions, procedures, and abstract ideas. Science has not yet completely described the internal mechanisms of the brain to the point where they can simply be replicated by computer programmers.

Various artificial languages and notations have been proposed for representing knowledge. They are typically based on logic and mathematics, and have easily parsed grammars to ease machine processing. They usually fall into the broad domain of ontologies.

Ontology Engineering

After CycL, a number of ontology languages have been developed. Most are declarative languages, and are either frame languages, or are based on first-order logic. Most of these languages only define an upper ontology with generic concepts, whereas the domain concepts are not part of the language definition. These languages all use special-purpose knowledge engineering because as stated by Tom Gruber, "Every ontology is a treaty- a social agreement among people with common motive in sharing." There are always many competing and differing views that make any general purpose ontology impossible. A general purpose ontology would have to be applicable in any domain and different areas of knowledge need to be unified. Gellish English is an example of an ontological language that includes a full engineering English Dictionary.

There is a long history of work attempting to build good ontologies for a variety of task domains, including early work on an ontology for liquids , the lumped element model widely used in representing electronic circuits (e.g.,), as well as ontologies for time, belief, and even programming itself. Each of these offers a way to see some part of the world. The lumped element model, for instance, suggests that we think of circuits in terms of components with connections between them, with signals flowing instantaneously along the connections. This is a useful view, but not the only possible one. A different ontology arises if we need to attend to the electrodynamics in the device: Here signals propagate at finite speed and an object (like a resistor) that was previously viewed as a single component with an I/O behavior may now have to be thought of as an extended medium through which an electromagnetic wave flows.

Ontologies can of course be written down in a wide variety of languages and notations (e.g., logic, LISP, etc.); the essential information is not the form of that language but the content, i.e., the set of concepts offered as a way of thinking about the world. Simply put, the important part is notions like connections and components, not whether we choose to write them as predicates or LISP constructs.

The commitment we make by selecting one or another ontology can produce a sharply different view of the task at hand. Consider the difference that arises in selecting the lumped element view of a circuit rather than the electrodynamic view of the same device. As a second example, medical diagnosis viewed in terms of rules (e.g., MYCIN) looks substantially different from the same task viewed in terms of frames (e.g., INTERNIST). Where MYCIN sees the medical world as made up of empirical associations connecting symptom to disease, INTERNIST sees a set of prototypes, in particular prototypical diseases, to be matched against the case at hand.

Commitment begins with the earliest choices

The INTERNIST example also demonstrates that there is significant and unavoidable ontological commitment even at the level of the familiar representation technologies. Logic, rules, frames, etc., each embody a viewpoint on the kinds of things that are important in the world. Logic, for instance, involves a (fairly minimal) commitment to viewing the world in terms of individual entities and relations between them. Rule-based systems view the world in terms of attribute-object-value triples and the rules of plausible inference that connect them, while frames have us thinking in terms of prototypical

objects. Each of these thus supplies its own view of what is important to attend to, and each suggests, conversely, that anything not easily seen in those terms may be ignored. This is of course not guaranteed to be correct, since anything ignored may later prove to be relevant. But the task is hopeless in principle--every representation ignores something about the world--hence the best we can do is start with a good guess. The existing representation technologies supply one set of guesses about what to attend to and what to ignore. Selecting any of them thus involves a degree of ontological commitment: the selection will have a significant impact on our perception of and approach to the task, and on our perception of the world being modeled.

The commitments accumulate in layers

The ontologic commitment of a representation thus begins at the level of the representation technologies and accumulates from there. Additional layers of commitment are made as we put the technology to work. The use of frame-like structures in INTERNIST offers an illustrative example. At the most fundamental level, the decision to view diagnosis in terms of frames suggests thinking in terms of prototypes, defaults, and a taxonomic hierarchy. But prototypes of what, and how shall the taxonomy be organized? An early description of the system shows how these questions were answered in the task at hand, supplying the second layer of commitment:

The knowledge base underlying the INTERNIST system is composed of two basic types of elements: disease entities and manifestations.... [It] also contains a...hierarchy of disease categories, organized primarily around the concept of organ systems, having at the top level such categories as "liver disease," "kidney disease," etc.

The prototypes are thus intended to capture prototypical diseases (e.g., a "classic case" of a disease), and they will be organized in a taxonomy indexed around organ systems. This is a sensible and intuitive set of choices but clearly not the only way to apply frames to the task; hence it is another layer of ontological commitment.

At the third (and in this case final) layer, this set of choices is instantiated: which diseases will be included and in which branches of the hierarchy will they appear? Ontologic questions that arise even at this level can be quite fundamental. Consider for example determining which of the following are to be considered diseases (i.e., abnormal states requiring cure): alcoholism, homosexuality, and chronic fatigue syndrome. The ontologic commitment here is sufficiently obvious and sufficiently important that it is often a subject of debate in the field itself, quite independent of building automated reasoners. Similar sorts of decisions have to be made with all the representation technologies, because each of them supplies only a first order guess about how to see the world: they offer a way of seeing but don't indicate how to instantiate that view. As frames suggest prototypes and taxonomies but do not tell us which things to select as prototypes, rules suggest thinking in terms of plausible inferences, but don't tell us which plausible inferences to attend to. Similarly logic tells us to view the world in terms of individuals and relations, but does not specify which individuals and relations to use. Commitment to a particular view of the world thus starts with the choice of a

representation technology, and accumulates as subsequent choices are made about how to see the world in those terms.

Reminder: A KR is not a data structure

Note that at each layer, even the first (e.g., selecting rules or frames), the choices being made are about representation, not data structures. Part of what makes a language representational is that it carries meaning, i.e., there is a correspondence between its constructs and things in the external world. That correspondence in turn carries with it constraint. A semantic net, for example, is a representation, while a graph is a data structure. They are different kinds of entities, even though one is invariably used to implement the other, precisely because the net has (should have) a semantics. That semantics will be manifest in part because it constrains the network topology: a network purporting to describe family memberships as we know them cannot have a cycle in its parent links, while graphs (i.e., data structures) are of course under no such constraint and may have arbitrary cycles.

While every representation must be implemented in the machine by some data structure, the representational property is in the correspondence to something in the world and in the constraint that

Links and structures

While hyperlinks have come into widespread use, the closely related semantic link is not yet widely used. The mathematical table has been used since Babylonian times. More recently, these tables have been used to represent the outcomes of logic operations, such as truth tables, which were used to study and model Boolean logic, for example. Spreadsheets are yet another tabular representation of knowledge. Other knowledge representations are trees, graphs and hypergraphs, by means of which the connections among fundamental concepts and derivative concepts can be shown.

Visual representations are relatively new in the field of knowledge management but give the user a way to visualise how one thought or idea is connected to other ideas enabling the possibility of moving from one thought to another in order to locate required information.

Notation

The recent fashion in knowledge representation languages is to use XML as the low-level syntax. This tends to make the output of these KR languages easy for machines to parse, at the expense of human readability and often space-efficiency.

First-order predicate calculus is commonly used as a mathematical basis for these systems, to avoid excessive complexity. However, even simple systems based on this simple logic can be used to represent data that is well beyond the processing capability of current computer systems.

Examples of notations:

- DATR is an example for representing lexical knowledge
- RDF is a simple notation for representing relationships between and among objects

Storage and manipulation

One problem in knowledge representation is how to store and manipulate knowledge in an information system in a formal way so that it may be used by mechanisms to accomplish a given task. Examples of applications are expert systems, machine translation systems, computer-aided maintenance systems and information retrieval systems (including database front-ends).

Semantic networks may be used to represent knowledge. Each node represents a concept and arcs are used to define relations between the concepts. The Conceptual graph model is probably the oldest model still alive. One of the most expressive and comprehensively described knowledge representation paradigms along the lines of semantic networks is MultiNet (an acronym for Multilayered Extended Semantic Networks).

From the 1960s, the knowledge frame or just *frame* has been used. Each frame has its own name and a set of **attributes**, or **slots** which contain values; for instance, the frame for *house* might contain a *color* slot, *number of floors* slot, etc.

Using frames for expert systems is an application of object-oriented programming, with inheritance of features described by the "is-a" link. However, there has been no small amount of inconsistency in the usage of the "is-a" link: Ronald J. Brachman wrote a paper titled "What IS-A is and isn't", wherein 29 different semantics were found in projects whose knowledge representation schemes involved an "is-a" link. Other links include the "part-of" link.

Frame structures are well-suited for the representation of schematic knowledge and stereotypical cognitive patterns. The elements of such schematic patterns are weighted unequally, attributing higher weights to the more typical elements of a schema. A pattern is activated by certain expectations: If a person sees a big bird, he or she will classify it rather as a sea eagle than a golden eagle, assuming that his or her "sea-scheme" is currently activated and his "land-scheme" is not.

Frame representations are object-centered in the same sense as semantic networks are: All the facts and properties connected with a concept are located in one place - there is no need for costly search processes in the database.

A behavioral script is a type of frame that describes what happens temporally; the usual example given is that of describing going to a restaurant. The steps include waiting to be seated, receiving a menu, ordering, etc. The different solutions can be arranged in a so-called semantic spectrum with respect to their semantic expressivity.

Chapter-2

Belief Revision

Belief revision is the process of changing beliefs to take into account a new piece of information. The logical formalization of belief revision is researched in philosophy, in databases, and in artificial intelligence for the design of rational agents.

What makes belief revision non-trivial is that several different ways for performing this operation may be possible. For example, if the current knowledge includes the three facts "*A* is true", "*B* is true" and "if *A* and *B* are true then *C* is true", the introduction of the new information "*C* is false" can be done preserving consistency only by removing at least one of the three facts. In this case, there are at least three different ways for performing revision. In general, there may be several different ways for changing knowledge.

Revision and update

Two kinds of changes are usually distinguished:

update

the new information is about the situation at present, while the old beliefs refer to the past; update is the operation of changing the old beliefs to take into account the change;

revision

both the old beliefs and the new information refer to the same situation; an inconsistency between the new and old information is explained by the possibility of old information being less reliable than the new one; revision is the process of inserting the new information into the set of old beliefs without generating an inconsistency.

The main assumption of belief revision is that of minimal change: the knowledge before and after the change should be as similar as possible. In the case of update, this principle formalizes the assumption of inertia. In the case of revision, this principle enforces as much information as possible to be preserved by the change.

Example

The following classical example shows that the operations to perform in the two settings of update and revision are not the same. The example is based on two different interpretations of the set of beliefs $\{a \vee b\}$ and the new piece of information a :

update

in this scenario, two satellites, Unit A and Unit B, orbit around Mars; the satellites are programmed to land while transmitting their status to Earth; Earth has received a transmission from one of the satellites, communicating that it is still in orbit; however, due to interference, it is not known which satellite sent the signal; subsequently, Earth receives the communication that Unit A has landed; this scenario can be modeled in the following way; two propositional variables a and b indicate that Unit A and Unit B, respectively, are still in orbit; the initial set of beliefs is $\{a \vee b\}$ (either one of the two satellites is still in orbit) and the new piece of information is $\neg a$ (Unit A has landed, and is therefore not in orbit); the only rational result of the update is $\neg a$; since the initial information that one of the two satellites had not landed yet was possibly coming from the Unit A, the position of the Unit B is not known;

revision

the play "Six Characters in Search of an Author" will be performed in one of the two local theatres; this information can be denoted by $\{a \vee b\}$, where a and b indicates that the play will be performed at the first or at the second theatre, respectively; a further information that "Jesus Christ Superstar" will be performed at the first theatre indicates that $\neg a$ holds; in this case, the obvious conclusion is that "Six Characters in Search of an Author" will be performed at the second but not the first theatre, which is represented in logic by $\neg a \wedge b$.

This example shows that revising the belief $a \vee b$ with the new information $\neg a$ produces two different results $\neg a$ and $\neg a \wedge b$ depending on whether the setting is that of update or revision.

Contraction, expansion, revision, consolidation, and merging

In the setting in which all beliefs refer to the same situation, a distinction between various operations that can be performed is made:

contraction

removal of a belief;

expansion

addition of a belief without checking consistency;

revision

addition of a belief while maintaining consistency;

consolidation

restoring consistency of a set of beliefs;

merging

fusion of two or more sets of beliefs while maintaining consistency.

Revision and merging differ in that the first operation is done when the new belief to incorporate is considered more reliable than the old ones; therefore, consistency is maintained by removing some of the old beliefs. Merging is a more general operation, in that the priority among the belief sets may or may not be the same.

Revision can be performed by first incorporating the new fact and then restoring consistency via consolidation. This is actually a form of merging rather than revision, as the new information is not always treated as more reliable than the old knowledge.

The AGM postulates

The AGM postulates (named after the names of their proponents, Alchourrón, Gärdenfors, and Makinson) are properties that an operator that performs revision should satisfy in order for being considered rational. The considered setting is that of revision, that is, different pieces of information referring to the same situation. Three operations are considered: expansion (addition of a belief without a consistency check), revision (addition of a belief while maintaining consistency), and contraction (removal of a belief).

The first six postulates are called "the basic AGM postulates". In the settings considered by Alchourrón, Gärdenfors, and Makinson, the current set of beliefs is represented by a deductively closed set of logical formulae K called belief base, the new piece of information is a logical formula P , and revision is performed by a binary operator $*$ that takes as its operands the current beliefs and the new information and produces as a result a belief base representing the result of the revision. The $+$ operator denoted expansion: $K + P$ is the deductive closure of $K \cup \{P\}$. The AGM postulates for revision are:

1. $K * P$ is a belief base (i.e., a deductively closed set of formulae);
2. $P \in K * P$
3. $K * P \subseteq K + P$
4. If $(\neg P) \notin K$, then $K * P = K + P$
5. $K * P$ is inconsistent only if P is inconsistent
6. If P and Q are logically equivalent, then $K * P = K * Q$
7. $K * (P \wedge Q) \subseteq (K * P) + Q$
8. If $(\neg Q) \notin K * P$ then $(K * P) + Q \subseteq K * (P \wedge Q)$

A revision operator that satisfies all eight postulates is the full meet revision, in which $K * P$ is equal to $K + P$ if consistent, and to the deductive closure of P otherwise. While satisfying all AGM postulates, this revision operator has been considered to be too conservative, in that no information from the old knowledge base is maintained if the revising formula is inconsistent with it.

Conditions equivalent to the AGM postulates

The AGM postulates are equivalent to several different conditions on the revision operator; in particular, they are equivalent to the revision operator being definable in terms of structures known as selection functions, epistemic entrenchments, systems of spheres, and preference relations. The latter are reflexive, transitive, and total relations over the set of models.

Each revision operator $*$ satisfying the AGM postulates is associated to a set of preference relations \leq_K , one for each possible belief base K , such that the models of K are exactly the minimal of all models according to \leq_K . The revision operator and its associated family of orderings are related by the fact that $K * P$ is the set of formulae whose set of models contains all the minimal models of P according to \leq_K . This condition is equivalent to the set of models of $K * P$ being exactly the set of the minimal models of P according to the ordering \leq_K .

A preference ordering \leq_K represents an order of unplausibility among all situations, including those that are conceivable but yet currently considered false. The minimal models according to such an ordering are exactly the models of the knowledge base, which are the models that are currently considered the most likely. All other models are greater than these ones, and are indeed considered less plausible. In general, $I <_K J$ indicates that the situation represented by the model I is believed to be more plausible than the situation represented by J . As a result, revising by a formula having I and J as models should select only I to be a model of the revised knowledge base, as this model represent the most likely scenario among those supported by P .

Contraction

Contraction is the operation of removing a belief P from a knowledge base K ; the result of this operation is denoted by $K - P$. The operators of revision and contractions are related by the Levi and Harper identities:

$$\begin{aligned} K * P &= (K - \neg P) + P \\ K - P &= K \cap (K * \neg P) \end{aligned}$$

Eight postulates have been defined for contraction. Whenever a revision operator satisfies the eight postulates for revision, its corresponding contraction operator satisfies the eight postulates for contraction, and vice versa. If a contraction operator satisfies at least the first six postulates for contraction, translating it into a revision operator and then back into a contraction operator using the two identities above leads to the original contraction operator. The same holds starting from a revision operator.

One of the postulates for contraction has been longly discussed: the recovery postulate:

$$K = (K - P) + P$$

According to this postulate, the removal of a belief P followed by the reintroduction of the same belief in the belief base should lead to the original belief base. There are some examples showing that such behavior is not always reasonable: in particular, the contraction by a general condition such as $a \vee b$ leads to the removal of more specific conditions such as a from the belief base; it is then unclear why the reintroduction of $a \vee b$ should also lead to the reintroduction of the more specific condition a . For example, if George was previously believed to have German citizenship, it was also believed to be European. Contracting this latter belief amounts to stop believing that George is European; therefore, that George has German citizenship is also retracted from the belief base. If George is later discovered to have Austrian citizenship, then the fact that he is European is also reintroduced. According to the recovery postulate, however, the belief that he also has German citizenship should also be reintroduced.

The correspondence between revision and contraction induced by the Levi and Harper identities is such that a contraction not satisfying the recovery postulate is translated into a revision satisfying all eight postulates, and that a revision satisfying all eight postulates is translated into a contraction satisfying all eight postulates, including recovery. As a result, if recovery is excluded from consideration, a number of contraction operators are translated into a single revision operator, which can be then translated back into exactly one contraction operator. This operator is the only one of the initial group of contraction operators that satisfies recovery; among this group, it is the operator that preserves as much information as possible.

The Ramsey test

The evaluation of a counterfactual conditional $a > b$ can be done, according to the Ramsey test, to the hypothetical addition of a to the set of current beliefs followed by a check for the truth of b . If K is the set of beliefs currently held, the Ramsey test is formalized by the following correspondence:

$$a > b \text{ if and only if } b \in K * a$$

If the considered language of the formulae representing beliefs is propositional, the Ramsey test gives a consistent definition for counterfactual conditionals in terms of a belief revision operator. However, if the language of formulae representing beliefs itself includes the counterfactual conditional connective $>$, the Ramsey test leads to the Gardnefors triviality result: there is no non-trivial revision operator that satisfies both the AGM postulates for revision and the condition of the Ramsey test. This result holds in the assumption that counterfactual formulae like $a > b$ can be present in belief bases and revising formulae. Several solutions to this problem have been proposed.

Non-monotonic inference relation

Given a fixed knowledge base K and a revision operator $*$, one can define a non-monotonic inference relation using the following definition: $P \vdash Q$ if and only if

$K * P \models Q$. In other words, a formula P entails another formula Q if the addition of the first formula to the current knowledge base leads to the derivation of Q . This inference relation is non-monotonic.

The AGM postulates can be translated into a set of postulates for this inference relation. Each of these postulates is entailed by some previously considered set of postulates for non-monotonic inference relations. Vice versa, conditions that have been considered for non-monotonic inference relations can be translated into postulates for a revision operator. All these postulates are entailed by the AGM postulates.

Foundational revision

In the AGM framework, a belief set is represented by a deductively closed set of propositional formulae. While such sets are infinite, they can always be finitely representable. However, working with deductively closed sets of formulae leads to the implicit assumption that equivalent belief bases should be considered equal when revising. This is called the *principle of irrelevance of syntax*.

This principle has been and is currently debated: while $\{a, b\}$ and $\{a \wedge b\}$ are two equivalent sets, revising by $\neg a$ should produce different results. In the first case, a and b are two separate beliefs; therefore, revising by $\neg a$ should not produce any effect on b , and the result of revision is $\{\neg a, b\}$. In the second case, $a \wedge b$ is taken a single belief. The fact that a is false contradicts this belief, which should therefore be removed from the belief base. The result of revision is therefore $\{\neg a\}$ in this case.

The problem of using deductively closed knowledge bases is that no distinction is made between pieces of knowledge that are known by themselves and pieces of knowledge that are merely consequences of them. This distinction is instead done by the *foundational* approach to belief revision, which is related to foundationalism in philosophy. According to this approach, retracting a non-derived piece of knowledge should lead to retracting all its consequences that are not otherwise supported (by other non-derived pieces of knowledge). This approach can be realized by using knowledge bases that are not deductively closed and assuming that all formulae in the knowledge base represent self-standing beliefs, that is, they are not derived beliefs. In order to distinguish the foundational approach to belief revision to that based on deductively closed knowledge bases, the latter is called the *coherentist* approach. This name has been chosen because the coherentist approach aims at restoring the coherence (consistency) among *all* beliefs, both self-standing and derived ones. This approach is related to coherentism in philosophy.

Foundationalist revision operators working on non-deductively closed belief bases typically select some subsets of K that are consistent with P , combined them in some way, and then conjoined them with P . The following are two non-deductively closed base revision operators.

WIDTIO

(When in Doubt, Throw it Out) the maximal subsets of K that are consistent with P are intersected, and P is added to the resulting set; in other words, the result of revision is composed by P and of all formulae of K that are in all maximal subsets of K that are consistent with P ;

Ginsberg-Fagin-Ullman-Vardi

the maximal subsets of $K \cup \{P\}$ that are consistent and contain P are combined by disjunction;

Nebel

similar to the above, but a priority among formulae can be given, so that formulae with higher priority are less likely to being retracted than formulae with lower priority.

A different realization of the foundational approach to belief revision is based on explicitly declaring the dependences among beliefs. In the truth maintenance systems, dependence links among beliefs can be specified. In other words, one can explicitly declare that a given fact is believed because of one or more other facts; such a dependency is called a *justification*. Beliefs not having any justifications play the role of non-derived beliefs in the non-deductively closed knowledge base approach.

Model-based revision and update

A number of proposals for revision and update based on the set of models of the involved formulae were developed independently of the AGM framework. The principle behind this approach is that a knowledge base is equivalent to a set of *possible worlds*, that is, to a set of scenarios that are considered possible according to that knowledge base. Revision can therefore be performed on the sets of possible worlds rather than on the corresponding knowledge bases.

The revision and update operators based on models are usually identified by the name of their authors: Winslett, Forbus, Satoh, Dalal, Hegner, and Weber. According to the first four of these proposal, the result of revising/updating a formula K by another formula P is characterized by the set of models of P that are the closest to the models of K . Different notions of closeness can be defined, leading to the difference among these proposals.

Dalal

the models of P having a minimal Hamming distance to models of K are selected to be the models that result from the change;

Satoh

similar to Dalal, but distance between two models is defined as the set of literals that are given different values by them; similarity between models is defined as set containment of these differences;

Winslett

for each model of K , the closest models of P are selected; comparison is done using set containment of the difference;

Borgida

equal to Winslett's if K and P are inconsistent; otherwise, the result of revision is $K \wedge P$;

Forbus

similar to Winslett, but the Hamming distance is used.

The revision operator defined by Hegner makes K not to affect the value of the variables that are mentioned in P . What results from this operation is a formula K' that is consistent with P , and can therefore be conjoined with it. The revision operator by Weber is similar, but the literals that are removed from K are not all literals of P , but only the literals that are evaluated differently by a pair of closest models of K and P according to the Satoh measure of closeness.

Iterated revision

The AGM postulates are equivalent to a preference ordering (an ordering over models) to be associated to every knowledge base K . However, they do not relate the orderings corresponding to two non-equivalent knowledge bases. In particular, the orderings associated to a knowledge base K and its revised version $K * P$ can be completely different. This is a problem for performing a second revision, as the ordering associated with $K * P$ is necessary to calculate $K * P * Q$.

Establishing a relation between the ordering associated with K and $K * P$ has been however recognized not to be the right solution to this problem. Indeed, the preference relation should depend on the previous history of revisions, rather than on the resulting knowledge base only. More generally, a preference relation gives more information about the state of mind of an agent than a simple knowledge base. Indeed, two states of mind might represent the same piece of knowledge K while at the same time being different in the way a new piece of knowledge would be incorporated. For example, two people might have the same idea as to where to go on holiday, but yet they differ on how they would change this idea if they win a million-dollar lottery. Since the basic condition of the preference ordering is that their minimal models are exactly the models of their associated knowledge base, a knowledge base can be considered implicitly represented by a preference ordering (but not vice versa).

Given that a preference ordering allows deriving its associated knowledge base but also allows performing a single step of revision, studies on iterated revision have been concentrated on how a preference ordering should be changed in response of a revision. While single-step revision is about how a knowledge base K has to be changed into a new knowledge base $K * P$, iterated revision is about how a preference ordering (representing both the current knowledge and how much situations believed to be false are considered possible) should be turned into a new preference relation when P is learned. A single step of iterated revision produces a new ordering that allows for further revisions.

Two kinds of preference ordering are usually considered: numerical and non-numerical. In the first case, the level of plausibility of a model is representing by a non-negative integer number; the lower the rank, the more plausible the situation corresponding to the

model. Non-numerical preference orderings correspond to the preference relations used in the AGM framework: a possibly total ordering over models. The non-numerical preference relation were initially considered unsuitable for iterated revision because of the impossibility of reverting a revision by a number of other revisions, which is instead possible in the numerical case.

Darwiche and Pearl formulated the following postulates for iterated revision.

1. if $\alpha \models \mu$ then $(\psi * \mu) * \alpha \equiv \psi * \alpha$;
2. if $\alpha \models \neg\mu$, then $(\psi * \mu) * \alpha \equiv \psi * \alpha$;
3. if $\psi * \alpha \models \mu$, then $(\psi * \mu) * \alpha \models \mu$;
4. if $\psi * \alpha \not\models \neg\mu$, then $(\psi * \mu) * \alpha \not\models \neg\mu$.

Specific iterated revision operators have been proposed by Spohn, Boutilier, Williams, Lehmann, and others.

Spohn rejected revision

this non-numerical proposal has been first considered by Spohn, who rejected it based on the fact that revisions can change some orderings in such a way the original ordering cannot be restored with a sequence of other revisions; this operator change a preference ordering in view of new information P by making all models of P being preferred over all other models; the original preference ordering is maintained when comparing two models that are both models of P or both non-models of P ;

Natural revision

while revising a preference ordering by a formula P , all minimal models (according to the preference ordering) of P are made more preferred by all other ones; the original ordering of models is preserved when comparing two models that are not minimal models of P ; this operator changes the ordering among models minimally while preserving the property that the models of the knowledge base after revising by P are the minimal models of P according to the preference ordering;

Transmutations

these are two forms of revision, conditionalization and adjustment, which work on numerical preference orderings; revision requires not only a formula but also a number indicating its degree of plausibility; while the preference ordering is still inverted (the lower a model, the most plausible it is) the degree of plausibility of a revising formula is direct (the higher the degree, the most believed the formula is);

Ranked revision

a ranked model, which is an assignment of non-negative integers to models, has to be specified at the beginning; this rank is similar to a preference ordering, but is not changed by revision; what is changed by a sequence of revisions are a current set of models (representing the current knowledge base) and a number called the rank of the sequence; since this number can only monotonically non-decrease,

some sequences of revision lead to situations in which every further revision is performed as a full meet revision.

Merging

The assumption implicit in the revision operator is that the new piece of information P is always to be considered more reliable than the old knowledge base K . This is formalized by the second of the AGM postulates: P is always believed after revising K with P . More generally, one can consider the process of merging several pieces of information (rather than just two) that might or might not have the same reliability. Revision becomes the particular instance of this process when a less reliable piece of information K is merged with a more reliable P .

While the input to the revision process is a pair of formulae K and P , the input to merging is a multiset of formulae K, T , etc. The use of multisets is necessary as two sources to the merging process might be identical.

When merging a number of knowledge bases with the same degree of plausibility, a distinction is made between arbitration and majority. This distinction depends on the assumption that is made about the information and how it has to be put together.

arbitration

the result of arbitrating two knowledge bases K and T entails $K \vee T$; this condition formalizes the assumption of maintaining as much as the old information as possible, as it is equivalent to imposing that every formula entailed by both knowledge bases is also entailed by the result of their arbitration; in a possible world view, the "real" world is assumed one of the worlds considered possible according to at least one of the two knowledge bases;

majority

the result of merging a knowledge base K with other knowledge bases can be forced to entail K by adding a sufficient number of other knowledge bases equivalent to K ; this condition corresponds to a kind of vote-by-majority: a sufficiently large number of knowledge bases can always overcome the "opinion" of any other fixed set of knowledge bases.

The above is the original definition of arbitration. According to a newer definition, an arbitration operator is a merging operator that is insensitive to the number of equivalent knowledge bases to merge. This definition makes arbitration the exact opposite of majority.

Postulates for both arbitration and merging have been proposed. An example of an arbitration operator satisfying all postulates is the classical disjunction. An example of a majority operator satisfying all postulates is that selecting all models that have a minimal total Hamming distance to models of the knowledge bases to merge.

A merging operator can be expressed as a family of orderings over models, one for each possible multiset of knowledge bases to merge: the models of the result of merging a multiset of knowledge bases are the minimal models of the ordering associated to the multiset. A merging operator defined in this way satisfies the postulates for merging if and only if the family of orderings meets a given set of conditions. For the old definition of arbitration, the orderings are not on models but on pairs (or, in general, tuples) of models.

Social choice theory

Many revision proposals involve orderings over models representing the relative plausibility of the possible alternatives. The problem of merging amounts to combine a set of orderings into a single one expressing the combined plausibility of the alternatives. This is similar with what is done in social choice theory, which is the study of how the preferences of a group of agents can be combined in a rational way. Belief revision and social choice theory are similar in that they combine a set of orderings into one. They differ on how these orderings are interpreted: preferences in social choice theory; plausibility in belief revision. Another difference is that the alternatives are explicitly enumerated in social choice theory, while they are the propositional models over a given alphabet in belief revision.

Complexity

The problem about belief revision that is the most studied from the point of view of computational complexity is that of query answering in the propositional case. This is the problem of establishing whether a formula follows from the result of a revision, that is, $K * P \models Q$, where K , P , and Q are propositional formulae. More generally, query answering is the problem of telling whether a formula is entailed by the result of a belief revision, which could be update, merging, revision, iterated revision, etc. Another problem that has received some attention is that of model checking, that is, checking whether a model satisfies the result of a belief revision. A related question is whether such result can be represented in space polynomial in that of its arguments.

Since a deductively closed knowledge base is infinite, complexity studies on belief revision operators working on deductively closed knowledge bases are done in the assumption that such deductively closed knowledge base are given in the form of an equivalent finite knowledge base.

A distinction is made among belief revision operators and belief revision schemes. While the former are simple mathematical operators mapping a pair of formulae into another formula, the latter depend on further information such as a preference relation. For example, the Dalal revision is an operator because, once two formulae K and P are given, no other information is needed to compute $K * P$. On the other hand, revision based on a preference relation is a revision scheme, because K and P do not allow determining the result of revision if the family of preference orderings between models is not given. The complexity for revision schemes is determined in the assumption that the extra

information needed to compute revision is given in some compact form. For example, a preference relation can be represented by a sequence of formulae whose models are increasingly preferred. Explicitly storing the relation as a set of pairs of models is instead not a compact representation of preference because the space required is exponential in the number of propositional letters.

The complexity of query answering and model checking in the propositional case is in the second level of the polynomial hierarchy for most belief revision operators and schemas. Most revision operators suffer from the problem of representational blow up: the result of revising two formulae is not necessarily representable in space polynomial in that of the two original formulae. In other words, revision may exponentially increase the size of the knowledge base.

Chapter-3

Colon Classification

Colon classification (CC) is a system of library classification developed by S. R. Ranganathan. It was the first ever faceted (or analytico-synthetic) classification. The first edition was published in 1933. Since then six more editions have been published. It is especially used in libraries in India.

Its name "Colon classification" comes from the use of colons to separate facets in class numbers. However, many other classification schemes, some of which are completely unrelated, also use colons and other punctuation in various functions. They should not be confused with Colon classification.

As an example, the subject "research in the cure of tuberculosis of lungs by x-ray conducted in India in 1950" results in a call number

L,45;421:6;253:f.44'N5

The components of this call number represent

Medicine,Lungs;Tuberculosis:Treatment;X-ray:Research.India'1950

Organization

The Colon Classification uses 42 main classes that are combined with other letters, numbers and marks in a manner resembling the Library of Congress Classification to sort a publication.

Facets

CC uses five primary categories, or facets to further specify the sorting of a publication. Collectively, they are called *PMEST*:

, personality
; matter or property
: energy

. space
' time

Classes

The following are the main classes of CC, with some subclasses, the main method used to sort the subclass using the PMEST scheme and examples showing application of PMEST.

- z Generalia
 - 1 Universe of Knowledge
 - 2 Library Science
 - 3 Book science
 - 4 Journalism
- B Mathematics
 - B1 Arithmetic
 - B13 Theory of numbers
 - B2 Algebra
 - B23 Algebraic equations
 - B25 Higher Algebra
 - B3 Analysis
 - B33 Differential Equations *[equation]* , *[degree]* , *[order]* : *[problem]*
 - B331,1,2:1 Numerical solutions (:1) of ordered (331) linear (,1) second order (,2) differential equations
 - B37 Real Variables
 - B38 Complex Variables
 - B4 Other Methods
 - B6 Geometry
 - B7 Mechanics
 - B8 Physico-Mathematics
 - B9 Astronomical Mathematics
 - C Physics
 - C1 Fundamentals of Physics
 - C2 Properties of Matter
 - C3 Sound
 - C4 Heat
 - C5 Light and Radiation
 - C6 Electricity
 - C7 Magnetism
 - C8 Cosmic Hypothesis
 - D Engineering
 - E Chemistry
 - facets*
 - :1 General Chemistry
 - :2 Physical Chemistry
 - :3 Analytical Chemistry

:33 Qualitative Chemistry
:34 Quantitative Chemistry
:35 Volumetric Chemistry
:4 Synthetic Chemistry
:5 Extraction Chemistry
E1 Inorganic Chemistry
E10 Group 0
E11 Group 1
E110 Hydrogen
E1109 Lithium
E111 Sodium
E2 Hydroxyl (Base)
E3 Acid
E4 Salt
E5 Organic Substance
F Technology
G Biology
H Geology
H1 Mineralogy
H2 Petrology
H3 Structural Geology
H4 Dynamic Geology
H5 Stratigraphy
H6 Paleontology
H7 Economic Geology
H8 Cosmic Hypothesis
HX Mining
I Botany
J Agriculture
facets
:1 Soil
:3 Propagation
:4 Disease
:5 Development
:6 Breeding
:7 Harvest
:91 Nomenclature
:92 Morphology
:93 Physiology
:95 Ecology
material facets
,2 Bulb
,3 Root
,4 Stem
,5 Leaf
,6 Flower

,7 Fruit
,8 Seed
,97 Whole Plant
J1 Horticulture
J2 Feed
J3 Food
J4 Stimulant
J5 Oil
J6 Drug
J7 Fabric
J8 Dye
K Zoology (*same facet schedule as I Botany*)
KZ Animal Husbandry (*same facet schedule as I Botany*)
L Medicine [*organ*]:[*problem*],[*cause*]:[*handling*]
LZ3 Pharmacology [*substance*].[*action*],[*organ*]
LZ5 Pharmacopoeia
M Useful arts
M7 Textiles [*material*]:[*work*]
Δ Spiritual experience and mysticism [*religion*],[*entity*]:[*problem*]
N Fine arts
NA Architecture [*style*] , [*utility*] , [*part*] : [*technique*]
ND Sculpture [*style*] , [*figure*] ; [*material*] : [*technique*]
NN Engraving
NQ Painting [*style*] , [*figure*] ; [*material*] : [*technique*]
NR Music [*style*] , [*music*] ; [*instrument*] : [*technique*]
O Literature
P Linguistics
Q Religion
R Philosophy
S Psychology
T Education
U Geography
V History
W Political science
X Economics
Y Sociology
Z Law

Example

The most commonly cited example of the Colon Classification is the classification for:

- "Research in the cure of the tuberculosis of lungs by x-ray conducted in India in 1950s":
- Main classification is Medicine

- (Medicine)
- Within Medicine, the Lungs are the main concern
 - (Medicine,Lungs)
- The property of the Lungs is that they are afflicted with Tuberculosis
 - (Medicine,Lungs;Tuberculosis)
- The Tuberculosis is being performed (:) on, that is the intent is to cure (Treatment)
 - (Medicine,Lungs;Tuberculosis:Treatment)
- The matter that we are treating the Tuberculosis with are X-Rays
 - (Medicine,Lungs;Tuberculosis:Treatment;X-ray)
- And this discussion of treatment is regarding the Research phase
 - (Medicine,Lungs;Tuberculosis:Treatment;X-ray:Research)
- This Research is performed within a geographical space (.) namely India
 - (Medicine,Lungs;Tuberculosis:Treatment;X-ray:Research.India)
- During the time (') of 1950
 - (Medicine,Lungs;Tuberculosis:Treatment;X-ray:Research.India'1950)
- And translating into the codes listed for each subject and facet the classification becomes
 - L,45;421:6;253:f.44'N5

Chapter-4

Default Logic

Default logic is a non-monotonic logic proposed by Raymond Reiter to formalize reasoning with default assumptions.

Default logic can express facts like “by default, something is true”; by contrast, standard logic can only express that something is true or that something is false. This is a problem because reasoning often involves facts that are true in the majority of cases but not always. A classical example is: “birds typically fly”. This rule can be expressed in standard logic either by “all birds fly”, which is inconsistent with the fact that penguins do not fly, or by “all birds that are not penguins and not ostriches and ... fly”, which requires all exceptions to the rule to be specified. Default logic aims at formalizing inference rules like this one without explicitly mentioning all their exceptions.

Syntax of default logic

A default theory is a pair $\langle D, W \rangle$. W is a set of logical formulae, called *the background theory*, that formalize the facts that are known for sure. D is a set of *default rules*, each one being of the form:

$$\frac{\text{Prerequisite} : \text{Justification}_1, \dots, \text{Justification}_n}{\text{Conclusion}}$$

According to this default, if we believe that *Prerequisite* is true, and each of *Justification_i* is consistent with our current beliefs, we are led to believe that *Conclusion* is true.

The logical formulae in W and all formulae in a default were originally assumed to be first-order logic formulae, but they can potentially be formulae in an arbitrary formal logic. The case in which they are formulae in propositional logic is one of the most studied.

Examples

The default rule “birds typically fly” is formalized by the following default:

$$D = \left\{ \frac{Bird(X) : Flies(X)}{Flies(X)} \right\}$$

This rule means that, if X is a bird, and it can be assumed that it flies, then we can conclude that it flies. A background theory containing some facts about birds is the following one:

$$W = \{Bird(Condor), Bird(Penguin), \neg Flies(Penguin), Flies(Eagle)\}$$

According to this default rule, a condor flies because the precondition $Bird(Condor)$ is true and the justification $Flies(Condor)$ is not inconsistent with what is currently known. On the contrary, $Bird(Penguin)$ does not allow concluding $Flies(Penguin)$: even if the precondition of the default $Bird(Penguin)$ is true, the justification $Flies(Penguin)$ is inconsistent with what is known. From this background theory and this default, $Bird(Eagle)$ cannot be concluded because the default rule only allows deriving $Flies(X)$ from $Bird(X)$, but not vice versa. Deriving the antecedents of an inference rule from the consequences is a form of explanation of the consequences, and is the aim of abductive reasoning.

A common default assumption is that what is not known to be true is believed to be false. This is known as the Closed World Assumption, and is formalized in default logic using a default like the following one for every fact F .

$$\frac{: \neg F}{\neg F}$$

For example, the computer language Prolog uses a sort of default assumption when dealing with negation: if a negative atom cannot be proved to be true, then it is assumed to be false. Note, however, that Prolog uses the so-called negation as failure: when the interpreter has to evaluate the atom $\neg F$, it tries to prove that F is true, and conclude that $\neg F$ is true if it fails. In default logic, instead, a default having $\neg F$ as a justification can only be applied if $\neg F$ is consistent with the current knowledge.

Restrictions

A default is categorical or prerequisite-free if it has no prerequisite (or, equivalently, its prerequisite is tautological). A default is normal if it has a single justification that is equivalent to its conclusion. A default is supernormal if it is both categorical and normal. A default is seminormal if all its justifications entail its conclusion. A default theory is called categorical, normal, supernormal, or seminormal if all defaults it contains are categorical, normal, supernormal, or seminormal, respectively.

Semantics of default logic

A default rule can be applied to a theory if its precondition is entailed by the theory and its justifications are all *consistent with* the theory. The application of a default rule leads to the addition of its consequence to the theory. Other default rules may then be applied to the resulting theory. **When the theory is such that no other default can be applied, the theory is called an extension of the default theory.** The default rules may be applied in different order, and this may lead to different extensions. The Nixon diamond example is a default theory with two extensions:

$$\left\langle \left\{ \frac{Republican(X) : \neg Pacifist(X)}{\neg Pacifist(X)}, \frac{Quaker(X) : Pacifist(X)}{Pacifist(X)} \right\}, \{Republican(Nixon), Quaker(Nixon)\} \right\rangle$$

Since Nixon is both a Republican and a Quaker, both defaults can be applied. However, applying the first default leads to the conclusion that Nixon is not a pacifist, which makes the second default not applicable. In the same way, applying the second default we obtain that Nixon is a pacifist, thus making the first default not applicable. This particular default theory has therefore two extensions, one in which $Pacifist(Nixon)$ is true, and one in which $Pacifist(Nixon)$ is false.

The original semantics of default logic was based on the fixed point of a function. The following is an equivalent algorithmic definition. If a default contains formulae with free variables, it is considered to represent the set of all defaults obtained by giving a value to

$$\alpha : \beta_1, \dots, \beta_n$$

all these variables. A default $\frac{\alpha : \beta_1, \dots, \beta_n}{\gamma}$ is applicable to a propositional theory T if $T \models \alpha$ and all theories $T \cup \{\beta_i\}$ are consistent. The application of this default to T leads to the theory $T \cup \{\gamma\}$. An extension can be generated by applying the following algorithm:

```

T=W          /* current theory */
A=0          /* set of defaults applied so far */

              /* apply a sequence of defaults */
while there is a default d that is not in A and is applicable to T
  add the consequence of d to T
  add d to A

              /* final consistency check */
if
  for every default d in A
    T is consistent with all justifications of d
then
  output T

```

This algorithm is non-deterministic, as several defaults can alternatively be applied to a given theory T . In the Nixon diamond example, the application of the first default leads to a theory to which the second default cannot be applied and vice versa. As a result, two

extensions are generated: one in which Nixon is a pacifist and one in which Nixon is not a pacifist.

The final check of consistency of the justifications of all defaults that have been applied implies that some theories do not have any extensions. In particular, this happens whenever this check fails for every possible sequence of applicable defaults. The following default theory has no extension:

$$\left\langle \left\{ \begin{array}{l} : A(b) \\ \neg A(b) \end{array} \right\}, \emptyset \right\rangle$$

Since $A(b)$ is consistent with the background theory, the default can be applied, thus leading to the conclusion that $A(b)$ is false. This result however undermines the assumption that has been made for applying the first default. Consequently, this theory has no extensions.

$$\frac{\phi : \psi}{\psi}$$

In a normal default theory, all defaults are normal: each default has the form $\frac{\phi : \psi}{\psi}$. A normal default theory is guaranteed to have at least one extension. Furthermore, the extensions of a normal default theory are mutually inconsistent, i.e., inconsistent with each other.

Entailment

A default theory can have zero, one, or more extensions. Entailment of a formula from a default theory can be defined in two ways:

Skeptical

a formula is entailed by a default theory if it is entailed by all its extensions;

Credulous

a formula is entailed by a default theory if it is entailed by at least one of its extensions.

Thus, the Nixon diamond example theory has two extensions, one in which Nixon is a pacifist and one in which he is not a pacifist. Consequently, neither $Pacifist(Nixon)$ nor $\neg Pacifist(Nixon)$ are skeptically entailed, while both of them are credulously entailed. As this example shows, the credulous consequences of a default theory may be inconsistent with each other.

Alternative default inference rules

The following alternative inference rules for default logic are all based on the same syntax as the original system.

Justified

differs from the original one in that a default is not applied if thereby the set T becomes inconsistent with a justification of an applied default;

Concise

a default is applied only if its consequence is not already entailed by T (the exact definition is more complicated than this one; this is only the main idea behind it);

Constrained

a default is applied only if the set composed of the background theory, the justifications of all applied defaults, and the consequences of all applied defaults (including this one) is consistent;

Rational

similar to constrained default logic, but the consequence of the default to add is not considered in the consistency check;

Cautious

defaults that can be applied but are conflicting with each other (like the ones of the Nixon diamond example) are not applied.

The justified and constrained versions of the inference rule assign at least an extension to every default theory.

Variants of default logic

The following variants of default logic differ from the original one on both syntax and semantics.

Assertional variants

An assertion is a pair $\langle p : \{r_1, \dots, r_n\} \rangle$ composed of a formula and a set of formulae. Such a pair indicates that p is true while the formulae r_1, \dots, r_n have been assumed consistent to prove that p is true. An assertional default theory is composed of an assertional theory (a set of assertional formulae) called the background theory and a set of defaults defined as in the original syntax.

Whenever a default is applied to an assertional theory, the pair composed of its consequence and its set of justifications is added to the theory. The following semantics use assertional theories:

- Cumulative default logic
- Commitment to assumptions default logic
- Quasi-default logic

Weak extensions

rather than checking whether the preconditions are valid in the theory composed of the background theory and the consequences of the applied defaults, the preconditions are checked for validity in the extension that will be generated; in other words, the algorithm for generating extensions starts by guessing a theory and using it in place of the background theory; what results from the process of extension generation is actually an extension only if it is equivalent to the theory

guessed at the beginning. This variant of default logic is related in principle to autoepistemic logic, where a theory $\Box x \rightarrow x$ has the model in which x is true just because, assuming $\Box x$ true, the formula $\Box x \rightarrow x$ supports the initial assumption.

Disjunctive default logic

the consequence of a default is a set of formulae instead of a single formula. Whenever the default is applied, at least one of its consequences is nondeterministically chosen and made true.

Priorities on defaults

the relative priority of defaults can be explicitly specified; among the defaults that are applicable to a theory, only one of the most preferred ones can be applied. Some semantics of default logic do not require priorities to be explicitly specified; rather, more specific defaults (those that are applicable in fewer cases) are preferred over less specific ones.

Statistical variant

a statistical default is a default with an attached upper bound on its frequency of error; in other words, the default is assumed to be an incorrect inference rule in at most that fraction of times it is applied.

Translations

Default theories can be translated into theories in other logics and vice versa. The following conditions on translations have been considered:

Consequence-Preserving

the original and the translated theories have the same (propositional) consequences;

Faithful

this condition only makes sense when translating between two variants of default logic or between default logic and a logic in which a concept similar to extension exists, e.g., models in modal logic; a translation is faithful if there exists a mapping (typically, a bijection) between the extensions (or models) of the original and translated theories;

Modular

a translation from default logic to another logic is modular if the defaults and the background theory can be translated separately; moreover, the addition of formulae to the background theory only leads to adding the new formulae to the result of the translation;

Same-Alphabet

the original and translated theories are built on the same alphabet;

Polynomial

the running time of the translation or the size of the generated theory are required to be polynomial in the size of the original theory.

Translations are typically required to be faithful or at least consequence-preserving, while the conditions of modularity and same alphabet are sometimes ignored.

The translatability between propositional default logic and the following logics have been studied:

- classical propositional logic;
- autoepistemic logic;
- propositional default logic restricted to seminormal theories;
- alternative semantics of default logic;
- circumscription.

Translations exist or not depending on which conditions are imposed. Translations from propositional default logic to classical propositional logic cannot always generate a polynomially sized propositional theory, unless the polynomial hierarchy collapses. Translations to autoepistemic logic exists or not depending on whether modularity or the use of the same alphabet is required.

Complexity

The computational complexity of the following problems about default logic is known:

Existence of extensions

deciding whether a propositional default theory has at least one extension is NP^{NP} -complete;

Skeptical entailment

deciding whether a propositional default theory skeptically entails a propositional formula is $co - NP^{NP}$ -complete;

Credulous entailment

deciding whether a propositional default theory credulously entails a propositional formula is NP^{NP} -complete;

Extension checking

deciding whether a propositional formula is equivalent to an extension of a propositional default theory is $P^{NP[\log]}$ -complete;

Model checking

deciding whether a propositional interpretation is a model of an extension of a propositional default theory is NP^{NP} -complete.

Chapter-5

E-Services

The concept of **E-service** (short for electronic service), represents one prominent application of utilizing the use of Information and communication technologies (ICTs) in different areas. However, providing an exact definition of e-service is hard to come by as researchers have been using different definitions to describe e-service. Despite these different definitions, it can be argued that they all agree about the role of technology in facilitating the delivery of services which make them more of electronic services.

It seems compelling to adopt Rowley (2006) approach who defines e-services as: “...deeds, efforts or performances whose delivery is mediated by information technology. Such e-service includes the service element of e-tailing, customer support, and service delivery”. This definition reflect three main components- service provider, service receiver and the channels of service delivery (i.e., technology). For example, as concerned to public e-service, public agencies are the service provider and citizens as well as businesses are the service receiver. The channel of service delivery is the third requirement of e-service. Internet is the main channel of e-service delivery while other classic channels (e.g. telephone, call center, public kiosk, mobile phone, television) are also considered.

Origin of the term E-service

Since its conceptual inception in the late 1980s in Europe and formal introduction in 1993 by the US Government, the term ‘E-Government’ has now become one of the recognized research domains especially in the context of public policy and now has been rapidly gaining strategic importance in public sector modernization. E-service is one of the branches of this domain and its attention has also been creeping up among the practitioners and researchers.

E-service benefits

Lu (2001) identifies a number of benefits for e-services, some of these are:

- Accessing a greater customer base

- Broadening market reach
- Lowering of entry barrier to new markets and cost of acquiring new customers
- Alternative communication channel to customers
- Increasing services to customers
- Enhancing perceived company image
- Gaining competitive advantages
- Potential for increasing customer knowledge

E-service domain

The term ‘E-service’ has many applications and can be found in many disciplines. The two dominant application areas of e-services are:

E-Business (or E-Commerce): e-services mostly provided by businesses or Non-government Organizations (NGOs) (private sector).

E-government: e-services provided by government to citizens or business (public sector is the supply side). The use and description of the e-service in this page will be limited to the context of e-government only where of the e-service is usually associated with prefix “public”: Public e-services. In some cases, we will have to describe aspects that are related to both fields like some conferences or journals which cover the concept of “e-Service” in both domains of e-government and e-business.

Architecture

Depending on the types of services, there are certain functionalities required in the certain layers of e-service architectural framework, these are but not limited to – Data layer (data sources), processing layers (customer service systems, management systems, data warehouse systems, integrated customer content systems), exchange layer (Enterprise Application Integration– EAI), Interaction layer (integrating e-services), and presentation layer (customer interface through which the web pages and e-services are linked).

E-service Quality

Measuring service quality and service excellence are important in a competitive organizational environment. The SERVQUAL- service quality model is one of the widely used tools for measuring quality of the service on various aspects. The five attributes of this model are: reliability, responsiveness, assurance, tangibles, and empathy. The following table summarizes some major of these:

SERVQUAL	Kaynama & Black (2000)	Zeithaml (2002)	Janda et al. (2002)	Alawattagama & Wattedegama (2008)
Reliability	Content	Access	Access	Factual

Responsiveness	Access	Ease of navigation	Security	information Business information
Assurance	Navigation	Efficiency	Sensation	General information
Tangibles	Design	Flexibility	Information/content	Consumer - related information
Empathy	Response Background Personalization	Reliability Personalization Security/privacy Responsiveness Assurance/trust Site aesthetics Price knowledge		

The LIRNEasia study on benchmarking national telecom regulator websites focuses on content than on accessibility and ease of use, unlike the other studies mentioned here. Websites are increasingly important portals to government agencies, especially in the context of information society reforms. Stakeholders, including businesses, investors and even the general public, are interested in information produced by these agencies, and websites can help to increase their transparency and accountability. The quality of its website also demonstrates how advanced a regulatory agency is.

E-service cost factor

Some major cost factors are (Lu, 2001):

- Expense of setting up applications
- Maintaining applications
- Internet connection
- Hardware/software
- Security concerns
- legal issues
- Training; and
- Rapid technology changes

Practical Examples of E-Services in the Developing World

Information Technology is a powerful tool for accelerating economic development. Developing countries have focused on the development of ICT during the last two decades and as a result, it has been recognized that ICT is critical to economy and is as a catalyst of economic development. So, in recent years there seems to have been efforts

for providing various e-services in many developing countries since ICT is believed to offer considerable potential for the sustainable development of e-Government and as a result, e-Services.

Many government agencies in developed countries have taken progressive steps toward the web and ICT use, adding coherence to all local activities on the Internet, widening local access and skills, opening up interactive services for local debates, and increasing the participation of citizens on promotion and management of the territory (Graham and Aurigi, 1997).

But the potential for eGovernment in developing countries remains largely unexploited, even though. ICT is believed to offer considerable potential for the sustainable development of eGovernment. Different human, organizational and technological factors, issues and problems pertain in these countries, requiring focused studies and appropriate approaches. ICT, in general, is referred to as an “enabler”, but on the other hand it should also be regarded as a challenge and a peril in itself. The organizations, public or private, which ignore the potential value and use of ICT may suffer pivotal competitive disadvantages. Nevertheless, some eGovernment initiatives have flourished in developing countries too, e.g. Brazil, India, Chile, etc. What the experience in these countries shows, is that governments in the developing world can effectively exploit and appropriate the benefits of ICT, but eGovernment success entails the accommodation of certain unique conditions, needs and obstacles. The adaptive challenges of eGovernment go far beyond technology, they call for organizational structures and skills, new forms of leadership, transformation of public-private partnerships (Allen et al., 2001).

Following are a few examples regarding e-services in some developing countries:

E-services in Rwanda

Only a decade after emerging from the fastest genocide of the 20th Century, Rwanda, a small country in Eastern Central Africa, has become one of the continent’s leaders in, and model on, bridging the digital divide through e-government. Rwanda has undergone a rapid turnaround from one of the most technologically deficient countries only a decade ago to a country where legislative business is conducted online and wireless access to the Internet is available anywhere in the country. This is puzzling when viewed against the limited progress made in other comparable developing countries, especially those located in the same region, sub-Saharan Africa, where the structural and institutional constraints to e-government diffusion are similar.

E-services in South Africa

In the South African post-apartheid era, there continues to be high expectations of government in respect of improved delivery of service and of closer consultation with citizens. Such expectations are not unique to this country, and in this regard there is a need for governments to recognise that the implementation of e-government systems and e-services affords them the opportunity to enhance service delivery and good governance.

The implementation of e-Government has been widely acclaimed in that it provides new impetus to deliver services quickly and efficiently (Evans & Yen, 2006:208). In recognition of these benefits, various arms of the South African government have embarked on a number of e-government programmes for example the Batho Pele portal, SARS e-filing, the e-Natis system, electronic processing of grant applications from remote sites, and a large number of departmental information websites. Also a number of well publicised e-government ventures such as the latter, analysts and researchers consider the state of e-government in South Africa to be at rudimentary stages. There are various factors which collectively contribute to such an assessment. Amongst these, key factors relate to a lack of a clear strategy to facilitate uptake and adoption of e-government services as well as evaluation frameworks to assess expectations of citizens who are one of the primary user groups of these services.

E-services in Malaysia

E-Services is one of the pilot projects under the Electronic Government Flagship within the Multimedia Super Corridor (MSC) initiative. With E-Services, one can now conduct transactions with Government agencies, such as the Road Transport Department (RTD) and private utility companies such as Tenaga Nasional Berhad (TNB) and Telekom Malaysia Berhad (TM) through various convenient channels such as the eServices kiosks and internet. No more queuing, traffic jams or bureaucratic hassles and one can now conduct transaction at one's own convenience. Also, Electronic Labour Exchange (ELX) is one stop-centre for labor market information, as supervised by the Ministry of Human Resource (MOHR), to enable employers and job seekers to communicate on the same platform.

e-Syariah is the seventh project under the Electronic Government flagship application of the Multimedia Super Corridor (MSC). A case management system that integrates the processes related to management of cases for the Syariah Courts.

Challenges to E-services in the Developing World

The future of e-service is bright but some challenges remain. There are some challenges in e-service, as Sheth & Sharma (2007) identify, are:

- Low penetration of ICT especially in the developing countries;
- Fraud on the internet space which is estimated around 2.8 billion USD
- Privacy due the emergence of various types of spyware and security holes, and
- intrusive characteristics of the service (e.g. mobile phones based) as customers may not like to be contacted with the service providers at any time and at any place.

The first challenge and primary obstacle to the e-service platform will be penetration of the internet. In some developing countries, the access to the internet is limited and speeds are also limited. In these cases firms and customers will continue to use traditional platforms. The second issue of concern is fraud on the internet. It is anticipated that the

fraud on the e-commerce internet space costs \$2.8 billion. Possibility of fraud will continue to reduce the utilization of the internet. The third issue is of privacy. Due to both spyware and security holes in operating systems, there is concern that the transactions that consumers undertake have privacy limitations. For example, by stealthily following online activities, firms can develop fairly accurate descriptions of customer profiles. Possibility of privacy violations will reduce the utilizations of the internet. The final issue is that e-service can also become intrusive as they reduce time and location barriers of other forms of contract. For example, firms can contact people through mobile devices at any time and at any place. Customers do not take like the intrusive behavior and may not use the e-service platform. (Heiner and Iyer, 2007)

Major E-service Keywords

A considerable amount of research efforts already exists on the subject matter exploring different aspects of e-service and e-service delivery ; one worth noting effort is Rowley's study (2006) who did a review study on the e-service literature. The key finding of his study is that there is need to explore dimensions of e-service delivery not focusing only on service quality "In order to understand e-service experiences it is necessary to go beyond studies of e-service quality dimensions and to also take into account the inherent characteristics of e-service delivery and the factors that differentiate one service experience from another."

Some of the major keywords of e-service as found in the e-government research are as follows:

Acceptance

User acceptance of technology is defined according to Morris (1996, referred by Wu 2005, p. 1) as "the demonstrable willingness within a user group to employ information technology for the tasks it is designed to support". This definition can be brought into the context of e-service where acceptance can be defined as the users' willingness to use e-service or the willingness to decide when and how to use the e-service.

Accessibility

Users' ability to access to the e-service is important theme in the previous literature. For example, Huang (2003) finds that most of the websites in general fail to serve users with disabilities. Recommendation to improve accessibility is evident in previous literature including Jaeger (2006) who suggests the following to improve e-services' accessibility like: design for accessibility from the outset of website development, Involve users with disabilities in the testing of the site ...Focus on the benefits of an accessible Web site to all users.

Administrative literacy

According to Grönlund et al. (2007), for a simple e-service, the needs for knowledge and skills, content and procedures are considerably less. However, in complicated services there are needed to change some prevailed skills, such as replacing verbal skills with skill in searching for information online.

Benchmarking

This theme is concerned with establishing standards for measuring e-services or the best practices within the field. This theme also includes the international benchmarking of e-government services (UN reports, EU reports); much critic has been targeting these reports being incomprehensive and useless. According Bannister (2007) "... benchmarks are not a reliable tool for measuring real e-government progress. Furthermore, if they are poorly designed, they risk distorting government policies as countries may chase the benchmark rather than looking at real local and national needs"

Digital divide

Digital divide is considered one of the main barriers to implementing e-services; some people do not have means to access the e-services and some others do not know how to use the technology (or the e-service). According to Helbig et al. (2009), "we suggest E-Government and the digital divide should be seen as complementary social phenomena (i.e., demand and supply). Moreover, a serious e-government digital divide is that services mostly used by social elites."

E-readiness

Most of the reports and the established criteria focus on assessing the services in terms of infrastructure and public policies ignoring the citizen participation or e-readiness. According to by Shalini (2009), "the results of the research project reveal that a high index may be only indicating that a country is e-ready in terms of ICT infrastructure and info-structure, institutions, policies, and political commitment, but it is a very poor measure of the e-readiness of citizens. To summarize the findings, it can be said that Mauritius is ready but the Mauritians are not"

"E-readiness, as the Economist Intelligence Unit defines, is the measure of a country's ability to leverage digital channels for communication, commerce and government in order to further economic and social development. Implied in this measure is the extent to which the usage of communications devices and Internet services creates efficiencies for business and citizens, and the extent to which this usage is leveraged in the development of information and communications technology (ICT) industries. In general terms, the definition of e-readiness is relative, for instance depending on a country in question's priorities and perspective.

Efficiency

As opposed to effectiveness, efficiency is focused on the internal competence within the government departments when delivering e-services. There is a complaint that researchers focus more on effectiveness “There is an emerging trend seemingly moving away from the efficiency target and focusing on users and governance outcome. While the latter is worthwhile, efficiency must still remain a key priority for eGovernment given the budget constraints compounded in the future by the costs of an ageing population. Moreover, efficiency gains are those that can be most likely proven empirically through robust methodologies”

Security

Security is the most important challenge that faces the implementation of e-services because without a guarantee of privacy and security citizens will not be willing to take up e-government services. These security concerns, such as hacker attacks and the theft of credit card information, make governments hesitant to provide public online services. According to the GAO report of 2002 “security concerns present one of the toughest challenges to extending the reach of e-government. The rash of hacker attacks, Web page defacing, and credit card information being posted on electronic bulletin boards can make many federal agency officials—as well as the general public—reluctant to conduct sensitive government transactions involving personal or financial data over the Internet.” By and Large, Security is one of the major challenges that faces the implementation and development of electronic services. people want to be assured that they are safe when they are conducting online services and that their information will remain secure and confidential

Stakeholders

Axelsson et al. (2009) argue that the stakeholder concept-which was originally used in private firms-, can be used in public setting and in the context of e-government. According to them, several scholars have discussed the use of the stakeholder theory in public settings. The stakeholder theory suggests that need to focus on all the involved stakeholder s when designing the e-service; not only on the government and citizens.

Usability

Compared to Accessibility, There is sufficient literature that addresses the issue of usability; researchers have developed different models and methods to measure the usability and effectiveness of eGovernment websites. However, But still there is call to improve these measures and make it more compressive

“The word usability has cropped up a few times already in this unit. In the context of biometric identification, usability referred to the smoothness of enrollment and other tasks associated with setting up an identification system. A system that produced few false matches during enrollment of applicants was described as usable. Another meaning

of usability is related to the ease of use of an interface. Although this meaning of the term is often used in the context of computer interfaces, there is no reason to confine it to computers.”

Social, Cultural and Ethical Implications of E-services

The perceived effectiveness of e-Service can be influenced by public’s view of the social and cultural implications of e-Technologies and e-Service.

Impacts on Individuals’ Rights and Privacy – as more and more companies and government agencies use technology to collect, store, and make accessible data on individuals, privacy concerns have grown. Some companies monitor their employees' computer usage patterns in order to assess individual or workgroup performance. Technological advancements are also making it much easier for businesses, government and other individuals to obtain a great deal of information about an individual without their knowledge. There is a growing concern that access to a wide range of information can be dangerous within politically corrupt government agencies.

Impact on Jobs and Workplaces - in the early days of computers, management scientists anticipated that computers would replace human decision-makers. However, despite significant technological advances, this prediction is no longer a mainstream concern. At the current time, one of the concerns associated with computer usage in any organization (including governments) is the health risk – such as injuries related to working continuously on a computer keyboard. Government agencies are expected to work with regulatory groups in order to avoid these problems.

Potential Impacts on Society – despite some economic benefits of ICT to individuals, there is evidence that the computer literacy and access gap between the haves and have-nots may be increasing. Education and information access are more than ever the keys to economic prosperity, yet access by individuals in different countries is not equal - this social inequity has become known as the digital divide.

Impact on Social Interaction – advancements in ICT and e-Technology solutions have enabled many government functions to become automated and information to be made available online. This is a concern to those who place a high value on social interaction.

Information Security - technological advancements allow government agencies to collect, store and make data available online to individuals and organizations. Citizens and businesses expect to be allowed to access data in a flexible manner (at any time and from any location). Meeting these expectations comes at a price to government agencies where it concerns managing information – more specifically, ease of access; data integrity and accuracy; capacity planning to ensure the timely delivery of data to remote (possibly mobile) sites; and managing the security of corporate and public information.

E-service Awards

The benefits of e-services in advancing businesses efficiency and in promoting good governance are huge; recognizing the importance of these benefits has resulted in number of international awards that are dedicated to recognize the best designed e-services. In the section, we will provide description of some international awards

Best Online E-service in Europe

European eGovernment Awards program started 2003 to recognize the best online public service in Europe. The aim of Awards is to encourage the deployment of e-services and to bring the attention to best practices in the field. The winners of the European eGovernment Awards were announced in the award ceremony that took place at the Ministerial eGovernment Conference on 19 November 2009 (Sweden); the winners in their respective categories are:

- Category 1. eGovernment supporting the Single Market: EU-OPA, the European Order for Payment Application ( Austria and  Germany)
- Category 2a. eGovernment empowering citizens: Genvej ( Denmark)
- Category 2b. eGovernment empowering businesses: MEPA, the Public Administration eMarketplace ( Italy)
- Category 3. eGovernment enabling administrative efficiency and effectiveness: Licensing of Hunters via the “Multibanco” ATM Network ( Portugal)
- Public prize: SMS Information System ( Turkey)

Other Awards

eGovernment Excellence Awards  Bahrain(Started 2007) The program has three categories: Government Awards: Best eContent, Best eService, Best eProject, eEconomy, eEducation, eMaturity Business Awards: Best ICT solution Provider, eEconomy,eEducation Citizen Awards: Best eContent, eCitizen.

e-Service Awards  Philippines(Started 2001) Categories: Outstanding Client Application of the Year, Outstanding Customer Application of the year, Groundbreaking Technology of the Year, Most Progressive Homegrown Company of the Year.

Chapter-6

Event calculus and Categorization

Event calculus

The **event calculus** is a logical language for representing and reasoning about actions and their effects first presented by Robert Kowalski and Marek Sergot in 1986. It was extended by Murray Shanahan and Rob Miller in the 1990s. The basic components of the event calculus, as with other similar languages for reasoning about actions and change are fluents and actions. In the event calculus, one can specify the value of fluents at some given time points, the actions that took place at given time points, and their effects.

Fluents and actions

In the event calculus, fluents are reified. This means that statements are not formalized as predicates but as functions. A separate predicate *HoldsAt* is used to tell which fluents hold at a given time point. For example, *HoldsAt(on(box,table),t)* means that the box is on the table at time t ; in this formula, *HoldsAt* is a predicate while *on* is a function.

Actions are also represented as terms. The effects of actions are given using the predicates *Initiates* and *Terminates*. In particular, *Initiates(a,f,t)* means that, if the action represented by the term a is executed at time t , then the fluent f will be true after t . The *Terminates* predicate has a similar meaning, with the only difference being that f will be false and not true after t .

Domain-independent axioms

Like other languages for representing actions, the event calculus formalizes the correct evolution of the fluent via formulae telling the value of each fluent after an arbitrary action has been performed. The event calculus solves the frame problem in a way that is similar to the successor state axioms of the situation calculus: a fluent is true at time t if and only if it has been made true in the past and has not been made false in the meantime.

$$\text{HoldsAt}(f, t) \leftarrow [\text{Happens}(a, t_1) \wedge \text{Initiates}(a, f, t_1) \wedge (t_1 < t) \wedge \neg \text{Clipped}(t_1, f, t)]$$

This formula means that the fluent represented by the term f is true at time t if:

1. an action a has taken place: $\text{Happens}(a, t_1)$;
2. this took place in the past: $t_1 < t$;
3. this action has the fluent as an effect: $\text{Initiates}(a, f, t_1)$;
4. the fluent has not been made false in the meantime: $\text{Clipped}(t_1, f, t)$

A similar formula is used to formalize the opposite case in which a fluent is false at a given time. Other formulae are also needed for correctly formalizing fluents before they have been effects of an action. These formulae are similar to the above, but $\text{Happens}(a, t_1) \wedge \text{Initiates}(a, f, t_1)$ is replaced by $\text{HoldsAt}(f, t_1)$.

The *Clipped* predicate, stating that a fluent has been made false during an interval, can be axiomatized, or simply taken as a shorthand, as follows:

$$\text{Clipped}(t_1, f, t_2) \equiv \exists a, t [\text{Happens}(a, t) \wedge (t_1 \leq t < t_2) \wedge \text{Terminates}(a, f, t)]$$

Domain-dependent axioms

The axioms above relate the value of the predicates *HoldsAt*, *Initiates* and *Terminates*, but do not specify which fluents are known to be true and which actions actually make fluents true or false. This is done by using a set of domain-dependent axioms. The known values of fluents are stated as simple literals $\text{HoldsAt}(f, t)$. The effects of actions are stated by formulae relating the effects of actions with their preconditions. For example, if the action *open* makes the fluent *isopen* true, but only if *haskey* is currently true, the corresponding formula in the event calculus is:

$$\text{Initiates}(a, f, t) \equiv [a = \text{open} \wedge f = \text{isopen} \wedge \text{HoldsAt}(\text{haskey}, t)] \vee \dots$$

The right-hand expression of this equivalence is composed of a disjunction: for each action and fluent that can be made true by the action, there is a disjunct saying that a is actually that action, that f is actually that fluent, and that the precondition of the action is met.

The formula above specifies the truth value of $\text{Initiates}(a, f, t)$ for every possible action and fluent. As a result, all effects of all actions have to be combined in a single formulae. This is a problem, because the addition of a new action requires modifying an existing formula rather than adding new ones. This problem can be solved by the application of circumscription to a set of formulae each specifying one effect of one action:

$$\begin{aligned} \text{Initiates}(\text{open}, \text{isopen}, t) &\leftarrow \text{HoldsAt}(\text{haskey}, t) \\ \text{Initiates}(\text{break}, \text{isopen}, t) &\leftarrow \text{HoldsAt}(\text{hashammer}, t) \\ \text{Initiates}(\text{break}, \text{broken}, t) &\leftarrow \text{HoldsAt}(\text{hashammer}, t) \end{aligned}$$

These formulae are simpler than the formula above, because each effect of each action can be specified separately. The single formula telling which actions a and fluents f make $Initiates(a,f,t)$ true has been replaced by a set of smaller formulae, each one telling the effect of an action to a fluent.

However, these formulae are not equivalent to the formula above. Indeed, they only specify sufficient conditions for $Initiates(a,f,t)$ to be true, which should be completed by the fact that $Initiates$ is false in all other cases. This fact can be formalized by simply circumscribing the predicate $Initiates$ in the formula above. It is important to note that this circumscription is done only on the formulae specifying $Initiates$ and not on the domain-independent axioms. The predicate $Terminates$ can be specified in the same way $Initiates$ is.

A similar approach can be taken for the $Happens$ predicate. The evaluation of this predicate can be enforced by formulae specifying not only when it is true and when it is false:

$$Happens(a,t) \equiv (a = open \wedge t = 0) \vee (a = exit \wedge t = 1) \vee \dots$$

Circumscription can simplify this specification, as only necessary conditions can be specified:

$$\begin{aligned} &Happens(open,0) \\ &Happens(exit,1) \end{aligned}$$

Circumscribing the predicate $Happens$, this predicate will be false in all points in which it is not explicitly specified to be true. This circumscription has to be done separately from the circumscription of the other formulae. In other words, if F is the set of formulae of the kind $Initiates(a, f, t) \leftarrow \dots$, G is the set of formulae $Happens(a,t)$, and H are the domain independent axioms, the correct formulation of the domain is:

$$Circ(F; Initiates, Terminates) \wedge Circ(G; Happens) \wedge H$$

The event calculus as a logic program

The event calculus was originally formulated as a set of Horn clauses augmented with negation as failure and could be run as a Prolog program. In fact, circumscription is one of the several semantics that can be given to negation as failure, and is closely related to the completion semantics.

Extensions and applications

The original event calculus paper of Kowalski and Sergot focused on applications to database updates and narratives. Extensions of the event calculus can also formalize non-

deterministic actions, concurrent actions, actions with delayed effects, gradual changes, actions with duration, continuous change, and non-inertial fluents.

Kave Eshghi showed how the event calculus can be used for planning, using abduction to generate hypothetical events in abductive logic programming. Van Lambalgen and Hamm showed how the event calculus can also be used to give an algorithmic semantics to tense and aspect in natural language using constraint logic programming.

Categorization

Categorization is the process in which ideas and objects are recognized, differentiated and understood. Categorization implies that objects are grouped into categories, usually for some specific purpose. Ideally, a category illuminates a relationship between the subjects and objects of knowledge. Categorization is fundamental in language, prediction, inference, decision making and in all kinds of environmental interaction.

There are many categorization theories and techniques. In a broader historical view, however, three general approaches to categorization may be identified:

- Classical categorization
- Conceptual clustering
- Prototype theory

The classical view

Classical categorization comes to us first from Plato, who, in his Statesman dialogue, introduces the approach of grouping objects based on their similar properties. This approach was further explored and systematized by Aristotle in his Categories treatise, where he analyzes the differences between classes and objects. Aristotle also applied intensively the classical categorization scheme in his approach to the classification of living beings (which uses the technique of applying successive narrowing questions such as "Is it an animal or vegetable?", "How many feet does it have?", "Does it have fur or feathers?", "Can it fly?"...), establishing this way the basis for natural taxonomy.

The classical Aristotelian view claims that categories are discrete entities characterized by a set of properties which are shared by their members. In analytic philosophy, these properties are assumed to establish the conditions which are both necessary and sufficient conditions to capture meaning.

According to the classical view, categories should be clearly defined, mutually exclusive and collectively exhaustive. This way, any entity of the given classification universe belongs unequivocally to one, and only one, of the proposed categories.

Conceptual clustering

Conceptual clustering is a modern variation of the classical approach, and derives from attempts to explain how knowledge is represented. In this approach, classes (clusters or entities) are generated by first formulating their conceptual descriptions and then classifying the entities according to the descriptions.

Conceptual clustering developed mainly during the 1980s, as a machine paradigm for unsupervised learning. It is distinguished from ordinary data clustering by generating a concept description for each generated category.

Categorization tasks in which category labels are provided to the learner for certain objects are referred to as supervised classification, supervised learning, or concept learning. Categorization tasks in which no labels are supplied are referred to as unsupervised classification, unsupervised learning, or data clustering. The task of supervised classification involves extracting information from the labeled examples that allows accurate prediction of class labels of future examples. This may involve the abstraction of a rule or concept relating observed object features to category labels, or it may not involve abstraction (e.g., exemplar models). The task of clustering involves recognizing inherent structure in a data set and grouping objects together by similarity into classes. It is thus a process of *generating* a classification structure.

Conceptual clustering is closely related to fuzzy set theory, in which objects may belong to one or more groups, in varying degrees of fitness.

Prototype Theory

Since the research by Eleanor Rosch and George Lakoff in the 1970s, categorization can also be viewed as the process of grouping things based on prototypes - the idea of necessary and sufficient conditions is almost never met in categories of naturally occurring things. It has also been suggested that categorization based on prototypes is the basis for human development, and that this learning relies on learning about the world via embodiment.

A cognitive approach accepts that natural categories are graded (they tend to be fuzzy at their boundaries) and inconsistent in the status of their constituent members.

Systems of categories are not objectively "out there" in the world but are rooted in people's experience. Conceptual categories are not identical for different cultures, or indeed, for every individual in the same culture.

Categories form part of a hierarchical structure when applied to such subjects as taxonomy in biological classification: higher level: life-form level, middle level: generic or genus level, and lower level: the species level. These can be distinguished by certain traits that put an item in its distinctive category. But even these can be arbitrary and are subject to revision.

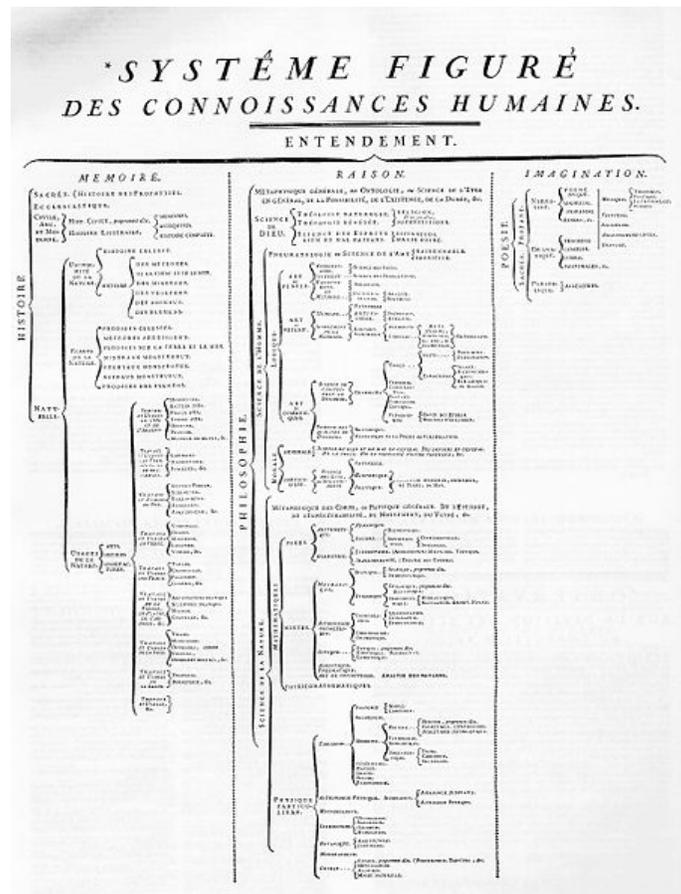
Categories at the middle level are perceptually and conceptually the more salient. The generic level of a category tends to elicit the most responses and richest images and seems to be the psychologically basic level. Typical taxonomies in zoology for example exhibit categorization at the embodied level, with similarities leading to formulation of "higher" categories, and differences leading to differentiation within categories.

Miscategorisation

Miscategorization can be a logical fallacy in which diverse and dissimilar objects, concepts, entities, etc. are grouped together based upon illogical common denominators, or common denominators that virtually any concept, object or entity have in common. A common way miscategorization occurs is through an over-categorization of concepts, objects or entities, and then miscategorization based upon over-similar variables that virtually all things have in common.

Chapter-7

Figurative System of Human Knowledge



The original "figurative system of human knowledge" tree, in French.

The "figurative system of human knowledge", sometimes known as **the tree of Diderot and d'Alembert**, was a tree developed to represent the structure of knowledge itself, produced for the *Encyclopédie* by Jean le Rond d'Alembert and Denis Diderot.

The tree was a taxonomy of human knowledge, inspired by Francis Bacon's *The Advancement of Learning*. The three main branches of knowledge in the tree are: "Memory"/History, "Reason"/Philosophy, and "Imagination"/Poetry.

Notable is the fact that theology is ordered under 'Philosophy'. The historian Robert Darnton has argued that this categorization of religion as being subject to human reason, and not a source of knowledge in and of itself (revelation), was a significant factor in the controversy surrounding the work. Additionally notice that 'Knowledge of God' is only a few nodes away from 'Divination' and 'Black Magic'.

The original version, in French, can be seen in the graphic on the right. An image of the diagram with English translations superimposed over the French text is available. Another example of English translation of the tree is available in literature. Below is a version of it rendered in English as a bulleted outline.

The Tree of Diderot and d'Alembert

"Detailed System of Human Knowledge" from the Encyclopédie.

- Understanding
 - Memory.
 - History.
 - Sacred (History of Prophets).
 - Ecclesiastical.
 - Civil, Ancient and Modern.
 - Civil History, properly said.
 - Literary History.
 - Memoirs.
 - Antiquities.
 - Complete Histories.
- Natural.
 - Uniformity of Nature.
 - Celestial History.
 - History.
 - of Meteors.
 - of the Earth and the Sea.
 - of Minerals.

- of Vegetables.
- of Animals.
- of the Elements.

- Deviations of Nature.

- Celestial Wonders.
- Large Meteors.
- Wonders of Land and Sea.
- Monstrous Minerals.
- Monstrous Vegetables.
- Monstrous Animals.
- Wonders of the Elements.

- Uses of Nature

- Arts, Crafts, Manufactures.

- Work and Uses of Gold and Silver.

- Minting.
- Goldsmith.
- Gold Spinning.
- Gold Drawing.
- Silversmith
- Planisher, etc.

- Work and Uses of Precious Stones.

- Lapidary.
- Diamond Cutting.
- Jeweler, etc.

- Work and Uses of Iron.

- Large Forges.
- Locksmith.
- Tool Making.
- Armorer.
- Gun Making, etc.

- Work and Uses of Glass.

- Glassmaking.
- Plate-Glassmaking.
- Mirror Making.

- Optician.
- Glazier, etc.

- Work and Uses of Skin.

- Tanner.
- Chamios Maker.
- Leather Merchant.
- Glove Making, etc.

- Work and Uses of Stone, Plaster, Slate, etc.

- Practical Architecture.
- Practical Sculpture.
- Mason.
- Tiler, etc.

- Work and Uses of Silk.

- Spinning.
- Milling.
- Work like.
- Velvet.
- Brocaded Fabrics, etc.

- Work and Uses of Wool.

- Cloth-Making.
- Bonnet-Making, etc.

- Working and Uses, etc.

- Reason

- Philosophy

- General Metaphysics, or Ontology, or Science of Being in General, of Possibility, of Existence, of Duration, etc.
- Science of God.

- Natural Theology.
- Revealed Theology.
- Science of Good and Evil Spirits.

- Divination.
- Black Magic.

- Science of Man.
- Pneumatology or Science of the Soul.
- Reasonable.
- Sensible.
- Logic.
- Art of Thinking.
- Apprehension.
- Science of Ideas
- Judgement.
- Science of Propositions.
- Reasoning.
- Induction.
- Method.
- Demonstration.
- Analysis.
- Synthesis.
- Art of Remembering.
- Memory.
- Natural.
- Artificial.
- Prenotion.
- Emblem.
- Supplement to Memory.
- Writing.
- Printing.
- Alphabet.

- Cipher.
- Arts of Writing, Printing, Reading, Deciphering.
- Orthography.
- Art of Communication
- Science of the Instrument of Discourse.
- Grammar.
- Signs.
- Gesture.
- Pantomime.
- Declamation.
- Characters.
- Ideograms.
- Hieroglyphics.
- Heraldry or Blazonry.
- Prosody.
- Construction.
- Syntax.
- Philology.
- Critique.
- Pedagogy.
- Choice of Studies.
- Manner of Teaching.
- Science of Qualities of Discourse.
- Rhetoric.
- Mechanics of Poetry.
- Ethics.
- General.

- General Science of Good and Evil, of duties in general, of Virtue, of the necessity of being Virtuous, etc.
- Particular.
- Science of Laws or Jurisprudence.
- Natural.
- Economic.
- Political.
- Internal and External.
- Commerce on Land and Sea.
- Science of Nature
- Metaphysics of Bodies or, General Physics, of Extent, of Impenetrability, of Movement, of Word, etc.
- Mathematics.
- Pure.
- Arithmetic.
- Numeric .
- Algebra.
- Elementary.
- Infinitesimal.
- Differential.
- Integral.
- Geometry.
- Elementary (Military Architecture, Tactics).
- Transcendental (Theory of Courses).
- Mixed.
- Mechanics.
- Statics.
- Statics, properly said.
- Hydrostatics.

- Dynamics.
- Dynamics, properly said.
- Ballistics.
- Hydrodynamics.
- Hydraulics.
- Navigation, Naval Architecture.
- Geometric Astronomy.
- Cosmography.
- Uranography.
- Geography.
- Hydrography.
- Chronology.
- Gnomonics.
- Optics.
- Optics, properly said.
- Dioptrics, Perspective.
- Catoptrics.
- Acoustics.
- Pneumatics.
- Art of Conjecture. Analysis of Chance.
- Physicomathematics.
- Particular Physics.
- Zoology.
- Anatomy.
- Simple.
- Comparative.
- Physiology.
- Medicine.
- Hygiene.

- Hygiene, properly said.
- Cosmetics (Orthopedics).
- Athletics (Gymnastics).

- Pathology.
- Semiotics.
- Treatment.

- Diete.
- Surgery.
- Pharmacy.

- Veterinary Medicine.
- Horse Management.
- Hunting.
- Fishing.
- Falconry.

- Physical Astronomy.

- Astrology.

- Judiciary Astrology.
- Physical Astrology.

- Meteorology.
- Cosmology.

- Uranology.
- Aerology.
- Geology.
- Hydrology.

- Botany.

- Agriculture.
- Gardening.

- Mineralogy.
- Chemistry.

- Chemistry, properly said, (Pyrotechnics, Dyeing, etc.).
- Metallurgy.
- Alchemy.
- Natural Magic.

- Imagination.
- Poetry.
- Profane.
- Narrative.
- Epic Poem
- Madrigal
- Epigram
- Novel, etc.
- Dramatic
- Tragedy
- Comedy
- Pastoral, etc.
- Parable
- Allegory

(NOTE: THIS NEXT BRANCH SEEMS TO BELONG TO BOTH THE NARRATIVE AND DRAMATIC TREE AS DEPICTED BY THE LINE DRAWN CONNECTING THE TWO.)

- Music
 - Theoretical
 - Practical
 - Instrumental
 - Vocal
- Painting
- Sculpture
- Engraving

Chapter-8

General Architecture for Text Engineering

General Architecture for Text Engineering or **GATE** is a Java suite of tools originally developed at the University of Sheffield beginning in 1995 and now used worldwide by a wide community of scientists, companies, teachers and students for all sorts of natural language processing tasks, including information extraction in many languages.

GATE includes:

- an IDE, GATE Developer: an integrated development environment for natural language processing components bundled with a very widely used information extraction system and a comprehensive set of other plugins
- a web app, GATE Teamware: a collaborative annotation environment for factory-style semantic annotation projects built around a workflow engine and a heavily-optimised backend service infrastructure
- a framework, GATE Embedded: an object library optimised for inclusion in diverse applications giving access to all the services used by GATE Developer and more
- an architecture: a high-level organisational picture of language processing software composition
- a process for the creation of robust and maintainable services.

Under development:

- a cloud computing solution for hosted large-scale text processing, GATE Cloud

GATE aims to remove the necessity for solving common engineering problems before doing useful research, or re-engineering before deploying research results into applications. Core functions of GATE take care of the lion's share of the engineering:

- modelling and persistence of specialised data structures
- measurement, evaluation, benchmarking
- visualisation and editing of annotations, ontologies, parse trees, etc.

- a finite state transduction language for rapid prototyping and efficient implementation of shallow analysis methods
- extraction of training instances for machine learning
- pluggable machine learning implementations (Weka, SVM Light, an in-house uneven margins SVM implementation and more.)

On top of the core functions, GATE includes components for diverse natural language processing tasks, e.g. parsers, morphology, tagging, information retrieval tools, information extraction components for various languages, and many others. It has been widely applied in fields such as bioinformatics and others. GATE Developer and Embedded are supplied with an information extraction system (ANNIE) which has been adapted and evaluated very widely (numerous industrial systems, research systems evaluated in MUC, TREC, ACE, DUC, Pascal, NTCIR, etc.). ANNIE is often used to create RDF or OWL (metadata) for unstructured content (semantic annotation). GATE has been compared to NLTK, R and RapidMiner. As well as being widely used in its own right, it forms the basis of the KIM semantic platform.

GATE community and research has been involved in several European research projects including TAO, SEKT, NeOn, Media-Campaign, Musing, Service-Finder, LIRICS and KnowledgeWeb, as well as many other projects.

As of December 4, 2009, 691 people are on the gate-users mailing list at SourceForge.net, and 98,858 downloads from SourceForge are recorded since the project moved to SourceForge in 2005. The paper "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications" has received over 800 citations in the seven years since publication (according to Google Scholar). Books covering the use of GATE, in addition to the GATE User Guide, include "Building Search Applications: Lucene, LingPipe, and Gate", by Manu Konchady, and "Introduction to Linguistic Annotation and Text Analytics", by Graham Wilcock.

Features

GATE includes an information extraction system called ANNIE (A Nearly-New Information Extraction System) which is a set of modules comprising a tokenizer, a gazetteer, a sentence splitter, a part of speech tagger, a named entities transducer and a coreference tagger. ANNIE can be used as-is to provide basic information extraction functionality, or provide a starting point for more specific tasks.

Languages currently handled in GATE include English, Spanish, Chinese, Arabic, French, German, Hindi, Italian, Cebuano, Romanian, Russian.

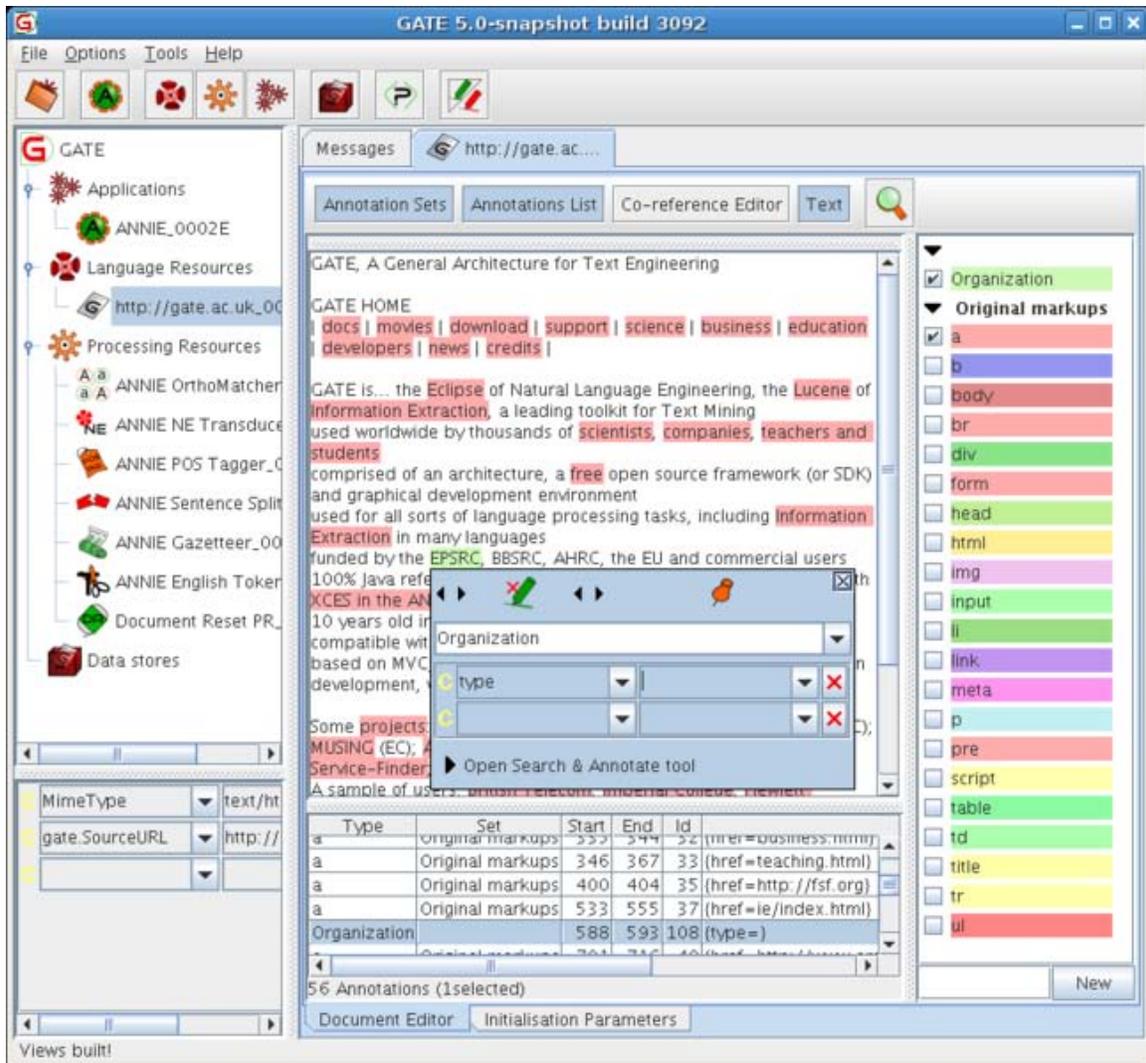
Plugins are included for machine learning with Weka, RASP, MAXENT, SVM Light, as well as a fast LibSVM integration and an in-house perceptron implementation, for managing Ontologies like WordNet, for querying search engines like Google or Yahoo, for part of speech tagging with Brill or TreeTagger, and many more.

GATE can handle input in various formats, such as TXT, HTML, XML, Doc, PDF documents, and Java Serial, PostgreSQL, Lucene, Oracle Databases with help of RDBMS storage over JDBC.

It also uses the JAPE (Java Annotation Patterns Engine) language for building rules in order to annotate documents with tags. JAPE stands for "Java Annotation Patterns Engine". JAPE provides finite state transduction over annotations based on regular expressions. JAPE is a version of CPSL – Common Pattern Specification Language. JAPE transducers are used within GATE to manipulate annotations on text. Documentation is provided in the GATE User Guide. A tutorial has also been written by Press Association Images.

GATE Developer

GATE Developer is the GATE graphical user interface. It is analogous to systems like Mathematica for mathematicians, or Eclipse for Java programmers, providing a convenient graphical environment for research and development of language processing software. As well as being a powerful research tool in its own right, it is also useful in conjunction with GATE Embedded (the GATE API by which GATE functionality can be included in applications); for example, GATE Developer can be used to create applications that can then be embedded via the API.



GATE 5 main window.

The GATE Developer GUI consists of a top menu and row of icons, a left vertical resources tree, a central-right tabbed pane of the resource viewers and a message field at the bottom.

The resources tree and the menu are used to load, save and run resources. The resources tree displays the loaded resources and allows to show a resource in a resource viewer by double-clicking on it or pressing Enter key.

Each loaded resource can be displayed in a specific resource viewer that takes most of the space in the GUI.

Here you can see the document viewer used to display a document and its annotations. In pink are <A> hyperlink annotations from an HTML file. The right list is the annotation

sets list and the bottom table is the annotation list. In the center is the annotation editor window.

GATE Teamware

Teamware is a web-based management platform for collaborative annotation & curation. GATE Teamware delivers a multi-function user interface over the Internet for viewing, adding and editing text annotations. The web-based management interface allows for project set-up, tracking, and management:

- Loading document collections (a "corpus" or "corpora")
- Creating re-usable project templates
- Initiating projects based on templates
- Assigning project roles to specific users
- Monitoring progress and various project statistics in real time
- Reporting of project status, annotator activity and statistics
- Applying GATE-based processing routines (automatic annotations or post-annotation processing)

Chapter-9

Geopolitical Ontology

A geopolitical ontology is a mechanism to describe, manage and exchange data related to geopolitical entities such as countries, territories, regions and other similar areas.

Definitions and examples

An Ontology is a kind of dictionary that describes information in a certain domain using concepts and relationships. It is often implemented using OWL (Web Ontology Language), an XML-based standard language that can be interpreted by computers.

- A *Concept* is defined as abstract knowledge. For example, in the geopolitical ontology a non-self-governing territory or a geographical group are concepts. Concepts are explicitly implemented in the ontology with individuals and classes:
 - An *individual* is defined as an object perceived from the real world. In the geopolitical domain Ethiopia or the least developed countries group are individuals.
 - A *class* is defined as a set of individuals sharing common properties. In the geopolitical domain, Ethiopia, Republic of Korea or Italy are individuals of the class *self-governing* territory; and least developed countries is an individual of the class *special group*.
- Relationships between concepts are explicitly implemented by:
 - *Object properties* between individuals of two classes. For example, *has member* and *is in group* properties, as shown in Figure 1.
 - *Datatype properties* between individuals and literals or XML datatypes. For example, the individual Afghanistan has the datatype property *CodeISO3* with the value "AFG".
 - *Restrictions* in classes and/or properties. For example, the property *official English name* of the class *self-governing* territory has been restricted to have only *one* value, this means that a self-governing territory (or country) can only have one internationally recognized official English name.

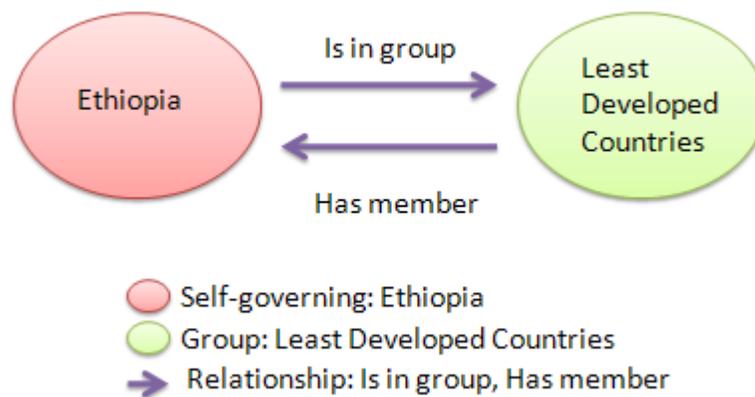


Figure 1. An example of concepts and relationship in the geopolitical ontology.

The advantage of describing information in an ontology is that it enables to acquire domain knowledge by defining hierarchical structures of classes, adding individuals, setting object properties and datatype properties, and assigning restrictions.

FAO ontology

The geopolitical ontology, developed by the Food and Agriculture Organization of the United Nations (FAO), provides names in seven languages (Arabic, Chinese, French, English, Spanish, Russian and Italian) and identifiers in various international coding systems (ISO2, ISO3, AGROVOC, FAOSTAT, FAOTERM, GAUL, UN, UNDP and DBPediaID codes) for territories and groups. Moreover, the FAO geopolitical ontology tracks historical changes from 1985 up until today; provides geolocation (geographical coordinates); implements relationships among countries and countries, or countries and groups, including properties such as *has border with*, *is predecessor of*, *is successor of*, *is administered by*, *has members*, and *is in group*; and disseminates country statistics including country area, land area, agricultural area, GDP or population.

The FAO geopolitical ontology provides a structured description of data sources. This includes: source name, source identifier, source creator and source's update date. Concepts are described using the Dublin Core vocabulary.

In summary, the main objectives of the FAO geopolitical ontology are:

- To provide the most updated geopolitical information (names, codes, relationships, statistics)
- To track historical changes in geopolitical information
- To improve information management and facilitate standardized data sharing of geopolitical information
- To demonstrate the benefits of the geopolitical ontology to improve interoperability of corporate information systems

It is possible to **download** the FAO geopolitical ontology in OWL and RDF formats. Documentation is available in the FAO Country Profiles Geopolitical information web page.

Features of the FAO ontology

The geopolitical ontology contains :

- Area types:
 - Territories: self-governing, non-self-governing, disputed, other.
 - Groups: organizations, geographic, economic and special groups.
- Names (official, short and names for lists) in Arabic, Chinese, English, French, Spanish, Russian and Italian.
- International codes: UN code – M49, ISO 3166 Alpha-2 and Alpha-3, UNDP code, GAUL code, FAOSTAT, AGROVOC FAOTERM and DBPediaID.
- Coordinates: maximum latitude, minimum latitude, maximum longitude, minimum longitude.
- Basic country statistics: country area, land area, agricultural area, GDP, population.
- Currency names and codes.
- Adjectives of nationality.
- Relations:
 - Groups membership.
 - Neighbours (land border), administration of non-self-governing.
 - Historic changes: predecessor, successor, valid since, valid until.

Implementation into OWL

The FAO geopolitical ontology is implemented in OWL. It consists of classes, properties, individuals and restrictions. Table 1 shows all classes, gives a brief description and lists some individuals that belong to each class. Note that the current version of the geopolitical ontology does not provide individuals of the class "disputed" territories. Table 2 and Table 3 illustrate datatype properties and object properties.

Class		Description	Individual	
area	territory	Self-governing	Politically Independent territories	Ethiopia and 208 other countries
		Non-self-governing	Territories dependent to self-governing	American Samoa and 22 non-self-governings
		Disputed	Territories subject to disagreement and debate	West Sahara
		Others	Territories not belonging to self-governing, non-self-governing, and disputed	Antarctica
	Group	Geographical region	Region divided by a demarcated area of the Earth	Middle Africa and other 27 regions
		Economic region	Region divided by economic factors	Caribbean Forum of African, Caribbean and Pacific States (CARIFORUM) and other 21 regions
		Organization	International organizations	FAO
		Special group	Group created for special needs	Least developed countries and other 4 groups

Table 1. Classes and instances in the geopolitical ontology.

Datatype property	Sub datatype properties	Subject classes to be applied (domain)
Has official name	Official English name and other 4 language names	Area and following sub classes
Has list name	List and table English name and other 4 language names	Area and following sub classes
Has short name	Short English name and other 4 language names	Area and following sub classes
Has code	UNDP code	Area and following sub classes
	UN code	Area and following sub classes
	ISO 3166-2 code	Area and following sub classes
	ISO 3166-3 code	Area and following sub classes
	AGROVOC code	Area and following sub classes
	FAOSTAT code	Area and following sub classes
	GAUL code	Area and following sub classes
	FAO TERM code	Area and following sub classes
Has coordinate	minimum latitude	Territory and following sub classes
	Minimum longitude	Territory and following sub classes
	Maximum latitude	Territory and following sub classes
	Maximum longitude	Territory and following sub classes
Valid since		Area and following sub classes
Valid Until		Area and following sub classes

Table 2. Datatype properties in the geopolitical ontology.

Object property	Subject class to be applied (domain)	Object class to be applied (range)	Inverse property
Has border with	Territory and following sub classes	Territory	Has border with
Is predecessor of	Area and following sub classes	Area	Is successor of
Is successor of	Area and following sub classes	Area	Is predecessor of
Is administrated by	Non-self-governing	Self-governing	
Has member	Group	Territory	Is in group
Is in group	Territory	Group	Has members

Table 3. Object properties in the geopolitical ontology.

Example of use

The screenshot shows the FAO Country Profiles website. At the top, there are language options: العربية, 中文, english, français, and español. The main header includes the FAO logo and the text "Food and Agriculture Organization of the United Nations for a world without hunger". Below the header, there is a navigation menu with options like "FAO home", "Country profiles home", "About", "Inventory of resources", "News", "Resources", "FAO member countries and flags", "Low-Income Food-Deficit Countries", "Geopolitical information", and "Select a country".

The main content area is titled "Geopolitical information" and contains the following text:

The FAO geopolitical ontology and related services have been developed to facilitate data exchange and sharing in a standardized manner among systems managing information about countries and/or regions.

The geopolitical ontology ensures that FAO and associated partners can rely on a master reference for geopolitical information, as it manages names in multiple languages (English, French, Spanish, Arabic, Chinese, Russian and Italian); maps standard coding systems (UN, ISO, FAOSTAT, AGROVOC, etc); provides relations among territories (land borders, group membership, etc); and tracks historical changes.

In addition, in order to facilitate the exploitation of this reference data by information systems, various services have been developed on top of this resource:

- The geopolitical ontology module maker: this service allows extracting smaller modules of the ontology and exporting them in RDF, XML or CSV formats, using a web interface.
- The geopolitical ontology web services: these RESTfull web services allow to access ontology data on demand.

Geopolitical ontology overview

The geopolitical ontology, in beta version 0.9, has been populated using FAO, UN and internationally recognized data sources.

On the right side of the page, there are links for "Geopolitical ontology", "Browse HTML version", "Download OWL version", "Previous versions", "Articles", and "FAO sources".

Figure 2. a website of introducing the geopolitical ontology in FAO Country Profiles.

The FAO Country Profiles is an information retrieval tool which groups the FAO's vast archive of information on its global activities in agriculture and rural development in one single area and catalogues it exclusively by country.

The FAO Country Profiles system provides access to country-based heterogeneous data sources. By using the geopolitical ontology in the system, the following benefits are expected:

- Enhanced system functionality for content aggregation and synchronization from the multiple source repositories.
- Improved information access and browsing through comparison of data in neighbor countries and groups.

Chapter-10

Minimum Information Standards and Defeasible Reasoning

Minimum Information Standards

Minimum Information (MI) standards or reporting guidelines specify the minimum amount of meta data (information) and data required to meet a specific aim or aims. Usually the aim is to provide enough meta data and data to enable the unambiguous reproduction and interpretation of an experiment. MI guidelines are normally informal human readable specifications that inform the development of formal data models (e.g. XML or UML), data exchange formats (e.g. FuGE, MAGE-ML, MAGE-TAB) or knowledge models such as an ontology (e.g. OBI, MGED-Ontology).

MI standards are developed by working bodies of practitioners working in a particular scientific domain. The MI standards listed below are all from the life sciences, largely driven by the development of high throughput experimental technologies.

These MI standards groups have been brought together in 2007 to form the "Minimum Information about a Biomedical or Biological Investigation" (MIBBI) umbrella community. More information about the MIBBI initiative and the MIBBI Foundry can be found below and on the MIBBI homepage

MI Standards

MIAME, gene expression microarray

Minimum Information About a Microarray Experiment (MIAME) describes the Minimum Information About a Microarray Experiment that is needed to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment and is aimed at facilitating the dissemination of data from microarray experiments.

MIAME contains a number of extensions to cover specific biological domains

MINI: Minimum Information about a Neuroscience Investigation

MINI: Electrophysiology

Electrophysiology is a technology used to study the electrical properties of biological cells and tissues. Electrophysiology typically involves the measurements of voltage change or electrical current flow on a wide variety of scales from single ion channel proteins to whole tissues. This document is a single module, as part of the Minimum Information about a Neuroscience investigation (MINI) family of reporting guideline documents, produced by community consultation and continually available for public comment. A MINI module represents the minimum information that should be reported about a dataset to facilitate computational access and analysis to allow a reader to interpret and critically evaluate the processes performed and the conclusions reached, and to support their experimental corroboration. In practice a MINI module comprises a checklist of information that should be provided (for example about the protocols employed) when a data set is described for publication. The full specification of the MINI module can be found here .

MIARE, RNAi experiment

Minimum Information About an RNAi Experiment (MIARE) is a data reporting guideline which describes the minimum information that should be reported about an RNAi experiment to enable the unambiguous interpretation and reproduction of the results.

MIACA, cell based assay

Advances in genomics and functional genomics have enabled large-scale analyses of gene and protein function by means of high-throughput cell biological analyses. Thereby, cells in culture can be perturbed in vitro and the induced effects recorded and analyzed. Perturbations can be triggered in several ways, for instance with molecules (siRNAs, expression constructs, small chemical compounds, ligands for receptors, etc.), through environmental stresses (such as temperature shift, serum starvation, oxygen deprivation, etc.), or combinations thereof. The cellular responses to such perturbations are analyzed in order to identify molecular events in the biological processes addressed and understand biological principles. We propose the Minimum Information About a Cellular Assay (MIACA) for reporting a cellular assay, and CA-OM, the modular cellular assay object model, to facilitate exchange of data and accompanying information, and to compare and integrate data that originate from different, albeit complementary approaches, and to elucidate higher order principles. Documents describing MIACA are available and provide further information as well as the checklist of terms that should be reported.

MIAPE, proteomic experiments

The Minimum Information About a Proteomic Experiment documents describe information which should be given along with a proteomic experiment.

MIGS/MIMS, genome/metagenome sequences

This specification is being developed by the Genomic Standards Consortium

MIFlowCyt, flow cytometry

Minimum Information about a Flow Cytometry Experiment

The fundamental tenet of any scientific research is that the published results of any study have to be open to independent validation or refutation. The Minimum Information about a Flow Cytometry Experiment (MIFlowCyt) establishes the criteria to record information about the experimental overview, samples, instrumentation and data analysis. It promotes consistent annotation of clinical, biological and technical issues surrounding a flow cytometry experiment by specifying the requirements for data content and by providing a structured framework for capturing information.

More information can be found at:

- The Flow Informatics and Computational Cytometry Society (FICCS) MIFlowCyt page.
- The Bioinformatics Standards for Flow Cytometry MIFlowCyt web page.

MIRIAM, Minimum Information Required in the Annotation of Models

The Minimal Information Required In the Annotation of Models (MIRIAM), is a set of rules for the curation and annotation of quantitative models of biological systems.

MIASE, Minimum Information About a Simulation Experiment

The Minimum Information About a Simulation Experiment (MIASE) is an effort to standardize the description of simulation experiments in the field of systems biology.

Defeasible reasoning

Defeasible reasoning is a kind of reasoning that is based on reasons that are *defeasible*, as opposed to the *indefeasible* reasons of deductive logic. Defeasible reasoning is a particular kind of non-demonstrative reasoning, where the reasoning does not produce a full, complete, or final demonstration of a claim, i.e., where fallibility and corrigibility of

a conclusion are acknowledged. Other kinds of non-demonstrative reasoning are probabilistic reasoning, inductive reasoning, statistical reasoning, abductive reasoning, and paraconsistent reasoning. Defeasible reasoning is also a kind of ampliative reasoning because its conclusions reach beyond the pure meanings of the premises.

The differences between these kinds of reasoning correspond to differences about the conditional that each kind of reasoning uses, and on what premise (or on what authority) the conditional is adopted:

- *Deductive* (from meaning postulate, axiom, or contingent assertion): if p then q (i.e., q or *not-p*)
- *Defeasible* (from authority): if p then (defeasibly) q
- *Probabilistic* (from combinatorics and indifference): if p then (probably) q
- *Statistical* (from data and presumption): the frequency of qs among ps is high (or inference from a model fit to data); hence, (in the right context) if p then (probably) q
- *Inductive* (theory formation; from data, coherence, simplicity, and confirmation): (inducibly) "if p then q "; hence, if p then (deducibly-but-revisably) q
- *Abductive* (from data and theory): p and q are correlated, and q is sufficient for p ; hence, if p then (abducibly) q as cause

Some have thought that defeasible reasoning could be connected to qualitative probabilistic reasoning, but such efforts have not borne great insights.

Defeasible reasoning finds its fullest expression in jurisprudence, ethics and moral philosophy, epistemology, pragmatics and conversational conventions in linguistics, constructivist decision theories, and in knowledge representation and planning in artificial intelligence. It is also closely identified with *prima facie* (presumptive) reasoning (i.e., reasoning on the "face" of evidence), and *ceteris paribus* (default) reasoning (i.e., reasoning, all things "being equal").

History

Though Aristotle differentiated the forms of reasoning that are valid for logic and philosophy from the more general ones that are used in everyday life, 20th Century philosophers mainly concentrated on deductive reasoning. At the end of the 19th Century, logic texts would typically survey both demonstrative and non-demonstrative reasoning, often giving more space to the latter. However, after the blossoming of mathematical logic at the hands of Bertrand Russell, Alfred North Whitehead and Willard van Orman Quine, latter-20th Century logic texts paid little attention to the non-deductive modes of inference.

There are several notable exceptions. John Maynard Keynes wrote his dissertation on non-demonstrative reasoning, and influenced the thinking of Ludwig Wittgenstein on this subject. Wittgenstein, in turn, had many admirers, including the positivist legal scholar H.L.A. Hart and the speech act linguist John L. Austin, Stephen Toulmin in rhetoric

(Chaim Perelman too), the moral theorists W.D. Ross and C.L. Stevenson, and the vagueness epistemologist/ontologist Friedrich Waismann.

The etymology of *defeasible* usually refers to Middle English law of contracts, where a condition of defeasance is a clause that can invalidate or annul a contract or deed. Though *defeat*, *dominate*, *defer*, *defy*, *deprecate* and *derogate* are often used in the same contexts as *defeasible*, the verbs *annul* and *invalidate* (and *nullify*, *overturn*, *rescind*, *vacate*, *repeal*, *debar*, *void*, *cancel*, *countermand*, *preempt*, etc.) are more properly correlated with the concept of defeasibility than those words beginning with the letter *d*. Many dictionaries do contain the verb, *to defease* with past participle, *defeased*.

Philosophers in moral theory and rhetoric had taken defeasibility largely for granted when American epistemologists rediscovered Wittgenstein's thinking on the subject: John Ladd, Roderick Chisholm, Roderick Firth, Ernest Sosa, Robert Nozick, and John L. Pollock all began writing with new conviction about how *appearance as red* was only a defeasible reason for believing something to be red. More importantly Wittgenstein's orientation toward language games (and away from semantics) emboldened these epistemologists to manage rather than to expurgate *prima facie* logical inconsistency.

At the same time (in the mid-1960s), two more students of Hart and Austin at Oxford, Brian Barry and David Gauthier, were applying defeasible reasoning to political argument and practical reasoning (of action), respectively. Joel Feinberg and Joseph Raz were beginning to produce equally mature works in ethics and jurisprudence informed by defeasibility.

By far the most significant works on defeasibility by the mid-1970s were in epistemology, where John Pollock's 1974 *Knowledge and Justification* popularized his terminology of *undercutting* and *rebutting* (which mirrored the analysis of Toulmin). Pollock's work was significant precisely because it brought defeasibility so close to philosophical logicians. The failure of logicians to dismiss defeasibility in epistemology (as Cambridge's logicians had done to Hart decades earlier) landed defeasible reasoning in the philosophical mainstream.

Defeasibility had always been closely related to argument, rhetoric, and law, except in epistemology, where the chains of reasons, and the origin of reasons, were not often discussed. Nicholas Rescher's *Dialectics* is an example of how difficult it was for philosophers to contemplate more complex systems of defeasible reasoning. This was in part because proponents of informal logic became the keepers of argument and rhetoric while insisting that formalism was anathema to argument.

About this time, researchers in artificial intelligence became interested in non-monotonic reasoning and its semantics. With philosophers such as Pollock and Donald Nute (e.g., defeasible logic), dozens of computer scientists and logicians produced complex systems of defeasible reasoning between 1980 and 2000. No single system of defeasible reasoning would emerge in the same way that Quine's system of logic became a de facto standard. Nevertheless, the 100-year head start on non-demonstrative logical calculi, due to George

Boole, Charles Sanders Peirce, and Gottlob Frege was being closed: both demonstrative and non-demonstrative reasoning now have formal calculi.

There are related (and slightly competing) systems of reasoning that are newer than systems of defeasible reasoning, e.g., belief revision and dynamic logic. The dialogue logics of Charles Hamblin and Jim Mackenzie (logician), and their colleagues, can also be tied closely to defeasible reasoning. Belief revision is a non-constructive specification of the desiderata with which, or constraints according to which, epistemic change takes place. Dynamic logic is related mainly because, like paraconsistent logic, the reordering of premises can change the set of justified conclusions. Dialogue logics introduce an adversary, but are like belief revision theories in their adherence to deductively consistent states of belief.

Political and judicial use

Many political philosophers have been fond of the word *indefeasible* when referring to rights, e.g., that were *inalienable*, *divine*, or *indubitable*. For example, in the 1776 Virginia Declaration of Rights, "community hath an indubitable, inalienable, and indefeasible right to reform, alter or abolish government..." (also attributed to James Madison); and John Adams, "The people have a right, an indisputable, unalienable, indefeasible, divine right to that most dreaded and envied kind of knowledge - I mean of the character and conduct of their rulers." Also, Lord Aberdeen: "indefeasible right inherent in the British Crown" and Gouverneur Morris: "the Basis of our own Constitution is the indefeasible Right of the People." Scholarship about Abraham Lincoln often cites these passages in the justification of secession. Philosophers who use the word *defeasible* have historically had different world views from those who use the word *indefeasible* (and this distinction has often been mirrored by Oxford and Cambridge zeitgeist); hence it is rare to find authors who use both words.

In judicial opinions, the use of *defeasible* is commonplace. There is however disagreement among legal logicians whether *defeasible reasoning* is central, e.g., in the consideration of *open texture*, precedent, exceptions, and *rationales*, or whether it applies only to explicit defeasance clauses. H.L.A. Hart in *The Concept of Law* gives two famous examples of defeasibility: "No vehicles in the park" (except during parades); and "Offer, acceptance, and memorandum produce a contract" (except when the contract is illegal, the parties are minors, inebriated, or incapacitated, etc.).

Specificity

One of the main disputes among those who produce systems of defeasible reasoning is the status of a *rule of specificity*. In its simplest form, it is the same rule as subclass inheritance preempting class inheritance:

(R1) if <i>p</i> then (defeasibly) <i>q</i> not-flies	e.g., if penguin then
(R2) if <i>r</i> then (defeasibly) <i>not-q</i> flies	e.g., if bird then

(O1) if p then (deductively) r bird	e.g., if penguin then
(M1) arguably, p	e.g., arguably, penguin
(M2) R1 is a more specific reason than R2 R2	e.g., R1 is better than
(M3) therefore, arguably, q arguably, not-flies	e.g., therefore,

Approximately half of the systems of defeasible reasoning discussed today adopt a rule of specificity, while half expect that such *preference* rules be written explicitly by whomever provides the defeasible reasons. For example, Rescher's dialectical system uses specificity, as do early systems of multiple inheritance (e.g., David Touretzky) and the early argument systems of Donald Nute and of Guillermo Simari and Ronald Loui. Defeasible reasoning accounts of precedent (*stare decisis* and case-based reasoning) also make use of specificity (e.g., Joseph Raz and the work of Kevin D. Ashley and Edwina Rissland). Meanwhile, the argument systems of Henry Prakken and Giovanni Sartor, of Bart Verheij and Jaap Hage, and the system of Phan Minh Dung do not adopt such a rule.

Nature of defeasibility

There is a distinct difference between those who theorize about defeasible reasoning as if it were a system of confirmational revision (with affinities to belief revision), and those who theorize about defeasibility as if it were the result of further (non-empirical) investigation. There are at least three kinds of further non-empirical investigation: progress in a lexical/syntactic process, progress in a computational process, and progress in an adversary or legal proceeding.

Defeasibility as corrigibility: Here, a person learns something new that annuls a prior inference. In this case, defeasible reasoning provides a constructive mechanism for belief revision, like a truth maintenance system as envisioned by Jon Doyle.

Defeasibility as shorthand for preconditions: Here, the author of a set of rules or legislative code is writing rules with exceptions. Sometimes a set of defeasible rules can be rewritten, with more cogency, with explicit (local) pre-conditions instead of (non-local) competing rules. Many non-monotonic systems with fixed-point or preferential semantics fit this view. However, sometimes the rules govern a process of argument (the last view on this list), so that they cannot be re-compiled into a set of deductive rules lest they lose their force in situations with incomplete knowledge or incomplete derivation of preconditions.

Defeasibility as an anytime algorithm: Here, it is assumed that calculating arguments takes time, and at any given time, based on a subset of the potentially constructible arguments, a conclusion is defeasibly justified. Isaac Levi has protested against this kind of defeasibility, but it is well-suited to the heuristic projects of, for example, Herbert Simon. On this view, the *best move so far* in a chess-playing program's analysis at a particular depth is a defeasibly justified conclusion. This interpretation works with either the prior or the next semantical view.

Defeasibility as a means of controlling an investigative or social process: Here, justification is the result of the right kind of procedure (e.g., a fair and efficient hearing), and defeasible reasoning provides impetus for pro and con responses to each other. Defeasibility has to do with the alternation of verdict as locutions are made and cases presented, not the changing of a mind with respect to new (empirical) discovery. Under this view, defeasible reasoning and defeasible argumentation refer to the same phenomenon.

Chapter-11

Attempto Controlled English

Attempto Controlled English (ACE) is a controlled natural language, i.e. a subset of standard English with a restricted syntax and a restricted semantics described by a small set of construction and interpretation rules .

ACE can serve as knowledge representation, specification, and query language, and is intended for professionals who want to use formal notations and formal methods, but may not be familiar with them. Though ACE appears perfectly natural – it can be read and understood by anybody – it is in fact a formal language.

ACE and its related tools have been used in the fields of software specifications, theorem proving, text summaries, ontologies, rules, querying, medical documentation and planning. In 2004, ACE was adopted as the controlled natural language of the EU Network of Excellence REWERSE (Reasoning on the Web with Rules and Semantics).

Here are some simple examples:

- (1) * Women are human.
- (2) Every woman is a human.
- (3) A man is a human.
- (4) A man tries-on a new tie. If the tie pleases his wife then the man buys it.

ACE construction rules require that each noun be introduced by a determiner (*a, every, no, some, at least 5, ...*). This excludes (1) as indicated by the * preceding the sentence. ACE interpretation rules decide that (2) is interpreted as universally quantified, while (3) is interpreted as existentially quantified. Other interpretation rules resolve the anaphoric references in (4): *the tie* and *it* of the second sentence refer to *a new tie* of the first sentence, while *his* and *the man* of the second sentence refer to *a man* of the first sentence. Thus an ACE text is a coherent entity of anaphorically linked sentences.

The Attempto Parsing Engine (APE) translates ACE texts unambiguously into discourse representation structures DRSs that use a variant of the language of first-order logic. DRSs can be further translated into other formal languages, for instance AceRules with

various semantics, RuleML, OWL, and SWRL. Translating DRSs into the standard language of first-order logic allows users to reason about ACE texts, for instance to verify, to validate, and to query them.

ACE in a nutshell

This section is a survey of ACE 6.0 in a nutshell that

- briefly describes the vocabulary
- gives an account of the syntax
- summarises the handling of ambiguity
- explains the processing of anaphoric references

Vocabulary

The vocabulary of ACE comprises

- predefined function words (e.g. determiners, conjunctions, prepositions),
- content words (nouns, proper names, verbs, adjectives, adverbs).

The Attempto system provides a basic lexicon of content words. Users can import domain-specific lexicons of content words. User-defined words take precedence over words found in the basic lexicon. Alternatively, users can let the ACE parser guess unknown words, or users can prefix unknown words by their word-class, for instance *n:kitkat*, *p:Thomas*, *v:google*, *a:trusted*, *a:undeviatingly*.

Grammar

The grammar of ACE defines and constrains the form and the meaning of ACE sentences and texts. ACE's grammar is expressed as a set of construction rules. The meaning of sentences is described as a small set of interpretation rules. A Trouble Shooting Guide describes how to use ACE and how to avoid pitfalls.

ACE texts

An ACE text is a sequence of declarative sentences that can be anaphorically interrelated. Furthermore, ACE supports questions and commands.

Simple sentences

A simple sentence describes that something is the case — a fact, an event, a state.

```
The temperature is -2 °C.  
A customer inserts 2 cards.  
A card and a code are valid.
```

Simple ACE sentences have the following general structure:

subject + verb + complements + adjuncts

Every sentence has a subject and a verb. Complements (direct and indirect objects) are necessary for transitive verbs (*insert something*) and ditransitive verbs (*give something to somebody*), whereas adjuncts (adverbs, prepositional phrases) are optional.

All elements of a simple sentence can be elaborated upon to describe the situation in more detail. To further specify the nouns *customer* and *card*, we could add adjectives:

A trusted customer inserts two valid cards.

possessive nouns and *of*-prepositional phrases

John's customer inserts a card of Mary.

or variables as appositions

John inserts a card A.

Other modifications of nouns are possible through relative sentences

A customer who is trusted inserts a card that he owns.

which are described below since they make a sentence composite. We can also detail the insertion event, e.g. by adding an adverb

A customer inserts some cards manually.

or, equivalently

A customer manually inserts some cards.

or, by adding prepositional phrases

A customer inserts some cards into a slot.

We can combine all of these elaborations to arrive at

John's customer who is trusted inserts a valid card of Mary manually into a slot A.

Composite sentences

Composite sentences are recursively built from simpler sentences through coordination, subordination, quantification, and negation. Note that ACE composite sentences overlap with what linguists call compound sentences and complex sentences.

Coordination

Coordination by *and* is possible between sentences and between phrases of the same syntactic type.

A customer inserts a card and the machine checks the code.
There is a customer who inserts a card and who enters a code.
A customer inserts a card and enters a code.
An old and trusted customer enters a card and a code.

Note that the coordination of the noun phrases *a card* and *a code* represents a plural object.

Coordination by *or* is possible between sentences, verb phrases, and relative clauses.

A customer inserts a card or the machine checks the code.
A customer inserts a card or enters a code.
A customer owns a card that is invalid or that is damaged.

Coordination by *and* and *or* is governed by the standard binding order of logic, i.e. *and* binds stronger than *or*. Commas can be used to override the standard binding order. Thus the sentence

A customer inserts a VisaCard or inserts a MasterCard, and inserts a code.

means that the customer inserts a VisaCard and a code or, alternatively a MasterCard and a code.

Subordination

There are four constructs of subordination: relative sentences, *if-then* sentences, modality, and sentence subordination.

Relative sentences starting with *who*, *which*, and *that* allow to add detail to nouns, e.g.

A customer who is trusted inserts a card that he owns.

With the help of *if-then* sentences we can specify conditional or hypothetical situations, e.g.

If a card is valid then a customer inserts it.

Note the anaphoric reference via the pronoun *it* in the *then*-part to the noun phrase *a card* in the *if*-part.

Modality allows us to express possibility and necessity.

A trusted customer can/must insert a card.
It is possible/necessary that a trusted customer inserts a card.

Sentence subordination comes in various forms.

It is true/false that a customer inserts a card.
It is not provable that a customer inserts a card.
A clerk believes that a customer inserts a card.

Quantification

Quantification allows us to speak about all objects of a certain class (universal quantification), or to denote explicitly the existence of at least one object of this class (existential quantification). The textual occurrence of a universal or existential quantifier opens its scope that extends to the end of the sentence, or in coordinations to the end of the respective coordinated sentence.

To express that all involved customers insert cards we can write

Every customer inserts a card.

This sentence means that each customer inserts a card that may, or may not, be the same as the one inserted by another customer. To specify that all customers insert the same card — however unrealistic that situation seems — we can write

A card is inserted by every customer.

or, equivalently

There is a card that every customer inserts.

To state that every card is inserted by a customer we write

Every card is inserted by a customer.

or, somewhat indirectly

For every card there is a customer who inserts it.

Negation

Negation allows us to express that something is not the case, e.g.

A customer does not insert a card.
A card is not valid.

To negate something for all objects of a certain class one uses *no*

No customer inserts more than 2 cards.

or, *there is no*

There is no customer who inserts a card.

To negate a complete statement one uses sentence negation

It is false that a customer inserts a card.

These forms of negation are logical negations, i.e. they state that something is provably not the case. Negation as failure states that a state of affairs cannot be proved, i.e. there is no information whether the state of affairs is the case or not.

It is not provable that a customer inserts a card.

Queries

ACE supports two forms of queries: *yes/no*-queries and *wh*-queries.

Yes/no-queries ask for the existence or non-existence of a specified situation. If we specified

A customer inserts a card.

then we can ask

Does a customer insert a card?

to get a positive answer. Note that interrogative sentences need always a question mark at the end.

With the help of *wh*-queries, i.e. queries with query words, we can interrogate a text for details of the specified situation. If we specified

A trusted customer inserts a valid card manually in the morning in a bank.

we can ask for each element of the sentence with the exception of the verb.

Who inserts a card?

Which customer inserts a card?

What does a customer insert?

How does a customer insert a card?

When does a customer enter a card?

Where does a customer enter a card?

Queries can also be constructed by a sequence of declarative sentences followed by one interrogative sentence, for example:

There is a customer and there is a card that the customer enters. Does a customer enter a card?

Commands

ACE also supports commands. Some examples:

```
John, go to the bank!  
John and Mary, wait!  
Every dog, bark!  
A brother of John, give a book to Mary!
```

A command always consists of a noun phrase (the addressee), followed by a comma, followed by an uncoordinated verb phrase. Furthermore, a command has to end with an exclamation mark.

Constraining ambiguity

To constrain the ambiguity of full natural language ACE employs three simple means

- some ambiguous constructs are not part of the language; unambiguous alternatives are available in their place,
- all remaining ambiguous constructs are interpreted deterministically on the basis of a small number of interpretation rules,
- users can either accept the assigned interpretation, or they must rephrase the input to obtain another one.

Avoidance of ambiguity

Here is an example how ACE replaces ambiguous constructs by unambiguous constructs. In full natural language relative sentences combined with coordinations can introduce ambiguity, e.g.

```
A customer inserts a card that is valid and opens an account.
```

In ACE the sentence has the unequivocal meaning that the customer opens an account. This is reflected by the paraphrase

```
A card is valid. A customer inserts the card. The customer opens an account.
```

To express the alternative — though not very realistic — meaning that the card opens an account the relative pronoun *that* must be repeated, thus yielding a coordination of relative sentences.

```
A customer inserts a card that is valid and that opens an account.
```

with the paraphrase

A card is valid. The card opens an account. A customer inserts the card.

Interpretation rules

However, not all ambiguities can be safely removed from ACE without rendering it artificial. To deterministically interpret otherwise syntactically correct ACE sentences we use a small set of interpretation rules. For example, if we write

A customer inserts a card with a code.

then *with a code* attaches to the verb *inserts*, but not to *a card*. However, this is probably not what we meant to say. To express that *the code* is associated with *the card* we can employ the interpretation rule that a relative sentence always modifies the immediately preceding noun phrase, and rephrase the input as

A customer inserts a card that carries a code.

yielding the paraphrase

A card carries a code. A customer inserts the card.

or — to specify that the customer inserts a card and a code — as

A customer inserts a card and a code.

Anaphoric references

Usually ACE texts consist of more than one sentence, e.g.

A customer enters a card and a code. If a code is valid then SimpleMat accepts a card.

To express that all occurrences of card and code should mean the same card and the same code, ACE provides anaphoric references via the definite article, i.e.

A customer enters a card and a code. If the code is valid then SimpleMat accepts the card.

During the processing of the ACE text, all anaphoric references are replaced by the most recent and most specific accessible noun phrase that agrees in gender and number. What does "most recent and most specific" mean?

Given the sentence

A customer enters a red card and a blue card.

then

The card is correct.

refers to the second card, while

The red card is correct.

refers to the first card.

What does "accessible" mean? Noun phrases introduced in *if-then* sentences, universally quantified sentences, negations, modality, and subordinated sentences cannot be used anaphorically in subsequent sentences. Thus for each of the sentences

If a customer owns a card then he enters it.

Every customer enters a card.

A customer does not enter a card.

A customer can enter a card.

A clerk believes that a customer enters a card.

we cannot refer to *a card* with

The card is correct.

Anaphoric references are also possible via personal pronouns

A customer enters a card and a code. If it is valid then SimpleMat accepts the card.

or via variables

A customer enters a card X and a code Y. If Y is valid then SimpleMat accepts X.

Anaphoric references via definite articles and variables can be combined.

A customer enters a card X and a code Y. If the code Y is valid then SimpleMat accepts the card X.

Note that proper names like *SimpleMat* always refer to the same object.

History

ACE has been under development at the University of Zurich since 1995. In 2004, ACE was adopted as the controlled natural language of the EU Network of Excellence REVERSE (Reasoning on the Web with Rules and Semantics). In 2008, ACE version 6.0 was announced.

Chapter-12

Upper Ontology (information science)

In information science, an **upper ontology** (**top-level ontology**, or **foundation ontology**) is an ontology which describes very general concepts that are the same across all knowledge domains. The most important function of an upper ontology is to support very broad semantic interoperability between a large number of ontologies accessible "under" this upper ontology. As the metaphor suggests, it is usually a hierarchy of entities and associated rules (both theorems and regulations) that attempts to describe those general entities that do not belong to a specific problem domain.

The seemingly conflicting use of metaphors implying a solid rigorous bottom-up "foundation" or a top-down imposition of somewhat arbitrary and possibly political decisions is no accident - the field is characterized by controversy, politics, competing approaches and academic rivalry.

Debates notwithstanding, it can be said that a very important part of each upper ontology can be considered as the computational implementation of natural philosophy, which itself is a more empirical method for investigating the topics within the philosophical discipline of physical ontology.

Library classification systems predate these upper ontology systems. Though library classifications organize and categorize knowledge using general concepts that are the same across all knowledge domains, neither system is a replacement for the other.

Development

Upper ontologies are also commercially valuable, creating competition to define them. Peter Murray-Rust has claimed that this leads to "semantic and **ontological warfare** due to competing standards" , and accordingly any standard foundation ontology is likely to be contested among commercial or political parties, each with their own idea of 'what exists'.. An important factor exacerbating the failure to arrive at a common approach is the absence of open-source applications that would permit the testing of different ontologies in the same computational environment. The differences are debated largely

on theoretical grounds, or are merely the result of personal preferences, with no method to objectively compare practical performance.

No one upper ontology has yet gained widespread acceptance as a de facto standard. Different organizations are attempting to define standards for specific domains. The 'Process Specification Language' (PSL) created by the National Institute for Standards and Technology (NIST) is one example.

An important factor leading to the absence of wide adoption of any existing upper ontology is the complexity. An upper ontology typically has from 2000 to 10,000 elements (classes, relations), with complex interactions among them. The resulting complexity is similar to that of a human language, and the learning process can be even longer than for a human language because of the unfamiliar format and logical rules. The motivation to overcome this learning barrier is largely absent because of the paucity of publicly accessible examples of use. As a result, those building domain ontologies for a local application tend to create the simplest possible domain-specific ontology, not related to any upper ontology. Such domain ontologies may function adequately for the local purpose, but are very time-consuming to relate accurately to other domain ontologies.

There is debate over whether the concept of using a single, shared upper ontology is even feasible or practical at all. There is further debate over whether the debates are valid - often leading to outright censorship and boosterism of particular approaches in supposedly neutral sources *including this one*. Some of these arguments are outlined below, with no attempt to be comprehensive. Please do not censor them because you promote some ontology.

Arguments for the infeasibility of an upper ontology

Historically, many attempts in many societies have been made to impose or define a single set of concepts as more primal, basic, foundational, authoritative, true or rational than others.

In the kind of modern societies that have computers at all, the existence of academic and political freedoms imply that many ontologies will simultaneously exist and compete for adherents. While the differences between them may be narrow and appear petty to those not deeply involved in the process, so too did many of the theological debates of medieval Europe, but they still led to schisms or wars, or were used as excuses for same. The tyranny of small differences that standard ontologies seek to end may continue simply because other forms of tyranny are even less desirable. So private efforts to create competitive ontologies that achieve adherents by virtue of better communication may proceed, but tend not to result in long standing monopolies.

A deeper objection derives from ontological constraints that philosophers have found historically inescapable. Some argue that a transcendent perspective or omniscience is implied by even searching for any general purpose ontology - since it is a social / cultural

artifact, there is no purely objective perspective from which to observe the whole terrain of concepts and derive any one standard.

A narrower and much more widely held objection is implicature: the more general the concept and the more useful in semantic interoperability, the less likely it is to be reducible to symbolic concepts or logic and the more likely it is to be simply accepted by the complex beings and cultures relying on it. In the same sense that a fish doesn't perceive water, we don't see how complex and involved is the process of understanding basic concepts.

- There is no self-evident way of dividing the world up into concepts, and certainly no non-controversial one
- There is no neutral ground that can serve as a means of translating between specialized (or "lower" or "application-specific") ontologies
- Human language itself is already an arbitrary approximation of just one among many possible conceptual maps. To draw any *necessary correlation* between English words and any number of intellectual concepts we might like to represent in our ontologies is just asking for trouble. (WordNet, for instance, is successful and useful precisely because it does not pretend to be a general-purpose upper ontology; rather, it is a tool for semantic / syntactic / linguistic disambiguation, which is richly embedded in the particulars and peculiarities of the English language.)
- Any hierarchical or topological representation of concepts must begin from some ontological, epistemological, linguistic, cultural, and ultimately pragmatic perspective. Such pragmatism does not allow for the exclusion of politics between persons or groups, indeed it requires they be considered as perhaps more basic primitives than any that are represented.

Those who doubt the feasibility of general purpose ontologies are more inclined to ask “what specific purpose do we have in mind for this conceptual map of entities and what practical difference will this ontology make?” This pragmatic philosophical position surrenders all hope of devising the encoded ontology version of “everything that is the case,” (Wittgenstein, *Tractatus Logico-Philosophicus*).

According to Barry Smith in *The Blackwell Guide to the Philosophy of Computing and Information* (2004), "the initial project of building one single ontology, even one single top-level ontology, which would be at the same time non-trivial and also readily adopted by a broad population of different information systems communities, has largely been abandoned." (p. 159)

Finally there are objections similar to those against artificial intelligence. Technically, the complex concept acquisition and the social / linguistic interactions of human beings suggests any axiomatic foundation of "most basic" concepts must be cognitive, biological or otherwise difficult to characterize since we don't have axioms for such systems. Ethically, any general-purpose ontology could quickly become an actual tyranny by recruiting adherents into a political program designed to propagate it and its funding

means, and possibly defend it by violence. Historically, inconsistent and irrational belief systems have proven capable of commanding obedience to the detriment of harm of persons both inside and outside a society that accepts them. How much more harmful would a consistent rational one be, were it to contain even one or two basic assumptions incompatible with human life?

Arguments for the feasibility of an upper ontology

Many of those who doubt the possibility of developing wide agreement on a common upper ontology fall into one of two traps: (1) they assert that there is no possibility of universal agreement on any conceptual scheme; **but** they ignore the fact that a practical common ontology does not need to have universal agreement, it only needs a large enough user community to make it profitable for developers to use it as a means to general interoperability, and for third-party developer to develop utilities to make it easier to use; and (2) they point out that developers of data schemes find different representations congenial for their local purposes; **but** they do not demonstrate that these different representation are in fact logically inconsistent. In fact, different representations of assertions about the real world (though not philosophical models), if they accurately reflect the world, must be logically consistent, even if they focus on different aspects of the same physical object or phenomenon. If any two assertions about the real world are logically inconsistent, one or both must be wrong, and that is a topic for experimental investigation, not for ontological representation. In practice, representations of the real world are created as and known to be approximations to the basic reality, and their use is circumscribed by the limits of error of measurements in any given practical application. Ontologies are entirely capable of representing approximations, and are also capable of representing situations in which different approximations have different utility. Objections based on the different ways people perceive things attack a simplistic, impoverished view of ontology. The objection that there are logically incompatible models of the world are true, but in an upper ontology those different models can be represented as different theories, and the adherents of those theories can use them in preference to other theories, while preserving the logical consistency of the *necessary* assumptions of the upper ontology. The *necessary* assumptions provide the logical vocabulary with which to specify the meanings of all of the incompatible models. It has never been demonstrated that incompatible models cannot be properly specified with a common, more basic set of concepts, while there are examples of incompatible theories that can be logically specified with only a few basic concepts.

Many of the objections to upper ontology refer to the problems of life-critical decisions or non-axiomatized problem areas such as law or medicine or politics that are difficult even for humans to understand. Some of these objections do not apply to physical objects or standard abstractions that are defined into existence by human beings and closely controlled by them for mutual good, such as standards for electrical power system connections or the signals used in traffic lights. No single general metaphysics is required to agree that some such standards are desirable. For instance, while time and space can be represented many ways, some of these are already used in interoperable artifacts like maps or schedules.

Objections to the feasibility of a common upper ontology also do not take into account the possibility of forging agreement on an ontology that contains all of the **primitive** ontology elements that can be combined to create any number of more specialized concept representations. Adopting this tactic permits effort to be focused on agreement only on a limited number of ontology elements (under 10,000). By agreeing on the meanings of that inventory of basic concepts, it becomes possible to create and then accurately and automatically interpret an infinite number of concept representations as combinations of the basic ontology elements. Any domain ontology or database that uses the elements of such an upper ontology to specify the meanings of its terms will be automatically and accurately interoperable with other ontologies that use the upper ontology, even though they may each separately define a large number of domain elements not defined in other ontologies. In such a case, proper interpretation will require that the logical descriptions of domain-specific elements be transmitted along with any data that is communicated; the data will then be automatically interpretable because the domain element descriptions, based on the upper ontology, will be properly interpretable by any system that can properly use the upper ontology. An upper ontology based on such a set of primitive elements can include alternative views, provided that they are logically compatible. Logically incompatible models can be represented as alternative theories, or represented in a specialized extension to the upper ontology. The proper use of alternative theories is a piece of knowledge that can itself be represented in an ontology.

Most proponents of an upper ontology argue that several good ones may be created with perhaps different emphasis. Very few are actually arguing to discover just one within natural language or even an academic field. Most are simply standardizing some existing communication. Another view advanced is that there is almost total overlap of the different ways that upper ontologies have been formalized, in the sense that different ontologies focus on a different aspect of the same entities, but the different views are complementary and not contradictory to each other; as a result, an internally consistent ontology that contains all the views, with means of translating the different views into the other, is feasible. Such an ontology has not thus far been constructed, however, because it would require a large project to develop so as to include all of the alternative views in the separately developed upper ontologies, along with their translations. The main barrier to construction of such an ontology is not the technical issues, but the reluctance of funding agencies to provide the funds for a large enough consortium of developers and users.

Several common arguments against upper ontology can be examined more clearly by separating issues of concept definition (ontology), language (lexicons), and facts (knowledge). For instance, people have different terms and phrases for the same concept. However, that does not necessarily mean that those people are referring to different concepts. They may simply be using different language or idiom. Formal ontologies typically use linguistic labels to refer to concepts, but the terms that label ontology elements mean no more and no less than what their axioms say they mean. Labels are similar to variable names in software, evocative rather than definitive. The proponents of a common upper ontology point out that the meanings of the elements (classes, relations, rules) in an ontology depend only on their logical form, and not on the labels, which are

usually chosen merely to make the ontologies more easily usable by their human developers. In fact, the labels for elements in an ontology need not be words - they could be, for example, images of instances of a particular type, or videos of an action that is represented by a particular type. It cannot be emphasized too strongly that words are **not** what are represented in an ontology, but entities in the real world, or abstract entities (concepts) in the minds of people. Words are not equivalent to ontology elements, but words **label** ontology elements. There can be many words that label a single concept, even in a single language (synonymy), and there can be many concepts labeled by a single word (ambiguity). Creating the mappings between human language and the elements of an ontology is the province of Natural Language Understanding. But the ontology itself stands independently as a logical and computational structure. For this reason, finding agreement on the structure of an ontology is actually easier than developing a controlled vocabulary, because all different interpretations of a word can be included, each **mapped** to the same word in the different terminologies.

A second argument is that people believe different things, and therefore can't have the same ontology. However, people can assign different truth values to a particular assertion while accepting the validity of certain underlying claims, facts, or way of expressing an argument with which they disagree. (Using, for instance, the issue/position/argument form.) This objection to upper ontologies ignores the fact that a single ontology can represent different belief systems, representing them as different belief systems, without taking a position on the validity of either.

Even arguments about the existence of a thing require a certain sharing of a concept, even though its existence in the real world may be disputed. Separating belief from naming and definition also helps to clarify this issue, and show how concepts can be held in common, even in the face of differing belief. It is also argued that most people share a common set of "semantic primitives", fundamental concepts, to which they refer when they are trying to explain unfamiliar terms to other people. An ontology that includes representations of those semantic primitives could in such a case be used to create logical descriptions of any term that a person may wish to define logically. That ontology would be one form of upper ontology, serving as a logical "interlingua" that can translate ideas in one terminology to its logical equivalent in another terminology.

Advocates argue that most disagreement about the viability of an upper ontology can be traced to the conflation of ontology, language and knowledge, or too-specialized areas of knowledge: many people, or agents or groups will have areas of their respective internal ontologies that do not overlap. If they can cooperate and share a conceptual map at all, this may be so very useful that it outweighs any disadvantages that accrue from sharing. To the degree it becomes harder to share concepts the deeper one probes, the more valuable such sharing tends to get. If the problem is as basic as opponents of upper ontologies claim, then, it applies also to a group of humans trying to cooperate, who might need machine assistance to communicate easily.

If nothing else, such ontologies are implied by machine translation, used when people cannot practically communicate. Whether "upper" or not, these seem likely to proliferate.

Available ontologies

Cyc

A well-known and quite comprehensive ontology available today is Cyc, a proprietary system under development since 1986, consisting of a foundation ontology and several domain-specific ontologies (called *microtheories*). A subset of that ontology has been released for free under the name OpenCyc, and a more or less unabridged version is made available for non-commercial use under the name ResearchCyc.

UMBEL

UMBEL is an ontology of 28,000 reference concepts that maps to a simplified subset of the OpenCyc ontology, that is intended to provide a way of linking the precise OpenCyc ontology with less formal ontologies.

Basic Formal Ontology (BFO)

The BFO or Basic Formal Ontology framework developed by Barry Smith and his associates consists in a series of sub-ontologies at different levels of granularity. The ontologies are divided into two varieties: **SNAP** (or snapshot) ontologies, comprehending continuant entities such as three-dimensional enduring objects, and **SPAN** ontologies, comprehending processes conceived as extended through (or as spanning) time. BFO thus incorporates both three-dimensionalist and four-dimensionalist perspectives on reality within a single framework. Interrelations are defined between the two types of ontologies in a way which gives BFO the facility to deal with both static/spatial and dynamic/temporal features of reality. Each SNAP ontology is an inventory of all entities existing at a time. Each SPAN ontology is an inventory (processory) of all the processes unfolding through a given interval of time. Both types of ontology serve as basis for a series of sub-ontologies, each of which can be conceived as a window on a certain portion of reality at a given level of granularity. An example of an application of BFO can be seen in the Ontology for Biomedical Investigations (OBI). A list of the large number of ontologies based on BFO can be found [here](#).

DOLCE and DnS

Developed by Nicola Guarino and his associates at the Laboratory for Applied Ontology (LOA), the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) is the first module of the WonderWeb foundational ontologies library. As implied by its acronym, DOLCE has a clear *cognitive bias*, in that it aims at capturing the ontological categories underlying natural language and human commonsense. DOLCE, however, does not commit to a strictly referentialist metaphysics related to the intrinsic nature of the world. Rather, the categories it introduces are thought of as cognitive artifacts, which are ultimately depending on human perception, cultural imprints and social conventions. In this sense, they intend to be just *descriptive* (vs *prescriptive*) notions, that assist in making already formed conceptualizations explicit. DOLCE is an ontology of particulars,

in the sense that its domain of discourse is restricted to them. Of course, universals are used to organize and characterize the particulars, but they are not themselves subject to being organized and characterized (e.g., by means of metaproperties).

DnS (Descriptions and Situations), developed by Aldo Gangemi (STLab, Rome), is a *constructivist* ontology that pushes DOLCE's descriptive stance even further. DnS does not put restrictions on the type of entities and relations that one may want to postulate, either as a domain specification, or as an upper ontology, and it allows for context-sensitive '*redescriptions*' of the types and relations postulated by other given ontologies (or 'ground' vocabularies). The current OWL encoding of DnS assumes DOLCE as a ground top-level vocabulary. DnS and related modules also exploit 'CPs' (Content ontology design Patterns), which provide a framework to annotate 'focused' fragments of a reference ontology (i.e., the parts of an ontology containing the types and relations that underlie 'expert reasoning' in given fields or communities). The combination of DOLCE and DnS has been used to build a planning ontology known as DDPO (DOLCE+DnS Plan Ontology).

Both DOLCE and DnS are particularly devoted to the treatment of social entities, such as e.g. organizations, collectives, plans, norms, and information objects. The DOLCE-2.1-Lite-Plus OWL version, including a number of DnS-based modules, has been and is being applied to several ontology projects.

A lighter OWL axiomatization of DOLCE and DnS, which also simplifies the names of many classes and properties, adds extensive inline comments, and thoroughly aligns to the repository of Content patterns is now available as DOLCE+DnS-Ultralite (abbreviated: DUL). Despite its simplification, which greatly speeds up consistency checking and classification of OWL domain ontologies that are plugged to it, the expressivity of DUL is not significantly different from the previous DOLCE-Lite-Plus. DOLCE OWL versions, DOLCE+DnS-Ultralite and the pattern repository are developed and maintained by Aldo Gangemi and his associates at Rome's Semantic Technology Lab.

General Formal Ontology (GFO)

The General Formal Ontology (GFO), developed by Heinrich Herre and his colleagues of the research group Onto-Med in Leipzig, is a realistic ontology integrating processes and objects. It attempts to include many aspects of recent philosophy, which is reflected both in its taxonomic tree and its axiomatizations. GFO allows for different axiomatizations of its categories (such as the existence of atomic time-intervals vs. dense time). The basic principles of GFO are published in the Onto-Med Report Nr. 8 and in General Formal Ontology (GFO): A Foundational Ontology for Conceptual Modelling.

Two GFO specialties, among others, are its account of persistence and its time model. Regarding persistence, the distinction between endurants (objects) and perdurants (processes) is made explicit within GFO by the introduction of a special category, a persistant. A persistant is a special category with the intention that its instances "remain

identical" (over time). With respect to time, time intervals are taken as primitive in GFO, and time-points (called "time boundaries") as derived. Moreover, time-points may coincide, which is convenient for modelling instantaneous changes.

IDEAS

The upper ontology developed by the IDEAS Group is higher-order, extensional and 4D. It was developed using the BORO Method. The IDEAS ontology is not intended for reasoning and inference purposes; its purpose is to be a precise model of business.

WordNet

WordNet, a freely available database originally designed as a semantic network based on psycholinguistic principles, was expanded by addition of definitions and is now also viewed as a dictionary. It qualifies as an upper ontology by including the most general concepts as well as more specialized concepts, related to each other not only by the subsumption relations, but by other semantic relations as well, such as part-of and cause. However, unlike Cyc, it has not been formally axiomatized so as to make the logical relations between the concepts precise. It has been widely used in Natural language processing research.

Suggested Upper Merged Ontology

The Suggested Upper Merged Ontology ("SUMO") is another comprehensive ontology project. It includes an upper ontology, created by the IEEE working group P1600.1 (predominantly by Ian Niles and Adam Pease). It is extended with many domain ontologies and a complete set of links to WordNet. It is freely available.

Biomedical ontology

Examples of *domain ontologies* can be found at the Open Biomedical Ontology site. They act as an umbrella organisation for many ontologies specific to biological topics (such as cellular organelles).

- Open Biomedical Ontologies
- Search, browse and visualise the OBO ontologies online via the NCBO Bioportal
- Ontology browser for most of the Open Biomedical Ontologies

COSMO

COSMO (COmmon Semantic MOdel) is an ontology that was initiated as a project of the COSMO working group of the Ontology and taxonomy Coordinating Working Group. The current version is an OWL ontology, but a Common-Logic compliant version is anticipated in the future. The ontology and explanatory files are available at the COSMO site. The goal of the COSMO working group was to develop a foundation ontology by a collaborative process that will allow it to represent all of the basic ontology elements that

all members feel are needed for their applications. The development of COSMO is fully open, and any comments or suggestions from any sources are welcome. After some discussion and input from members in 2006, the development of the COSMO has been continued primarily by Patrick Cassidy, the chairman of the COSMO Working Group. Contributions and suggestions from any interested party are still welcome and encouraged. Many of the types (OWL classes) in the current COSMO have been taken from the OpenCyc OWL version 0.78, and from the SUMO. Other elements were taken from other ontologies (such as BFO and DOLCE), or developed specifically for COSMO. Recent development of the COSMO has focused on including representations of all of the words in the Longman Dictionary of Contemporary English (LDOCE) controlled defining vocabulary (2148 words). These words are sufficient to define (linguistically) all of the entries in the LDOCE. It is hypothesized that the ontological representations of the concepts represented by those terms will be sufficient to specify the meanings of any specialized ontology element, thereby serving as a basis for general Semantic Interoperability. The current (May 2009) OWL version of COSMO has over 6400 types (OWL classes), over 700 relations, and over 1400 restrictions.

OCHRE

The Object-Centered High-level REference ontology (OCHRE) was developed by Luc Schneider at the Institute for Formal Ontology and Medical Information Science at the University of Leipzig. This ontology was developed not only to create a particular basic ontological framework, but also to demonstrate how the quality of a foundational ontology depends on descriptive adequacy and on formal simplicity and transparency. The ontology identifies objects, attributes, and events as describing reality. The ontology distinguishes thin objects (a core of enduring characteristics) and thick objects (having spatio-temporal bulk that undergo change; being stages or phases or snapshots of thin objects). Attributes (properties and relations) can be regarded either as repeatables (universals that apply to more than one case) or as non-repeatables (property-instances or tropes that are single characteristics of individuals). Events (changes or state-transitions) are accounted for as a succession of object stages or phases. The ontology has a focus on conceptual simplicity, so that the number of basic (primitive) concepts and relations is as small as possible in order to simplify the theory.

PROTON

PROTON (PROTo ONtology) is a basic subsumption hierarchy which provides coverage of most of the upper-level concepts necessary for semantic annotation, indexing, and retrieval.