The Dimensions of Context-Space

Doug Lenat

October 28, 1998

Contexts have historically been either ignored completely or else treated as black boxes, as indivisible atoms. About a decade ago, as part of our work on building the large Cyc® knowledge base of human common sense and common knowledge, our group began to study and harness the internal structure of that “atom”. Each context was said to have assumptions and content; there was a theory of importing assertions across contexts; contexts were fully reified first-class terms in the CycL representation language; they were partially ordered by specialization to control visibility and access to content; and so on. That 1989-91 work turned out to be inadequate: it was too expensive to do nontrivial lifting (importing); to explicate the assumptions of each context; and to place each assertion/query into the proper context.

Over the last few years, as the number of Cyc contexts grew into the thousands, we gained a better understanding of the problem – and a possible solution has emerged. There is a finer internal structure to a context than just those two parts, assumptions and content. There are a dozen mostly-independent dimensions along which contexts vary; conversely, each region of that 12-dimensional space implicitly defines a context. In effect that space is the space of assumptions, and each assertion can be thought to hold true in some region of that space. A more advanced calculus of contexts is required to handle those 12-dimensional constructs, but it should be worth the cost: it should enable a much more efficient, much more focused sort of “virtual lifting” of assertions from one context to another, and – by providing a superstructure that can serve as a principled guide to orient the working KB builder or per user – it should make it easier to specify the proper context in which an assertion (or question) should be stated.

In this paper, we discuss contexts in general, and delineate the dimensions of context-space. We introduce specific terminology for describing points and regions and comparative locations along each dimension. We then consider what ontological engineering will be like in that world.

©1998 CYCORP, 3721 EXECUTIVE CENTER DRIVE, SUITE 100, AUSTIN, TX 78731

EMAIL: info@cyc.com PHONE: (512) 342-4000

WEB: www.cyc.com FAX: (512)-342-4040

Contents

1 Summary

Throughout the history of AI (and of software, of science, or for that matter of practically any human endeavor) *contexts* have usually been either ignored completely or else treated as black boxes [McCarthy&Hayes 69], as indivisible atoms. We talk about "the context" of a statement, event, etc. without actually ever delving into what comprises a context. Intuitively, we all understand the danger of taking things "out of context", since assertions true in one context might well be false in a different one. And we understand the potential usefulness and power of contexts, of being in – and reasoning within – a context:

- Enabling us to ignore 99.999% of our knowledge so we can focus on "the task at hand"
- Enabling us to be terse and sloppy in our communications and yet expect our readers/listeners to understand our intent
- Enabling us to accommodate apparently contradictory information, by partitioning it out to different contexts

This power is not a luxury. During the 1984-1989 time period, as the Cyc® common sense knowledge base [Lenat&Guha 90] grew ever larger, it became increasingly difficult to shoehorn every fact and rule into the same flat "world." Finally, in 1989, as Cyc exceeded 100,000 "rules" in size, we found it necessary to introduce an explicit context mechanism. That is, we divided the KB up into a lattice[1] of hundreds of contexts, placing each Cyc assertion in whichever context(s) it belonged.

These early Cyc contexts were full-fledged first-class reified (named) objects. There was a simple theory of "lifting" assertions across contexts, to import an assertion from one context into another. Each context was said to consist of 2 parts:

◊ assumptions: a set of conjuncts to the antecedants of *all* the rules in that context
◊ content: the set of rules/assertions said to *hold*, or be *true* in, that context]

That late-1980's body of work culminated in a PhD thesis for R. V. Guha, the person who primarily implemented the Cyc context mechanism [Guha 91], and sparked a rebirth of interest, lively discussion, and published articles about contexts [McCarthy 93], [Buvac 96].

Unfortunately, that first foray into contexts turned out to be inadequate in several ways:

[1] More precisely a DAG or "directed acyclic graph" – this means a node&link graph in which links have a direction (e.g., in a company org chart, having a link from Jane to Ann means Ann's boss is Jane, not vice versa), each node can have multiple parent-nodes and multiple child-nodes, and if you follow links from A to B to C… you will never wind up back at A.

- The cost of lifting was computationally very high, so only trivial special cases of it were ever permitted.
- Partly because of that restriction, even when reified contexts were identified there was little short-term payoff in painstakingly explicating their assumptions. Not surprisingly, then, very little such assumption-articulating work was ever done.
- Many of the problems familiar to programming language “packages” veterans recurred; e.g., asserting P⇨Q in one context C1 and asserting P in a different context C2, and failing to have Q concluded in some asking context C3 because one or both of the assertions were not importable into the asking context C3.
- Related to that previous point was the difficulty of deciding in what context an assertion belonged, or deciding from what point of view (context) an ASK was being made.

Over the last few years, we have gained a better understanding of the problem – and a possible solution has emerged. To wit:

There is a finer internal structure to a context than just those two aforementioned parts, assumptions and content. There are a dozen mostly-independent dimensions along which contexts vary (Absolute Time, Type of Time, Absolute Place, Type of Place, Culture, Sophistication/Security, Granularity, Epistemology, Argument-Preference, Topic, Justification, and Anthropacity.) Conversely, each region of that 12-dimensional space implicitly defines a context.

In effect that space is the space of *assumptions*, and each assertion can be thought to hold true in some region of that space.

There is also some finer structure associated with the *content* of a context, such as whether the assertion holds somewhere in (versus everywhere in) that context.

There is also some finer structure to each of the 12 dimensions; many of them are really a bundle of somewhat-mutually-dependent finer-grained dimensions; e.g., the Epistemology dimension encompasses several related dimensions such as Modality (belief versus expectation versus desirability versus intention versus knowledge) and Disposition (if x is saying something untrue to y, is this untruth being communicated as a lie, an error, a joke, a simplification, etc.?)

A much larger calculus of contexts is required to handle 12- versus 1-dimensional constructs, but it should be worth the cost: it should enable a much more efficient, much more focused sort of lifting of assertions from one context to another; and it should make it easier to place a rule or fact (or question) in the proper context.

In the next section, we discuss contexts in general, and then, in the 12 subsections of Section 3, we treat each of the 12 dimensions of context-space. We introduce specific terminology for describing points and regions and comparative locations along each dimension. Finally, in Section 4, we consider what ontological engineering will be like in that world.

2 Thinking about Contexts

2.1 We are swaddled by an aether of context

To communicate a thought to another thinking being – be it flesh or silicon – we must encode it in some representation. We might encode it in an English sentence, a database, a spreadsheet, a C++ program, etc.

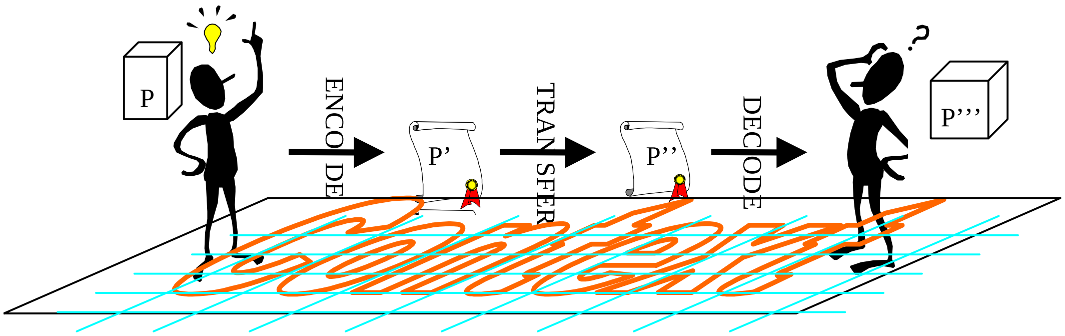


Figure 1. Communication involves encoding and decoding, each of which are done in some context.

If you pluck an isolated line of code from a computer program, or an isolated sentence from a book, or an isolated cell from a spreadsheet, it will likely lose some or all of its meaning. I.e., if you show it – "out of context" – to someone else, they will likely miss some or all of its intended significance.

Thus, much of the meaning of a represented piece of information derives from the context in which the information is encoded and decoded. This can be a tremendous advantage. To the extent that the two thinking beings are sharing a common rich context, they may utilize terse signals to communicate complex thoughts. For example:

- One member of an old married couple inhales a certain way, or blinks a certain way, and their spouse knows they want to go home now.

- Three lights are lit in a church steeple, and Paul Revere knows that the British are coming both by land and by sea.
- Each technical field develops its own jargon, notation, and acronyms, making it difficult for a non-specialist to decipher texts in that field.

In a more common, but scarcely less powerful, case, an author employs a pronoun or an ambiguous noun, and the reader disambiguates it effortlessly. E.g., if someone writes (or utters) a sentence like “Fred told the waiter he wanted some chips” the reader (or listener) would be expected to infer many things. To cite just a few of them:

◊ Fred wants potato chips, not wood chips, cow chips, bone chips, etc.

◊ There’s no particular set of chips that he wants.

◊ Fred and the waiter were a few feet apart at the time.

◊ This telling event took place in a restaurant.

◊ Fred was a customer dining there at that time.

◊ The waiter was at work there, waiting on Fred at that time.

◊ Fred will start eating the chips very shortly after he gets them.

◊ Fred wants and expects the chips in the next few minutes.

◊ Fred wants and expects the waiter to bring him the chips.

◊ Fred wants and expects a single portion (1-5 oz, 5-25 chips)

◊ Fred and the waiter speak the same language.

◊ Fred accomplished this by speaking words to the waiter.

◊ Fred might have said “I want chips”, or “Chips”, or even “Okay”

◊ Fred assumes the waiter knows/infers all the above things as well.

◊ Fred and the waiter were a few feet apart at the time.

◊ Fred and the waiter are both human beings.

◊ Fred is old enough to talk (2+ years of age).

◊ The waiter is old enough to work (4+ years, probably 15+)

◊ This took place after the date of invention of potato chips (1853)

◊ “he” means Fred. I.e., it’s Fred who wants the chips, not the waiter.

Etc.

We human beings all get by, today, in the real world, speaking and writing such terse, syntactically ambiguous utterances – such as Fred saying "I want chips" or, in response to a question from the waiter, possibly just saying "Okay" – because we all draw on the same seven elements of shared *context*:

1. The content of the previous sentences that have just gone by, in the dialogue.
2. The form of the previous sentences (word choices, sentence structure, tone, etc.)
3. the underlying substrate of general real-world[2] knowledge that we assume practically everyone knows. In modern America, this encompasses recent history and current affairs, everyday physics, "household" chemistry, famous books and movies and songs and ads, famous people, nutrition, addition, weather, etc., etc.
4. The underlying substrate of common sense "rules of thumb" largely derived from shared experiences (dating, driving, dining, daydreaming, etc.) and human cognitive economies/limitations (misremembering, misunderstanding, etc.), and shared modes of reasoning both high (induction, intuition, inspiration, incubation) and low (modus ponens, dialectic argument, superficial analogy, pigeon-holing, etc.)
5. The current short-term real-world[2] situation/problem/task/environs[3] that the speaker [or author] and listener [or reader] are in, or are talking about, and their respective roles in that situation/task/etc., and what each presumes the short-term goals of the other to be in that conversation.
6. The long-term background/credentials/occupation/role of each party – at least those that the other party is aware of or, more importantly, believes to be true.
7. The history of any memorable experiences they shared together (and the roles they played in those events), any memorable[4] prior conversations they had with each other.

[2] Of course, in the case of a work of fiction, or an old chronicle, etc., the "real world" means the world in which that utterance was set. E.g., the sentences spoken by the narrator in *Dracula* are set in a fictional world akin to 19th century Europe, but with real vampires in that world. Even in that flight of fantasy, 99.9% of all the objects, events, places, relationships, etc. have the same "true real world" structure and rules about them.

[3] "Environs" includes lighting conditions, crowdedness, noisiness; each other's appearance, dress, stance, etc.

[4] The quality "memorable" often derives from some combination of unexpected, significant, and recent.

This sharing of context enables extraordinary degrees of ambiguity and metonymy to be injected by the author/speaker and tolerated by the reader/listener. For example, Figures 2a-2e, on the next few pages, contain five sentences from this afternoon's (1/28/98) *USA Today*. In each of those five figures, we point out the uses of context. The context enables the sentences to be relatively terse and syntactically ambiguous. The sentence *s* then becomes part of the context – a new skin on the onion – enabling *subsequent* sentences to be even more terse. So in each case (Figures 2a-2e), the context has two different effects:

- It forestalls the need to say additional things in (or near) that sentence *s* – things which the author expects the reader to already know or to infer for themselves based on this sentence (and preceding ones).
- It enables the next few sentences to immediately address the questions that this sentence *s* raises. I.e., sentence *s* passes into the context, after it's read. It becomes part of the context, thereby enabling the next sentences to be even more terse and even more syntactically ambiguous.

This paper focuses primarily on shared aspects 3 & 4: the common body of real-world knowledge and common sense that each person assumes that "everyone" else already has, about the world. For example :

◊ you should carry a glass of water open-end-up;
◊ the USA is a big country;
◊ every animal has two parents, a female mother and a male father;
◊ if something is true for all people/cars/dates/…, it's true for any particular one;
etc.

That many of these are simplifications or misconceptions is irrelevant; what matters is the universality of the agreement, not the truth or accuracy of the information. After all, there can be, and are, other contexts that hold that more precise but less widely known model of that bit of the world.

The other five aspects of shared context (1, 2, 5, 6, 7) are more properly part of a theory of natural language conversation and dialogue, and will be dealt with – by us and by Cyc – separately.

Figure 2: Context in a few typical sentences from the 1/28/98 *USA Today*

Fig. 2a.: “*Monica Lewinsky's former lover says she is untruthful.*”

To whom does the pronoun “she” refer: to Monica or to the former lover?

Does the person claim that Monica always lies, or just sometimes? What sorts of things is she unlikely to lie about (e.g., her name, her gender, her home planet, the date, etc.)?

Might a former lover have a grudge or bias against Monica?

Might such a person have some privileged inside knowledge about her?

Is the former lover actually claiming to know that Monica is currently lying about (the nonexistence of) that particular affair with Clinton, or just that in the past Monica has lied?

Of all the things that Monica said, which specific thing is this statement actually suggesting that the reader alter their confidence in? *[Answer: Monica’s January 27, 1998 claim to the press – later retracted – that she did not have sex with Bill Clinton in the White House.]*

Once this sentence is stated, in the article, it becomes part of the context of the article. The next sentence might answer questions, tersely, that this one raises. E.g., “However, their separation was not amicable.”

How would the meaning change if “says” had been “admits”? “claims”?

Fig. 2b: “Claris will concentrate on its FileMaker Pro database software, changing its name to FileMaker.”

Consider that first “its”. Is Claris renaming an entire product line, or just one particular copy of a program? Obviously the whole product line.

Consider the second “its”. Is FileMaker the new name for FileMaker Pro or for Claris?

What is likely to follow this sentence? Probably some sort of history and rationale.

Fig. 2c: “College hoops are looking for new faces to carry the game, as stars leave early for the NBA.”

Metaphors are packed into practically every word and phrase in that sentence.

From the wording and/or from prior knowledge, the reader is expected to infer:

- Was the phenomenon being described always so prevalent?
- Why do the stars leave college early?
- Why do the NBA teams lure them away?
- Why do college teams mind this?
- What will the stars be doing at/for the NBA?

This sentence also sets a strong context; the reader now is ready for the next sentences to address: Which colleges? Which NBA teams? Which notable stars? How much money?

Notice that the reader is expected to know what the NBA is, and a fair amount about basketball in late 20th century America. Think how baffling that sentence would be to someone 100, or even 50 years ago, even if the “NBA” acronym were spelled out.

Fig. 2d: “Stay on top of the latest hurricanes and tropical storms in the Atlantic and Pacific basins using the following links.”

The writer assumes the reader will infer the answer to questions like:

- Is this literally, physically “on top”?
- Does this only promise things about the past storms, or about future ones too?
- Does a storm have to span both Atlantic and Pacific basins to be covered?
- How does a reader use the links? Do they have to use them all?
- To stay on top, do they have to use the links once or more than once?

By contrast, there are some less obvious questions that might be explicitly talked about, in the next few sentences in the article, such as:

- How often is each linked information source updated?
- How often is the set of links themselves reviewed and updated?
- What the source (and reliability) of the links?

The next sentence can be even more terse, as a result, if it answers one of those questions. E.g., “They change each hour, on the hour”, meaning the linked information gets updated then.

Fig. 2e: “U.S. air fatalities fell in 1997.”

Since “fatality” can refer to the deceased person, this sentence could just be stating the obvious (i.e., that’s why those people died: they fell), but that is unlikely for two reasons:

- Good versus bad taste in making jokes about death
- All readers of the newspaper are expected to already know that that’s generally what happens when an airplane crashes – that’s how and why those people died

So presumably, the sentence is talking about the rate of air fatalities per year, which number apparently was lower for the year 1997 than for 1996. This raises several questions for which the answer is not obvious, however, such as:

- Is that statistic for U.S. air carriers, or for American passengers on all carriers?
- How much lower is it?
- Why was it lower?
- What was the trend in pre-1996 years?

Because of this, the reader is half-expecting to have some of those questions answered in the next few sentences of the article. Because of that expectation, the author can be even more terse. E.g., the next sentence might be something like: “They fluctuate wildly”, meaning that this statistic fluctuates a lot from year to year.

Summary of section 2.1: In 1989 we added a context mechanism to Cyc. Each context had its own content (set of assertions) and set of assumptions (some explicit and some as-of-yet unstated.) This provided some relief for knowledge-entering and for efficient reasoning – but not enough. After we examine what worked out as part of that early context mechanism (in section 2.2 and 2.3), we’ll turn to why and how we’re changing it.

2.2 Contexts catalyze knowledge-entering

When we write an assertion P – for example, **P999: “If it’s raining outside, carry an umbrella”** – there are certain circumstances in which it applies (is a good idea, is valid, etc.) and otherwise it doesn’t apply (is a bad idea, is invalid, etc.) You can think of those “applicability circumstances” as a set of tests one could conjoin onto the antecedent of P.[5,6]

- Some of these tests will already be explicitly part of P, namely whatever was already considered the antecedent of P. E.g., in the case of P999 (“If it’s raining outside, carry an umbrella”), there is one test of this sort: the condition that it be raining outside.
- Some tests will be implicit but are so essential/common/fundamental/… that they are used unconsciously. These tests may be revealed by introspection, by having the system get the wrong parses to sentences, or by getting the wrong answers to questions posed to it. A few of those tests to add to assertion P999 are collected together in Figure 2f, on the next page. E.g., the test that the listener/reader is not going to have both their arms full. Before you turn the page, take a moment to introspect on what you think some of the other tests are; there are probably more of them than you might expect. At any given moment, one has the illusion of having found them all. This is akin to programmers believing they’ve found the last bug in their code.
- These conditions are not likely to come to mind right away, and are certainly not worth stating over and over again, so it’s good to have a more global way to write, store, and add to them (the *assumptions* of a context), rather than having them repeated over and over again for each member of a large set of assertions. Skipping these common assumptions (until and unless a problem is caused by skipping them) greatly simplifies and speeds up the process of KE-ing (knowledge entering – the process of codifying and formalizing a set of axioms, rules, etc.)
- In some cases, you won’t know what exactly the assumptions are, even after you spend time and effort trying to ferret them out, but you may still know that *some* set of assertions all make more or less the same assumptions. You could then clump those assertions together into a context, and provide at least an informal comment explaining the sense of what the context is meant to capture. [E.g., “dining in a restaurant”, or “War and Peace”, or “what Egypt believes the USA wants Egypt and Israel to agree to”.] Most computer programs that have ever been written fall into this category; the

[5] If P structurally had no antecedent, then the conjunction of those tests t would *form* its antecedent: $t \Rightarrow P$

[6] These extra tests are all true in the context where P naturally arises and gets used, so they generally aren’t even thought about explicitly; you might need to think about them, and conjoin them onto the antecedent of P, if you were using P outside its usual context.

modules of the computer program share a set of never-explicated assumptions in common with each other and, hopefully, in common with the user of the application program.

Figure 2f. Some implicit assumptions made by the innocuous-looking rule P999: "If it's raining outside, carry an umbrella."

- the performer is a human being,
- the performer is sane,
- the performer can carry an umbrella; thus:
- the performer is not a baby,
- the performer is not quadriplegic,
- the performer is not dead,
- the performer is going to go outdoors now/soon,
- their actions permit them a free hand (e.g., not wheelbarrowing)
- their actions wouldn't be unduly hampered by it (e.g., marathon-running)
- the wind outside is not too fierce (e.g., hurricane strength)
- the time period of the action is after the invention of the umbrella
- the culture is one that uses umbrellas as a rain- (not just sun-)protection device,
- the performer has access to an umbrella; thus:
- not too destitute,
- not someone who lives where it practically never rains,
- not somewhere such as a movie theater where it's too late to take an umbrella if they hadn't already done so on their way to the place, earlier
- the performer is going to be unsheltered for some period of time, while outside,
- the more waterproof their clothing, the longer that time period
- the gentler the rain, the longer that time period
- the warmer the air, the longer that time period
- the umbrella will be used practically the whole time unsheltered
- in the air (i.e., not outdoors scuba-diving, or swimming, etc.)
- the rain is merely annoying; thus:
- not ammonia rain on Venus, radioactive post-apocalyptic rain
- not biblical (Noah's-ark-sized, or frogs/blood as rained on Pharaoh)
- not a hydrophobic person, gingerbread man, etc.,
- not a hydrophilic person, Gene Kelley in love, etc.

For the past nine years, we've been carving the Cyc KB up into a set of "contexts" or "microtheories." Has it reduced the time needed to enter knowledge into the knowledge base? Has it been enough? Those questions have different answers. As for the first question – has the context mechanism actually sped up knowledge-entering? – the answer is Yes, it has had two positive effects:

1. Assertions (facts, rules, etc.) entered into a specialized context tended to be more terse and simpler in form than if they had been entered into one flat KB.
2. There's a good chance that some similar and relevant assertions will be close at hand, in that same (or nearby) context, which someone can copy & edit.

But turning to the second question – is this speedup in the rate of ontologizing *sufficient*? – the answer is No, the gains have been undermined and offset by this cost:

3. It is distracting and time-consuming for the knowledge enterer to select (or, even worse, *create*) precisely the right context each time he/she enters a new piece of domain knowledge. And it can't be skipped: doing a slipshod job of placing each new assertion just leads to bugs (wrong answers, failure to apply relevant knowledge, etc.) and thence a large amount of "debugging" that must go on – i.e., where a "wrong answer" turns out to be due to someone simply having placed an assertion into the wrong context. Or, just as bad, the knowledge enterer ignores the context mechanism and the result is a small number of enormous contexts.

So the challenge is to find a good intermediate-granularity carving of the KB into contexts:
too coarse-grained is like having no contexts at all: rules are huge and difficult to debug
too fine-grained makes it hard to find the right context for a new piece of knowledge

This subsection has focused on how contexts can speed up KE-ing (entering knowledge into the KB: browsing, editing, composing new rules, creating new terms, debugging, etc.) The next subsection addresses the impact of contexts on speeding up the inference engine – the automated reasoning processes of Cyc itself.

2.3 Contexts catalyze efficient inferencing

When we are asked to solve a problem P – for example, "Is someone *a priori* more likely to be carrying an umbrella open or closed, if it's raining hard?" – there is some information which is (likely to be) more relevant than other information. In this case, the purpose and operation of umbrellas is more relevant than the numbers of legs a spider has, or the date that Julius Caesar died. To make the inference as efficient and sensible[7] as possible, heuristics are brought to bear. Here are two such special-purpose guidance mechanisms that Cyc employs:

1. Some of those heuristics are compiled into a set of so-called heuristic-level (HL) modules; they mainly increase the efficiency of the inference process. They recognize special cases – certain kinds of questions/statements – and when such a case arises the HL module grabs control and handles that special case by dint of some customized, optimized data structures and algorithms, rather than falling back logical deduction (the method of last resort, despite – because of? – its generality). E.g., answer questions like "Are any ducks also elk?" by quick tree-searching, not slow theorem-proving.

2. Some of the heuristics are explicit search-guidance heuristics invoked by the inference engine; they increase both efficiency and sensibility[5] of the inference process, though it might be prohibitive to have a huge number of these. These heuristics guide the inference engine in deciding which sub-problem to tackle next, at any given moment; what tack to take on that problem; how much resources (cpu time, elapsed real time, disk space, real-time queries and email messages sent to human beings) to spend before giving up or trying a new approach on it, and so on.

Contexts provide a third mechanism Cyc can use to guide inference:

3. Some of the heuristics can be implicitly encoded in a subsetting of the knowledge base; that is, carving it up into meaningful subsets such that intra-subset deductions are more likely to be productive than cross-subset deductions***. Each of those subsets is a context.*** Thus, carving the KB up into a set of contexts should increase both the efficiency and sensibility of inference.

[7] Here, the term "*efficient*" refers to the amount of time or processing power used to get an answer; and the term "*sensible*" refers to how sane *versus* silly the current inference sub-sub-…-sub-goal looks to a human being, if you interrupt the inference process at random and ask about it. E.g., if you ask whether Iran might use biological weapons against Iraq, then it's probably not very sensible if a sub-…sub-goal is to decide whether or not Nelson Mandela is a piece of lawn furniture, or how many legs a spider has. More precisely, "*sensible*" has to do with the success rate, in hindsight, of sub-…-sub-goals, where "*success*" means both satisfiability and inclusion on a final solution path.

If the carving-up of the knowledge base into contexts is **too coarse** (an extreme case would be if every assertion were placed into one big context) that's suboptimal because (i) lots of irrelevant assertions will be placed in the same context, and (ii) many of the assumptions will only actually be *relied on* by a small fraction of the assertions in that context. If the carving-up of the KB into contexts is **too fine-grained** (an extreme case would be if every assertion were placed in its own separate context) then that's suboptimal due to the high cost of lots of liftings (importings from one context into another) that would need to get done. So the challenge is to find a very good intermediate-granularity. This reinforces the conclusion of Section 2.3, where – to maximize the speed of knowledge entering and KB building – we also didn't want there to be too coarse- or fine-grained a carving up of the KB into contexts.

2.4 "Lifting" (Importing) Assertions From One Context to Another

In section 2.2 and 2.3, we saw that contexts may be the key to drastically faster knowledge-entering and also to drastically faster inferencing; they have already made the difference between huge knowledge-based systems existing – namely Cyc – or not existing.

At any given moment, some assumptions will not yet be articulated or even suspected. When this lack causes some subsequent wrong answer or behavior, the missing assumption will be painfully clear, and can then be formalized and added as an explicit assumption. This is a form of knowledge debugging, in many ways analogous to program debugging. It is a correctness issue, not just an efficiency issue.

When we lift an axiom P from context C1 into context C2, and use it there in some inference (with, say, axioms Q, R,… from C2), we need to ensure that there is no logical inconsistency between the C1 assumptions and the C2 assumptions. If assertions only depend on a few of their context's assumptions, this will prevent a lot of otherwise-useful and valid lifting from taking place. So: most (if not all) of the assertions in a context should actually rely on most (if not all) of the *domainAssumptions* of that context. Otherwise we will get hamstrung by some *irrelevant* assumption from C1, when we try to use P outside of C1.

So what do you do if a C1 "assumption" α only pertains to a sliver of C1's assertions?

- Split off that sliver as its own little context C1' (and make α one of C1''s assumptions)
- Or just conjoin α to the antecedent of each of those assertions and leave them in C1.

Here's another way to visualize this situation. Suppose some assertion P109, e.g., is true in context C1; we want to "import" P109 into context C2 and use it there. What we really want to do is assert some P109' whose meaning there (in context C2) is more or less the same as the meaning of P109 in context C1. See Figure 2h.

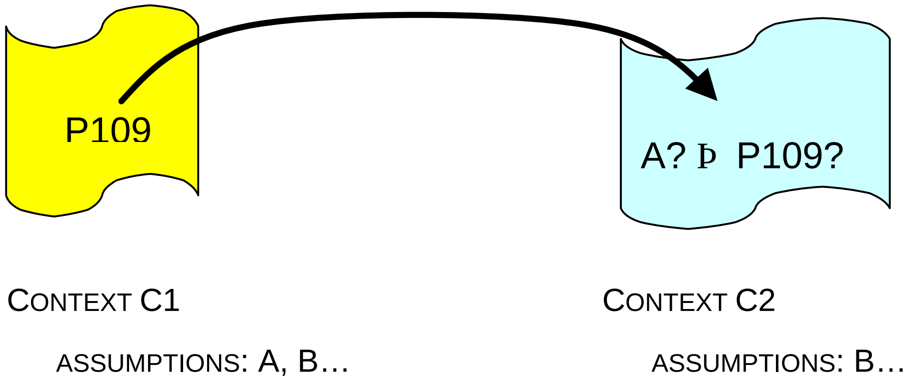


Figure 2h. Lifting an axiom P109 from one context to another.

Sometimes we can just go ahead and assert P109, over into context C2, with no changes, and it will all work out fine. But often it turns out that some of C1's assumptions – on which P109 depends – are not made in C2, and it would be better to conjoin them on the antecedent of P109. For instance, suppose

C1 = the context of the Heaven's Gate cult beliefs;
A35 = "The Heaven's Gate cult beliefs are right."
C2 = the editorial position of the New York Times in the 1990's
P109 = "The comet Hale-Bopp has a UFO hiding behind it."

One of C1's assumptions is A35: This is a core assumption of the entire Heaven's Gate microtheory C1. An unsigned New York Times editorial mentioning P109 would never just assert P109, as though it were factual, it would say something like "If the Heaven's Gate cult's beliefs are right, then the comet Hale-Bopp has a UFO hiding behind it." In other words, the *lifted* form of the assertion P109 is *not* identical to assertion P109 but rather: "If <A35> then <P109>." Assumption A35 was tacked onto the antecedent of P109, because A35 is not one of the assumptions of C2.[8]

We can never fully explicate all assumptions underlying the knowledge we express. No matter how hard we try, some assumptions are bound to be left implicit. Contexts, just like the real-world situations they describe, are "rich objects;" i.e., there is an infinity of things

[8] There was no antecedent before, so the antecedent was effectively just "TRUE". I.e., TRUE $\Rightarrow$ P1. So it became (A1 $\wedge$ TRUE) $\Rightarrow$ P1, which is of course just A1 $\Rightarrow$ P1. In Figure 2h, above, we actually wrote this as A?$\Rightarrow$ P109?, to reflect the possibility that some other aspects of P109, and even of A, might be slightly changed in C2 compared to C1. E.g., the arity of a predicate might be different, or the spelling of a term.

one could say about a context, and of course only a small finite number of those will ever be explicated.

One extremely special case is when one wants to import an assertion from a context to a more specific context (e.g., sporting event to baseball game; or baseball game to professional baseball game.) In that case, all the domain assumptions are (by default) still satisfied in the more specialized context, so any assertion can just be imported (unchanged) from the general context to the more specialized one. This is still just a default, of course; e.g., there could be an assertion in the specialized context that contradicts and overrides it.

A special case of “importing to a more specialized context” is the case of importing from the BaseKB (the most general context of all). There should be precious little in the BaseKB, since all of it must be freely importable – unchanged – to every other context.

A more complicated case arises when a domain assumption has the form of an implication – i.e., it is some general “if/then rule”, intuitively. For example, suppose one of the assumptions of a military infantry-combat context (call it `MilInfantryCombatMt)` is that people mentioned anywhere in that context are (by default) male:

```
(domainAssumptions MilInfantryCombatMt
                   (implies
                         (isa ?x Person)
                         (isa ?x MalePerson)))
```
[9]

And suppose we’re lifting an assertion P1 which has a conjunct `(isa ?x Person)` in it – that is, in its antecedent or its consequent. Then the lifted form of P1 in some more gender-neutral context (one which doesn’t assume the bias that people are by default male) should have that conjunct replaced by `(isa ?x MalePerson).` For example, consider a rule in `MilInfantryCombatMt` like “almost everyone over 30 has a wife.” Suppose this rule happens to be true there – remember, in that context “everyone” refers to male infantry soldiers. Lifted into a less gender-biased military context, it would become “almost every man over 30 has a wife.” Lifted into a gender-biased but non-military context, it would become “almost every infantry soldier over 30 has a wife.” And if the rule were lifted into a context that was neither gender-biased nor military, then it would become “almost every male infantry soldier over 30 has a wife.”

[9] The notation (F a b c d) is so-called prefix notation for a function or predicate being applied to arguments; i.e., it is the same as F(a,b,c,d). Thus, e.g., (implies p q) means implies(p,q) which means p ⇒ q. As another example, (isa x Y) means isa(x,Y) which means x e Y; i.e., x is an element of the set Y.

A related, but even more complicated, case is where there is an *implicit* "personhood" of x, either because of other assumptions, or argument-type constraints, or information lifted from some third context, etc. In that case, a whole new clause of the form `(isa ?x MalePerson)` must be added (to the antecedent of the rule). In our last example, if the original rule in `MilInfantryCombatMt` just had the variable *x* be universally bound, then when lifted into a context in which some of the performers were non-humans (animals, corporations, etc.) a whole new clause `(isa ?x Person)` would have to get added.[10]

In cases where it's not so clear how exactly to insert the old context's assumption into the assertion being lifted, one can always fall back on the crude but logically equivalent approach of conjoining, to the antecedent of the assertion, all the assumptions of the old context it's being lifted from. E.g., in the case discussed above, one of the conjuncts that would get added would be `(implies (isa ?x Person) (isa ?x MalePerson)).`

2.5 Each assertion has a domain (context) of applicability

So far we've been talking about contexts as though they're the real objects, and they happen to have (i.e. contain) assertions. There is an alternate way to view things, a "dual" view, in which assertions are the real objects, and they happen to have (i.e. hold true in) some contexts. This section explores and expands that point of view.

Each assertion P is true in some contexts, and false in others. So in theory you could write down any old random assertion P – for instance, "Pigs can talk"– and then assert it to Cyc, so long as you thought about and articulated (to Cyc) the contexts in which P holds – such as the novel *Animal Farm* and the movie *Babe* and the TV show *Sesame Street* and various other fictional stories and delusional belief systems. You'd then eventually make this more parsimonious by finding and asserting the most general mutually-incommensurable contexts in which P held.

Letting each assertion define its own unique context is not such a great idea efficiency-wise, though. As we discussed in Section 2.2., finding the right context would then be a nightmare. And for reasons discussed in Section 2.3, inference would be slower rather than faster, because of all the lifting that would constantly have to get done.

[10] If the new context lacked gender-bias, then the new term added wold be `(isa ?x MalePerson).`

The basic idea still is valid, though: when you decide to enter some new assertion P into the KB, you should think about, and articulate to Cyc, the (most general) contexts in which P holds.[11] It's just that there should be some faster way to do that than by browsing through a million contexts and pointing at the right one. Snazzy graphical browsing interface tools won't by themselves help enough here; after all, even $\log_2(1{,}000{,}000)$ is an awful lot of things to go through by hand every time you enter an assertion.

Similarly, each question Q has an answer (such as one or more sets of variable bindings) in some contexts, and the answer may vary from context to context. If the context of the question is unspecified (or under-specified) then the "answer" could/should really be a set of `<context, answer>` pairs. E.g., "What will the weather be like?" has different answers, depending on who's asking, where they're going, the time of day they're asking about, the season of the year, etc. Even the question "Who's president of the USA?" has dozens of answers, depending on the date for which an answer is sought to that question.

2.6 Those domains are approximated by regions of an n-dimensional space

When we say `(ist C P)`, we mean that P holds in context C. That is, assertion P is true in context C. But what sort of beast is C? We can reify a context, and then give that specific name in place of C. As we discussed above, that would be a bad idea: almost every assertion in the KB could have its own unique context, even though only a minuscule fraction of those contexts actually merit their own names.

On the other hand, we can easily articulate several properties that separate one context from another: level of granularity, time, place, topic, etc. These can be thought of as the labels on different coordinate axes in *n* dimensions. Then each context is a region of that *n*-space.

Point 1: Besides thinking of a context as "a named node in an ontology[12]", let's also think of it as being[13] "a region in some n-dimensional space".

[11] This is akin to your thinking about – and articulating to Cyc – how generally P can/should be stated, how likely P is to be true (absolutely or compared to other assertions), how explosive it could/would be to have P applied in a forward-reasoning direction, what new vocabulary should/must be introduced in order to efficiently state P to Cyc, whether the English paraphrase of P is okay, etc.

[12] A DAG (directed acyclic graph) of reified context "nodes" related by generalization/specialization "links".

[13] A context may not be 100% specified just by giving its coordinates in 12-dimensional space. The point is that it is *mostly* specified, it is *nearly* unique, once you assign those coordinates. That's why the title of this section says "approximated by" rather than "are".

Specifying a context therefore entails specifying or locating a point or region along each of those *n* dimensions.[14] This applies to asserting some proposition, to asking for an answer to some question, to asking for a justification, etc.

The old Cyc *genlMt* predicate is just one simple relationship that can hold among two contexts; it roughly corresponds to "proper subset-of (or similar restriction) in some *m* dimensions, and no change (that is, equality) in all the other *n-m* dimensions." Thus, in one situation *genlMt* might hold between *The1920sUSA* and *TheTwentiethCenturyUSA* – in this case meaning "is a temporal slice of"; in another situation it might hold between *HumanActivitiesMicrotheory* and *AnimalActivitiesMicrotheory*, meaning "has a more restricted set of default actors than"; in another situation it might hold between *NewtonianPhysicsMt* and *NaivePhysicsMt*, meaning "is a more precise and accurate theory than"; and so on. In many ways, our naiveté in the early 1990's in having just this one vague *genlMt* predicate is akin to the error that other ontologists make when they have just one *a-kind-of* relation, instead of distinguishing – as we've done for 15 years – the relations *isa (element-of), genls (subset-of), genlPreds (restriction-of-predicate), partOf, subRegionOf,…*

So What's The Big Deal about Point 1? Point 1 sounds rather tame, but in fact it embodies a highly unorthodox alternative way of looking at the world; possibly a full-fledged paradigm shift. The three best analogies that come to mind are these:

Analogy 1: Besides thinking of real world substances as named substances in a hierarchy (with nodes like wood, chocolate, iron, silk, etc.), let's also think of each as being a region in some "chemical-formula-space." A set of just 100 chemical elements then generates the huge set of formulae for all possible chemical compounds – plus many impossible ones.

Analogy 2: For those of you trained in physics, another good analogy is to the way that quarks (with their dimensions of strangeness, charm, direction, etc.) are an alternative and – even without their being detectable directly – a much simpler, more organized way of describing the otherwise tangled plethora of elementary particles that have been experimentally observed and measured in cloud chambers and particle accelerators.

[14] *n* is a positive integer denoting the number of largely-independent properties that can be used to partition/classify assertions; i.e., that can be used to distinguish one context from another.

Analogy 3: In the mid-1980's a former Stanford Ph.D student of mine, Paul Cohen, who was by then a University of Massachusetts professor, came up with the idea that the tens of thousands of relations (predicates, slots,…) that we see and name and organize might all be just surface manifestations of "allowed points" in an *n*-dimensional matrix, a "slot-space" if you will. The dimensions of that slot-space are attributes true for some predicates and false for others; e.g., transitivity, insensitivity to time's arrow, and so on.

If you have say a dozen such dimensions, and each one has even just a few different values, that already is a million distinct points in slot-space: in principle far more than enough to differentiate the several thousands of slots/predicates/… named in English, and in Cyc.

Professor Cohen, and his student Cindy Loiselle, and later the Cyc group and others at MCC (notably Michael Huhns and Larry Stephens), put this to use doing something astounding: automatically guessing what the composition of two slots (binary predicates) might be, in cases where that composition turned out to equal some known slot. Consider Figure 2i, below. For each dimension d, they analyzed what the possible d-values could possibly legally be, for the composition R = $p_2 \circ p_1$, *given the d-values for* p_1 *and* p_2. For instance, if both p_1 and p_2 are transitive, then so must be R. This set of constraints is applied to a particular pair of relations such as "owns" and "physical-parts", and yields a set of constraints on the value of R in each dimension. By looking over the set of known predicates, a very small number of predicates may be consistent with all those constraints; in this case, e.g., all that might be left as potentially legal predicates R is the predicate "owns", so the system can hypothesize that if x owns y, and one of y's physical parts is z, then x owns z. E.g., if you own a car, then you also own its steering wheel. Of course this algorithm doesn't always work – there might be more than one consistent predicate, or the composition might not be any known predicate at all – but it's like the tap-dancing bear: the fact it does it at all is amazing. Thus ends Analogy 3.

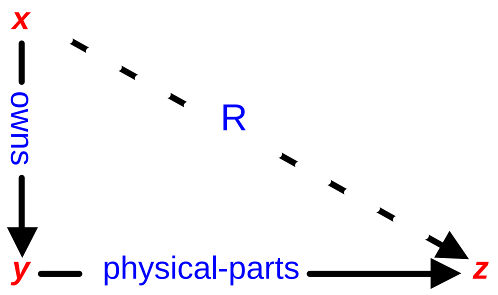


Figure 2i. What is the composition of owns and physical-parts? I.e., if *x* owns *y*, and one of *y*'s physical-parts is *z*, then what relation R holds between *x* and *z*? Is **R** owns? physical-parts? or some other known relation? How could a program make a good *automatic* guess about the identity of that relation **R**?

Criteria for Preferring One Dimension Over Another. The following sort of utility consideration must ultimately be the criteria for deciding which "dimensions" we support or don't support:

Utility of a dimension should be measured by:

- success in separating out mutually-irrelevant portions of knowledge from each other.
- ease of computing the overlap / disjointness of subsections of the dimension.
- how much they tease apart notorious large KB clumps (e.g., HumanActivitiesMt)
- whether the points and regions (especially the extreme ones) correspond to semantically familiar and important real-world concepts and considerations

Utility of a dimension should *not* be measured by:

- how closely it satisfies some arbitrary philosophical/linguistic/… model
- whether there happens to be an existing English word succinctly capturing it
- notions of "cleverness", "cleanness", "novelty", "symmetry", "academic interest"
- the need to have 10 dimensions, or 42, or 666, or any particular favorite number.

Let's stop and recognize that there will be no true limit to the number *n* of dimensions of context-space that one could identify. Although the general domain assumptions/ relevance/lifting problem is arbitrarily hard, we are going to use an 80/20 approach to get the most bang for the buck: There is a small set of *n* useful, important categories along which context-space can be crudely partitioned. The first dozen or so of these dimensions will provide the maximal utility for the minimal cost. We arrived at the figure of a dozen dimensions by enumerating a much larger set of dimensions (over 100), and then clumping together the two most closely related, again and again, until there were no closely related dimensions any more.

The Order of Dimensions Matters, and There Can Be Repeated Dimensions

When we have a set of universal and existential quantifiers, the order matters of course. The same is true when we have a set of dimensional qualifiers. Thus, e.g., "For all moments during the 1800's, for some place in each world capital, …" is different than if those clauses had been reversed. There is no fixed order in which the dimensions are evaluated; rather, that order is specified for a context by **(earlierDimension** C d1 d2), which defines a partial order of the dimensions d_i such as Time, GeoLocation, etc. However, our expectation is that the order of dimension-constraints will rarely matter.

We also note that it therefore might happen that some dimension gets qualified two or more times, presumably at different places in the layerings of qualifications, above. E.g., ., "For all moments during the 1800's, for some place in each world capital, sometime on Christmas eve, P" where the exact time on Christmas eve that P holds depends on the country. And exactly where, in each capital, depends on what year during the 1800's it is. Those two temporal qualifications (year in 1800s, time of day on Christmas eve) can't be merged; all 3 qualifications need to be there, in that order: time/place/time.

2.7 The Top 12 Dimensions of Context-Space

Well, okay, but what *are* the dozen most useful dimensions of context-space? And what sorts of values are there along each dimension? Are certain regions of this *n*-space "special" or "prohibited"? Are there some facets (attributes, modifiers) for those values? What about 100 sub-dimensions[23] that led to this set of a dozen "clumps" of dimensions?

The 12 (classes of) dimensions of context-space we will discuss in Sections 3.1 – 3.12 are:

1. **Absolute Time**: a particular time interval in which events occur
2. **Type Of Time**: a non-absolute type of time period, such as "just after eating"
3. **Absolute Place**: a particular location where events occur, such as "Paris"
4. **Type Of Place**: a non-absolute type of place, such as "in bed"
5. **Culture**: linguistic, religious, ethnic, age-group, wealth, etc. of typical actors
6. **Sophistication/Security**: who already knows this, who could learn it, etc.
7. **Topic/Usage**: drilling down into aspects and applications – not subsets
8. **Granularity**: phenomena and details which are (and are not) ignored
9. **Modality/Disposition/Epistemology**: who wants/believes this content to be true?
10. **Argument-Preference**: local rules for how to resolve pro-con argument disputes
11. **Justification**: are things in this context generally proven, observed, on faith…
12. **Let's**: local bindings of variables etc. that hold true in that context

Point 2: Placing each new assertion P into the ideal context is expensive today. Idea to speed it up: have it be mostly just stating/choosing 12 meta-level values.

Point 3: Automated inference over a million-rule KB is expensive today. Idea to speed it up: localize the search, favoring the content in same/nearby contexts.

As Point 2 (in the box, above) says, this should speed up Knowledge-Entering. As Point 3 says, this should also speed up Inferencing.

Before treating each of the dozen dimensions in detail, let's address the question of what some also-ran would-be dimensions are, and why they weren't chosen, and how they'll be handled under the new system.

Non-Dimensions. There are several *non*-dimensions or *pseudo*-dimensions; that is, things that at first glance might appear to be possible dimensions, but at second glance appear to *not* be such after all. E.g., the following phenomena must be handled somehow, we just suggest that these *not* be done by trying to tease them out as 8 more dimensions:

- a tag about the probability/certainty/monotonicity of a set of assertions
- a tag about whether a set of assertions is meant to be run forward/backward
- a tag about whether typical propositions in some context – which has *some* time/place/sophistication/… interval specified, such as "The1920's" – are claimed to hold throughout that entire interval or just at some points/subsets of that interval. (Or even worse, having a "dimension" like this for each of the 12 other dimensions.)
- other tags, similar to the last one, but talking about whether intervals should be assumed to be open (not including their endpoints) or closed, solid or disjointed, finite or infinite or semi-infinite (infinite in one direction only), and so on.
- a specific region of any (valid) dimension
- quoting. This means that we want to represent the fact that #$Lenat is a Cyc term, and #$Lenat is a person, and yet people are tangible and Cyc terms aren't. What's the solution? We can use a kind of *quoting* operator; so if we write the expression `(quote #$Lenat)` or `'#$Lenat,` that *represents* the Cyc term `#$Lenat` (which then in turn represents the person). Without the quote-mark, `#$Lenat` *is* a Cyc term and *represents* the person, not the Cyc term.
- physical attributes, such as: Temperature, Pressure, Orientation, Stability, etc. of the typical objects and events talked about in this context.
- The level of "anthropacity" of the context, by which we mean what does it presume that there "is" in the world: a physical universe? Like ours? People? Human history as we know it? Etc. This can largely be computed from the various assertions (and assumptions) of the context. E.g., if assertions talk about *I Love Lucy* shows then the world must be pretty close to ours.

The Final 99% of the Dimensions. Specifying values for the most important dozen dimensions doesn't always fully specify the context, but it does dramatically cut down on the number of contexts (worth distinguishing) once you've specified the values for all 12 dimensions. This is analogous to a hashing function, where there still might be a few input items that hash into the same "bucket", but if the programmer has done his/her job well, then that's usually not a problem. In this case, each 12-space region might be considered a "bucket" of contexts; the contexts in that bucket would organize (via *genlMt* and other inter-context relations) into hierarchies. The seven non-dimension features bulleted above would not be distinguished in that imagined hash table, even if there were several reified contexts at one of those terminal buckets.

A note on the “number of buckets”: even if we only distinguished ten points in each dimension, that means that trillions of distinct context “buckets” would thus be defined. Each context in that bucket would, presumably, make some additional assumptions that didn’t quite fit along any of the 12 dimensions; e.g., making assumption about the temperature, or pressure, etc. in which events (talked about in assertions in that context) transpire. Temperature and Pressure may not be worthy of being dimensions in their own right, hence assumptions involving them are stuck on, at the end, as domain assumptions of that context, after it’s been situated into 12-space. As with hashing, if we do a good job in selecting the top 12 dimensions, then all the rest, the final 99% of the dimensions, won’t matter that often, *vis* efficient KE-ing and inferencing. So some of the domain assumptions of a context will now be specified by its 12-space location, and the leftover domain assumptions will be specified explicitly (when they are known/thought about) as Cyc formulas which are then the listed *domainAssumptions* of that context.

The Plan for Sections 3 and 4. In the 12 subsections of Section 3, we will treat these chosen 12 dimensions one at a time, covering such topics as:

- the basic idea behind each dimension
- the reasons why it is an important dimension
- examples of extreme and typical points/regions along that dimension
- the new vocabulary of terms (collections, predicates, functions, individuals, etc.) that we have now in Cyc to handle that dimension. These are terms are used:
    - to specify points on that dimension
    - to specify regions of that dimension
    - to specify relative positions of points on that dimension
    - to specify relationships between regions of that dimension
- our current ideas and plans for future extensions of that vocabulary, if any

Finally, Section 4 discusses some pragmatic issues about how to thrive in this more complex 12-dimensional context world – from both a KE-ing and Inferencing point of view.

3 Dimension-by-Dimension Discussion

3.1 Dimension: Time

Often, one context C1 is separated from another C2 in that the things true in C1 are true at a different point/region of the timeline than the things true in C2. E.g., one context might be temporally situated “during the Crusades”, and another might be talking about events “during the 1990’s”. Obviously many of the things that are true in one of those contexts will be flat out false in the other.

So this – Time – will be our first dimension. Each context in Cyc will have a piece of time associated with it. The assertions that hold true in that context will be presumed to be occurring during that piece of time.

What makes this such a great dimension to have is the fact that, in the real world, causes precede effects by a small (or zero) amount of time. Whether the action is spilling a glass of water, or fighting a battle, the objects interacting with each other will generally be intersecting in time, and their interactions will take place in that piece of time.

That piece of time can be as simple (e.g., a point, a solid interval) or as complex (e.g., a series of discontinuous pieces such as “every second Wednesday morning”) as any piece of time in Cyc. It can be named (e.g., TheYear1956) or not. It can have some significance that goes beyond being the Time locus of the assertion (e.g., WorldWarII), or not.

There are a few thorny issues here. For the purpose of this discussion, let C be a context with Time value t. And let’s suppose that assertions A1, A2,… hold true in C.

3.1.1 Always-True-During *vs.* Sometimes-True-During

Is A1 true during the entire piece of time t, or just true sometime during t? Often the answer is pretty clear (e.g., “JFK died on November 23, 1963” means sometime during that day; “Doug was a 6th-grader on November 23, 1963” means for that entire day).

Sometimes it’s not so clear what’s meant (e.g., “Jane was married during the 1980’s” – though even in an ambiguous case like this one, “during the 1980’s” means presumably that there is some single unbroken interval M of time that intersects the 1980-90 decade-long interval such that Jane was married during that entire interval M.)

3.1.2 Temporal Projection.

Given that assertion A1 holds at a certain time, it’s likely to continue holding for “a while” and it’s likely that it had already been true for “a while.” How can Cyc automatically decide how far into the future/past a certain condition persists? More precisely, there is a probability or likelihood distribution over time – both before and after the stated time period in which an assertion is stated to hold – of whether it was/will be still true.

For example, consider “Billy was in pain when he fell and sprained his ankle.” You know that he probably wasn’t in pain just before that, not even a second before that moment (unless we have some other argument for why he was); and you know that Billy probably was in pain 10 seconds later. How about 2 minutes later? 20 minutes? 2 hours? 2 days? 2 weeks? 2 months? 2 years? Somewhere along the way, your answer changed from Yes to Maybe to No (meaning: at least not just because of this specific accident).

The “proper” way to treat this would be for someone to specify some sort of distribution in which the likelihood of Billy being in pain (because of this accident) is zero before the accident occurs, then discontinuously jumping up to about 100%, and then falling off for a while until it reaches a near-zero asymptote after a day or two.

This is a dangerous tarpit; we don’t want our ontological engineers sitting around all day agonizing over nuances in probability distributions – that would be even worse than the mid-1980’s bout we went through of agonizing over numeric certainty factors. So what’s our proposed solution? We present it as Point 4:

Point 4: Each assertion points to one of a handful of persistence distributions (spike, step, uniform, normal, etc.) and gives a crude estimate of the parameters of that distribution (e.g., mean and standard deviation in the case of a normal distribution, or max/min and overall length in the case of a less regular one).

3.1.3 Relative *vs.* Absolute time

Much of this Time information will be relative, not absolute. What do we mean by “relative”, here? Let’s explain this by considering an example. Suppose you drop a raw egg on a hard stone floor from a height of about 6’. What will happen?

- First, for a small period of time thereafter, it will be moving downward.
- Then, when it contacts the stone, it will stop moving and start breaking
- In less time than it would take me to jump out of the way, the contents will spatter out.
- For a short time interval following the splattering, I’ll have wet egg on my clothes.

The durations and starting/ending times of these various events could be specified in an absolute sense, by citing a particular duration (e.g., "0.532 seconds"), or a particular point on the timeline (e.g., "14:52:09.003 on Tuesday, October 13, 1998."). But they aren't. They're specified relative to other events' durations (such as "less time than it would take me to jump out of the way") and starting/ending time points (such as "when it contacts the stone.") These are *relative* ways of specifying duration and starting/ending times.

3.1.4 Inferring (e.g., inheriting) those 3 types of meta-level information

So how does Cyc decide whether assertion A1 is true sometimes-during or always-during the time period of context C1? How does Cyc decide whether it applies to some particular absolute piece of time, and/or standing in some relation to some other event or piece of time? How does Cyc decide how long in the future that assertion will remain true? To make the task of specifying these three types of meta-level information much less time-consuming, each of them will almost always be inferable (i.e., computable) from the predicates and other terms used in the assertion, and/or be a default attached to the context as a whole. In rare cases some special meta-level assertion about A1 will have to be made. A similar tripartite scheme (context-wide unless predicates-involved unless idiosyncratic-for-that-assertion) applies to most aspects of most context-space dimensions, not just Time.

Point 5: Most assertions' persistence-distributions are "inherited" from a single default distribution attached to their context (and, in turn, even that is typically inherited/inferred); a few assertions get this overridden by dint of their particular predicates (such as **gender** ***and*** **birthDate*****) [usually their main predicate] possibly with some typing of arguments to that predicate.***[15] ***And, finally, there is the very tiny fraction of the KB's assertions that need to have a manually-entered persistence-distribution. A similar tripartite scheme (context-wide unless predicates-involved unless idiosyncratic-for-that-assertion) applies to deciding if it's sometimes-true-during C1*** **versus** ***always-true-during C1; etc.***

So the ontological engineer will hopefully only rarely have to attend to the persistence distribution type, and its parameters – for example, they might need to attend to it explicitly during debugging. And, during debugging, a good interface might let them modify those parameters (semi)automatically, by complaining that the truth of a certain assertion should – or should not – have persisted to a certain point in time.

[15] For example, (wetnessOfObject x Wet) will have very different temporal projection if x is meant to be in the water (e.g., a buoy) *versus* might be in the water a while (e.g., a swimsuit) *versus* meant never to get wet.

That same argument applies to the absolute and relative specification of the Time interval itself: In most cases, both the duration and starting/ending times ought to be guessable (semi)automatically by the system, based on terms used in the assertion. At the least, relative constraints on the Time value should be inferred from the assertion, connecting that piece of time to other pieces of time. In the case of "George Washington wears wooden teeth", e.g., the Time period of the objective content is during that person's lifetime, and also during the time period of commercial use of wooden teeth.

Sometimes a piece of time is so simple that it's just a continuous block of time from some starting point to some ending point. In that case it can be represented by two numbers, if the absolute endpoints are known. Sometimes one or both of the endpoints won't be "in" that piece of time. E.g., the time *after* 3pm and *before* 4pm today does not include either endpoint. Even in less trivial cases, pieces of time can be naturally partially-ordered by inclusion, and they thereby form into a lattice.[16]

Most of the above remarks apply to more dimensions than just Time. E.g., in a New Jersey context, someone states "Gambling is legal"; does that mean it's legal somewhere in the state, or everywhere in the state? Does New Jersey include its borders or just the land *inside* those borders?

3.1.5 True-in-time *versus* Relevant-in-time

One final issue to address here is to distinguish "true-in-time-x" from "relevant-in-time-x". This is a somewhat subtle point, but turns out to be very important for efficiency reasons – including both efficiency of knowledge-entering, browsing, and inferencing.

Consider the sentence "Bill Clinton is the President of the USA." This is true in some contexts (e.g., TheYear1997) and false in other contexts (e.g., TheYear1900). Now consider the sentence "Bill Clinton is the President of the USA during 1997." This statement is true for all time, not just during 1997. In the year 9999 it will still be true. In 1900 no one knew this fact, but it was still true. It was largely irrelevant in 1900, in the sense that no one then acted differently because of it (since no one knew it.)

Thus: Many assertions are strictly speaking *true* at all times, or for a wide swatch of the timeline, but are only *relevant* to some small sliver(s) of time. This is a strong enough schism that there should in effect be two Time dimensions, one for truth and one for relevance. At any moment, P is not true iff (if and only if) not-P is true. But it is often the case that both P and not-P are irrelevant, at the same moment.

[16] If the time intervals have no gaps, they form an inter-ordinal lattice, otherwise they form a Boolean lattice (i.e., the power-set of all time points.)

3.1.6 Temporal Granule Size

Each context has some notion of what it means, in that context, to be a “moment” of time. Differences in time smaller than that granule size are just not even talked about in that context. For a track and field event, it might be 0.01 second; for a macroeconomic theory of nineteenth century Europe it might be 1 day or even 1 year.

The temporal granule for a context will almost always have a known duration. It will often have some tie-in to some absolute calendar-type time. Furthermore, some contexts will have granules of a few different sizes. E.g., for a New York Stock Exchange context, the granules might be tied to one trading-day; hence one would be an 8-hour block of time on a non-holiday weekday starting at 9am and ending at 5pm East Coast time; one would be the 16-hour period between most trading-days; and one would be the 64-hour period “over the weekend” between close of business Friday and the market’s opening on Monday morning.

3.1.7 Vocabulary for expressing this in Cyc

Putting the above discussion all together, we have for Time the following:

- One portion of a context’s content is just “true”, the rest is both “true” and “relevant.” One portion of a context’s content holds throughout a whole time-interval t_i, the rest just holds sometime during that interval.[17] These two binary choices effectively carve each context up into four discernable regions.[18] In Cyc there are 4 functions that connect a context to these 4 more specialized (and probably non-reified) contexts. See Figure 3a.

(#$**Relevant+AlwaysFn C t) = C1** means that the assertions in context C1 are precisely those assertions that hold in C and which happen to be both true and relevant throughout the entire temporal extent of the TemporalThing **t**.

[17] The temporal quantifier could be For Some Of, For All Of, For Most Of, Occurs Cyclically Within With Cycle *n*, Occurs *n* Times In, etc. A new Collection called TemporalQuantifier could be created, with those instances. But for now we will try and get by with just For Some Of (…-Sometimes…) and For All Of (…-Always...)

[18] Given *n* associated temporal objects t_1, t_2, t3, …, t_n the content of a context C is divided up into 4*n* binary partitions: (Relevant+AlwaysFn C t_i), (AlwaysFn C t_i), (SometimesFn C t_i), (Relevant+SometimesFn C t_i), for each of those *n* temporal objects. Often, there will just be one associated temporal object; i.e., often n=1.

(#$**Relevant+SometimesFn C t) = C2** means that the assertions in context C2 are precisely those assertions that hold in C and which happen to be both true and relevant during at least *some* portion of the temporal extent of **t** (a #$TemporalThing.) Note that (for all C and t) (genlMt (#$Relevant+SometimesFn C t) (#$Relevant+AlwaysFn C t)).

(#$**AlwaysFn C t) = C3** means that the assertions in context C3 are precisely those assertions that hold in C and which hold true throughout the entire temporal extent of **t**. Note that (for all C and t) (#$genlMt (#$AlwaysFn C t) (#$Relevant+AlwaysFn C t))

(#$**SometimesFn C t) = C4** means that the assertions in context C4 are precisely those assertions that hold in C and which are true during at least *some* portion of the temporal extent of **t**. Note that C4 will be a specMt of C1, C2, C3, above. For *some* **t**, C4 = C.

- Another component of each "value along the Time dimension" is the granule size and, if known, a more absolute calendar-based characterization of the granules. This indicates how small a piece of time is before it's considered an indivisible moment. We write **(mtTemporalGranuleSize C t)** to indicate that context C has, as such a granule size, the size of the duration of temporal object t.

If we know not just the right duration, but the particular granules themselves, we write **(mtTemporalGranule C TSET)**, where TSET is the Cyc collection of such granules. For example, (mtTemporalGranule NYSE-Mt NYSETradingDay). We might tell Cyc that the granule *size* of the Winter1998OlympicsMt is .01 seconds, but we wouldn't have a collection of the particular hundredths of seconds involved. So we could make a mtTemporalGranuleSize assertion about Winter1998OlympicsMt, but not one about mtTemporalGranule of Winter1998OlympicsMt.

Thus the meaning of SometimesFn and Relevant+SometimesFn is slightly more complicated than we mentioned above. Namely, they mean that assertions hold for some *moment* in that context.

- Another component of each "value along the Time dimension" is a persistence-distribution. This is a default for how likely it is that a typical assertion that's made into context C will be true earlier/later than the time associated with the C. That distribution d is specified via an assertion of the form **(mtTimeDistribution C d),** where d itself will be specified in terms of the appropriate parameters (see Section 3.1.2.) There will likely be exceptions that override it, explicitly. That is, an assertion P in context C having a different persistence distribution: **(timeDistribution P d′)**

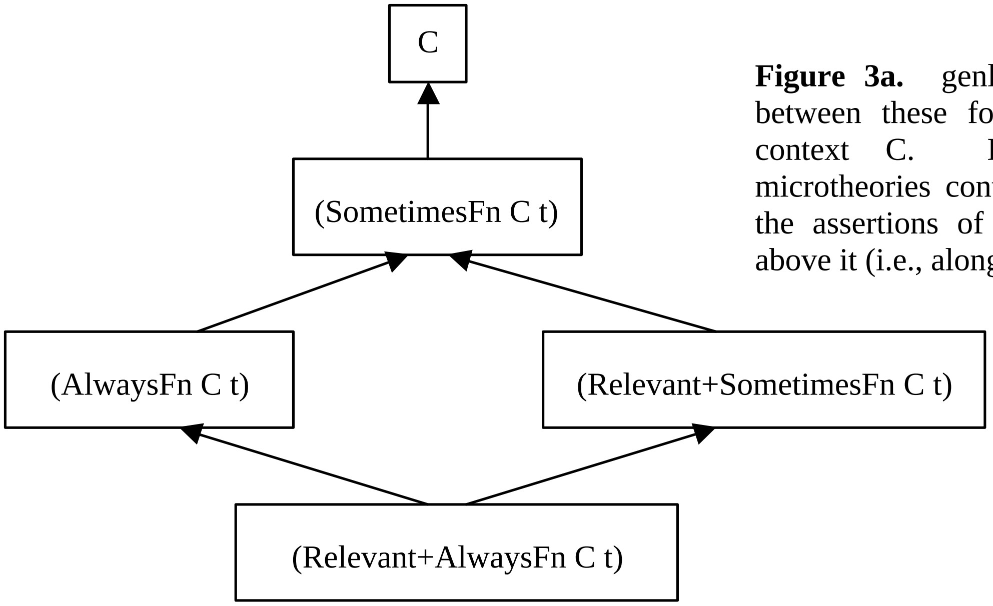


Figure 3a. genlMt relationships between these four regions of a context C. Each of these microtheories contains a subset of the assertions of the microtheory above it (i.e., along an arrow).

For example, if we want to have a 1920sUSA context, it's as if there were 4 contexts:

TrueThroughout1920sUSA-Mt	TrueSometimeIn1920sUSA-Mt
RelevantThroughout1920sUSA-Mt	RelevantSometimeIn1920sUSA-Mt

From the user's point of view, an interface should let them quickly and easily find/create a single USA context – USAMt – and then associate *n* temporal objects with that context, and then (for each one of those *n* temporal objects) enable them to make two yes/no choices (True/Relevant? and Sometimes/Always?) when assertions are entered, questions are asked, etc. Those choices could have the form of in/out pushbuttons whose settings get guessed at, so the user rarely has to take the time and trouble to change any of those settings. If no temporal object is explicitly specified, the presumption is that *n*=1 and that the implicit temporal object associated with the context is the largest one possible, consistent with all other constraints (such as the date of invention of the objects mentioned in assertions in the context).[19]

[19] It would be unusual for assertions to be made by hand, locally, into a …True… part of a theory. I.e., if something's true in context C, and it was worth your telling Cyc about it, then it's probably going to also be *relevant* in context C. But as Cyc infers more and more true but irrelevant assertions, this quadripartite scheme gives Cyc a place to keep them around without having them start to slow inference way down, since inference should and can and will strongly favor known-true&relevant information over merely known-true.

In the case of 1920sUSA, we would create a USAMt context, then associate it with the temporal object that is the decade-long solid block of time (DecadeFn 1920), which might get reified as, e.g., The1920s. We would then make assertions about 1920s America into the four contexts defined by these four non-atomic terms:

```
(Relevant+AlwaysFn        USAMt   The1920s)
(Relevant+SometimesFn     USAMt   The1920s)
(AlwaysFn                 USAMt   The1920s)
(SometimesFn              USAMt   The1920s)
```

Facts and rules that were true throughout the decade would get asserted into the first of those four contexts, and things true just sometime during the decade would get asserted into the second context. Something asserted into the first context would be known to hold in the second context, because of the guaranteed genlMt relationship between an …Always… context and the corresponding …Sometimes… context. It would be very unusual for anyone to want to hand-assert anything into the latter 2 contexts; after all, if it's not relevant to 1920's USA why in the world bother asserting it into a 1920's USA context?

That takes care of the first part of specifying the Time dimension value for 1920sUSA. The second and final part is to specify a temporal persistence distribution. E.g., if you are told that P was true throughout the entire 1920's, how likely is it that P also held in December of 1919? January of 1933? Etc. To specify that distribution, we would assert, e.g., (mtTimeDistribution (AlwaysFn USAMt The1920s) ModernHistoricDecayDistrib).

There would be several assertions about ModernHistoricDecayDistrib, which would say that it's a normal distribution, its mean is 2 years, its standard deviation is 3 months, etc. Those assertions about ModernHistoricDecayDistrib would have to hold in some context at least as general as (AlwaysFn USAMt The1920s); probably they would hold in a context broader than just the USA and broader than just the 1920's.

In rare cases we would assert a different distribution for some individual assertions; e.g., every assertion about who the leader of a country is, gets a step distribution.

There will almost certainly be different default persistence distributions for the four specMts of USAMt above, (Relevant+AlwaysFn USAMt The1920s), (AlwaysFn USAMt The1920s)**,** (Relevant+SometimesFn USAMt The1920s), and (SometimesFn USAMt The1920s) . E.g., the probability distribution of the typical …Sometimes… assertion is generally much more rapid and spike-like than a typical …Always… assertion.

So: to specify the decay pattern of the truth of a typical assertion in a context C, we cite an instance d of TemporalPersistenceDistribution; we assert (mtTimeDistribution C d). For an exceptional assertion P, we override this by asserting (timeDistribution P d'). TemporalPersistenceDistribution is a subset of PersistenceDistribution (there are spatial persistence distributions as well.) Initially, each instance d of PersistenceDistribution will be spike, step, ramp, uniform, or normal (bell-curve-shaped), or the appending of two persistence distributions d1 and d2 – i.e., the value of (appendedDistributionFn d1 d2).

1. #$Spike-PersistenceDistribution – The parameters for this sharp spike-shaped distribution are specified using the predicates #$spikeWidth, #$spikePeakValue, #$spikeNonPeakValue, and #$spikePeakLocation.
2. #$Step-PersistenceDistribution – The parameters for this step-shaped distribution are specified using the predicates #$stepLocation, #$stepInitialValue, and #$stepFinalValue. For a distribution that is a series of steps, we may either introduce a new type persistence distribution, but a more general solution is to just have a standard way of forming a compound distribution by appending simpler ones thus: (#$appendedDistributionFn d1 d2).
3. #$Ramp-PersistenceDistribution – The parameters for this smooth ramp-shaped distribution are specified using the predicates #$rampInitialValue and #$rampFinalValue. The ramp's initial starting/ending location is presumed to be the starting/ending-point of the temporal object.
4. #$Uniform-PersistenceDistribution – The parameter for this flat distribution is specified using the predicate #$uniformValue.[20]
5. #$Normal-PersistenceDistribution – The parameters for this bell-curve-shaped distribution are specified using the predicates #$meanDistributionValue and #$standardDeviation. The location of the mean is assumed to be the midpoint of the piece of time corresponding to the TemporalObject. In cases where that's not true – i.e., the distribution is "lopsided", use #$LopsidedNormal-PersistenceDistribution.
6. #$LopsidedNormal-PersistenceDistribution – The parameters for this lopsided-bell-curve-shaped distribution are specified using the predicates #$meanDistributionValue, #$leftStandardDeviation, #$rightStandardDeviation, and #$meanLocation. In effect this distribution is like two normal bell-shaped curves (but with different shapes) that meet at the mean value, wherever that is.

It is not always necessary to create a context just to temporally qualify an assertion. There are two temporally qualifying modal operators (the first of which is today's "holdsIn"):

- **(holdsAlwaysDuring** TTHING P). This is true iff formula P is true at every moment in the temporal extent of the temporal thing TTHING.

- **(holdsSometimesDuring** TTHING P). This is true iff formula P is true at some moment in the temporal extent of the temporal thing TTHING.

[20] . Notice that this is a special case of #$Ramp-PersistenceDistribution, where the #$rampInitialValue and #$rampFinalValue are both the same value. It's also a special case of #$Spike-PersistenceDistribution, where the #$spikePeakValue and the #$spikeNonPeakValue are both the same value. It's also a special case of #$Step-PersistenceDistribution, where the #$stepInitialValue and #$stepFinalValue are both the same value.

3.1.8 When is a formula True?

Suppose P is a Cyc formula, and t is a "moment" (in some contexts C1, C2,…) If P is a compound "molecular" formula, then:

- (not P) is true at moment t if and only if P is false at moment t
- (and P1 P2) is true at moment t if and only if P1 is true at t and P2 is true at t

etc.

For an "atomic" formula, let's consider the typical case (#$owns term1 term2). That formula holds true at moment t if and only if the denotations of term1 and term2 stand in the "owns" relationship at moment t. E.g., suppose Fred owned Spot from 1945-50; i.e., for that whole time, but not before or after that time period. Then the Cyc formula (owns Fred Spot) is true at each moment in 1946, and false at each moment in 1940 and at each moment in 1960. Notice that it doesn't matter whether either or both of them (Fred and Spot) happen to be alive in 1940, nor 1960.

In the (AlwaysFn USA TheYear1948), the assertion (owns Fred Spot) is true, and its negation is false. The same for (SometimesFn USA TheYear1948).

In the (AlwaysFn USA TheYear1908), the assertion (*owns* Fred Spot) is false, and its negation is true. The same for (SometimesFn USA TheYear1908). Note that it doesn't matter whether either or both of Fred and Spot were alive in 1908. That does affect the *relevance* of those assertions, of course. If either wasn't yet born, then the assertion is true but most likely not going to be relevant to reasoning about America in 1908.

Neither (owns Fred Spot) nor its negation would hold true in the context (AlwaysFn USA The1940s). *Both* (owns Fred Spot) and its negation would hold true in the (SometimesFn USA The1940s). That means that Cyc had better not use most rules of inference, such as modus ponens, within that context. There is a class of contexts NonInferrableMt to which this belongs, having the property that in general just because one can deduce Q from assertions in that context does *not* means that one can then safely just add Q there. Another example of such a context would be the one representing sentences that Fred has uttered. Just because he's said "A and B" doesn't mean that he's ever said "B and A." In the case of a …Sometimes… context SC, we can do some reasoning safely, such as combining one of its assertions with one that's known to hold across the entire time interval of SC.

The denotations of the expressions term1 and term2 might change over time. For example, the assertion (livesIn (PresidentFn USA) (CapitalCityFn USA)) is true at all (and only) those moments when the then-current President of the United States lives in the city that is then the capital of the USA. E.g., it is true for George Washington living in Philadelphia for a few years, and for Jerry Ford living in Washington, D.C. for a few years, and so on.

Subabstractions haven't gone away, but the preceding semantics should enable us to deal directly with whole individuals even when making temporally qualified assertions.

Just because (p x y) is true at moment t doesn't mean necessarily that x temporally subsumes t. For example, (inventor ThomasEdison Phonograph) is true today.

3.2 Dimension: TypeOfTime

The default type of time period in which the action takes place, the actors are situated, etc, for the assertions in this theory. For example, a context of ModernWesternSleeping might have Night or NapTime as its TypeOfTime. Other possible values along this dimension will be NATs (non-atomic terms) constructed by applying some sort of "during" operator to some event or process; for example: during an intermission, during an election campaign, and so forth. Cyc's current vocabulary for talking about events may already suffice for the purpose of stating these values. If not, the …Sometimes… functions and operator, below, should fill any gap.

Note that this dimension is relatively independent of Time. Since the TypeOfTime is going to be (one or more) Collections, the genls predicate induces a partial order on the TypeOfTime dimension values.

Putting this all together, a setting for this dimension will be a set of one or more TypeOfTime instances. Appropriate superTimeInterval assertions will be made in the KB.

We use four functions here similar to what we did for Time, in the previous section. Namely:

(#$**Relevant+AlwaysTypeFn C TypeOfTime) = C1** means that the assertions in context C1 are precisely those assertions that hold in **C** and which happen to be both true and relevant throughout the entire temporal extent of each temporal object t which is an instance of **TypeOfTime** (which must itself be a TemporalThingType and is usually a TemporalObjectType). C1 is akin to a "during an entire…" restriction; e.g., could be used to express the idea that P is true during every moment of every intermission, during every instant of every Friday, etc. (Naturally, these can be stated as defaults, with exceptions.)

Here are some sample legal values for what TypeOfTime could be – i.e., these are TemporalThings and (except for Breathing) also TemporalObjectTypes: Swallowing, 1998WinterOlympicEvent, USCabinetMeeting, ChristmasHoliday, and Breathing.

Here are some examples of subsets of TemporalObjectType; instances of these could be legal values for TypeOfTime: #$SportsCategoryType, #$AnnualEventType, #$WeeklyEventType, #$ClimateCycleType, #$ExclusiveTreatment, #$CourseOfStudyType, #$DrugAdministrationRouteType, #$CalendarCoveringType, #$DayOfMonthType, #$TimeOfDayType, #$TemporallyDisjointIntervalType, #$ConveningEventType, $DayOfYearType, #$MonthOfYearType, and #$CalendarSeasonType.

(#$**Relevant+SometimesTypeFn C TypeOfTime) = C2** means that the assertions in context C2 are precisely those assertions that hold in **C** and which happen to be both true and relevant during at least *some* portion of the temporal extent of each **t** which is an instance of **TypeOfTime** (which must itself be a TemporalThingType (a spec of TemporalThing) and usually also turns out to be an instance of TemporalObjectType). Note that (genlMt (#$Relevant+SometimesTypeFn C x) (#$Relevant+AlwaysTypeFn C x)) – that holds true for all C and x. Note that C2 will be akin to a "at some time during a…" restriction; e.g., could be used to express the idea that P is true at some time during an intermission, at some time during a meal, at some time on a Friday, etc.

(#$**AlwaysTypeFn C TypeOfTime) = C3** means that the assertions in context C3 are precisely those assertions that hold in **C** and which hold true throughout the entire temporal extent of every **t** which is an instance of **TypeOfTime** (which must itself be an instance of TemporalThingType and usually also TemporalObjectType). Note that – for all C and x – the following will necessarily be true: (#$genlMt (#$AlwaysTypeFn C x) (#$Relevant+AlwaysTypeFn C x)).

(#$**SometimesTypeFn C TypeOfTime) = C4** means that the assertions in context C4 are precisely those assertions that hold in **C** and which are true during at least *some* portion of the temporal extent of each temporal object **t** which is an instance of **TypeOfTime** (which must itself be an instance of TemporalThingType and usually a TemporalObjectType). Note: C4 will be a specMt of C1, C2, C3, above. For *some* value of TypeOfTime, C4 = C.

It is not always necessary to create a context just to temporally qualify an assertion to hold during a certain type of time. There are two new temporally-qualifying modal operators (besides holdsAlwaysDuring and holdsSometimesDuring) for this purpose:

- **(holdsAlwaysDuringA** TOBJTYPE P). This is true iff formula P is true at <u>every</u> moment in the temporal extent of each TemporalThing which is an instance of TOBJTYPE (which itself must be a TemporalThingType and is probably a TemporalObjectType.
- **(holdsSometimesDuringA** TOBJTYPE P). This is true iff formula P is true at <u>some</u> moment in the temporal extent of each TemporalThing which is an instance of TOBJTYPE (which itself must be a TemporalThingType and is probably a TemporalObjectType.

3.3 Dimension: GeoLocation

The default geographic location of the actors, events, etc. mentioned in the assertions in this theory. In many ways this is analogous to the Time dimension (see Section 3.1).

- Elements of this dimension are partially ordered by containment.

- Each assertion can have a Some/All/... tag that indicates whether it's true somewhere in, or everywhere in, that geographic region. That can often be inferred by examining the main predicate (and other terms) appearing in the assertion, or may even be weakly inferred by the context as a whole. Thus a legal context might by default predict that its assertions will have an "All" GeoLocation bit; but the predicate *livesIn* might predict that *its* assertions are by default "Some" (for example, Ronald Reagan *lives in* California – meaning that he lives somewhere in California but he doesn't live everywhere in California.)

- Each assertion has a spatial range over which its truth value changes from True to False. That might be a crisp step function, but often is more like a distribution. of probabilities. For instance, in a context having New York State as its GeoLocation, we might assert "It's snowy in the winter." This remains true outside New York, but by the time one gets to Ecuador it's false. Rather than handcraft a million special-purpose distributions, and have some complicated reasoning mechanism, we expect that it will suffice to have a handful of commonly occurring distributions, and to specify one of those and give zero or more parameters, just as we do for the Time dimension. In fact, they are the *same* distributions as used there – instances of PersistenceDistribution.

- Location is typically given relative to something else, such as:
  the North Pole and other features on the surface of the earth;
  position in our solar system relative to the sun and the planets;
  on the dark side of the moon;
  on the surface of the sun;
  in the Gulf Stream;
  etc.

- Again as with Time, there is a difference between where geographically something is True versus where it is Relevant. The temperature at which igloos melt is more relevant in Alaska than Ecuador, though it's equally true in both places.

Of course there are some differences between Time and GeoLocation, if nothing else because space is more than just one-dimensional. Richer notions of connectedness, convexity, compactness, paths, junctions/branchings, etc. are possible than in the one-dimensional Time case.

Putting that all together, then, here is how we represent GeoLocation in Cyc:

- **One portion of a context's content is just "true" for some spatial region (usually some geographic region) the rest is both true and "relevant." One portion of a context's content holds throughout a whole spatial region g_i, the rest just holds at some places in that region. These two binary choices effectively carve each context up into four discernable pieces.[21] In Cyc there are 4 functions that connect a context to these 4 more specialized (and probably non-reified) contexts. See Figure 3b.**

(#$**Relevant+EverywhereFn C t) = C1** means that the assertions in context C1 are precisely those assertions that hold in C and which happen to be both true and relevant throughout the entire spatial extent of the SpatialThing **t**.

(#$**Relevant+SomewhereFn C t) = C2** means that the assertions in context C2 are precisely those assertions that hold in C and which happen to be both true and relevant in at least *some* portion of the spatial extent of **t** (a #$SpatialThing.) Note that (for all C and t) (genlMt (#$Relevant+SomewhereFn C t) (#$Relevant+EverywhereFn C t)).

(#$**EverywhereFn C t) = C3** means that the assertions in context C3 are precisely those assertions that hold in C and which hold true throughout the entire spatial extent of **t**. Note that (for all C and t) (#$genlMt (#$EverywhereFn C t) (#$Relevant+EverywhereFn C t))

(#$**SomewhereFn C t) = C4** means that the assertions in context C4 are precisely those assertions that hold in C and which are true in at least *some* portion of the spatial extent of **t**. Note that C4 will be a specMt of C1, C2, C3, above. For *some* **t**, C4 = C.

[21] Given *n* associated spatial objects t_1, t_2, t3, …, t_n the content of a context C is divided up into 4*n* binary partitions: (Relevant+EverywhereFn C t_i), (Relevant+SomewhereFn C t_i), (EverywhereFn C t_i), and (SomewhereFn C t_i), for each of those *n* spatial objects. Often, there will just be one associated spatial object; i.e., often n=1.

- **Another component of each “value along the GeoLocation dimension” is the granule size and, if known, a more absolute geographic-location-based characterization of the granules.** This indicates how small a piece of space is before it’s considered an indivisible point. We write **(mtSpatialGranuleSize C t)** to indicate that context C has, as such a granule size, the size of the duration of spatial object t.

If we know not just the right size, but the particular spatial granules themselves, we write **(mtSpatialGranule C GSET)**, where GSET is the Cyc collection of such granules. For example, (mtSpatialGranule GlobalWeatherForecastingMt HadleyCell), where the world is divided up into Hadley cells but no finer. We might tell Cyc that the spatial granule *size* of the KentuckyDerbyHorseraceMt is the size of a horse’s nose, but we wouldn’t carve up space into coordinate axes labeled in terms of horse’s noses. So we could make a mtSpatialGranuleSize assertion about KentuckyDerbyHorseraceMt, but not one about mtSpatialGranule of KentuckyDerbyHorseraceMt.

Thus the meaning of SomewhereFn and Relevant+SomewhereFn is slightly more complicated than we mentioned above. Namely, they mean that assertions hold for some *point* in that context C (i.e., for 1+ spatial points in each of C’s associated spatial objects.)

- **Another component of each “value along the GeoLocation dimension” is a persistence-distribution.** This is a default for how likely it is that a typical assertion that’s made into context C will be true somewhere outside the GeoLocation of C. That distribution d is specified via an assertion of the form **(mtSpaceDistribution C d),** where d itself will be specified in terms of the appropriate parameters (see Section 3.1.2.) There will likely be exceptions that override it, explicitly. That is, an assertion P in context C having a different persistence distribution: **(spaceDistribution P d′)**

NOTE: In this and the following sections, we use “SpatialThing” repeatedly but, in the current vocabulary, this would actually be #$SpatialThing-Localized. Furthermore, we use the term SpatialThingType which does not exist in the current Cyc vocabulary, and – if it is not created and added there – should be taken to be just Thing; where we say “…a SpatialThingType”, or “an instance of SpatialThingType”, please take that as “a spec of SpatialThing”. Furthermore, we use the term SpatialObjectType which, in the current Cyc vocabulary, does not exist and should be taken to be just ObjectType.

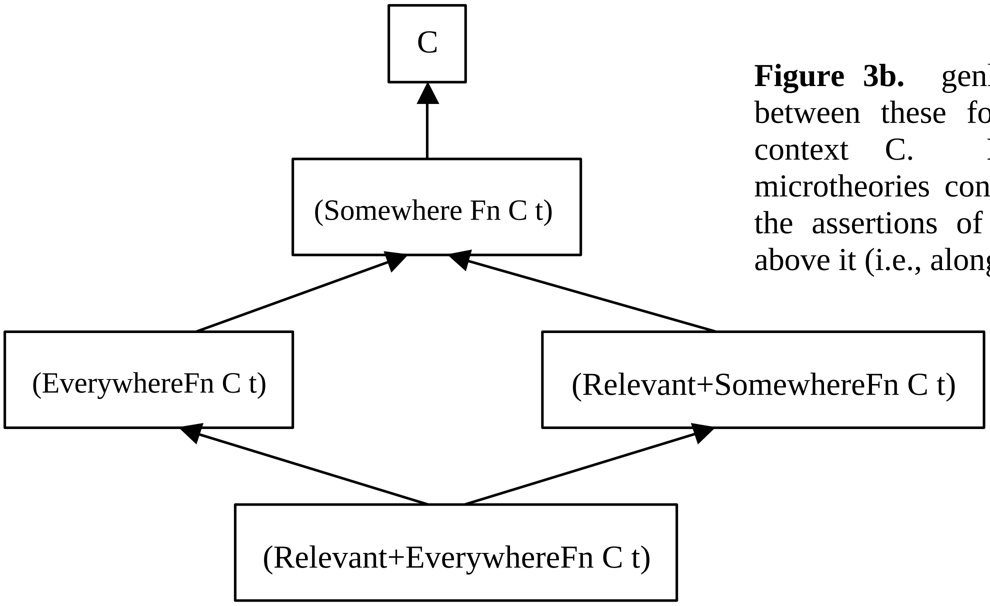


Figure 3b. genlMt relationships between these four regions of a context C. Each of these microtheories contains a subset of the assertions of the microtheory above it (i.e., along an arrow).

For example, if we want to have a 1920sUSA context, it's as if there were 4 contexts:

TrueEverywhereIn1920sUSA-Mt	TrueSomewhereIn1920sUSA-Mt
RelevantEverywhereIn1920sUSA-Mt	RelevantSomewhereIn1920sUSA-Mt

From the user's point of view, an interface should let them quickly and easily find/create a single context – e.g., Farming – and then associate *n* spatial objects with that context – e.g., the continental USA – and then (for each one of those *n* spatial objects) enable them to make two yes/no choices (<u>True/Relevant?</u> and <u>Somewhere/Everywhere?</u>) when assertions are entered, questions are asked, etc. Those choices could have the form of in/out pushbuttons whose settings get guessed at, so the user rarely has to take the time and trouble to change any of those settings. If no spatial object is explicitly specified, the presumption is that *n*=1 and that the implicit spatial object associated with the context is the largest one possible, consistent with all other constraints (such as the date of invention of the objects mentioned in assertions in the context). Note that TemporalThings associated with the context would also exist, and each have a pair of pushbutton settings to specify true/relevant and sometimes/always for the Time dimension of that context.

In the case of the context C99 for “Automobiles in the 1920s USA”, we would create an AutomobilesMt context, then associate it with the temporal object that is the decade-long solid block of space (DecadeFn 1920), and associate it with the spatial object that is the continental United States of America – say that’s been reified as USA-Continental.. We would then make assertions about automobiles in 1920’s America, asserting them into the appropriate one (or more) contexts defined by these four non-atomic terms:

```
(Relevant+EverywhereFn     AutomobilesMt   USA-Continental)
(Relevant+SomewhereFn      AutomobilesMt   USA-Continental)
(EverywhereFn              AutomobilesMt   USA-Continental)
(SomewhereFn               AutomobilesMt   USA-Continental)
```

Facts and rules that were true throughout the country (that decade) would get asserted into the first of those four contexts, and things true just in certain parts of the USA during the decade would get asserted into the second context. Something asserted into the first context would be known to hold in the second context, because of the guaranteed genlMt relationship between an …Everywhere… context and the corresponding …Somewhere… context. It would be very unusual for anyone to want to hand-assert anything into the latter two contexts; after all, if it’s not *relevant* to “Automobiles in the 1920’s USA” why would anyone deem it worth asserting it into that context?

Point 6 describes how this dimensionalizing of contexts is already beginning to apply, even though we’ve just started enumerating the dimensions:

> ***Point 6: Starting with a context about*** **“daytime driving automobiles in rural 1920’s America…”** ***we can use the Time dimension to factor out the “1920’s” part of what the context was, the TypeOfTime dimension to factor out the “daytime” part, the GeoLocation dimension to factor out the “America” part, and we will use the TypeOfPlace and Topic dimensions, respectively, to filter out the “rural” and “driving automobiles” parts. In this case, the whole context will be uniquely specified by specifying a set of values on a handful of our dimensions. There may not be any extra assumptions at all that have to get stated; the context C may just receive some assertions. In other words, C is just a set of assertions plus a set of values along a few of our dimensions.***

That takes care of the first part of specifying the GeoLocation dimension value for this context. The second and final part is to specify a spatial persistence distribution. E.g., if you are told that P was true throughout the entire continental USA, how likely is it that P also held in Toronto? In Cairo? Etc. To specify that distribution, we would assert, e.g.,

```
    (mtSpaceDistribution
         (EverywhereFn AutomobilesMt USA-Continental)
         ModernCultureSpatialDecayDistrib)
```

There would be several assertions about ModernCultureSpatialDecayDistrib, which might say that it’s a normal distribution, its mean is 10 miles, its standard deviation is 1 mile, etc.

Those assertions about ModernCultureSpatialDecayDistrib would have to hold in some context at least as general as (EverywhereFn AutomobilesMt USA-Continental); probably they would hold in a context broader than just the USA, broader than just cars, etc.

In rare cases we would assert a different distribution for some individual assertions; e.g., every assertion about who the leader of a country is, gets a step distribution. That was true in Time (x is the leader starting and ending at crisp time-points, usually) and it's also true in GeoLocation (x is the leader of a region with precise borders, usually within those borders, s/he *is* the leader, and outside those borders s/he is *not* the leader. Period.)

There will almost certainly be different default persistence distributions for the four specMts of AutomobilesMt above.

So: to specify the spatial decay pattern of the truth of a typical assertion in a context C, we cite an instance d of SpatialPersistenceDistribution; we assert (mtSpaceDistribution C d). For an exceptional predicate p, we override this by asserting (predSpaceDistribution p d'). For an exceptional assertion P, we override this by asserting (spaceDistribution P d'').

SpatialPersistenceDistribution is a subset of PersistenceDistribution (as we discussed above, there are temporal persistence distributions as well.) Initially, each instance d of PersistenceDistribution will be spike, step, ramp, uniform, or normal (bell-curve-shaped), lopsided-normal, or the appending of two persistence distributions d1 and d2 – i.e., the value of (appendedDistributionFn d1 d2). For the parameters corresponding to how one specifies each type of distribution, please see the listing above, near the end of Sec. 3.1.7.

It is not necessary to create a new context just to spatially qualify an assertion. There are two spatially qualifying modal operators that can do that directly:

- **(holdsEverywhereIn** STHING P). This is true iff formula P is true at every point in the spatial extent of the spatial thing STHING.
- **(holdsSomewhereIn** STHING P). This is true iff formula P is true at some point in the spatial extent of spatial thing STHING.

3.4 Dimension: TypeOfPlace

The default type of place in which the action takes place, the actors are located, etc, for the assertions in this theory. For example, a context of ModernWesternSleeping might have Bedroom and HotelRoom as its TypeOfPlace. A Cooking context might have Kitchen (private or commercial) as its TypeOfPlace.

Note that this dimension is relatively independent of GeoLocation. More precisely, it depends somewhat on Culture, and Culture in turn correlates with GeoLocation. For instance, most activities in Japan are part of JapaneseCulture, which in turn has the sleeping take place by default in the living room, not in a separate bedroom.

Not all the values here are culture-related. E.g., Underwater, Underground, Arboreal, Airborne, Outdoors, a small claustrophobic place, a large agoraphobic place, a place with no (significant) atmosphere, a dark place, etc.

Since the TypeOfPlace is going to be (one or more) Collections, the genls predicate induces a partial order on the TypeOfPlace dimension values. Typically, combinations of repeated genls (e.g., Kitchen to Room) and superLocations (e.g., Kitchen to House) will be allowed. Another useful partial order on Type Of Place instances A and B will be provided by axioms of the form

(=> (isa X A) (thereExists Y (and (isa Y B) (physicalDecompositions X Y))))

and

(=> (isa X A) (thereExists Y (and (isa Y B) (physicalDecompositions Y X)))).

Putting this all together, how do we represent TypeOfPlace in Cyc? We use four functions here similar to what we did for the previous dimensions. Namely:

(#$**Relevant+EverywhereTypeFn C SPLACETYPE) = C1** means that the assertions in context C1 are precisely those assertions that hold in **C** and which happen to be both true and relevant throughout the entire spatial extent of every spatial object **x** which is an instance of SPLACETYPE (which must itself be a SpatialThingType and is usually a TemporalObjectType). C1 is akin to a "throughout an entire…" restriction; e.g., could be used to express the idea that P is true throughout all parts of every city (such as the presence of buildings might be), in every part of every restaurant (such as a no-smoking policy might be, in the context whose GeoLocation was California and whose Time was the 1990's), etc.

Here are some sample legal values for what SPLACETYPE could be – i.e., these are SpatialThingTypes and (except for the last) also SpatialObjectTypes: Room, Lawn, Jet Fighter Cockpit, Wooded Area.

(#$**Relevant+SomewhereTypeFn C SPLACETYPE) = C2** means that the assertions in context C2 are precisely those assertions that hold in **C** and which happen to be both true and relevant at least somewhere in the spatial extent of every **x** which is an instance of SPLACETYPE (which must itself be an instance of SpatialThingType, and is probably also an instance of SpatialObjectType.) Note: (genlMt (#$Relevant+SomewhereTypeFn C x) (#$Relevant+EverywheresTypeFn C x)) for all C and x. Note that C2 will be akin to a "somewhere in a…" restriction; e.g., could be used to express the idea that P is true at some place in each hotel lobby (such as the presence of a checkin desk), somewhere in each restaurant (such as a kitchen door), somewhere in each city (such as a city hall), etc.

(#$**EverywhereTypeFn C SPLACETYPE) = C3** means that the assertions in context C3 are precisely those assertions that hold in **C** and which hold true throughout the entire spatial extent of every **x** which is an instance of **SPLACETYPE** (which must itself be an instance of SpatialThingType and usually also be an instance of SpatialObjectType). Note that (for all C and x) the following will necessarily be true:

(#$genlMt (#$EverywhereTypeFn C x) (#$Relevant+EverywhereTypeFn C x)).

(#$**SomewhereTypeFn C SPLACETYPE) = C4** means that the assertions in context C4 are precisely those assertions that hold in **C** and which are true in at least *some* portion of the spatial extent of every spatial thing **x** which is an instance of **SPLACETYPE** (which must itself be an instance of SpatialThingType and usually also SpatialObjectType). Note that C4 will be a specMt of C1, C2, C3, above. For *some* value of SPLACETYPE, C4 = C.

It is not always necessary to create a context just to temporally qualify an assertion to hold during a certain type of space. There are two new spatially-qualifying modal operators (besides holdsAlwaysDuring and holdsSometimesDuring) for this purpose:

- **(holdsEverywhereInA** SOBJTYPE P). This is true iff for each spatial object SOBJ which is an instance of SOBJTYPE (which in turn must be an instance of SpatialThingType and usually also is an instance of SpatialObjectType), the formula P is true at every spatial point x in the spatial extent of SOBJ.
- **(holdsSomewhereInA** SOBJTYPE P). This is true iff for each spatial object SOBJ which is an instance of SOBJTYPE (which in turn must be an instance of SpatialThingType and usually also is an instance of SpatialObjectType), the formula P is true at some spatial point x in the spatial extent of SOBJ.

Before we go on to the next dimension, let's pause and consider the issue of **"prohibited regions" of *n*-space**. Sometimes the assertions we've made about some context-dimension-setting predicates constrain or even fully determine the possible values for other such predicates. Here are a couple ways this can happen:

♦ for all P, (holdsEverywhereInA Building P) => (holdsEverywhereInA House P) which says: if P holds throughout every building, it must hold throughout every house. The fancy way to say that is: the dimension-value-setting predicate (in this case, *holdsEverywhereInA*) transfers through the partial ordering relation (in this case, *genls* – the supersets relation) that holds between the arg1types of the first predicate. A related example of (transfersThrough holdsEverywhereInA implies): for all P and Q,
if P=>Q and(holdsEverywhereInA Building P), then (holdsEverywhereInA Building Q)

♦ If someone has taken the trouble to assert [for some P], in context C, (holdsEverywhereInA Building P), then it's a pretty good bet that C's Time value intersects the time period starting when people began having and using buildings.

3.5 Dimension: Culture (restrictions on the AgentType)

This dimension is meant to specify the default culture to which intelligent actors (typically people) belong – i.e., the actors who are mentioned in assertions in this theory. Cultures are ordered by *subCulture*. Several types of assertions transfer through *subCulture*, including most aspects of codes of conduct. The points of this dimension therefore organize into a lattice. It is not at all tree-like, though, due to the large number of somewhat-independent subculture "types" or "sub-dimensions" [23]:

political culture (Democrat, Facist, Libertarian, ...)
sexual culture (male, female)
sexual orientation culture (heterosexual, ...)
age culture (young,…, adult ,…, old)
generation culture (Renaissance, of the 1960's, generation X...)
religious culture (Atheist, Fundamentalist Jew, Moslim, Catholic, ...)
ancestral culture (of Irish descent, etc.)
geo-political culture (American, European,…)
regional culture (midwestern, east-coast, west coast, ...)
region-type culture (urban, rural, ...)
economic/work culture (idle rich, working academicians, Protestant ethicees,…)
legal culture (based on aspects of the codes of conduct/enforcement)

Note that Culture need not be a geopolitical thing. There's a Hacker culture, a Cult culture, a FishingCulture, a QuiltingCulture, a GenerationX culture, etc.

In Cyc we specify the Culture constraint via a predicate called mtAgentType. I.e., by asserting: **(mtAgentType C HCT)** where C is a context, and HCT is a HumanCultureType. Multiple mtAgentType assertions are interpreted as conjoining; i.e., the default agent in that context belongs to all those agent-types. Some of these conjoinings are worth naming, such as PoorWhiteSouthernBaptistCulture, or HardWorkingKoreanImmigrantsToUSA.

A single theory (i.e., context) may very well have two or three specific Cultures whose *dis*junction (i.e., union) is meant to be the Culture of the context. For instance, consider an AmerianSoldiersInVietnam context, in which American and Vietnamese cultures are both relevant (the assertions assume *some* mixture of the two cultures, but not their intersection.) We have a special predicate to define that union of cultures, namely **(cultureUnion X Y)**. This is associative, so 3+ cultures can be "unioned" by just having nested cultureUnion assertions. Most of these unions aren't worth reifying. This raises the following point:

Point 7: Most "settings" for a context – for any one full value of any one single dimension of context-space – are not worth naming or remembering.

More complicated specifications may have to use the more general domainAssumptions predicate; see the Let's dimension, below. E.g., to express USSoldiers&LaotianCivilians we could use (cultureUnion USSoldiers LaotianCivilians) if they both are reified, otherwise we could use domainAssumptions to say that (in that context) if x was American they were a soldier, and if Laotian then a civilian, and in the culture dimension just assert the full cultureUnion.

3.6 Dimension: Sophistication / Security

This dimension captures the notion that only certain people can/may//should/do/will know the material that is the content (and assumptions) of a context. A detailed medical context might contain information known only by physicians who specialize in that area. A company's salary data might be known only by Human Resources staff of that company. A listing of the latest stock tips might be known only to those who have paid for them. So the limitations to someone knowing those assertions may be educational, legal, etc. There may be enough distinctions between Sophistication and Security that these should eventually be split into two completely separate dimensions[22] but in Cyc they are both specified, as we discuss below, using the same quinary predicate *mtAccessConstraint*.

3.6.1 Sophistication

"Sophistication" refers to how sophisticated someone would have to be, in order

- to readily learn them(in this form) if they are taught, without lots of prerequisites
- to understand them and recognize them as true, once they clearly stated
- to already be familiar with the assertions, at least theoretically (this can mean, e.g., being able to infer the assertions quickly and automatically, from a few hints)
- to have already deeply assimilated (implications of) the content of the assertions

This is"can", not "may." It generally means understanding/recognizing/ knowing/… the predicates, the collections, etc. mentioned in assertions in the theory. It also may depend on how verbose and complicated the assertions are, the sheer number of terms, etc.

We could define a plethora of more specialized attributes (of a user/peruser) that are related to and in a way come together to define that person's sophistication: educational level, age, job, places-visited, etc. In many ways, this set of attributes subsumes all of the previously-mentioned Culture dimension predicates, with two exceptions: (1) in this case we aren't

[22] E.g., one can lift an assertion from a less secure to a more secure context, and rarely have to make any changes in it, whereas lifting from a less to a more sophisticated context often means often major changes, or even outright contradiction with the assertions in the more sophisticated context. That happens, though to a lesser degree, when one lifts a low-security assertion about, e.g., the level of cooperation between Egypt and Israel, into a high-security context where secret deals, extra factors, additional specific people/places/objects/events/…etc. are known about.

describing the agents "talked about" in the content of the context, but rather the agents who already know, or could learn, or couldn't learn, that content; and (2) some of the Culture sub-dimensions[23] are at best only weakly relevant to comprehension: gender, religion, etc.

One bit of good news is that each of these in turn – education level, places-visited, age, etc. – is describable via a simple linear scale or a well-defined set of possible settings. The other good news, as with most other dimensions, is that most of the time the person doing the asserting (or asking) is not going to manually tinker with these settings, but rather have them copied from similar/related assertions (or questions) and/or have them inferred by the system automatically (from the prior context, the current task, a model of the person who is stating/asking, and of course from the form and terms of the assertion/question itself.)

3.6.2 Security

"Security" refers to what sorts of authorization someone would have to have, in order to be permitted to see, use, modify, etc. the assertions in this theory. This might depend on their being employed by a certain company, being over some legal minimum age, having qualified/registered in some fashion with some organization, etc. The types of rights are probably similar to those generally granted by database systems and operating systems.

Conversely, someone might wish to only browse through assertions which they have the right to edit. One interesting level of access might be "used in non-top-level inference", which means you can't see the assertion, or trivially see it being used, but it can be deeply used in solving some sub-sub-…-goal of the problem.

At any rate, the Security authorizations are likely to be specified in a "rights" matrix, whose rows are the contexts and whose columns form the Security sub-dimensions[23]: who can see this, who can use this, who can edit this, who can delete this, and so on.

In Cyc, here is how this is all expressed:

- There is a term called **#$TheUser** which denotes the `current user' who is talking to Cyc – i.e., logged in and running the current Cyc image. Note that at any moment there may actually be many current users of a single Cyc image, communicating with it through various sorts of API interfaces (HTML-based browser, telnet Application Program Interface (API) connection, etc.) Each communication in effect defines a local ephemeral value for #$TheUser. The point: here is a place for Cyc to keep track of each user's relevant characteristics (including authorization and sophistication level.) Much of this will be inferred from groups to which the user belongs, jobs s/he holds, their date of birth, their country of citizenship, and so on.

- (**#$mtAccessConstraint C P** ***necessary/sufficient? type? content/assumptions?*****)** expresses the constraint P. C is a context, P is a Cyc formula, *necessary/sufficient?* is either #$Necessary or #$Sufficient, *content/assumptions?* is either #$Content or #$Assumptions, and *type?* is an **#$MtAccessClass** which means it is one of these 6 values: #$CapableOfKnowing, #$TheoreticalKnowing, #$InternalizedKnowing, #$AccessRights, #$AdditionRights, #$DeletionRights. If *necessary/sufficient?* is #$Necessary, this means that #$TheUser must satisfy the constraint P, in order for the system to conclude that they can/should/do have access of type *type?* to the content or assumptions of context C. In other words, if the user fails that test P, then they must not have that mode of access to that material. If *necessary/sufficient?* is #$Sufficient, this means that if #$TheUser satisfies the constraint P then Cyc can infer that they can/should/do have access of type *type?* to that material: they could learn it, they probably already know it abstractly, they act as though they know it, they are allowed to inspect/browse it, they are allowed to assert or unassert it. Some of these imply others; some are inconsistent with others; but mostly they are independent combinations which each make sense.

Note that the formula P probably mentions #$TheUser. For example, P might be a formula like: (#$levelOfEducation #$TheUser #$HighSchoolEducation) or – omitting #$'s – (speaks TheUser French) or (employedBy The User GlaxoWellcomeInc). It's hard to argue that we should reify those sorts of P-values, but efficient inference will probably dictate doing so sooner or later, for the more common categories of access constraints.

We regret having to use a quinary (5-argument) predicate like this – mtAccessConstraint – but otherwise there would be 24 more specialized binary predicates (taking just C and P as their arguments) for the knowledge enterer to learn. We may decide to go ahead and also define 24 specialized predicates, or not – or at least the few which capture the most-often-used values of the final 3 arguments. E.g., #$necessaryMtContentAccessRightsConstraint.

Note that if there are multiple mtAccessConstraints with the same arguments for everything except P, then the meaning (which depends on the 3rd argument) is as follows: for multiple *necessary* tests P1, P2,… (i.e., the 3rd argument is #$Necessary), they must *all* be satisfied; for multiple *sufficient* tests P1, P2,… *any one* of those tests suffices.

3.7 Dimension: Topic

Given a context C, what is it "about"? It could be *about* a person, a group of people, a field of study, a type of event, a particular event, an occupation, an academic discipline, a novel or mythos, preparing for a certain event or kind of event, etc.

Each assertion in the context is, by default, relevant to and helping to talk about and explain and interrelate the things that C is *about.* The basic hypothesis about Topics – and for that matter about each dimension of context-space – goes something like this: If you limit your attention to one such topic, then fruitful inference will occur within that clump of assertions. I.e., most of the reasoning we will want and need to do, to solve some particular problem, is likely to be restricted to one or a small number of particular topics.

Conceptually, there might be several sub-dimensions[23] of this dimension, since there is no one correct way of organizing subject matter.

- We can have separate sub-dimensions[23] for each standard subject classification hierarchy (Dewey decimal system, LC, Colon, Bliss, UDC, Yahoo, UMLS, the Art & Architecture Thesaurus, Cyc itself, HPKB chart G1, newspaper classified ads, newspaper sections, encyclopedia topics, the Outline of Cultural Materials, etc.)
- Yet another (species of) sub-dimension[23] of Topic is one organized around a hierarchy of typical goals that actors have, typical subgoals that help achieve or advance those goals, and so on.
- Yet another sub-dimension[23] of Topic could be based on academic field boundaries, such as typically are found in universities today.

Consider a context whose topic is "shipwrecks and people and fish". Each assertion in the context will probably refer to one or more of those three terms, or to elements of one or more of those three collections; moreover, the assertions will likely interrelate (an element of) one of those three terms to (an element of) another of those three terms.

Note that having (some aspect of) time/location/etc. as a *Topic* is quite different from having a setting for the Time, Location, etc. dimensions. The topic of "time" is one which will have assertions made about it in many different cultures, different eras, different levels of granularity, etc. but it is far far less pervasive than the Time dimension, for which there will generally be some setting for every context.

[23] When we talk about a *"sub-dimension"* we mean a separate dimension of context-space. Instead of having 144 dimensions, though, it would be less daunting to have 12, each of which has a dozen sub-dimensions. Also, there is much in common among the sub-dimensions of a dimension, from the high-level conceptualization all the way down to the level of code that supports those sub-dimensions.

This may very well turn out to be the most useful dimension for separating out large existing “clumps” of assertions in the Cyc KB, like the content of the current huge HumanActivitiesMt. One could write arbitrary CycL expressions to precisely denote particular topics (the education of 5-year-olds, international arms negotiations, extended family social events that have a joyous tone to them, etc.)

But reasoning about the relative relevance between two topics if described via arbitrary CycL like that could be arbitrarily complicated. To cut through that complexity, we will just let each Topic “setting” be a set of Cyc terms, no more and no less. This means that new Cyc terms will be added to demarcate topics which are worth distinguishing but which (currently) don’t exist in Cyc. The interpretation is that all the assertions in a context are “about” (one or more terms in) its topic; in Cyc, this means that each assertion which is in the content of the context should explicitly mention one or more of the collections (or instances of such), predicates, etc. which are that context’s Topic setting.

The relation “subtopic” is much more complicated than “subset.” There are many cases where a topic’s subtopics are not subsets of it; e.g., “FreshwaterAlgae” and “Dam-building” are sub-topics of “Rivers”, though they aren’t subsets, elements, etc. of #$River. And there are many cases where a set’s subsets are not subtopics of it; e.g., “Thing” ⊃ “RealNumber” ⊃ “NaturalNumber” ⊃ “EmptySet” but they are part of the disparate topics of ontology, real analysis, number theory, and set theory.

There are a relatively small number of questions which characterize what we colloquially mean by a topic being “about” *x.* E.g., if *x* is a tangible thing, when was it discovered/invented, by whom, etc.; what’s it made of, its shape, its internal structure; how and where and when is it produced/destroyed/sold/kept, and why; what is it used for/in/by/…; and so on. If *x* is an action, when and where does it take place, and why; what pre- and post-requisite conditions are there on its performance; who are the performers and other actors and their roles; what about the first/next/last time it did/will occur; and so on. Regardless of what *x* is, What are some extreme cases of this, or other special cases of it? All of those questions also can lead from a topic to one of its subtopics, which is why that relation is so much more complicated than subset.

So there is a new predicate *subTopic*, which interrelates 2 Cyc contexts, and a new predicate *propositionalInfoAbout* which interrelates a context and a term (to express the notion that that context is “about” that term.)

Here is a more complete listing of questions that define the notion of sub-topic, and hence should strongly influence the *subTopic* and *propositionalInfoAbout* values.

- If the topic is some relatively short-term practical task at hand, such as opening a door:

  inputs needed (including energy, information, and physical materials)
  tools and equipment required/involved
  physical situation, configuration, and location
  currently governing codes and standards
  methods, plans, scenarios

particular problem set/situation faced
desired outcomes, products, effects, outputs, and by-products
persons/interests to be served
persons or other agents who might assist or thwart success
access and transport to the task activity area
financial/commercial consequences of task
any likely or feasible thing that would alter the outcome

- If the topic is some relatively long-term task at hand, then include these as well:
development of methodology over time
commercial practice in the field
spectrum of participants and players
changes in governing codes and standards
political (economic, military, etc.) consequences
building the skills needed; training, education

- If the topic is a task or even a tangible thing in the world, then include these:
the origins of the task/thing
how it was dealt with historically
the fundamental underlying principles (and prerequisite knowledge)
classifications of sub-varieties of task type or thing type
literary or cultural role of the task type or thing type
social role and effects of task type or thing type
fine physical or information structure of the things involved
implications for philosophy, science, history
representation of task or thing in formal systems
are the ultimate goals of the task/thing really worth pursuing?
analogous situations in other fields; parallels

So how exactly is this represented in Cyc? We use these two binary predicates:

(#$propositionalInfoAbout C TERM) which means that the context C is `about' the reifiable term TERM, i.e. it is a #$Microtheory with assertions concerning TERM. For example, a context which is the propositional content of a portrait of George Washington might have #$GeorgeWashington as the only TERM of this satisfying this predicate (i.e., the only value of TERM for which (#$propositionalInfoAbout C TERM) holds.)

(#$subTopic C1 C2) which means that context C2 is a subtopic of context C1. Note that propositionalInfoAbout *often*[24] transfers through subTopic.

[24] But not always: sometimes the subtopic C2 (e.g., teleneuromicrosurgery) is so much more specialized that the general topic C1 (e.g., medicine) that the general topic would not be said to be "about" some of the very specialized terms (e.g., microprobe-camera-latency-compensator) that C2 is said to be "about."

3.8 Dimension: Granularity

There are again several sub-dimensions[23] here; the two-page chart near the end of Miller's *Living Systems* book gives a good summary of some of these, as he goes from molecular to cellular to organ to organism to society level. There is a hierarchy of integrative levels, in which things at one level combine to form things at the next higher level, such as cells coming together to form an organ, organs coming together to form an organism, etc. The reason for keeping these levels separate, though – for having this dimension at all – is that assertions at one level do *not* simply translate up or down to the next level. Each level is almost its own world; atomic collisions don't directly come up often in solving problems in British Immigration Law, and vice versa. Moving from level to level even the name of the academic field changes – e.g., from Sociology to Psychology to Physiology to Molecular Biology to Chemistry to Physics. Here are some of the types or sub-dimensions involved:

- **What sized** objects, events, object-parts, time-periods, subevents, organizations, etc. are relevant to this context? In particular, more fine-grained ones are not mentioned nor needed for reasoning within this context. Also, it's usually the case that much coarser-grained ones are also not mentioned or needed within this context. For instance, when reasoning about a baseball's trajectory, neither particular individual atoms nor particular galaxies (even the Milky Way) are likley to be taken into account.

- **What specificity** of classes (collections), relationships, measurements, etc. are relevant to this context? In particular, more fine-grained ones are not mentioned nor needed for reasoning within this context. Also, it's usually the case that much larger classes, relationships, etc. are not mentioned or needed within this context.

- Those first two sub-dimensions[23] can be thought of as asking the question "What distinctions matter?" A different sort of sub-dimension arises from asking the question "**What phenomena** matter?" For example, does friction matter, do relativistic effects matter, does weather matter, etc.?

- The final sub-dimension[23] here has to do with representation systems and abstract classes. It can be thought of as a sort of "**Level of formal entity**" sub-dimension of Granularity.[25] The context may assume that the focal actors are at a certain level such

[25] This is reminiscent of the ActorSlotSense notion. The idea is that a context uses each actor slot in a particular way – either ActualPerformerSense, in which case the axioms are about the actual entities acting in the action, or in some other way. For example, the US would win a gold medal in RepresentedAgentSense if one of its citizens won the gold medal, even though individuals, not countries, win gold medals.

> as people, corporations, countries; a book versus an edition of a book versus the content of the book; a proposition or an entire theory or a representation of one of those two in some language; particular versus abstract game pieces; etc. Several real objects may embody one formal object, such as several copies of *Moby Dick* being "the same book" at some level of granularity, and different books at another level of granularity. Or a certain chess game may be nothing more than an abstract series of representations of moves at one level of granularity, and, at another level of granularity, it might be an actual event with specific physical players and pieces and movements.

Let's consider one simple example of Granularity, taken from the current Cyc KB. The assertion that the mainland USA (#$ContiguousUnitedStates) is #$spatiallyContinuous is asserted to be true in the #$UnitedStatesGeographyMt context. But what about all the small islands, sandbars, etc. just off the coast – such as Catalina Island off the coast of southern California? They are considered part of the contiguous 48 US States – they're part of the USA and not part of Alaska or Hawaii – but on the other hand they are clearly not spatially continuous with the rest of the mainland. The solution: the spatial granularity of the #$UnitedStatesGeographyMt context is larger (coarser) than any context in which those sandbars, islands, etc. are known about. In that context, they don't even exist. The #$UnitedStatesGeographyMt is talking about geographic regions that are state-sized (or larger), not small-island-sized (or smaller). That could be stated as a predicated domain-assumption of that context, but having the Granularity dimension enables us to state it succinctly as a Granularity value – i.e., that is one of the *n* "coordinates" of that context that situate it in a region of the *n*-dimensional context space.

The Granularity dimension is where we will integrate "order of magnitude" reasoning into Cyc. E.g., so Cyc won't worry about quantum wave phenomena when reasoning about the likely outcomes of a monster truck pull.

Here is how this is represented in Cyc: we use these 6 new binary predicates:

(#$mtLinearDistanceValue C INTERVAL) means that when reasoning in (or with) the context C, spatial objects for which the longest axis (longest linear dimension) falls within the range specified by the #$Distance INTERVAL should be attended to, or taken into account, and those which are much larger or smaller shouldn't. For example, (#$mtLinearDistanceValue #$SleepMt (#$Meter .001 10)) means that when reasoning about sleeping, it's unusual for objects < 1mm or > 10m in length to be relevant.

(#$mtVolumeValue C INTERVAL) means that when reasoning in (or with) the context C, spatial objects with a volume that falls within the range specified by INTERVAL should be attended to, or taken into account, but not those with larger or smaller volumes. For example, (#$mtVolumeValue C (#$CubicMeter 1 100)) means that when reasoning proceeds using the knowledge in C, only objects having a volume that measures greater than or equal to 1 cubic meter, and less than or equal to 100 cubic meters, should be considered relevant. Note: these 3 distance/area/volume predicates are weakly interdependent on each other; if 1 or 2 are unspecified, they can be guessed from the 3rd.

(#$mtAreaValue C INTERVAL) means that when reasoning in (or with) the context C, only objects with an area that falls within the range specified by INTERVAL need be attended to, or taken into account. For example, to indicate this value for one of the Geography contexts, namely the one that has maps, distances, etc. from the 1995 CIA World Fact Book, saying it dealt with areas between 10 and 100k square miles, we'd write: (#$mtAreaValue #$CIAWorldFactBook1995GeographyMt (#$SquareMile 10 100000)).

(#$mtSubsetAbstractionLevel CTXT COL) means that when reasoning in (or with) the context CTXT, (instances of) classes of objects at about the level of abstraction indicated by the collection COL should be attended to, or taken into account. If you consider the knowledge base as a lattice of abstract concepts arranged in a hierarchical structure via the subset/superset relation (#$genls), then the further away (up or down) in the hierarchy you go from COL, the less likely it is that (instances of) those classes should be used in inferencing in context CTXT. This is not symmetric, in that the fall-off of relevance is likely to be much more pronounced and absolute in the "upward" (more general) direction than the "downward" (more specific) direction. Here is a related new predicate:

(#$mtPartonomicAbstractionLevel CTXT COL) is similar to the last predicate. It means that when reasoning in (or with) the context CTXT, (instances of) classes of objects at about the level of partonomic abstraction indicated by the Collection COL should be attended to, or taken into account. If you consider the knowledge base as a lattice of abstract concepts arranged in a hierarchical structure via the subparts/superparts relation (#$parts), then the further 'up' (toward bigger assemblages) or 'down' (toward smaller and smaller subparts) the hierarchy you go from COL, the less likely it is that (instances of) these classes should be used in inferencing in context CTXT. For example: the assertions in #$AutoRepairCustomerMt might apply to wheel-and-axle assemblies, but not to universal bearings (which are part of a wheel-and-axle assembly), and not to cars or the overall urban transportation system for a city (which have wheel-and-axle assemblies as parts or parts-of-parts). This predicate, or the one above it, would be the ones to use to specify that, e.g., in one context a book such as Moby Dick should be considered to be the text of that novel, not individual copies, not the way it's typeset into a printing/edition, etc.

(#$mtRelevantPhenomena C TERM) means that when reasoning in (or with) the context C, the phenomenon indicated by the reifiable term TERM (or instances of TERM, if TERM is a #$Collection) should be attended to, or taken into account.

3.9 Dimension: Modality / Disposition / Epistemology

This is another situation where there are a few very closely related notions which *for the moment* we are clumping together as one dimension of context-space.

3.9.1 Modality

"Modality" refers to the (default) modal status of assertions locally made in this context: Are they beliefs, agreements, expectations, memories, etc., and, if so, whose? "Whose" means: belonging to, and represented in, some mind(s), some text, etc.

Naturally these may be nested, so one can have a context whose assertions are, for example, what Fred believes that Joe remembers that Sam and Fred agreed to. Weekly summaries of TV soap operas are even more deeply nested than that example.

Unlike most of the other dimensions, this one is extremely tricky when it comes to lifting; i.e., in deciding what assertions can be imported from one modal context to another, and how exactly they need to be modified when imported.[26] E.g., does Bill Clinton believe that Chelsea Clinton remembers the year that Ulysses S. Grant became president? In the Star Trek context, did Captain Kirk remember that Bill Clinton was elected President of the US? Did Dracula believe that Ulysses S. Grant commanded the Union army in the American Civil War? How old does someone growing up in Austin today have to be before you'd expect them to believe that Washington D.C. is the capital of the U.S.A.? Before you'd expect them to *know* it?

One of the most common undesirable phenomena in intelligence analysis, interpersonal relations, etc., is called "mirror imaging"; it refers to one agent ascribing their own beliefs, values, knowledge, desires, etc. to a second agent about whom they are reasoning, or with whom they are conversing/interacting. This often manifests as something which, in hindsight, we label as naivete; for example, Oliver North believed that the Iranis would keep their word, because *he* would have done so.

Why is this a separate, independent dimension from all the others? Consider the following phrase: "…the beliefs of most US Army officers stationed in Berlin in the 1960's…" The pieces of that phrase represent distinct dimensions, and one of them is "beliefs."

[26] This trickiness involving Lifting also means that there is some trickiness in deciding whether and how the modality-nesting is sibling-disjoint or not – i.e., whether and where it forms a tree, not just a poset (partially-ordered set – please see footnote 28 for further information about posets.)

To decide who is the believer, you might very well need the full expressive power of the Time, Culture, GeoLocation, Sophistication,…. dimensions. And this need resurfaces every time there is a recursive nesting of modality.

There are two pieces of good news, though:

(1) Most of Cyc's knowledge is modally flat – Cyc should believe assertion P as True, not as someone's beliefs. So, as with each of us humans, what Cyc's modal contexts bottom out in is not "what is true" but rather "what Cyc believes to be true."

(2) Most modal assertions, let alone the nested-modals assertions, are likely to involve mostly constant terms rather than quantified variables.

3.9.2 Disposition

"Disposition" is closely related to modality. Suppose an agent believes assertion P to be true; that's modality. Now suppose they are *disposed* to think that's a good thing, or a bad thing; that's disposition. The disposition of assertion P refers to the attitude of agents (who believe, expect, etc. P) toward P being true. Thus, coupled with modality, disposition includes dreads, desires, goals, annoyances, etc.

The disposition of a set of assertions becomes important when reasoning about (predicting, understanding) some agent's behavior/comprehension/mood. For example, to argue self-defense in a murder trial, the defense might be required to demonstrate that the defendant both believed "the victim was going to try to kill me" to be true and, moreover, that the defendant had a certain particular disposition toward it, namely an overwhelming dread – e.g., hadn't just a moment before been suicidal and begged/hired the victim to kill them.

3.9.3 Epistemology

"Epistemology" refers to the (default) epistemological status of assertions locally made in this context: Are they intended/taken to be true, intentional lies/works of fiction/humor/ erotica, rumors, etc. Or, alternatively: who believes these things to be true? Who believes them to be fictional? To be a joke? And so on.

This is closely related to Modality, but not quite the same thing. Modality deals with whether/why things are true, Epistemology with who believes them to be true (and why).

3.9.4 Putting these three components together

The three components of this dimension are, together, responsible for indicating various ways and means of awareness that an agent has of a piece of information: a conscious thought, a subconscious one; an internal sensation or biological urge; something they're aware of in their physical surroundings through one or more of their senses (such as peripheral vision). Some assertions may form only through the passage of time (e.g., "this movie is long.") and some may be instantaneous; some may be derived by the agent themselves on their own, and some may be communicated in messages from other agents.

Unlike the other, earlier dimensions, this is one where all the sub-dimensions[23] might be better just clumped together. Putting it all together, then, we have as a setting in this dimension just one single pair of attitude/who. Where "attitude" is one of BelievedBy, KnownBy, DesiredBy, FearedBy, etc. and "who" is an intelligent entity or collection of intelligent entities. Nested modality etc. is represented by nesting the contexts in this dimension. So: Here are the Cyc collections and predicates involved in this dimension:

**#$PropositionalAttitudeContext –** The collection of all contexts (all instances of #$Microtheory) which contain assertions expressing the propositional attitudes (beliefs, desires, intentions) of some agent or group of agents. Some subsets (that is, subtypes of propositional attitude contexts) are: #$GoalsContext, #$DesiresContext, #$BeliefsContext.

#$BeliefsContext – The collection of all contexts which contain assertions expressing the beliefs of some agent or group of agents. A subset of PropositionalAttitudeContext.

(#$believesMt AGT C) means that the Agent AGT's beliefs include all of the propositions in context C. This predicate is a #$PropositionalAttitudeSlot and #$MicrotheoryPredicate. In general C will be a BeliefsContext. Our current hypothesis is that we can express AGT's disbelief or scepticism in P by asserting the *negation* of P into a BeliefsContext C and then asserting (believesMt AGT C). I.e., AGT believes that not-P is true.

#$DesiresContext – The collection of all contexts which contain assertions expressing the desires of some agent or group of agents. This is a subset not only of PropositionalAttitudeContext but also of CounterfactualContext, since if the desire were true it would then not normally[27] still be what we call a desire.

(#$desiresMt AGT C) means that the Agent AGT's desires include all of the propositions in the context C. I.e., AGT desires (by default) that each assertion in C come true. This predicate is a #$PropositionalAttitudeSlot and #$MicrotheoryPredicate. In general C will be a DesiresContext. Our current hypothesis is that we can express AGT's fear or hate by asserting its *negation* in a DesiresContext C and then asserting (desiresMt AGT C).

#$GoalsContext – The collection of all contexts which contain assertions expressing the goals of some agent or group of agents. It is a MicrotheoryType, and a subset of

[27] One exception to this is: if the person doesn't know that the desired situation P has come to pass; then they of course may continue holding that situation as one of their desires. I.e., they *believe* it to be counterfactual.

PropositionalAttitudeContext. Unlike *DesiresContext*s, a GoalsContext can contain a mixture of assertions that are already true and not already true. This reflects the two major types of goals: those to sustain something that is already true, and those to attain something desirable but currently not true. Both types of goals are states of affairs that an agent is actively working toward, planning for, or at least genuinely aspiring to. Each not-yet-true goal is a desire, but not always vice versa: There are many things an agent might desire (e.g., to be Indiana Jones) that are not actual goals of that agent, because they're truly impossible (e.g., involving fictional characters, involving changing the past) or at least there's nothing the agent can do at the present time about them.

(#$goalsMt AGT C) means that the Agent AGT's goals include all of the propositions in context C. This is both a #$PropositionalAttitudeSlot and a #$MicrotheoryPredicate. C will in general be a GoalsContext. Our current hypothesis is that we can express each state of affairs P that AGT is actively working to avoid/deter/undo/… by negating P and then asserting that negation – (not P) – into a GoalsContext. If such a (not P) happens to currently hold, then the "goal" is to keep it from coming true, to keep (not P) remaining true; if the (not P) is currently false – i.e. P currently holds – then the agent's "goal" is to make P stop holding, to make (not P) become true.

3.10 Dimension: Argument-Preference

What is the (default) set of heuristics used in this context, to resolve pro/con argument clashes? What is the (default) ordering/weight placed on each of *them*? (E.g., suppose you have a short novice argument and a long expert argument; do you prefer "prefer short arguments to long ones" to "prefer expert arguments to novice ones"?)

One extreme sort of argument preference heuristic is to discount, or refuse to even consider, nonconstructive arguments.

This will be a poset[28] (partially ordered set) of posets of heuristics, the latter posets ordered by preference and the former ordered by refinement.

Note that it is possible that the mechanisms used to resolve argument clashes empirically turn out to be less related to the knowledge itself than to the way the knowledge will be used. This might vary from application to application, so instead of having Argument Preference be a separate dimension, it possibly should be an application parameter or "tag".

Putting it all together, then, a setting for this dimension in Cyc is an instance of a new collection called ArgumentPreferenceSet, which instance in turn will be a set of (1) assertions for preferring one argument over another, and (2) assertions for preferring one preference (of type (1)) over another; i.e., assertions of the the two following forms:

- P => (prefer A1 A2) where P is some condition involving arguments A1 and A1. E.g., (shorter A1 A2) => (prefer A1 A2)

- P1 => (overrides PREF1 PREF2) where PREF1 and PREF2 are each statements of that previous form (such as "(implies (shorter A1 A2) (prefer A1 A2))").

[28] "Poset" stands for "partially ordered set." An *ordering* on a set is a function that tells whether one element x should come before another element y of the set. Some orderings are total – e.g., the "<" ordering on the real numbers between 0 and 1. Some orderings are just partial: there are some "don't-know" or "don't-care" answers for certain elements x and y – that is, the ordering function doesn't tell you which should come first. For instance, consider the "⊂" ordering on a set of sets, or Cyc's *genls* predicate interrelating Cyc Collections. Sometimes one is a subset of the other, and "comes first", and sometimes vice versa, and "comes second", but sometimes neither set is a subset of the other, so the partial ordering says "don't-care". Partial orderings should be transitive (if a is "before" b, and b is "before" c, then a should be "before" c) and non-commutative (if a is "before" b, then b is *not* "before" a). Posets are slightly more general than "*Lattices*". In a lattice, if a is "before" b and c, then either b will be "before" c or else c will be "before" b; in a poset, that requirement isn't present. All lattices are posets. For an example of a poset which is not a lattice, consider a group of people – say all the people who've ever lived – partially ordered by the relation "descended from". Some person "a" might be descended from persons b and c, and yet neither b nor c is descended from the other; for instance, b and c might be two unrelated people who marry and have person "a" as their child.

3.11 Dimension: Justification

Which justification, kinds of justifications, sets of justifications, or sets of kinds of justifications, justify all/most/some of the assertions in this context? Are they definitional, causal, statistical, appeals to intuition, by faith, by assumption, etc.

Another way of looking at this is: sometimes it's useful to grab a set of similarly-justified assertions (from some large outer contexts) and clump them together into one new, smaller context.

Yet another way of looking at this is: if you place, or infer placement of, an assertion P into one of these contexts, then you can automatically infer the kind of justification that P will have; this may be all you need for certain purposes, it may be a help to the system in finding the full justification, etc.

Yet another reason for this to exist might be because a large number of assertions share a sub-justification (one large chain, one "lemma" if you will), and this saves space, makes it faster to do truth maintenance if something causes that lemma to be retracted, etc.

In Cyc we represent this dimension by the two predicates:

(#$mtSource C SOURCE) means that the information in context C originated with, or is authenticated by, SOURCE – a temporal thing which must either be an instance of #$Agent or of #$InformationBearingThing.

(#$mtSourceType C STYPE) means that the information in context C originated with, or is authenticated by, an information source of type STYPE – an #$ExistingObjectType which is a spec – a subset – of the collection #$TemporalThing (usually is moreover a spec of #$Agent or a spec of #$InformationBearingThing.)

3.12 Dimension: Let's (and misc. domain assumptions)

Often in print (or conversation) we preface a set of remarks by stating some assumptions:

a. "Suppose someone – let's call them Fred – trades in their car every year." Or:
b. "Let Z be a right triangle, whose hypotenuse is of length h." Or:
c. "Say the USA wins the gold medal in Ice-Dancing at the 2000 Olympics." Or:
d. "Let X be a person and suppose Y is that person's height."

This makes the ensuing remarks much terser than if we had to keep repeating those local assumptions over and over again. This dimension does that sort of thing for Cyc.

So one thing each Cyc context can have is a list of such "Let…" statements. Sometimes a variable is bound, as in a and b and d, above; sometimes none is bound, as in c. Sometimes the value of one bound variable depends on the value of an earlier binding, as in b an d.

Formally, the Let's of a context are just a set of assertions that are assumed to hold true in that context. The reason for pulling them out separately (rather than just asserting them as part of the content of the context) is that they are conceptually part of the "if", the setup of conditions that in effect *defines* the context. They are what we used to call "domain assumptions." In fact, to state one of these in Cyc we just say:

(domainAssumptions C P) – This means that P holds in C, by assumption. Presumably P does not neatly fit into any of the previous *n-1* dimensions, such as Time or GeoLocation, or else those more specialized predicates would be employed to state P more tersely.

There is one very special thing we can do in a domainAssumption that we generally can't do anywhere else in Cyc except within a single rule: we can introduce "skimpy" new terms that will only be local to that context (and its specMts). They're "skimpy" in that they don't need to be fully fleshed out, or given mnemonic names, or supplied with clear comments, etc. They *could* get all that, but likely won't.

The new terms are introduced just by using them – just as though this were all going on within a single assertion – as we shall now see.

We could assert (#$domainAssumptions #$USAMt (#$isa **#$X** #$Person)) which means: Let X be a person, throughout context #$USAMt..[29] Thus, any assertion in context #$USAMt could refer to #$X just like any constant, such as #$BillClinton. Within #$USAMt, e.g., it should be the case that Cyc could conclude that #$X has a mother.

[29] If instead we had asserted (#$domainAssumptions #$USAMt (#$isa **?X** #$Person)) that would mean that, within the #$USAMt context, everything is a person, since unbound variables (such as **?X**) are taken to be universally quantified. That would probably be a very bad thing to assert, as most things are *not* people.

(Since that draws only on knowledge about people in general, located in more general contexts than – i.e., genlMts of – #$USAMt.)

One could assert in the #$USAMt context (#$citizenOf #$X #$USA) and – if there were no further restrictions on #$X (e.g., in other domainAssumptions of that context) – that would mean that in that context every person mentioned in any way is, by default, a US citizen. Which is a good default for that context, though it would be a bad one for most contexts. Think of the domainAssumptions as the If- parts (antecedents) of axioms, and the various content assertions of a context as the Then- parts (consequents). All those domain assumptions are conjoined onto the antecedent (if any) of each content assertion of the context.

If instead we asserted (#$domainAssumptions #$USAMt (#$citizenOf #$X #$USA)) the effect would be quite different. That would just be further specifying specific assumed details about $X. Thus, within that context, $X would be a US citizen but that assertion would not imply that the average person was a US citizen (contrasted with the previous case, where we asserted that as a part of the content of #$USAMt, not as a further domain assumption.)

Outside C, however, #$X would be unknown. If #$X already exists in a context outer to C (that is, one of C's genlMt's) then don't try to "bind" #$X to some new thing within C, though it would be okay to add some new additional assumption about #$X that further restricts it.

We mentioned above that the domain assumptions were a list rather than a set. Actually it should be rare that order matters much, so we will use an explicit partial-ordering predicate when it does matter. This is a ternary predicate, modal in both 2nd and 3rd arguments:

(earlierDomainAssumption C A1 A2) meaning that assumption A1 precedes A2, for context C. This is likely to have no effect (except possibly slowing down inference a bit) unless both A1 and A2 are domain assumptions of C, but that is not enforced.

So to implement the Let (d) above – "Let X be a person and suppose Y is their height" – for context C, we would assert:

```
(#$domainAssumptions C (#$isa #$X #$Person))      and
(#$domainAssumptions C (#$height #$X #$Y))
```

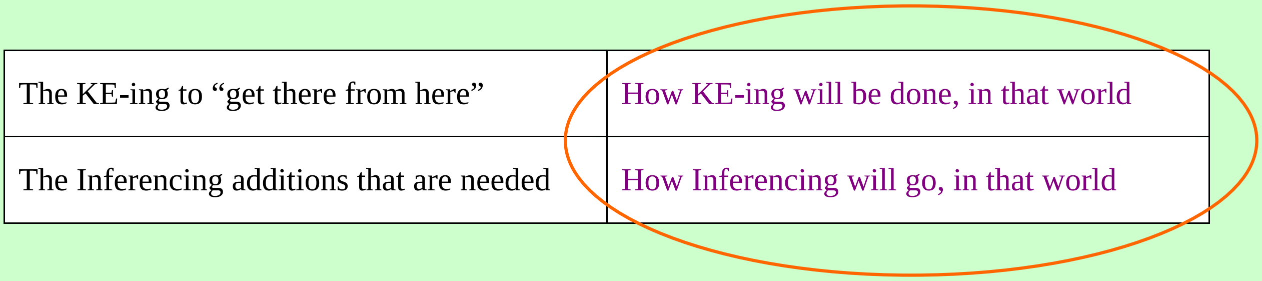
We could also assert, if we wanted to:

```
(#$earlierDomainAssumption
        (#$domainAssumptions C (#$isa #$X #$Person))
        (#$domainAssumptions C (#$height #$X #$Y)))
```

but frankly, why bother? It doesn't really matter which of those assertions precedes the other. This is what we meant above, when we said that it should be rare that we care about the order in which the assertions get made.

4 Living in this 12-Dimensional World

This section focuses on the righthand column of the following 2x2 table; namely, what is life like, in that new 12-dimensional context world, *vis* KE-ing and Inferencing.[30]



The KE-ing to "get there from here"	How KE-ing will be done, in that world
The Inferencing additions that are needed	How Inferencing will go, in that world

Figure 4a. Getting to that 12-dimensional world, and successfully living in it

4.1 How KE-ing Will be Done, in That World

Think of Knowledge Entering (KE-ing) as a two-step process:

1. Specify a context – that is, indicate the region of 12-space you're talking about.
2. Tell or ask Cyc something "in" that context.

All knowledge entering – whether it is formulating a large new body of axiomatized assertions, or editing a gap or mistake in the Knowledge Base (KB), or browsing around the KB, or asking a specific business-level question of the KB – can be broken down into (one or more) applications of that two-step process.

So the question of how KE-ing is done, in the 12-dimensional context space world, devolves into these two questions:

- How is a context specified?
- How are things asked/told to Cyc once the context is specified?

[30] Cycorp internal documents discuss the two left-hand boxes: the process of converting from the old 1-dimensional implementation of contexts, that Cyc had for the past decade, to this new 12-dimensional one, including the changes in inference, representation, additional knowledge that had to get added to the KB, etc.

4.1.1 Specifying a Context

This is not much different from the current #$ist, except that the context is not necessarily a reified term in Cyc's vocabulary – rather, you can think of it as akin to a large nonatomic term (NAT). There are actually several different ways to specify the context; here are 5 of them:

Case 1: In most cases, you will already know some other assertion that's true in the same – or at least somewhat similar – context. There should be tools[31] for easily locating such an assertion and, having located it, for examining and adopting any or all of its 12-space "settings."[32]

Point 8:** **One conceptual operation that needs to be supported efficiently is: given an assertion, return the contexts (the sets of 12 settings) in which it holds.

Case 2: In a less common, more complicated case, you may look at two (or more) assertions, find their contexts, and "adopt" some settings from each of those contexts. Sometimes, for dimension *m*, the right thing to do is to conjoin, or disjoin, or intersect, or union, the values from those assertions' contexts' dimension *m*.

Case 3: One or more of those settings might then need to be adjusted a bit. For some dimensions, there will be a slider; for others, a set of pushbuttons (xor or not), for others a matrix, etc. For some dimensions, there will be some easy way (possibly a pushbutton) to specify meta-information such as Some *versus* All.

Point 9: One conceptual operation that needs to be supported efficiently/naturally is: given a particular context, modify one (or more) of those 12 dimension settings a little.

Case 4: The context is sufficiently unique that it's just specified from scratch. This is really just a specialization of Case 3 such that it's likely that all 12 settings will need adjustment,

[31] These tools include ones similar to the ones that Cycorp currently uses for browsing the Cyc KB.

[32] In case the dimension has several sub-dimensions, this entire Case 1/2/3/4/5/… procedure is recursively followed for them. Let's suppose that Topic has 4 sub-dimensions; then it is really a 4-dimensional matrix more than a list. In the first sub-dimension, Case 1 might apply for assertion P; in the second sub-dimension, Case 3 might apply; and so on.

and those adjustments may be more than just a little tweaking. Note that specifying a context from scratch in this fashion entails 12 separate, independent “specifyings.”[32]

Actually, it’s not quite that bad:

- For one thing, certain choices/settings in some dimensions may drastically constrain the possible/plausible settings in another dimension. In some cases, the only possible value in one dimension might be v1 (one specific value); in some cases, there might be no meaningful value whatsoever in that dimension, given the values in other dimensions. We might or might not want an explicit NULL value allowed.

- For another thing, even just the *appearance* of various terms in P can provide a good starting guess for the 12 settings specifying the context in which P should hold. Particularly good constraint information is provided by terms representing proper nouns (e.g., Gerald Ford), types of devices (e.g., microwave ovens) and other artifacts (e.g., bellbottom jeans), events, and so on.

Case 5: The context has been reified – given a name; i.e., there is a Cyc term denoting it, such as #$USAMt. Of course this means that in Cases 1-4, once a context has been specified, the KE-er should be able to give a specific name to the context, for later reference, if s/he wants to. Also, in cases 1-4, anytime some other/similar/… context was being referred to, any or all of those contexts might have been found by name in this Case 5 fashion. (E.g., in Case 2, instead of pointing to an assertion and thereby indirectly its context, and thereby its value for dimension *m*, one could instead point directly to a reified context by name, and thereby its value for dimension *m.*) Similar reifications may exist for particular values on particular dimensions, such as #$TheYear1998 for a Time dimension value.

One complicating factor is that the *order* in which dimensions are specified can sometimes matter, especially if some dimensions are *Some…* and some are *All…* -- just as when one has several quantifiers, the order of nesting matters if some are existential and some are universal. And the same dimension can occur more than once, at different places in this layering. So there needs to be some way to select one dimension after another, in order – sometimes returning to a dimension – and specify a value along that dimension. We will return to this issue below, when we discuss a possible user interface design.

4.1.2 Telling/Asking/Browsing “in” a Context

Once the context C has been specified, using the one or more of the 5 methods above (case 1-5 in Section 4.1.1), the person entering the knowledge then talks to Cyc “in” that context.

An assert of P “in” that context is akin to (ist C P). An ask of P (with some unbound variables) “in” that context means finding variable bindings that satisfy P in that context; in other words, if P’ is P with any such set of variable bindings substituted in, then (ist C P’). Browsing is done “in” that context. The terms,[33] assertions, etc. one finds are all and only those which are “in”[34] P.

Much of the KE-ing eveyone is likely to do in the coming few years has the flavor of adding a large number of assertions in a single new context, then moving on to a new – and possibly more specialized – context, and restarting that batch KE-ing process. Once the context is specified, the actual KE-ing process needn’t be that different from what it is today. The main discernable difference should be that sometimes some of the antecedent’s clauses can be replaced by a change in one (or more) of the 12 dimensions.

The interface for this could involve some sort of slider/dial/map… for the value for that dimension; then some of the KE-ing is done by writing the body of the assertion P and then adjusting those sliders/dials/… to maximize (or more generally to indicate) the region over which P is true. Somehow the *order* of the choosings has to be specified, too; the easiest way might be to superimpose a number in a corner of each slider/dial/… to indicate that ordering. In rare cases, a slider/dial/… would have to be duplicated because it would appear (usually with a different value) in more than one place in that layering of dimension values.

Browsing could make good use of the very same sort of interface. Namely, one can “browse” along any dimension, as well as within any context (fixed values for the dimensions). The former is done by moving the slider/dial/…, and noting the change in the shape/content of the resulting browsing tree/graph/table/… One could also reorder the numbering (sequence of layering) of the slider/dial/… settings, and note the changes.

There must also be a path to avoiding the sort of “backsliding” (to the all-in-BaseKB laziness) that has proven so harmful for both human users (looking through a massive

[33] A term is “in” C iff it appears inside some assertion P which is “in” C.

[34] If you’re “in” C then you’re also “in” all its more general contexts. E.g., being “in” a 1986 context also means you’re “in” a 1980s context, and “in” a 20thCentury context, etc.

number of assertions) and computers (inefficiency in inferencing, etc.) Each dimension will, if nothing else, present to the knowledge enterer a set of questions that can be asked of any proposed assertion: What's the smallest physical distance worth distinguishing, from this assertion's point of view? What is the applicable time period? And so forth.

4.2 How Inferencing Will be Done, in That World

This section deals with the lower right box in our 2x2 table, Figure 4a: how and why inferencing in that new world will be no worse than it is today, and usually will be a whole lot better.

Increasing constraints, dimensionwise, should continually reduce the set of assertions to worry about – sort of like conjoining multiple filters. Alternatively, think of contexts as determining not an absolute relevant/irrelevant black&white border, but rather as determining a partial ordering of relevance, so the most likely-to-be-relevant assertions can be found and considered before the less likely ones.

Obviously, inference must now take account of the dimensions of context-space, the meta-level information such as Some/All, and inter-context relations more specialized than just *genlMt*. Each dimension will have one or more special forms of generalization/ containment/… to indicate that one context is talking about, e.g., a subsuming geographic region, a subsuming time interval, a subsuming topic, a coarser granularity, or whatever. We didn't mention all of these by name, in section 3; e.g., the Granularity dimension will have the predicate *finer*. That is, (finer C1 C2) means context C1 has a finer granularity than context C2.

Some dimensions will have more than one special form of context-comparison predicates. E.g., *finer* should have 6 more specialized kin, one such predicate corresponding to each of mtRelevantPhenomena, mtSubsetAbstractionLevel, mtPartonomicAbstractionLevel, mtAreaValue, mtLinearDistanceValue, and mtVolumeValue. E.g., one of those would be **finerMtLinearDistanceValue**. This doesn't mean strictly-finer, but rather "no coarser than." The definition of *finer*, then, is the conjunction of all 6 of those more specialized *finer…* predicates.

A new sort of question that should be supported is, in effect:

- in which contexts is P true?
- in which contexts is P false? (conversely: in which could P *possibly* hold?)

Conceptually, this is probably nothing more than the current ASK functionality, with some aspects of context-space unspecified or at least underspecified. Thus, some of the variables being bound in the ASK might be contexts, or aspects of certain dimensions of contexts. Pragmatically, though, this might be costly to compute efficiently. There is even some trickiness in simply expressing ranges over which P holds, if it also holds for each subpart of that range, since in effect P holds for an infinite set of contexts; what we want, clearly, is the maximal range over which it holds.

A related question which should be supported is: in which contexts is term Z even present? For instance, you come outside one morning (blithely in some daily routine context) and find on your doorstep a baby, or a gun, or a jackhammer. Now suddenly you are "in" a larger context than you were a moment before, and you will soon make inferences that shift/enlarge your context even more (to involvement with the police, etc.)

Maybe a better way to think of this, instead of being "in" an ever-growing single context, is to think of it as a set of contexts that are "activated." A small number of lifted assertions then connects the two contexts, but most of the reasoning that goes on is still in just one narrow context. E.g., on the television program *Law&Order*, the first half of the show is usually a typical police whodunit, and the second half is usually a typical courtroom battle show, but there are a few indexical terms (such as the time of year, the defendant, the victim, the murder weapon, the locale) and assertions (such as the description of the crime, and of evidence found) that are shared by the two halves.

5 Conclusion

Contexts have the power to improve both knowledge entering and inference in Cyc.

- As regards, inference, contexts should help speed things up because

  (a) We will be able to generally limit search to one relevant context (or a very small number of relevant contexts) plus a very small number of imported assertions from other contexts. Even in cases where that is too simplistic, the hierarchies of contexts can help define a relevancy-ordering on the content of the knowledge base.
  (b) Inconsistent – not just irrelevant – information is kept isolated in other contexts, thereby reducing the frequency and severity of sophisticated reasoning needing to be done (for example, weighing pro and con arguments for a proposition.)
  (c) When reasoning in a particular context, all the shared assumptions of the assertions can be “factored out” and ignored, thereby making each assertion much terser and, probably, making the search faster.[35]

Contexts don’t do the whole job, of course, in speeding up inference. There still need to be other mechanisms, such as the entire EL/HL (Epistemological Level / Heuristic Level) split, the set of particular commonly-needed HL modules, etc.; and such as inference strategy rules like “prefer to examine assertions that share (more) terms with (many) already-active assertions used so far in this partial path to a solution.”

- As regards knowledge entering, contexts should help speed things up because

  (a) In a context, thanks to factored-out assumptions, assertions can be much simpler (terser, shorter, more certain, have fewer exceptions, etc.)
  (b) Even if the assumptions of a context are only much later – or even never – fully known and specified, it may be relatively easy and natural to tell that assertion P belongs in the same context as assertion Q.
  (c) The *n* dimensions we choose for context-space provide a set of ways to specify values in those dimensions. Thus, given interface tools that exploit those preconceived ways of specifying regions of context dimensions, it should be possible for a knowledge enterer stating some assertion P to easily broaden/narrow/ change the domain over which P is asserted to be true, along those *n* dimensions. I.e., to easily “write” some conjuncts on P’s antecedent just by sliding/dialing/...

[35] Namely, the inclusion of those shared conjuncts would just enlarge the search space with fruitless directions to explore, and would make each direction – fruitless or fruitful – longer and more complicated to explore once it was chosen.

Let's review exactly what a Cyc context "is". Anatomically, it has these 8 parts:

(1) content: a group of generally-consistent assertions, sharing many assumptions
(2) core assumptions: a group of assertions (consistent with each other and with the content assertions) that are nontrivially assumed by almost all the content assertions
(3) localized assumptions: a group of assertions (consistent with each other and with the content assertions and with the core assumptions) that are relied on only by some specific assertions in the content of the context. Each of these can be stated in the form of a class of exceptions to an existing rule, thanks to Minimization.
(4) term or nat: the context is a first-class IndividualObject (may or may not be reified)
(5) inter-context links: participates in many kinds of assertions (not just genlMt)[36]
(6) lifting rules: one class of (5), which "map" assertions from one context to another
(7) *n*-dimensional location: specification of regions & tags along various dimensions
(8) assertions that derive some of (2),(5),(6) from that *n*-dimensional location

As it becomes vital to specify the context dimensions' values for each assertion (as accurately as possible), we predict that the whole flavor of entering knowledge into Cyc will likely change back from the current style ("prepare a big file of assertions and dump it into one general context") to the original style, which was "tell Cyc things, one at a time, let Cyc help you dynamically as you are trying to tell it something, including reporting to you omissions, contradictions, and suggestions." We see this as a very important and positive change. In fact, let's make a Point out of it:

Point 10: Moving to* n*-dimensional context-space will likely change KE-ing back from "prepare a big KE file, statically" to "tell Cyc things, one at a time, and let Cyc help you by commenting/reacting to each thing you tell it." Good!

This is a good thing. It will make ever more powerful use of Cyc itself, to help with the KE-ing process. It is very much in line with the applications in our business plan, which involve clarificatory dialogues about database schemas, conversational dialogues about email messages and TV tapings, etc.

Since we want non-CycL-literate users to eventually be the ones carrying on some of these dialogues, the tools must eventually make context-specification as easy as content-specification. In fact, even for our own technical staff, the time has come for Cyc to

[36] For example, C1 models a discourse context in which a certain other context C2 becomes increasingly (ir)relevant as that discourse proceeds. There are *many* inter-context relations worth defining.

actively assist in its own development, and to more (than currently) actively assist in its own consistency-checking of individual assertions and of inter-assertion consequences. In particular, we should expect Cyc to do a decent job at semi-automatically positioning assertions among the *n* dimensions (of context space) we decide will give the most utility. Those may or may not be the dozen that are identified here; in any case, these should be a good start.

Even if Cyc can't do that task completely, now, we can write inference rules which handle particular cases, and which, taken together, will incrementally approach the goal of automatic context-setting. For example: If the Cyc KB contains the dateOfInvention for each DeviceType, and a new assertion P is entered which mentions an automobile, then Cyc could use the date of invention of automobiles to automatically constrain the relevant time of assertion P.

In summary, we expect this rich new context scheme will catalyze and revolutionize both KE-ing and inferencing.

Acknowledgements

We would like to thank everyone who provided comments on drafts of this document. We received useful detailed readings from Ken Murray, Andy Schwartz, Tony Davis, Doug Foxvog, Keith Goolsbey, and Bill Jarrold. A lot of credit goes to Fritz Lehmann and Karen Pittman for their work on the very earliest drafts, over the past two years. Nick Siegel made a huge contribution by coming up with specific vocabulary (predicates, collections, etc.) for talking about each dimension. We also thank in advance each of you who will comment on – or even better be a developer/user of – the design outlined here.

References

Buvac, Sasa (1996) Quantificational Logic of Context. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI, Menlo Park, Ca.

Guha, R. V. (1991) *Contexts: A Formalization and Some Applications*, Stanford University Computer Science Department PhD thesis, Stanford, Ca.

Lenat, D., and R. V. Guha, (1990), *Building Large Knowledge Based Systems*, Addison Wesley, NY.

Lenat, D., (1995), Artificial Intelligence, Scientific American, September.

McCarthy, J., and Hayes, P. J. (1969) "Some Philosophical Problems from the Standpoint of Artificial Intelligence." In (Bernard Meltzer and Donald Mitchie, editors.), *Machine Intelligence 4,* Edinburgh University Press.

McCarthy, J. (1993). Notes on Formalizing Context. *Proc. IJCAI 93*.Morgan-Kaufmann.

Miller, James G., (1978), *Living Systems*, McGraw-Hill: New York, 1978. An excellent summary chart (19 subsystems at 7 levels) is available as Table 13-1, on p. 1028.

Murdock, George P., C. Ford, A. Hudson, R. Kennedy, L. Simmons, J. Whiting, (1987), *Outline of Cultural Materials* (5th revised ed.) New Haven: Human Relations Area Files. [An anthropological taxonomy of concepts, updated regularly since 1938.]

National Library of Medicine (1996), *UMLS Knowledge Sources* , 7th ed., US Department of Health and Human Services.

Peterson, T., (1994) *Art & Architecture Thesarus*, 2nd ed., Oxford University Press.

APPENDIX A: Recap of Some of the Major Points

- Automated inference over a million-rule Knowledge Base (KB) such as Cyc's is very expensive today. To speed it up, localize the search by carving the KB up into a moderately large number of contexts, each containing a manageable number (e.g., 1000) of assertions/facts/rules, and then – when doing inference – always favor the content found in the same (or "nearby") context(s).

- Representing knowledge in a large knowledge based system such as Cyc today is very expensive. To speed up this knowledge-entering process, the person doing it should have some easy way to find related/similar assertions that they can copy and modify, to produce the new rule/fact/… that they want to enter into the KB. This can be achieved by carving the KB up into a moderately large number of contexts, and then – when doing knowledge-entering – doing it in three stages: (i) find the relevant context, (ii) find a similar assertion in it, (iii) copy it and alter that copy to turn it into the new rule.[37]

- Besides thinking of a context as "a named node in an ontology", let's also think of it as being "a region in some 12-dimensional space." Some good dimensions are Absolute Time, Type Of Time, Absolute Place, Type Of Place, Culture, Sophistication/Security, Topic/Usage, Granularity, Modality/Disposition/Epistemology, Argument-Preference, Justification, and Let's.

- The order of applying the 12 qualifications matters, and there can be multiple qualifications of the very same dimension at different places in that series. E.g., a temporal qualification, then a spatial one, then another temporal one.

- Placing each new assertion P into the ideal context is very expensive today – it may take a person longer to situate an assertion than it took them to mentally compose it in the first place! One way to speed up this knowledge-entering process would be to make it mostly just a process of stating/choosing 12 meta-level values (i.e., stating values along 12 well-defined dimensions such as those mentioned in the previous bullet item.)

- Each assertion points to one of a handful of persistence distributions (spike, step, uniform, normal, etc.) and gives a crude estimate of the parameters of that distribution (e.g., mean and standard deviation in the case of a normal distribution, or max/min and overall length in the case of a less regular one)

[37] To optimize (i) and (ii), a KB of size *n* would have Ö*n* contexts each with about Ö*n* assertions in them.

- Starting with a context about “daytime driving automobiles in rural 1920’s America…” we can use the Time dimension to factor out the “1920’s” part of what the context was, the TypeOfTime dimension to factor out the “daytime” part, the GeoLocation dimension to factor out the “America” part, and we will use the TypeOfPlace and Topic dimensions, respectively, to filter out the “rural” and “driving automobiles” parts. In this case, the whole context will be uniquely specified by specifying a set of values on a handful of our dimensions. There may not be any extra assumptions at all that have to get stated; the context C may just receive some assertions. In other words, C is just a set of assertions plus a set of values along a few of our dimensions

- Most “settings” for a context – for any one full value of any one single dimension of context-space – are not worth naming or remembering

- Several conceptual operations need to be supported efficiently/naturally, such as:
  - given an assertion, return the contexts (the sets of 12 settings) in which it holds.
  - given a particular context, modify 1+ of those 12 dimension settings a little.

- Moving to this sort of *n*-dimensional context-space scheme will likely change KE-ing back from “prepare a big KE file, statically” to “tell Cyc things, one at a time, and let Cyc help you by commenting/reacting to each thing you tell it.” This is a good thing.

APPENDIX B: An assertion only relies on *some* of the assumptions

R1: If it’s raining outside, then wipe your feet on the mat.
A1: Assume that the performer has (and can use their) feet.

R2: If it’s raining outside, then carry an umbrella.
A2: Assume that the performer has (and can use their) arms.

Fig. B1. Not every statement in a context “bundle” need rely on all the assumptions associated with that bundle.

Unfortunately, our characterization of a context as a bundle of assertions with “a set of shared assumptions” is a bit too simplistic. As shown in Figure B1, in the box to the left, the assumption A1 that the performer has (and can use their) feet in assertion R1, is irrelevant to the assumption A2 that the performer has (and can use their) arms in assertion R2, even though we might very well find both R1 and R2 bundled together in the same context.

R1’: “If it’s raining outside, and you have/use feet, then wipe your feet on the mat.”

R2’: “If it’s raining outside, and you have/use arms, then carry an umbrella.”.

Figure B2. A heavy-handed way of fixing the problem illustrated in Figure B1.

There is some connection here, intuitively, between A1 and A2: namely, we’re assuming that the people are physically normal. But R2 still applies to a polio victim in an electric wheelchair, though R1 doesn’t; and R1 still applies to someone whose arms are missing or immobilized, though R2 doesn’t. So it feels wrong to list A1 and A2 as assumptions of the context. What’s the solution?

We need a way to say that there’s one sort of exception for R1, and another sort of exception for R2. One heavy-handed way to do this would be modify R1 and R2, as shown above in Figure B2.

R1. “If it’s raining outside, then wipe your feet on the mat.”
E1. If a person has no (use of their) feet, then they are an exception to R1.

R2. “If it’s raining outside, then carry an umbrella.”
E2. If a person has no (use of their) arms, then they are an exception to R2.

Figure B3. A less heavy-handed Fig. B1 problem solution: specify which assertions rely on which assumptions.

Instead of pushing the exception-condition into the axiom itself, like that, it would be better style (i.e., less work) to state each axiom in its simple form – R1 and R2, in Figure B1, above – and then use some “defaults & exceptions” mechanism to state the exceptions. See Figure B3.

The exception axioms (*E1* and *E2*) could and should be lifted from one context to another, just like any other axioms. E1 and E2 make assumptions of their own, such as the performers being human beings, adults, living in the 20th century, magic not existing, etc. Just as we phrased the assumptions about R1 and R2 as exception rules (E1 and E2) capable of blocking the lifting of R1 and R2, so too the assumptions about E1 and E2 lead to second-order exception rules capable of blocking E1 and E2. For example, when trying to lift all four assertions (R1, R2, E1, E2) into Walt Disney's *Sorceror's Apprentice* context in which enchanted brooms and whatnot abound, that last exception condition about E2 – no magic – would trigger, and would block E2's being lifted into that context. Thus, in the *Sorceror's Apprentic*e context, it's not necessarily the case that Mickey needs to use his arms to carry an umbrella, as it could be magically enchanted to move without being held.

Aside: The defaults&exceptions scheme works because of the phenomenon of *minimization*. Think of that as one very general rule that says something like this: only worry about something being exceptional if you have some particular reason to. I.e., most things are not unusual or exceptional. See Figure B4. We indicate what is unusual by stating an axiom of the form (*minimizeAround* P) where P is a predicate; this axiom means that P should be assumed to not hold unless there is some explicit argument that it does hold. So if we define and name both **P** and its negation $\overline{\mathbf{P}}$, then only one (or possibly neither) will be minimized around. There is a *very* weak rule that says that by default all predicates are minimized around. I.e., if you don't know anything better, then just because of the way we humans tend to define predicates, they tend to be false more often than they are true. Of course that is just a weak default rule.

"Given a thing, and a way it could be abnormal, it probably isn't abnormal that way."

Figure B4. Minimization

So to summarize the main point of this appendix: At the crudest level, each context is divided into two parts: content (terms, taxonomic information connecting the terms, heuristic rules involving those terms, etc.) and assumptions shared by the content of that context. There are two sorts of assumptions:

Type-(a): The core shared assumptions of a context; those which practically all the assertions in that context really do depend on – e.g., the performer being a conscious human being living in modern times, etc.

Type-(b): A set of mutually non-contradictory "special-case" assumptions that *some* of the assertions in the context really do depend on, and which (almost all) the others don't care about one way or the other. E.g., people have (the use of) their feet.

The way to handle Type-(a) assumptions is to assert them about the context as a whole. The way to handle Type-(b) assumptions is to state them as separate "exception" rules in the style of E1 and E2, above (see Figure B3, above.)