Name=Struct: A Practical Approach to the Sorry State of Real-Life Chemical Nomenclature

J. Brecher

CambridgeSoft Corporation, 100 CambridgePark Drive, Cambridge, Massachusetts 02140

Received June 22, 1999

The interpretation (automatic or otherwise) of chemical names is complicated by the poor quality of such names in common usage. A practical interpretation system for chemical names must consider not only the published rules for chemical nomenclature but also their misinterpretations, corruptions, and popular extensions. This paper describes the design of Name=Struct, an automated system for the conversion of chemical names to their corresponding structures, and discusses the nomenclature interpretation problems addressed in its development. The comprehensiveness and accuracy of Name=Struct are analyzed in several contexts.

INTRODUCTION

Nobody speaks Chemist as a native language.

The nomenclature used to describe chemical structures *is* a language, as others have noted.[1−3] A specialized one, surely, but one that is systematic and rule-based, with specific syntax and semantics that allow complicated terms to be built from simpler morphemes taken from an agreed-upon (but otherwise arbitrary) lexicon, just like English or German or Japanese or any other spoken language. Unlike English or the others, however, nobody speaks Chemist in day-to-day activities, and nobody grows up learning it. As a language without native speakers, chemical nomenclature poses some interesting problems for those trying to identify the particular chemical structures described in textual form in the literature and elsewhere.

We first encountered these problems while constructing the *CS ChemFinder WebServer*,[4] an index of chemical information provided on the Internet. The information we found was raw to say the least, a diverse collection of names and other terms provided by people trying to communicate information about the substances in question. "Communicate" is the key word here, as communication is the primary purpose of any language, and "adequate" communication is more the norm than "optimal". In building the *ChemFinder WebServer*, we needed a way to figure out, for example, that phthalonitrile and *o*-dicyanobenzene—each perfectly adequate names in their own right—both represent the same substance even though they have not even a single pair of adjacent letters in common. We investigated some approaches for normalizing nearly-identical names[5] but made no progress in many common cases such as the above example. It became clear that the only long-term solution was to use some routine capable of interpreting chemical nomenclature, so that the invariant structures could be compared directly. We needed a system that was at least as clever as a trained chemist, that could take unmodified names of varying quality and display the corresponding structural diagrams. It needed to be as *comprehensive* as possible, meaning that it would understand a large variety of names and nomenclature types. At least as important, it had to be as *accurate* as possible, so that the user could have confidence that each structure truly was representative of the chemical name provided.

Such a system was described by Vander Stouw over 30 years ago[6,7] but was never commercialized and so was not available to us. Other approaches have been limited in scope, restricted to the interpretation of steroid[8] or stereochemical and isotopic[9] nomenclature. Rayner and co-workers published a series of papers[3,10−14] describing a grammar-based approach to the interpretation of "IUPAC systematic organic chemical nomenclature," but focused only on "certain classes of compounds of industrial importance, including some cases of semisystematic and trivial nomenclature." Even if it were a comprehensive interpreter of IUPAC organic chemical nomenclature, even if it also interpreted IUPAC inorganic and biochemical and other types of nomenclature, such a system would not have met our needs. People do not rigidly follow grammatical rules in their native spoken language—witness "gonna", "ain't", "y'all", split infinitives, dangling prepositions, and a host of other pitfalls in everyday English. Given that people do not follow the formalized rules of their native language, it would be reasonable to assume that neither do they follow IUPAC rules for grammar when creating chemical names. Unfortunately, that assumption was confirmed when we examined chemical sites on the Internet and later also when we examined catalogs produced by commercial chemical vendors.

Any system designed to interpret chemical names must understand "The Rules", but even a thorough understanding of any set of published rules will be insufficient to interpret the vast majority of chemical names in common usage. The Name=Struct system described in this paper represents our successful efforts to create a comprehensive system that understands and converts to structural form not only the systematic chemical names that *should* be used but also the semisystematic, asystematic, obsolete, ambiguous, and otherwise "creative" names that are the reality of modern chemical communication.

10.1021/ci990062c CCC: $18.00
Published on Web 09/28/1999

GENERAL DESIGN

The sole purpose of Name=Struct is to interpret chemical names. It makes no value judgments, focusing only on providing a structure that the name accurately describes. Name=Struct has no concept of a "correct" name. Instead, it should do its best to interpret any name presented. Chemically reasonable names should be interpreted in a chemically reasonable manner; other names should be interpreted as accurately as possible, whether or not such an interpretation appears at first glance to be chemically reasonable. Also, while correct interpretation of published nomenclature rules is important, the interpretation of "real-life" usage is most critical, whether or not the two coincide. We formulated six principles by which a general-purpose chemical nomenclature interpretation system should behave:

1. Anything Allowed by the Rules Is Acceptable. Regardless of how often the rules are violated, they are followed some of the time, and names that actually do follow the rules should be interpreted correctly. When speaking of "The Rules" for chemical nomenclature, chemists generally refer to those published by the International Union of Pure and Applied Chemistry (IUPAC), either alone or with its sister organization the International Union of Biochemistry and Molecular Biology (IUBMB), or by the Chemical Abstracts Service (CAS). Even though we found that these rules (or "recommendations" in the case of IUPAC/IUBMB) are broken as often as they are followed, they remain the most widely recognized guides for how chemical nomenclature should be formed. A nomenclature interpretation system had better get these right.

To complicate matters somewhat, IUPAC has never published a single comprehensive nomenclature reference. Its recommendations for organic, inorganic, and biochemical nomenclature were last published in book form in separate volumes during the early 1990s,[15−17] and even then the organic recommendations were not intended to be complete, but only an "outline of the main principles" of the earlier 1979 recommendations.[18] Additional papers have expanded upon or corrected portions of these works.[19] Fortunately, the IUPAC Nomenclature Home Page[20] is becoming increasingly thorough in its compilation of the relevant texts.

2. Anything Forbidden by the Rules Is Acceptable. The rules are written as a guideline of what should be done; forbidden nomenclature is discussed in the rules as counterexamples of what should not be done. Yet, if a rule is subtle enough that it warrants a counterexample in the first place, then surely it will be violated in common usage.

For example, both IUPAC and CAS rules stipulate the alphabetical ordering of substituents;[21,22] thus, 3-bromo-2-chlorobenzoic acid. There is nothing ambiguous about the (deprecated) 2-chloro-3-bromobenzoic acid. Such a name should be recognized, as it is in fact commonly found.

IUPAC allows unlimited substitution on acetic acid ("2-chloro-2-fluoroacetic acid"), but none at all on propionic acid and butyric acid. CAS also allows the use of acetic acid with unlimited substitution but does not allow its organic replacement analogues (use "ethanimidic acid" rather than "acetimidic acid") and forbids the use of propionic acid and butyric acid altogether. Such names should be recognized in all their variations, as should other members of this series ("valeric", "caproic", "enanthic", and so on).

This guideline also applies to nomenclatures that are recommended in one edition of the rules and deprecated in a subsequent one. A given name might have been created during the time that the first set of rules were current, and not updated in its printed form since then. Dated works (including journal articles) cannot be modified retroactively and can be corrected only with later editions. It should not be assumed, either, that everyone will be aware of the most recent rules. Both the current and the obsolete systems should be supported.

3. Any Rule That Can Be Extended Should Be. IUPAC recommends the use of Hantzsch−Widman nomenclature for heteromonocyclic rings. These recommendations allow only 19 different heteroelements.[23] CAS also uses Hantzsch−Widman nomenclature but allows only 14 different heteroelements (not mercury, fluorine, chlorine, bromine, or iodine).[24] Whether 19 or 14, a limitation of any type is arbitrary: If someone were to refer to "1,3-polonazole" they would clearly mean a five-membered unsaturated ring containing polonium and nitrogen, analogously with "1,3-oxazole" and "1,3-thiazole", and it should be interpreted as such.

4. If It Looks Like It Ought To Be a Rule, It Is a Rule. Neither the CAS nor the IUPAC rules are complete, and they do not pretend to be. CAS states that its reference to chemical substance names "is not a nomenclature manual. It has the more restricted aim of enabling a user of CA indexes to proceed from the structure of an individual chemical compound to the place in the current *Chemical Substance Index* where the particular index name and any associated index entries will be found."[25] The IUPAC recommendations are scattered through numerous publications, but the comments prefacing those for organic nomenclature are typical: "[T]his *Guide*...often presents alternative sets of rules, equally systematic,...to enable a user to fit the name to a particular need.... [T]he Commission recognizes that for certain types of compounds, there is significant disagreement among chemists in different fields as to what should be the preferred nomenclature....Therefore, the Commission's policy is to offer critically examined alternatives...and to observe how they are accepted and used."[26]

For example, the name "acetone" is commonly used, but the similar names "butyrone" (for di-propyl ketone), "valerone" (for di-butyl ketone), and the like are also seen. Published or not, it seems that there is a rule to the effect that (name of trivial acid) + (one) = di-(number of carbons in the trivial acid, minus 1)-yl ketone. "Enanthone", "stearone", "melissone", and the like should be recognized, and the appropriate structure should be provided.

5. The Meaning of Logically Ambiguous Names Is Determined by Common Usage. In the strictest sense, "propyl chloride" is ambiguous: It could refer to "prop-1-yl chloride" or "prop-2-yl chloride" or even some mixture of the two. Common usage assumes that undifferentiated alkanes are in the unbranched form.

In a more subtle example, consider the two underlined names in Figure 1. "2-Chloroethylbenzene" and "trifluoromethylsilane" each could reasonably refer to two or more substances. In the presence of suitable punctuation, the ambiguity disappears, but in the absence of punctuation, common usage demands the second interpretation in each case.

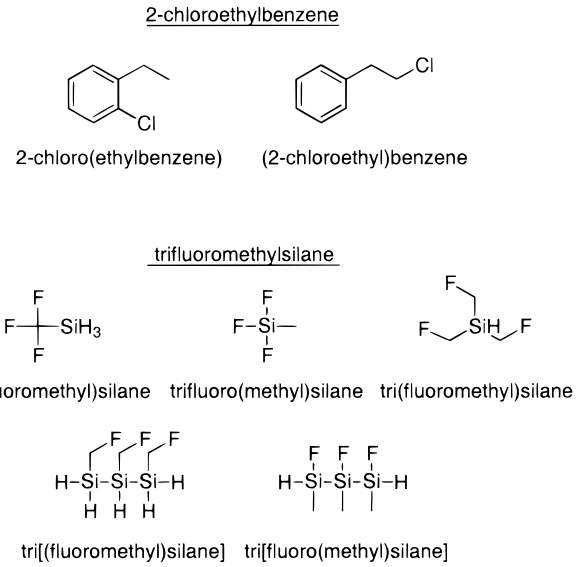

Figure 1. The meaning of ambiguous names can be clarified by the addition of appropriate punctuation.

Common usage is itself determined empirically. For each ambiguous case, we sought WWW sites or chemical vendor catalogs that had examples of similar usages. Journals were not helpful in this regard since their higher editorial quality reduced the frequency of ambiguity. As other cases were found, it was determined from context (usually from the presence of unambiguous synonyms or CAS registry numbers) what structure was intended for the ambiguous name.

6. Punctuation and Capitalization Do Not Matter; Spacing Matters as Little as Possible. The *ACS Style Guide* needs a dozen pages to discuss punctuation and capitalization just for regular English prose.[27] Even the most cursory examination of chemical names as they are actually used shows that punctuation and capitalization rules are the first ones abandoned. A selection of punctuation and capitalization varieties that we found frequently is shown in Table 1; other varieties are also seen, although less often.

Spacing matters when interpreting esters (where it differentiates between an ester and an anion) and in ethers and related compounds (where it serves as a sort of "weak" parenthetical marker). Figure 2 shows the importance of a space character—or its absence—in names of these sorts. Names are first interpreted as if the space character were a hard delimiter. If the fragments on either side of the space character can be fully interpreted, then the space is considered to be significant and is retained. Otherwise, the space character is ignored, and the fragments on either side are reinterpreted as one unit. The results of this process can be seen in Figure 3. "4-Chloro-2-ethyl aniline" is parsed as if

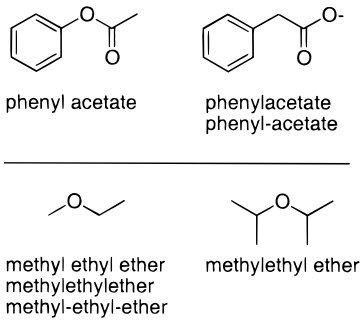

Figure 2. Spacing is important in the interpretation of some classes of nomenclature, including esters and ethers.

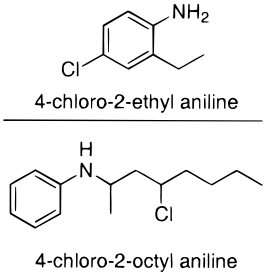

Figure 3. Inconsistent use of spacing can result in inconsistent structures.

the space were absent because "4-chloro-2-ethyl" cannot be interpreted when considered on its own. The similar name "4-chloro-2-octyl aniline" is treated differently, because "4-chloro-2-octyl" can be interpreted as equivalent to "4-chloro-oct-2-yl". The different treatment of similar names is unfortunate, but generally limited to names where a space character is used not only incorrectly (according to the rules) but also inconsistently. There are three primary fragments in the two names: "4-chloro", "2-*alkyl*", and "aniline": If space characters had been used consistently (either present before both the "2" and the "aniline" or absent in both places), the two names would have been interpreted similarly.

INTERPRETING THE LANGUAGE OF CHEMISTRY

Spoken languages have only a few parts of speech: nouns, verbs, adjectives, and a handful of others. Chemical nomenclature does not have the same parsimony of types. IUPAC lists nearly 600 rules just in Sections A, B, and C of the 1979 Organic recommendations, but they make many distinctions that are not observed in common usage. CAS is somewhat more concise, dividing their current rules under 122 headings, including such nontechnical categories as "Introduction". Name=Struct is currently based on a core of 58 principal types, many of which are subdivided further. These are shown in Table 2, with the semidescriptive

Table 1. Some Common Punctuation and Formatting Variances in Chemical Names

example	description
p-Dinitrobenzene	IUPAC recommended form
p-Dinitrobenzene	no italics
p-dinitrobenzene	all lower case
P-DINITROBENZENE	all upper case
p-DINITROBENZENE	locant in lower case, remainder in upper case
P-dinitrobenzene	first alphabetic character capitalized, remainder in lower case
pDinitrobenzene	missing hyphen
p-Di-nitro-benzene	added hyphens
p-Dinitro benzene	added space

Table 2. Chemical Nomenclature "Parts of Speech" Recognized by Name=Struct

category name	count[a]	percentage[a]	typical examples
Ordinal/Unknown[b]	121509	80.3	1, 2, alpha
Root[b]	105924	70.0	meth-, eth-, prop-
Suffix	88702	58.6	-ane, -ene, -ol
Prefix	61681	40.7	tri-, tetra-, penta-
Acid	40534	26.8	-oate, -amide, -ophenone
Infix	21364	14.1	-oxy-, -amino-
AcidLover	12809	8.5	phosphor-, mangan-, cinchonin-
Replacement	12802	8.5	oxa-, aza-
AcidPart2	12215	8.1	acid, amide (after -ic)
Counterion	10507	6.9	chloride, bromide, iodide
Cyclo	6510	4.3	cyclo-
Heterocyc	6384	4.2	-etine, -etane, -ocine
Hydro	6351	4.2	hydro-
Tert	5766	3.8	iso-, sec-, neo-, tert-
OPFuser	5570	3.7	naphtho-, anthra-
AminoAcidEnder	4838	3.2	-ine, -yl
Roman	4397	2.9	I, II, III, IV
Salt	3461	2.3	salt, salt of
Ester	3378	2.2	ester, ester of, ester with
Ylene	3314	2.2	-ylene
Toluene	2939	1.9	toluene, cumene, cumidene
Benzo	2900	1.9	benzo-
AminoAcid	2337	1.5	asparag-, ethion-
AcidDeriver	2234	1.5	-chlorido-, -cyanatido-
Sugar	2230	1.5	gluco-, manno-, galacto-
Chargegiver	1887	1.2	-onium, -yl cation
SugarEnder	1686	1.1	-ose, -itol, -oside
OrganometallicAnion	1181	0.8	-ate
PseudoSugar	1049	0.7	glycer-, thymin-
CyanicEnder	956	0.6	acid
DiacidDeriver	768	0.5	-am-, -aldehyd-
Thio	763	0.5	thio-, seleno-, telluro-
Deoxy	745	0.5	deoxy-
Nucleotide	716	0.5	adenos-, thymid-
Pyranose	695	0.5	-pyranose, -furanose, -oxetose
Cyanic	614	0.4	cyan-, fulmin-, hydrochlor-
Radical	484	0.3	radical
Glycol	476	0.3	glycol, chlorohydrin
Methanomaker	420	0.3	-ano-
Carbeth	403	0.3	carb- (as in carbethoxy-)
Azo	376	0.2	-azo-, -hydrazo-, -azoxy-
Imide	304	0.2	-imide, -chlorimide
Spiro	230	0.2	spiro-
NatDeriver	228	0.2	homo-, nor-, seco-
AcidPerModifier	208	0.1	per-, hypo-
Basic	200	0.1	basic (as in phosphoric acid, dibasic)
ReqIneAminoAcid	176	0.1	cyste-, glutam-
StructSugarEnder	165	0.1	-ulo-, -osamine
Imine	160	0.1	-imine
Acene	108	0.1	-acene, -aphene
Crown	94	0.1	crown
Toluidide	62	<0.1	toluidide, xylidide, anisidide
DiacidTailDeriver	44	<0.1	semialdehyde
Oin	42	<0.1	-oin (as in furoin)
AminoDiacid	39	<0.1	cyst-, lanthion-
Glycerin	35	<0.1	-in, -anoin (as in tributyrin, tributanoin)
Annulene	20	<0.1	annulene
MultSugar	17	<0.1	tetro-, pento-, hexo-

[a] Count and percentage columns indicate the frequency of each fragment type relative to 151 407 names extracted from the database of the *CS ChemFinder WebServer*. [b] Because unrecognized root names (including trade names, etc.) are classified as "Unknown", the counts for that category are artificially inflated while the counts for the "Root" category are artificially lowered.

category names used by Name=Struct and example members of each category. As with everything else in Name=Struct, the types were determined empirically. Each type identifies a nomenclature environment where different terms are used to the same or very similar structural end result.

Name=Struct divides input names into recognized fragments of maximum length, starting with the first character and proceeding sequentially. Thus, the name for a five-carbon alkane is interpreted as "pent + ane", not as "p + en + t + an + e" or any other combination of recognizable chemical terms. Paired parentheses, brackets, and braces are treated as absolute delimiters between name fragments, and everything within a pair is parsed as a unit, with recursive descent through all nested levels. Other punctuations, if present, are treated as delimiters between name fragments, so "pentane," "pent-ane," and "pent.ane" would be interpreted identically, while "p-entane" would not. All punctuation is discarded after the name fragments have been divided, since the "correct" syntactic use of punctuation cannot be assumed.

In rare cases, those simple fragmentation rules are insufficient. For "4-nitrosalicylate," strictly identifying the longest recognized fragments at each character would produce "4 + nitros + al + ic + yl + ate", where "nitros" is an elided form of "nitroso". This name is clearly uninterpretable. Incorrect fragmentations of this type are extremely rare, with fewer than 20 cases found to date, and they have been handled on an ad hoc basis. In this instance, the addition of a rule that disallows two consecutive "acid"-type fragments ("al" and "ic") produces the correct fragmentation ("4 + nitro + salicyl + ate").

Once an input name has been divided into smaller fragments of known type, its corresponding structure can be assembled according to rules that are type-specific and not associated with the actual text of the name. The fragment assembly process implies a hierarchy of types, or at least an ordered sequence of processing steps. The name "3-chloropropenoic acid" is divided into five main fragments: "3 + chloro + prop + en + oic acid". Each of the last two fragments modifies "prop", introducing an unsaturation and a functional group, respectively, onto the three-carbon chain. However, they must be applied in the reverse order. First placing the acid group at C-1 leaves the correct position (between C-2 and C-3) for the double bond. If the double bond were placed first between C-1 and C-2, the remainder of the name could not be interpreted. Attempting to place the acid group also on C-1 in that case would produce a 5-coordinate carbon; placing it at C-3 would leave no room to attach the "chloro" fragment.

To the best of our knowledge, no such hierarchy has ever been published. The one used by Name=Struct was determined empirically through hundreds of cases like the one shown above. Regrettably, a comprehensive discussion of the entire Name=Struct algorithm—some 20 000 lines of C++ just for the nomenclature interpretation, not to mention a similar amount again for the handling of the chemical structure data objects and an additional 10 000 lines needed to produce aesthetic structural diagrams as output—is beyond the scope of this paper, but an overview of the main analysis hierarchy can be presented.

The process begins by considering the most localized nomenclatural features and continues through progressively broader issues. Strictly lexicographical issues such as removal of punctuation and correction of a limited number of recognized typos are handled in a preprocessing step since they can be performed mechanically and without any

understanding of the underlying nomenclature. Uninversion of CAS-style names is also performed at this point, and then the input names are parsed into recognized fragments as discussed above.

Identification of stopwords also takes place early in the process. In this context, "stopwords" refer to fragments that always indicate the end of the useful portion of a chemical name. As we examined real-life uses of chemical nomenclature, we found notations such as "acetic acid, glacial" and "copper sulfate 99.5%," where the "glacial" and "99.5%" describe the state of the compound but do not contribute to the way its structure would be depicted. Stopwords are removed from the chemical name, as are any subsequent notations, which are likely also to be descriptive in nature.

Fragment consolidation continues with the most localized issues, addressing first a small set of nomenclature types that are fairly well-behaved. For example, annulenes always appear in the form "*number*-annulene", ignoring punctuation. There is no variation. Name fragments such as "annulene-*number*," "annul-*number*-ene," and especially "*number*-annul-*somethingelse*-ene" are never seen, and so any annulene moieties can be parsed with confidence. Once the fragment is parsed, it is reclassified as a regular type "Root" for subsequent interpretation such as the addition of suffixes and other functional groups.

Structural interpretation continues with a series of less-localized nomenclature types characterized as sometimes appearing in multiple nonadjacent fragments. Interpretation of atomic chains is typical of actions performed at this stage. The "penta" in "pentadiene" must refer to a five-carbon chain. On the other hand, the same "penta" in "decapentaene" must mean that the unsaturation—the "ene"—will occur five times in the *ten*-carbon chain described by the preceding "deca". Other major events taking place in this portion of analysis are the creation of cyclic systems and the fusion of aromatic rings. As with the interpretation process as a whole, order of interpretation is important even in the individual sections. A name like "benzocyclooctene" implies that chains must be interpreted first, then closed with "cyclo" prefixes all before they can participate in ring fusions. At the end of this section, Name=Struct has identified those portions within the greater set of name fragments that correspond to what most chemists would recognize as "root" or "core" structures.

Having identified the main root portions of the name, Name=Struct then attempts to interpret the prefixes and suffixes that directly modify them. Most functional groups are recognized at this stage, including acids in all their variations, radical suffixes like "-yl", and the prefixes of heterocyclic "aza" nomenclature. This is one of the messiest portions of the interpretation process because of the many special cases that have evolved over the years for describing chemical functionality. As an example, "tetrazine", "triazine", and "diazine" are six-membered aromatic rings containing four, three, and two nitrogens, respectively. The simple "azine" is acyclic and represents a functional group in its own right, related to the others only in that it also contains nitrogen.

In its final section, Name=Struct assembles the remaining fragments, which by this point represent complete ligands or core groups. So, while the acid functionality of esters would have been interpreted in the previous section, producing the two fragments "ethyl" and "acetate" for example, those fragments would be joined at this stage. Other multifragment nomenclature is also handled in this section, including salts and other pseudo-ionic compounds (hydrochlorides) and conjunctive nomenclature (benzenemethanol).

Standard chemistry rules are explicitly considered throughout the fragment assembly process. Valence conventions in particular play a key role in the arrangement of fragments and are responsible, for example, for the correct placement of the double bond in "propenoic acid" as discussed earlier. The interactions between standard chemistry rules, standard grammar rules, and common usage can be very subtle. In "methanol-d_1", the single deuterium is attached to the oxygen, while all four hydrogens of "methanol-d_4" are replaced with deuteriums. What about "methanol-d_2" and "methanol-d_3"? In those cases, common usage dictates that all of the deuteriums are attached to the carbon. A strictly grammar-based approach would confirm that it is possible to end a chemical name in "-d_{number}", but no procedure could identify the placement of the hydrogens without referring to the actual structure determined by the rest of the name.

A type-based approach has proved effective in allowing new nomenclatures to be added with essentially no programming effort. For example, "glycol" is mentioned in the IUPAC recommendations in the particular case of ethylene glycol; its only mention in the CAS rules is in the further limited context of ethylene glycol polymers. Its implementation in Name=Struct was extended according to the "Any Rule That Can Be Extended Should Be" guideline, allowing propylene glycol, butylene glycol, and so on. Once the behavior for the "glycol" type was defined in general, other glycol-like nomenclatures (such as the "halohydrin" nomenclature, which isn't mentioned at all by either IUPAC or CAS) follow analogously: propylene chlorohydrin, xylylene cyanohydrin.

Similarly, Name=Struct could readily be extended to handle non-English chemical names. Defining "-säure" as a member of the same type as "-ate" or "-ic acid" goes a long way towards handling many common German names: acetylsalicylsäure, acrylsäure, etc. Because the omission of the final "e" in chemical names is one of several common English typos that it recognizes anyway, Name=Struct already does interpret many other German chemical names: aceton, benzylchlorid, etc. The rules for punctuation and spacing are different in different languages, but since they are already ignored by Name=Struct, they are not a major cross-language issue.

Errors are reported to the user only when they are definite and cannot be resolved. Unbalanced parentheses are one example. In a small number of other cases, chemical names have internal redundancy: names of the form "bicyclo[*x.y.z*]-*mult*ane" must have $mult = x + y + z + 2$, and an error is returned in other cases. More often, the exact source of the problem cannot be identified. A name like "m-ethylamine" will fail to generate a structure without any further explanation because an accurate error message is not possible. There is no way to determine whether the user added an extra hyphen (from "methylamine"), mistyped a character (from "N-ethylamine"), omitted an entire name fragment (from "m-ethylbenzenamine") or made some other mistake. "Pentachloromethane" might appear to be "unreasonable" with its five chlorines attached to its single carbon, and so might be

considered an erroneous name. Still, such a structure could be important to someone, perhaps as a transition state. Since its structure is unambiguous (albeit a little strange-looking), the structure is displayed. The user is also presented with a warning to take "Caution: valence appears to be exceeded." Errors that can be identified and resolved with confidence are so resolved without notice to the user; Name=Struct is designed to display chemical structures and not designed to provide advice or coaching on alternate nomenclatural styles if the one used can be interpreted without question.

UNPARSABLE NAMES

Some names can have no structures associated with them even by the most sophisticated algorithms. In other cases, a structure is possible but not particularly useful. Name=Struct does not currently interpret any of the following types of names:

Mixtures, and particularly those natural products that are themselves mixtures: mineral oil, milk, sand, garlic oil, gasoline, and mixtures described by their properties, such as "buffer pH 9".

Isomeric mixtures that specifically indicate an unknown composition: "xylenes", "hexanes". Name=Struct will, however, interpret names without locants such as "dimethylbenzene", producing the structure that is commonly intended if there is a clear preference in common usage, or producing a random structure otherwise.

Macromolecules: agarose, lanolin, chitin, cellulose, and others including all proteins and enzymes. There is no particular reason that substances of this type could not be interpreted, but the current implementation of Name=Struct is designed to produce two-dimensional structures as output[28] and a two-dimensional projection of a protein was deemed to be of minimal use.

Fully asystematic names, especially of dyes and drugs: Brilliant Green, Bis-Tris, C.I. 75660, Viagra, "compound #3", "brown sludge from Thursday's reaction". These names have no systematic component and cannot be further modified (no name exists of the type "2,3-diphenyl-C.I. 75660"), even though they do refer to specific substances. The only possible way to interpret such names is via exact database lookup, and there are already many reference works that contain common names. Name=Struct in its current form has been limited to those names that are themselves systematic or that can serve as the basis for further modification.

VALIDATION OF RESULTS

Prior to the start of this project, we had collected over 150 000 unique chemical names within the *ChemFinder WebServer*. To that number we added a similar quantity of additional names from a variety of sources, primarily including the catalogs of commercial chemical vendors. From this complete set we selected a 30 000-name subset for diversity and examined it closely throughout development. After the smaller subset was interpreted satisfactorily, we then examined the entire list of approximately 300 000 names.

Development began with recognition of basic IUPAC nomenclature: alkanes and other carbon skeletons, simple functional groups and substituents, and the like. The IUPAC recommendations were selected as a starting point because they provided a reasonably concise list of major types of organic nomenclature. Additional IUPAC nomenclature recommendations were added incrementally. Repeated review of uninterpreted names revealed areas in which the IUPAC and CAS rules inadequately described common usage. We then extended the implementation of the rules according to the six principles discussed earlier so as to minimize the number of unparsable names without introducing incorrect interpretations.

Since it is difficult to enumerate a complete list of nomenclatures, it is consequently difficult to place a precise value on the comprehensiveness of Name=Struct. In a general sense, Name=Struct handles the vast majority of all chemical nomenclatures: organic, inorganic, organic with inorganic ions, natural products, radicals, and others too numerous to mention. Depending on how you count the individual rules and subrules, Name=Struct fully supports over 95% of Sections A, B, and C of the 1979 IUPAC Organic Nomenclature recommendations. It fully addresses the issues raised by more than 90% of the 1993 IUPAC Organic Nomenclature recommendations, plus most of the biochemical and inorganic recommendations. Since the CAS rules can be described as a restricted subset of the IUPAC recommendations without many of the most-obscure rules, Name=Struct actually supports a slightly higher percentage of the CAS rules (both for inverted and uninverted names). Rules were selected and implemented on the basis of their frequency in common usage; some rules remained unsupported in the initial release of Name=Struct not because of their difficulty but because of lack of time to add support for them. The unsupported rules range from somewhat esoteric to extremely so—one measure of their obscurity is that they are all less frequently observed than even the bottommost items in Table 2. Of course, these figures are relevant only when the names to be interpreted conform to the rules in the first place.

Name=Struct was used to generate compounds from the names in catalogs produced by commercial chemical vendors, providing a measure of comprehensiveness more applicable to real-life usage. All names and synonyms from several catalogs were interpreted, and the structures generated were examined individually for accuracy. Comprehensiveness varied from 72% to 92% of the total names interpreted, depending primarily on the overall systematicity of the names in each catalog. In particular, the catalog with the least comprehensive coverage by Name=Struct (Alfa-ÆSAR) contains a high proportion of entries with descriptive or trade names ("Chlorine, Aquaquant™ Test Kit"). If the failures on such unparsable names are disregarded, the coverage in all cases was in excess of 97% (Figure 4).

Name=Struct was also used to generate structures for each of the 150 000+ synonyms in the *ChemFinder WebServer* database. The total coverage of 55% in this database was significantly lower than was found in the catalogs, reflecting the high proportion of trade and other unparsable names in this database. Automated comparison of the 85 000 structures generated by Name=Struct against the corresponding structures in the database revealed over 8000 discrepancies (9%), but a further manual review showed that nearly all of these discrepancies were due to previously unnoticed errors in the existing database or reasonable interpretations of ambiguous names that happened not to match the particular structure in

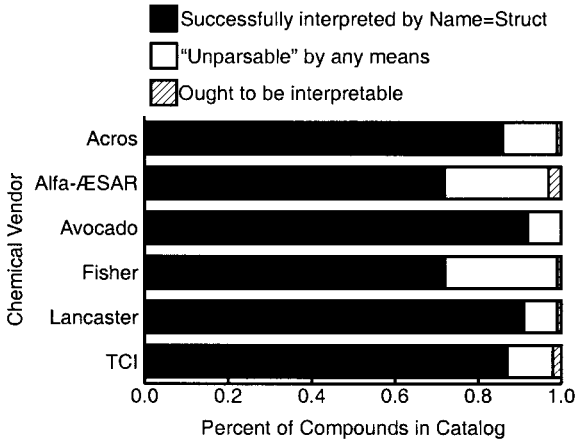

Figure 4. Success of Name=Struct in interpreting unedited chemical names extracted from catalogs of commercial chemicals vendors.

the database. At final analysis, the structures generated by Name=Struct were found to be accurate well over 99% of the time, with the final 1% consisting primarily of names whose meaning could reasonably be debated even by trained chemists.

In an effort to explore the worst-case performance of Name=Struct, we analyzed the usage logs of the *ChemFinder WebServer* over a period of several weeks. From these logs we extracted queries where the user performed a search by chemical name and where there were no corresponding matches found in the *ChemFinder WebServer* database. A total of 58 844 such queries were identified, yielding 40 670 unique names. We did not attempt a complete review of all names, but a cursory examination confirmed their extremely poor quality: Maleic anhydride alone was represented by at least 39 lexicographically distinct variants, ranging alphabetically from "maaleic anhydride" to "Msleic Snhydride". The collection also included many nonchemical names such as "acid rain and soils", "beer", "chicago university", and "chicken feces". Nonetheless, Name=Struct successfully generated structures for 19 729 (33.5%) of the names and 9244 (22.7%) of the unique names. Equally striking, 4239 (7.2%) of all structures and 3086 (7.6%) of unique structures represented compounds *that were in the database* and could be located by exact structure searches with the structures generated by Name=Struct. That is, a researcher was prevented from finding information strictly because of nomenclature differences and Name=Struct *was able to surmount those differences and identify the information sought*.

LIMITATIONS

The most significant limitation of this first release of Name=Struct is its lack of support for Cahn−Ingold−Prelog stereochemistry (*R*, *S*, *E*, *Z*), and relative stereochemical terms (*cis*, *trans*). These were identified as a discrete nomenclature topic that could be omitted without invalidating the remainder of the name interpretation process. Regrettably, time constraints did require its omission from the first release, a deficiency we expect to rectify in a later version. In its current form, Name=Struct identifies stereochemical terms as being present, then ignores them and produces the relevant structure without stereochemical indicators (hashed and wedged bonds). In such a case, the user is also warned to take "Caution: Stereochemical terms discarded." Stereochemistry is supported for amino acids, steroids, and carbohydrates, three classes of compounds whose structures require stereochemistry for any sort of meaning.

Other types of stereochemical indicators reference physical properties that cannot reliably be determined algorithmically; these include the (+) and (−) indicators for optical rotation, which will never be supported by Name=Struct. The D- and L- prefixes are recognized only for amino acids and carbohydrates; their use for other types of compounds is not predictable.

Some other limitations are in the interpretation of nomenclatures determined to be obscure in common usage. Support for isotopic labeling is the most noticeable example. Deuterium labeling (of both the "2,2-dideuteropropane" and the "propane-2,2-d_2" types) is commonly seen and is currently supported. While other types of isotopic labeling are commonly seen, labeling at specific positions is not. It is difficult to generate a reasonable structural diagram for compounds with unspecified labeling.

In a similar vein, Name=Struct does not currently support certain highly bridged ring systems, including cubane, fullerenes, and polyboranes, pending advances in structure generation algorithms to generate graphical depictions that look reasonable. Polymers as a general class are unsupported for similar reasons, as are inorganic coordination complexes (μ-, η-, and -ato- nomenclature), alloys, and elementary particles. These do not pose a particular challenge in interpretation but have been specifically disabled in this version of Name=Struct until more satisfactory diagrams can be generated.

Finally, subtractive nomenclature (de-, des-) is largely unsupported. Its correct usage is rare and it can easily be abused. Methane, after all, is "de(phenyl)toluene", "des-(ethyl)propane", "des(methylacetamido)dimethylacetamide", and so on.

SUMMARY

Through a combination of methods, Name=Struct is able to overcome the limitations of published nomenclature systems in providing a practical system for the interpretation of chemical names. Treating chemical nomenclature as a language is a useful approach. Even if treated as a language, however, it cannot be regarded as an "ideal" one. Mistakes are made by people producing chemical names, just as they are made by people when writing or speaking. Deviations from published nomenclature systems should not render a name as uninterpretable. Name=Struct is a practical system for the interpretation of the majority of chemical names in common usage, regardless of their provenance or quality.

REFERENCES AND NOTES

(1) Luque Ruiz, I.; Cruz Soto, J. L.; Gómez-Nieto, M. A. Error Detection, Recovery, and Repair in the Translation of Inorganic Nomenclautres. 1. A Study of the Problem. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 7−15.

(2) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. From names to diagrams−by computer. *Chem. Br.* **1985**, *21*, 467−471.

(3) Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 1. Introduction and Background to a Grammar-Based Approach. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 101−105.

(4) CambridgeSoft Corporation. http://www.chemfinder.com (accessed September 1999).
(5) Brecher, J. S. The ChemFinder WebServer: Indexing Chemical Data on the Internet. *Chimia* **1998**, *52*, 658−663.
(6) Vander Stouw, G. G.; Naznitsky, I.; Rush, J. E. Procedures for Converting Systematic Names of Organic Compounds into Atom-Bond Connection Tables. *J. Chem. Doc*. **1967**, *7*, 165−169.
(7) Vander Stouw, G. G.; Elliott, P. M.; Isenbert, A. C. Automated Conversion of Chemical Substance Names to Atom-Bond Connection Tables. *J. Chem. Doc*. **1974**, *14*, 185−193.
(8) Stillwell, R. N. Computer Translation of Systematic Chemical Nomenclature to Structural Formulas—Steroids. *J. Chem. Doc*. **1973**, *13*, 107−109.
(9) Ihlenfeldt, W. D.; Gasteiger, J. Augmenting Connectivity Information by Compound Name Parsing: Automatic Assignment of Stereochemistry and Isotope Labeling. *J. Chem. Inf. Comput. Sci*. **1995**, *35*, 663−674.
(10) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 2. Development of a Formal Grammar. *J. Chem. Inf. Comput. Sci*. **1989**, *29*, 106−112.
(11) Cooke-Fox, D. I.; Kirby, G. H.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 3. Syntax Analysis and Semantic Processing. *J. Chem. Inf. Comput. Sci*. **1989**, *29*, 112−118.
(12) Cooke-Fox, D. I.; Kirby, G. H.; Lord, M. R.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 4. Concise Connection Tables to Structure Diagrams. *J. Chem. Inf. Comput. Sci*. **1990**, *30*, 122−127.
(13) Cooke-Fox, D. I.; Kirby, G. H.; Lord, M. R.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 5. Steroid Nomenclature. *J. Chem. Inf. Comput. Sci*. **1990**, *30*, 128−132.
(14) Kirby, G. H.; Lord, M. R.; Rayner, J. D. Computer Translation of IUPAC Systematic Organic Chemical Nomenclature. 6. (Semi)-automatic Name Correction. *J. Chem. Inf. Comput. Sci*. **1991**, *31*, 153−160.
(15) International Union of Pure and Applied Chemistry. *A Guide to IUPAC Nomenclature of Organic Chemistry, Recommendations 1993*; Panico, R., Powell, W. H., Richer, J. C., Eds.; Blackwell Science: Oxford, U.K., 1993.
(16) International Union of Pure and Applied Chemistry. *Nomenclature of Inorganic Chemistry, Recommendations 1990*; Leigh, G. J., Ed.; Blackwell Science: Oxford, U.K., 1990.
(17) International Union of Biochemistry and Molecular Biology. *Biochemical Nomenclature and Related Documents*: *A Compendium*, 2nd ed.; Liebecq, C., Ed.; Portland Press: London, 1992.
(18) International Union of Pure and Applied Chemistry. *Nomenclature of Organic Chemistry*; Rigaudy, J., Klesney, S. P., Eds.; Pergamon: Oxford, U.K., 1979; Sections A-F and H.
(19) A comprehensive bibliography of IUPAC nomenclature publications is beyond the scope of this paper. Such a list is available, however, at the location specified in the next reference.
(20) International Union of Pure and Applied Chemistry Recommendations on Organic & Biochemical Nomenclature, Symbols & Terminology etc. http://www.chem.qmw.ac.uk/iupac/ (accessed September 1999).
(21) International Union of Pure and Applied Chemistry. *A Guide to IUPAC Nomenclature of Organic Chemistry, Recommendations 1993*; Panico, R., Powell, W. H., Richer, J. C., Eds.; Blackwell Science: Oxford, U.K., 1993; Recommendation R-0.1.8.3; pp 10−11.
(22) Chemical Abstracts Service, American Chemical Society. Chemical Substance Index Names; Appendix IV in the 1997 *Index Guide*; Chemical Abstracts Service: Columbus, OH, 1997; ¶121; p 246I.
(23) International Union of Pure and Applied Chemistry. *A Guide to IUPAC Nomenclature of Organic Chemistry, Recommendations 1993*; Panico, R., Powell, W. H., Richer, J. C., Eds.; Blackwell Science: Oxford, U.K., 1993; Recommendations R-2.3.3.1−R-2.3.3.1.3; pp 40−43.
(24) Chemical Abstracts Service, American Chemical Society. Chemical Substance Index Names; Appendix IV in the 1997 *Index Guide*; Chemical Abstracts Service: Columbus, OH, 1997; ¶146; pp 2591−260I.
(25) Chemical Abstracts Service, American Chemical Society. Chemical Substance Index Names; Appendix IV in the 1997 *Index Guide*; Chemical Abstracts Service: Columbus, OH, 1997; p 241I.
(26) International Union of Pure and Applied Chemistry. *A Guide to IUPAC Nomenclature of Organic Chemistry, Recommendations 1993*; Panico, R., Powell, W. H., Richer, J. C., Eds.; Blackwell Science: Oxford, U.K., 1993; p xiv.
(27) *The ACS Style Guide*: *A Manual for Authors and Editors*, 2nd ed.; Dodd, J. S., Ed.; American Chemical Society: Washington, DC, 1997; pp 56−67.
(28) Helson, H. E. Structure Diagram Generation. In *Reviews in Computational Chemistry*, Lipkowitz, K. B., Boyd, D. B., Eds.; Wiley-VCH: New York, 1999; Vol. 13, pp 313−398.

CI990062C