A Survey and Classification of Controlled Natural Languages

Tobias Kuhn*
ETH Zurich and University of Zurich

*What is here called **controlled natural language (CNL)** has traditionally been given many different names. Especially during the last four decades, a wide variety of such languages have been designed. They are applied to improve communication among humans, to improve translation, or to provide natural and intuitive representations for formal notations. Despite the apparent differences, it seems sensible to put all these languages under the same umbrella. To bring order to the variety of languages, a general classification scheme is presented here. A comprehensive survey of existing English-based CNLs is given, listing and describing 100 languages from 1930 until today. Classification of these languages reveals that they form a single scattered cloud filling the conceptual space between natural languages such as English on the one end and formal languages such as propositional logic on the other. The goal of this article is to provide a common terminology and a common model for CNL, to contribute to the understanding of their general nature, to provide a starting point for researchers interested in the area, and to help developers to make design decisions.*

1. Introduction

Controlled, *processable*, *simplified*, *technical*, *structured*, and *basic* are just a few examples of attributes given to constructed languages of the type to be discussed here. We will call them **controlled natural languages (CNL)** or simply **controlled languages**. Basic English, Caterpillar Fundamental English, SBVR Structured English, and Attempto Controlled English are some examples; many more will be presented herein. This article investigates the nature of such languages, provides a general classification scheme, and explores existing approaches.

As the variety of attributes suggests, there is no general agreement on the characteristic properties of CNL, making it a very fuzzy term. There are two main reasons for this. First, CNL approaches emerged in different environments (industry, academia, and government), in different disciplines (computer science, philosophy, linguistics, and engineering), and over many decades (from the 1930s until today). People from different backgrounds often used and continue to use different names for the same kind of language. Second, although controlled natural languages seem to share important

* Chair of Sociology, in particular of Modeling and Simulation, ETH Zurich, and Institute of Computational Linguistics, University of Zurich. E-mail: kuhntobias@gmail.com. Personal Web site: http://www.tkuhn.ch.

Submission received: 26 October 2012; revised version received: 7 March 2013; accepted for publication: 25 April 2013.

doi:10.1162/COLI_a_00168

properties, they also exhibit a very wide variety: Some are inherently ambiguous, others are as precise as formal logic; virtually everything can be expressed in some, only very little in others; some look perfectly natural, others look more like programming languages; some are defined by just a handful of grammar rules, others are so complex that no complete grammar exists. This variety makes it difficult to get a clear picture of the fundamental properties. This article aims at resolving this problem by giving an overview of existing CNLs and by providing a general classification scheme. Generally, this work has several, partly overlapping, goals, ranging from purely theoretical to more practical objectives (listed in this order):

- To give us a better understanding of the nature of CNL
- To establish a common terminology and a common model for CNL
- To provide a starting point for researchers interested in CNL
- To help CNL developers make design decisions

Although a wide variety of CNLs have been applied to a wide variety of problem domains, virtually all of them seem to be relevant to the field of computational linguistics. Among other techniques, they involve lexical analyses, grammar and style checking, ambiguity detection, machine translation, and computational semantics.

Unsurprisingly, most CNLs are based on English. For the sake of simplicity, the survey presented in this article is restricted to these languages and excludes existing approaches based on other natural languages, such as German and Chinese. The classification scheme to be presented, however, is general and not restricted to English in any way.

In what follows, the relevant background is discussed (Section 2), a classification scheme is introduced (Section 3), existing English-based CNLs are classified and described based on a small sample (Section 4), the results are analyzed (Section 5), and finally the conclusions are discussed (Section 6). The appendix shows the full list of languages with short descriptions for each of them.

2. Background

Controlled natural language being such a fuzzy term, it is important to clarify its meaning, to establish a common definition, and to understand the differences in related terms. In addition, it is helpful to review previous attempts to classify and characterize CNLs.

2.1 Definition

As mentioned earlier, there is no generally agreed-upon definition for controlled natural language and for closely related terms including controlled language, constrained natural language, simplified language, and controlled English. The following two quotations illustrate this:

> A controlled language (CL) is a restricted version of a natural language which has been engineered to meet a special purpose, most often that of writing technical documentation for non-native speakers of the document language. A typical CL uses a well-defined subset of a language's grammar and lexicon, but adds the terminology needed in a technical domain. (Kittredge 2003, page 441)

> Controlled natural language is a subset of natural language that can be accurately and efficiently processed by a computer, but is expressive enough to allow natural usage by non-specialists. (Fuchs and Schwitter 1995, page 1)

Both descriptions exhibit a strong bias towards one particular type of CNL (these types are discussed in more detail subsequently): The first quotation focuses on technical languages that are designed to improve comprehensibility, whereas the second one only covers languages that can be interpreted by computers. They agree, however, on the fact that a CNL is based on a certain natural language but is more restrictive. It is also generally agreed that CNLs are constructed languages, which means languages that did not emerge naturally but have been engineered. The use of the term *subset* is misleading though, because many CNLs are not proper subsets of the underlying natural language. Many of these languages have small deviations from natural grammar or semantics. Others make use of unnatural elements such as colors and parentheses to increase readability and precision. Some even consider the programming language COBOL a controlled natural language (Sowa 2000a). The subset relation in its mathematical sense is clearly too strict to cover a large part of the languages commonly called CNL. Although they all clearly share important properties, the specific languages can be quite different in their coverage and nature. It is not surprising that O'Brian (2003), who compared English-based CNLs of different types, came to the conclusion that no common core language can be identified. To meet these problems, the following definition is proposed here:

Definition 1 (long)
A language is called a **controlled natural language** if and only if it has all of the following four properties:

1. It is based on exactly one natural language (its "base language").
2. The most important difference between it and its base language (but not necessarily the only one) is that it is more restrictive concerning lexicon, syntax, and/or semantics.
3. It preserves most of the natural properties of its base language, so that speakers of the base language can intuitively and correctly understand texts in the controlled natural language, at least to a substantial degree.
4. It is a constructed language, which means that it is explicitly and consciously defined, and *is not* the product of an implicit and natural process (even though it is based on a natural language that *is* the product of an implicit and natural process).

Properties 2 and 3 are deliberately vague, because it is not possible or desirable to draw a strict line there. Properties 1 and 3 refer to the N in CNL: naturalness; Properties 2 and 4 refer to the C: control. We will later be able to be a little more precise concerning Property 3. We leave it for now, and we can summarize this relatively verbose definition in the form of the following short version:

Definition 2 (short)
A **controlled natural language** is a constructed language that is based on a certain natural language, being more restrictive concerning lexicon, syntax, and/or semantics, while preserving most of its natural properties.

As a further remark, we should note that the term *language* is used in a sense that is restricted to sequential languages and excludes visual languages such as diagrams and the like. We can verify that Definition 2 includes virtually all languages that have been called CNL, and it excludes natural languages (because they are not constructed), languages such as Esperanto (because they are not based on one particular natural language), and common formal languages (because they lack intuitive understandability).

2.2 Related Terms

Before we move on to examine the types and properties of languages, we should discuss a number of terms that are related to CNL and are easy to confuse: sublanguage, fragments of language, style guide, phraseology, controlled vocabulary, and constructed language.

Sublanguages are languages that naturally arise when "a community of speakers (i.e., 'experts') shares some specialized knowledge about a restricted semantic domain [and] the experts communicate about the restricted domain in a recurrent situation, or set of highly similar situations" (Kittredge 2003, page 432). As with controlled natural language, a sublanguage is based on exactly one natural language and is more restricted. The crucial difference between the two terms is that sublanguages emerge naturally, whereas CNLs are explicitly and consciously defined.

Fragments of language is a term denoting "a collection of sentences forming a naturally delineated subset of [a natural] language" (Pratt-Hartmann and Third 2006). The term is closely related to CNL and the difference seems to be mainly a methodological one: Fragments of language are *identified* rather than *defined*, they are closely kept in the context of the full natural language and related fragments, and their purpose is rather to theoretically study them than to directly use them to solve a particular problem. A CNL can be seen as a fragment of a language "developed for the purpose of supporting some technical activity" (Pratt-Hartmann 2009, page 1).

Style guides are documents containing instructions on how to write in a certain natural language. Some style guides such as "How to Write Clearly" (European Commission 2011) provide "hints, not rules" and therefore do not describe a new language, but only give advice on how to use the given natural language. However, other style guides such as the Plain Language Guidelines (PLAIN 2011) are stricter and *do* describe a language that is not identical to the respective full language. The question of whether such a language can be considered a CNL depends on whether the style guide defines a new language or whether it merely describes good practices that have emerged naturally.

Phraseology is a term that denotes a "set of expressions used by a particular person or group" (Houghton Mifflin Harcourt 2000). Typically, this term is used when the grammatical structure is simpler than in full natural language. In contrast to sublanguages and fragments of languages, a phraseology is not a selection of sentences but a selection of *phrases*. Phraseologies can be natural or constructed, and in the latter case they are usually considered CNLs.

Controlled vocabularies are standardized collections of names and expressions, including "lists of controlled terms, synonym rings, taxonomies, and thesauri" (ANSI/NISO 2005). Mostly, controlled vocabularies target a specific, narrow domain. In contrast to CNL, they do not deal with grammatical issues, that is, how to combine the terms to write complete sentences. Many CNL approaches, especially domain-specific ones, include controlled vocabularies.

Constructed languages (or **artificial languages** or **planned languages**) are languages that did not emerge naturally but have been consciously defined. In this broad sense, the term includes (but is not limited to) languages such as Esperanto, programming languages, and CNLs.

2.3 Types and Properties

Let us now turn to the nature of CNLs. To bring order to their seemingly chaotic variety, more than 40 properties of such languages and their environments have been identified (Wyner et al. 2010). Many of these properties, however, are fuzzy and do not allow for a strict categorization. For the survey to be presented in Section 4, we collect nine general and clear-cut properties and give them letter codes. As it turns out, however, these properties mainly describe the application environment of languages and not so much the languages themselves. For that reason, a classification scheme is introduced in the next section to describe the fundamental nature of CNLs and other languages.

In general, controlled natural languages can be roughly subdivided according to the problem they are supposed to solve (Schwitter 2002): to improve communication among humans, especially speakers with different native languages (we will use the letter code C for these languages); to improve manual, computer-aided, semi-automatic, or automatic translation (T); and to provide a natural and intuitive representation for formal notations (F). The last type includes approaches for automatic execution of texts, which requires, at least conceptually, a mapping to an executable formalism. As we will see, these three types emerged at different points in time: Type C is the oldest, type T emerged later, and type F is the most recent of the three. Although this seems to be a sensible and useful subdivision, a simpler version based on just two types dominates the literature. Huijsen (1998) introduced the distinction between "human-oriented" and "computer-oriented" languages. The former roughly corresponds to type C, the latter to the types T and F. However, Huijsen observes that "it is often difficult to qualify a controlled language as either human-oriented or machine-oriented, since often simplification works both ways" (page 2). Because these types describe *problems* rather than *languages*, reusing a language in a different problem domain can change its type even if the language itself has not changed at all. Other similar categorizations include the distinction between "naturalistic" (type C and T) and "formalistic" (type F) languages (Pool 2006; Clark et al. 2010) and the distinction between readability and translatability (Reuther 2003).

Another apparent fact is that some languages originated from academia (letter code A), some from industry (I), some from a government or a UN agency (G), and others from a combination of the three. In addition, the distinction between general purpose languages and those for a particular restricted domain is often discussed (Pool 2006). This is related to the distinction of whether the lexicon is open or closed (Adriaens and Schreors 1992). We will use the letter code D to denote languages targeting a specific and narrow domain. A further important difference is the one between written and spoken languages. We will use W to denote languages that are intended to be written, and S for those that are intended to be spoken. However, none of these distinctions seems to describe a fundamental language property: Languages that originated in one environment can later be used in another; the lexicon can later be declared open or closed; written languages can be read aloud; and spoken languages can be written down.

The rules that define a CNL can be proscriptive or prescriptive (Nyberg, Mitamura, and Huijsen 2003), or a combination of the two. Proscriptive rules describe what is

not allowed, whereas prescriptive rules describe what *is* allowed. Languages defined by proscriptive rules alone must have some starting point in the form of a given (natural) language. Languages with only prescriptive rules, in contrast, typically start from scratch. As we will see, there is a close connection of this distinction to the concept of simplicity as introduced in the next section.

Because of their lack of generality, we do not include here more specific low-level properties such as the support for subclauses and free compounding (Adriaens and Schreors 1992), specific restrictions on grammatical tenses and modal verbs (O'Brien 2003), and support for interrogative and imperative sentences (Wyner et al. 2010).

Table 1 summarizes the letter codes. Any two of these properties can overlap, and therefore any combination is possible in theory (with the exception that no language should be neither W nor S).

Finally, there is one additional aspect of constructed languages that deserves attention: their life cycle. Some languages are not much more than abstract ideas, others have left this stage being applied to concrete problems, and yet others have progressed to widespread application in productive environments. At different stages of maturity, languages can be discontinued or abandoned, which signifies the end of their life cycle. Obviously, these different stages flow into each other and it is often difficult to name a concrete year of birth or death (especially the latter, as most CNLs die silently). Where possible, we will keep track of these life cycle properties.

3. PENS Classification Scheme

As we have seen, the CNL properties introduced here describe application domains rather than the languages themselves. Certainly, several fundamental language properties have been identified and discussed in the literature, such as expressiveness (Mitamura and Nyberg 1995; Boyd, Zowghi, and Farroukh 2005; Pool 2006), complexity (Mitamura and Nyberg 1995), grammar modifications (Pool 2006), understandability, natural look-and-feel, ambiguity, predictability, and formality of definition (Wyner et al. 2010). However, these properties are all very fuzzy and do not allow for strict categorization.

To construct a principled classification scheme for such fundamental language properties, it makes sense to condense them to a few dimensions that are to a large degree (though not entirely) independent of each other. Ambiguity, predictability, and formality of definition can be subsumed by a dimension that we can call **precision**.

Table 1
Letter codes for properties of CNLs.

Code	Property
C	The goal is comprehensibility
T	The goal is translation
F	The goal is formal representation (including automatic execution)
W	The language is intended to be written
S	The language is intended to be spoken
D	The language is designed for a specific narrow domain
A	The language originated from academia
I	The language originated from industry
G	The language originated from a government

Expressiveness can make up a second dimension. Grammar modifications, understandability, and natural look-and-feel can be combined to a dimension of **naturalness**. A fourth dimension can be called complexity or—to have a dimension of the type "more is better"—**simplicity**. This is how we arrive at the four dimensions Precision, Expressiveness, Naturalness, and Simplicity that underlie the **PENS** classification scheme.[1]

It seems that all fundamental language properties mentioned in the existing literature fall into one of these general dimensions, or can be broken down into different aspects that can be mapped to these dimensions. There are no strong dependencies between any two dimensions (for any dimension pair, it is easy to imagine languages that are at the top, bottom, and opposite ends in these two dimensions). Furthermore, there is no obvious dimension pair that could be merged in a meaningful way. Together, this seems to indicate that this set of dimensions is minimal yet complete.

The development of this scheme originated from the insight that CNLs can be conceptually located somewhere in the gray area between natural languages on the one end and formal languages on the other. Generally, CNLs are more formal than natural languages but more natural than formal ones. For instance, a natural language such as English is very expressive, but complex and imprecise. A formal language such as propositional logic, in contrast, is very simple and precise, but at the same time unnatural and inexpressive. CNLs must be somewhere in the middle, but where exactly?

It seems obvious that all four of the dimensions are continuous in nature or at least very fine-grained. In fact, one can argue that each of the dimensions is actually multidimensional and that representing it in one dimension is a rough simplification. Such simplifications are necessary, however, in order to get a precise measure for such vague concepts such as expressiveness.

Intuitively, PENS uses a natural language such as English and a formal language such as propositional logic as pegs to span a conceptual space in which different kinds of controlled natural languages can be placed. In order to get a general but strict classification scheme, PENS drastically simplifies things by restricting each of its four dimensions to five classes, to be numbered from 1 to 5. These five classes are non-overlapping and consecutively cover the one-dimensional space between the two extremes: English on the one end and propositional logic on the other. For precision and simplicity, English is on the bottom end of the scale in class 1, which we write as P^1 and S^1. Propositional logic is on the opposite end of the scale in class 5, represented with P^5 and S^5. For expressiveness and naturalness, the roles are switched: English is at the top end (E^5 and N^5) and propositional logic at the bottom (E^1 and N^1). In this way, the scheme defines a conceptual space for CNLs that includes natural and formal languages as special cases. Combining the four dimensions gives $5^4 = 625$ classes, represented with shorthand such as $P^1E^5N^5S^1$ for English and $P^5E^1N^1S^5$ for propositional logic. The difficult and interesting part of this intellectual exercise is where and how to draw the borders between the five classes of each dimension.

The decision to use five classes for each dimension, and not four or six, is somewhat arbitrary. A larger number of classes allows for more detailed classifications, although it also gets more difficult to come up with strict and objective criteria to define these classes. Five seems to be a good middle ground.

1 These four dimensions had first been sketched as "design principles" in the author's doctoral thesis (Kuhn 2010), where "precision" was called "clearness."

3.1 Precision

The precision dimension of the PENS scheme captures the degree to which the meaning of a text in a certain language can be directly retrieved from its textual form, that is, the sequence of language symbols. Natural language is very imprecise in this sense, because a large amount of context information is needed to grasp the meaning of typical sentences. Formal logic languages, on the other hand, have maximal precision, because their meaning is strictly defined solely on the basis of the possible sequences of their language symbols. The symbol grounding problem, that is, the problem of mapping symbols to their counterparts in the real world, is not considered here, because it affects all languages, including both natural and formal ones. On this precision dimension, languages are divided into the five classes P^1, P^2, P^3, P^4, and P^5, as follows:

Imprecise languages (P^1). Virtually every sentence of these languages is vague to a certain degree. Without taking context into account, most sentences of a certain complexity are ambiguous. The automatic interpretation of such languages is "AI-complete," which means it is a problem for which no complete solutions are in sight. These languages require a human reader to check whether a given statement is syntactically correct, and include borderline statements on which readers disagree. The same applies to the semantic properties of the language. All natural languages belong to this category.

Less imprecise languages (P^2). For these languages, the degree of ambiguity and vagueness is considerably lower than in natural languages, and their interpretation depends much less on context. They restrict the use and/or the meaning of a wide range of the respective ambiguous, vague, or context-dependent constructs. However, these constructs are still too dominant to make automatic interpretation reliable. Such languages are typically not related to a formal (i.e., mathematically precise) underpinning.

Reliably interpretable languages (P^3). The syntax of these languages is heavily restricted, though not necessarily formally defined. The restrictions are strong enough to make automatic interpretation reliable. There is a logical underpinning or at least a formal conceptual scheme, in which the semantics of sentences can be represented. However, the mapping of sentences to their formal representations is itself not defined in a fully formal way, but requires external background knowledge, heuristics, or user feedback.

Deterministically interpretable languages (P^4). Such languages are fully formal on the syntactic level; that is, they are (or can be) defined by a formal grammar. Each text in such a language can be deterministically parsed to a formal logic representation, or a small set of all possible representations (including all and only the possible ones). Based on the underlying formalism, these representations describe the meaning of the sentences, but they may be underspecified in the sense that they require certain parameters, background axioms, external resources, or heuristics to enable sensible deductions.

Languages with fixed semantics (P^5). These languages are fully formal and fully specified on both the syntactic and semantic levels. Each text has exactly one meaning, which can be automatically derived. The circumstances in which inferences hold or do not hold are fully defined. What conclusions follow from a given text in the language (e.g., whether it is consistent and which sentences of the language are a consequence of the text) can be defined with mathematical rigor, without the help of heuristics or external resources.

3.2 Expressiveness

The dimension of expressiveness describes the range of propositions that a certain language is able to express. A language X is more expressive than a language Y if language X can describe everything that language Y can, but not vice versa. The relation of "being more expressive" does not constitute a total order: For two given languages of nonequal expressiveness, it can be the case (and often is the case) that neither is more expressive than the other. This entails that ranking a general set of languages in a linear order according to their expressiveness cannot be done in a completely objective way. A classification scheme, such as the one presented here, must therefore rely on only a subset of all possible expressiveness features. These expressiveness features should be general and important ones, and at the same time allow for a balanced and clear discrimination between the languages to be classified. The PENS classification scheme employs the following five expressiveness features:

(a) universal quantification over individuals (possibly limited)

(b) relations of arity greater than 1 (e.g., binary relations)

(c) general rule structures (*if–then* statements with multiple universal quantification that can target all argument positions of relations)

(d) negation (strong negation or negation as failure)

(e) general second-order universal quantification over concepts and relations

For each of these features to be considered fulfilled, they should be an integral part of the language and not just manifested by a few special cases. There are a number of other important features that could be considered, for example, support for existential quantification, equality, and types of supported speech acts (such as declarative, interrogative, directive, and indirect speech acts). However, to achieve a simple classification into a sequence of five classes, these features will turn out to be sufficient and lead to a classification that seems consistent with the intuitive understanding of expressiveness.

Because this classification system should not only include declarative formal languages but also informal as well as procedural ones, it makes sense to apply a weaker notion of expressiveness than what is usually applied to logic languages. From the research on programming languages, we can adopt the convention that a certain language construct adds expressiveness if its removal would require "a global reorganization of the entire program" (Felleisen 1991). If a certain language construct allows us to express something locally which would otherwise require us to reorganize the entire text, then we say that this language construct makes the language more expressive. This means, for example, that a language with second-order features relying on Henkin semantics qualifies for the last criterion of the above list, even though Henkin semantics can be reduced to first-order. A given set of statements written in a language with Henkin-style second-order features cannot generally be reduced to first-order logic without global reorganization, that is, changing statements that do not actually use second-order features. With this qualification, we can define the five classes as follows:

Inexpressive languages (E^1). These are languages lacking one or both of the features (a) and (b): They have no universal quantification or no relations of arity greater than 1. Propositional logic belongs to this category.

Languages with low expressiveness (E^2). Such languages have both of the features (a) and (b), but are not E^3-languages: They have universal quantification over individuals and relations of arity greater than 1. Description logics belong to this category.

Languages with medium expressiveness (E^3). These languages have all of the features (a), (b), (c), and (d), but are not E^4-languages: They have general rule structures and negation, in addition to the features of E^2. First-order logic belongs to this category.

Languages with high expressiveness (E^4). Such languages have all listed features (a), (b), (c), (d), and (e), but are not E^5-languages: They have second-order universal quantification over concepts and relations, in addition to the features of E^3. Second-order predicate calculus belongs to this category.

Languages with maximal expressiveness (E^5). These languages can express anything that can be communicated between two human beings. Such languages cover any statement in any type of logic. Obviously, this includes all of the features. All natural languages belong to this category.

3.3 Naturalness

The dimension of naturalness describes how close the language is to a natural language in terms of readability and understandability to speakers of the given natural language. We define the five classes as follows:

Unnatural languages (N^1). These are languages that do not look natural, making heavy use of symbol characters, brackets, or unnatural keywords. It might be possible to use natural words or phrases as names for certain entities, but this is neither required nor further defined by the language.

Languages with dominant unnatural elements (N^2). Natural language words or phrases are an integral part of such languages, but are dominated by unnatural elements or unnatural statement structure, or have unnatural semantics. The natural elements do not connect in a natural way to each other, and speakers of the given natural language typically fail to intuitively understand the respective statements.

Languages with dominant natural elements (N^3). In such languages, natural elements are dominant over unnatural ones and the general structure corresponds to natural language grammar. Due to the remaining unnatural elements or unnatural combination of elements, however, the sentences cannot be considered valid natural sentences. Speakers of the given natural language do not recognize the statements as well-formed sentences of their language, but are nevertheless able to intuitively understand them to a substantial degree.

Languages with natural sentences (N^4). These are languages with sentences that can be considered valid natural sentences. Speakers of the respective natural language recognize the statements as sentences of their language and are able to correctly understand their essence without instructions or training. Minor or infrequent exceptions and unnatural means for clarification (including text color, indentation, hyphenation, and capitalization) are permitted as long as they do not disturb the natural look-and-feel

and the natural flow of the sentence. Parentheses and brackets in unnatural positions, however, in most cases *do* disturb the natural text flow considerably, and are therefore typically not present in this category. Although single sentences have a natural flow, this does not scale up to complete texts or documents. Complete texts in such languages seem very clumsy and repetitive, and lack a natural text flow.

Languages with natural texts (N^5). With these languages, complete texts and documents can be written in a natural style, with a natural text flow, and with natural semantics. In the case of spoken languages, complete dialogs can be produced with a natural flow and a natural combination of speech acts.

We can now be a little more precise concerning our definition of CNL. Property number 3 of the long version of the definition shown in Section 2.1 says that a CNL "preserves most of the natural properties of its base language, so that speakers of the base language can intuitively and correctly understand texts in the controlled natural language, at least to a substantial degree." We will interpret this in such a way that it only includes languages of naturalness N^3 and higher. Thus, by this definition, there are no CNLs with N^1 or N^2.

3.4 Simplicity

The fourth dimension is a measure of the simplicity or complexity of an exact and comprehensive language description covering syntax and semantics, if such a complete description is possible at all. This description should not presuppose intuitive knowledge about any natural language. It is therefore not primarily a measure for the effort needed by a human to learn the language, neither does it capture the theoretical complexity of the language (as, for example, the Chomsky hierarchy does). Rather, it is closely related to the effort needed to fully implement the syntax and the semantics of the language in a mathematical model, such as a computer program.

The PENS scheme applies a very pragmatic and simple indicator for simplicity: The number of pages in natural language needed to describe the language in an exact and comprehensive way. For languages for which no such exact and comprehensive descriptions exist or can be written (that do not presuppose linguistic knowledge on the side of the reader, and given the current state of science), we can distinguish languages with the complexity of natural language from languages with considerably lower complexity.

These "exact and comprehensive descriptions" should define all syntactic and semantic properties of the language using accepted grammar notations to define the syntax and accepted mathematical or logical notations to define the semantics. They are assumed to use scientific writing style as found in scientific articles or technical reports, and should allow a skilled grammar engineer to implement a correct and complete parser within a reasonable time. The page count should be based on a one-column format with up to about 700 words per page. It is important to note that the criterion is not the *presence* of such a description but whether it is *possible or not* to write one.

In order to treat languages with fixed vocabularies and those with extensible ones in an equal way, these language descriptions do not need to include the vocabularies. Concretely, the five classes are defined as follows:

Very complex languages (S^1). These languages have the complexity of natural languages. They cannot be described in an exact and comprehensive manner.

Languages without exhaustive descriptions (S^2). These are languages that are considerably simpler than natural languages, in the sense that a significant part of the complex structures are eliminated or heavily restricted. Still, they are too complex to be described in an exact and comprehensive manner. Usually, the definitions of such languages just describe restrictions on top of a given natural language that is taken for granted.

Languages with lengthy descriptions (S^3). Such languages can be defined in an exact and comprehensive manner, but it requires more than ten pages to do so.

Languages with short descriptions (S^4). These are languages for which an exact and comprehensive description requires more than one page but not more than ten pages.

Languages with very short descriptions (S^5). Such very simple languages can be described in an exact and comprehensive manner on a single page.

S^1 and S^2 are considered complex because they rely on a given natural language. Coming back to a distinction briefly introduced in the previous section, such languages are typically defined by *proscriptive* rules, describing what is not allowed compared with the full language. S^3, S^4, and S^5, in contrast, typically use *prescriptive* rules that define the language from scratch. For that reason, they are simpler in our sense of the word than languages of the first type, which "import" the complexity of full natural language.

Before we move on to apply this scheme, it should be stressed that PENS is designed to measure the *nature* of a language, not its *quality* or *usefulness*. It should be used to *describe* languages, not to *rank* them. As the "perfect" language does not exist, compromises have to be made. Depending on application area, environment, and goal, different weights are assigned to the PENS dimensions, and therefore different optimal levels result. In theory, more is better for each of the PENS dimensions, but this does not necessarily hold in practice. A certain level in any of the dimensions is often good enough for a given application domain, and going beyond that level brings no additional benefit. Furthermore, as we restrict ourselves to just five classes per dimension, there can be relatively large differences *within* one class. It is inevitable that two languages in the same class can be farther apart in the respective dimension than two languages in adjacent classes. Even if a language has higher PENS values in every dimension than another language, this does not mean that the former is "better" in any meaningful sense of the word. Having a high PENS score for expressiveness, for example, just means that the general expressiveness level is high, and not that the language is able to express each and every statement of all languages with a lower score. Similarly, having a high score for naturalness does not mean that all aspects of the language are more natural as compared to all languages with a lower score.

4. Languages

We can now turn to the actual survey. For practical reasons, we restrict ourselves here to English-based languages, leaving out CNLs that are based on other languages, such as Chinese, French, German, Greek, Spanish, and Japanese (Pool 2006). To give an overview of the different existing English-based CNLs, twelve important and influential languages are introduced here. The complete list can be found in the appendix; surprisingly, we ended up with exactly 100 languages. In addition, a handful of other languages for comparison are introduced in the following, such as natural English

and propositional logic. Each language is classified according to the nine properties with letter codes and the PENS scheme. A best guess is made in the cases where not enough information is available. The descriptions in the appendix are shorter in the case of similar languages or scarce information. This data set is also available online as a CSV table.[2]

There are many user interface approaches based on some sort of natural language input, and it could be argued that they all—at least indirectly—define and use a controlled language, because none of them is able to correctly process full natural language. Such approaches, however, are included here only if the restrictions on the language are considered an inherent property of the approach and not a shortcoming of its implementation. In other words, the following listing excludes languages whose restrictions are not design decisions of the general approach but practical concessions (e.g., Warren and Pereira 1982). The same criterion is applied to verbalization approaches, which inevitably define a restricted version of the respective language that could be considered a CNL (e.g., Halpin 2004; Jarrar, Keet, and Dongilli 2006; Lukichev and Wagner 2006). Other languages follow an approach called conceptual authoring or WYSIWYM (Hallett, Scott, and Power 2007) where texts are created by short cycles of language generation and user-triggered modification actions. We include such languages here, because in this case the restrictions on the language are an important aspect of the approach. Finally, it should be mentioned that we leave out fictional languages, such as Newspeak of George Orwell's *Nineteen Eighty-Four*.

Languages that do not have an official name are introduced by a "generic name in quotation marks." Unless stated otherwise, quotes and examples are taken from the publications cited in the beginning of each paragraph.

4.1 English-Based Controlled Languages

Below, twelve selected CNLs are introduced, roughly in chronological order of their first appearance or the first appearance of similar predecessor languages. For this small sample, languages are chosen that were influential, are well-documented, and/or are sufficiently different from the other languages of the sample.

"Sowa's syllogisms" (Sowa 2000b) are simple logic languages based on the syllogisms originally introduced by Aristotle (ca. 350 BC). Sowa was probably the first to bring them into the context of CNL, claiming that they are the first reported instance of a controlled natural language. Because this survey is restricted to English, Sowa's version of the syllogisms is listed here instead of Aristotle's original version in ancient Greek. The complete language can be described by just four simple sentence patterns:

Every A is a B. Some A is a B. No A is a B. Some A is not a B.

A and B can be any English common nouns such as *cat* and *animal*. This language is very similar to the language $\mathcal{E}_0$ presented and studied by Pratt-Hartmann (2004), who used some additional patterns:

Every A is not a B. No A is not a B. P is a B. P is not a B.

Here, P can be any English proper name such as *Socrates*. We will use the term "Sowa's syllogisms" in a sense that includes such similar approaches. The semantics

2 `http://purl.org/tkuhn/cnlsurvey/data`.

of syllogisms is also very easy to define. The first four patterns can be mapped to first-order logic like this (and similarly for the other patterns):

$$\forall x(\, A(x) \rightarrow B(x)\,) \quad \exists x(\, A(x) \wedge B(x)\,) \quad \neg\exists x(\, A(x) \wedge B(x)\,) \quad \neg\forall x(\, A(x) \rightarrow B(x)\,)$$

Hereby, we have an exact and comprehensive description of the language, taking just a couple of lines. Despite the simple structure of the language, the sentences are perfectly natural. Its expressiveness, however, is very restricted: Only very simple sentence structures are covered and only one-place relations are supported. — $P^5E^1N^4S^5$, F W A

Basic English (Ogden 1930) is a language presented in 1930 that should improve communication among people around the globe. It is the first reported instance of a controlled version of English, at least the first one that received broader recognition. It influenced Caterpillar Fundamental English, which became itself a very influential language. Basic English was designed as a common basis for communication in politics, economy, and science. It restricts the grammar and makes use of only 850 English root words. The restrictions are arguably most drastic in the case of verbs. Only 18 verbs are supported: *put*, *take*, *give*, *get*, *come*, *go*, *make*, *keep*, *let*, *do*, *be*, *seem*, *have*, *may*, *will*, *say*, *see*, and *send*. These verbs can be combined with prepositions to form more specific relations such as *put in* to express *insert*. Other verbs can be expressed with the help of nouns, such as *give a move* instead of using *move* as a verb. The usage of the given words and their variants is described by informal grammar rules, for example, "Collective nouns may be formed from adjectives when used with *the*." These are two examples of sentences in Basic English:

> The camera man who made an attempt to take a moving picture of the society women, before they got their hats off, did not get off the ship till he was questioned by the police.

> It was his view that in another hundred years Britain will be a second-rate power.

Many variations exist that use larger word sets. The Simple English version of Wikipedia,[3] for example, claims to use Basic English, but in fact uses a much less restricted language. Basic English is still used today and promoted by a dedicated Basic-English Institute.[4] Many texts have been written in this language, including textbooks, novels, and large parts of the bible. The drastic simplifications on the lexical level together with the grammatical restrictions constitute a significant gain in precision compared with full English. Still, any type of topic can be expressed with a natural text flow. The informal restrictions on the grammar, however, are not strong enough to significantly reduce the complexity of the language (in the PENS sense of complexity). — $P^2E^5N^5S^1$, C W

E-Prime or **E′** (Bourland 1965) is a restricted version of English with the only restriction being that the verb *to be* is forbidden to use. This includes all inflectional forms such as *are*, *was* and *being*, regardless of whether used as auxiliary or main verbs. The language was presented in 1965 but the idea goes back to the late 1940s. The motivation for the use of E-Prime is the belief that "dangers and inadequacies . . . can result from the careless, unthinking, automatic use of the verb *to be*." E-Prime is claimed by its proponents to enhance clarity. The statement *We do this because it is right* would

3 http://simple.wikipedia.org.
4 http://www.basic-english.org.

not be allowed, but one would have to rephrase it in a way that does not include *to be*, for example:

> We do this thing because we sincerely desire to minimize the discrepancies between our actions and our stated "ideals."

In the area of natural language processing, however, the verb *to be* is *not* considered one of the most difficult problems, which is good evidence that E-Prime is not considerably more precise than full English in the PENS sense. Also, in terms of complexity it is not considerably different from full English, because words such as *become* and *exist* are allowed that can replace the forbidden *to be* in most cases. On the other hand, it seems true that it is always possible to rephrase a text without the use of *to be* in a way that is fully natural though possibly longer than the original. — $P^1E^5N^5S^1$, C W A

Caterpillar Fundamental English (CFE) (Verbeke 1973) was an influential controlled language developed at Caterpillar. It was officially introduced in 1971, was based on Basic English (Smart 2003), and has been reported to be the earliest industry-based CNL (Wojcik and Hoard 1997). The need for a controlled language emerged because of the increasing sophistication of Caterpillar's products and the need to communicate with non-English speaking service personnel in different countries (Verbeke 1973): "To summarize the problem: There are more than 20,000 publications that must be understood by thousands of people speaking more than 50 different languages." The idea of CFE was "to eliminate the need to translate service manuals" (Kamprath et al. 1998). A trained, non-English speaking mechanic familiar with Caterpillar's products should be able to understand the language after completing a course on CFE consisting of 30 lessons. The vocabulary of the language is restricted to around 800 to 1,000 words (Crabbe 2009), with only one meaning defined for each of them (e.g., *right* only as the opposite of *left*). Still, many of the words "had broad semantic scope and it was assumed that they would be disambiguated in context by the human reader" (Kamprath et al. 1998). The following ten rules summarize the grammatical restrictions (Crabbe 2009):

1. Make positive statements.
2. Avoid long and complicated sentences.
3. Avoid too many subjects in one sentence.
4. Avoid too many successive adjectives and nouns.
5. Use uniform sentence structures.
6. Avoid complicated past and future tenses.
7. Avoid conditional tenses.
8. Avoid abbreviations, contractions, and colloquialisms.
9. Use punctuation correctly.
10. Use consistent nomenclature.

These are two examples of CFE sentences:

> The maximum endplay is .005 inch.

> Lift heavy objects with a lifting beam only.

CFE was discontinued by Caterpillar in 1982, because (among other reasons) "the basic guidelines of CFE were not enforceable in the English documents produced" (Kamprath et al. 1998). As a result, Caterpillar Technical English (see appendix) was developed following a different approach: The restrictions on the language should be enforceable, and should reduce translation costs instead of trying to eliminate the need for translations altogether. The strong lexical restrictions together with some grammatical constraints make CFE more precise than full English, but it is not considerably different in terms of expressiveness, naturalness, and complexity. — $P^2E^5N^5S^1$, C W D I

FAA Air Traffic Control Phraseology (FAA 2010) is a controlled language defined by the Federal Aviation Administration (FAA) and used for the communication in air traffic

coordination, going back to at least the early 1980s. There are other very similar languages for air traffic control such as the ICAO and CAA phraseologies. To a large extent, these languages are indistinguishable from each other, and together they are sometimes called **AirSpeak** (Robertson 1987). The FAA Phraseology is defined by more than 300 fixed sentence patterns such as "(ACID), IN THE EVENT OF MISSED APPROACH (issue traffic). TAXIING AIRCRAFT/VEHICLE LEFT/RIGHT OF RUNWAY." This is an example of a statement following that pattern:

> United 623, in the event of missed approach, taxiing aircraft right of runway.

In addition to these explicit patterns, there are many more implicit patterns defined in prose form, for example "Issue advisory information on … bird activity. Include position, species or size of birds, if known, course of flight, and altitude." The following statement is an example that corresponds to this implicit pattern:

> Flock of geese, one o'clock, seven miles, northbound, last reported at four thousand.

Vocabulary and semantics are restricted too, for example "Use the word *gain* and/or *loss* when describing to pilots the effects of wind shear on airspeed." Phraseology statements can be mixed with statements in full English in cases where no pattern exists to express the desired message. The language is heavily restricted and much less ambiguous than full English. It is inexpressive in the sense that no universal quantification is supported, and is not sufficiently restricted to make an exact and exhaustive description feasible. — $P^2E^1N^3S^2$, C S D G

ASD Simplified Technical English (ASD-STE) (ASD 2013), often abbreviated to **Simplified Technical English (STE)** or just **Simplified English**, is a CNL for the aerospace industry. Originally inspired by a language called ILSAM (Adriaens and Schreors 1992), the language had its origins in 1979, but it was only in 1986 when it was officially presented for the first time, then under the name **AECMA Simplified English**. It received its current name in 2004 when AECMA merged with two other associations to form ASD. The main purpose of the language is to make texts easier to understand, especially for non-native speakers. Although AECMA Simplified English was designed to make translation into other languages unnecessary, one of the original goals of ASD-STE was to improve translation. Today, the language is maintained by the Simplified Technical English Maintenance Group. ASD-STE is based on English with restrictions expressed in about 60 general rules. These rules restrict the language on the lexical level (e.g., "Use approved words from the Dictionary only as the part of speech given"), on the syntactic level (e.g., "Do not make noun clusters of more than three nouns"), as well as on the semantic level (e.g., "Keep to the approved meaning of a word in the Dictionary. Do not use the word with any other meaning."). There is a fixed vocabulary consisting of terms common to the aerospace domain. Additionally, user-defined "Technical Names" and "Technical Verbs" can be introduced. This is an exemplary excerpt of a text in ASD-STE:

> These safety precautions are the minimum necessary for work in a fuel tank. But the local regulations can make other safety precautions necessary.

Even though its restrictions make ASD-STE considerably more precise than full English, it does not allow for reliable automatic interpretation. Full expressiveness and full naturalness of unconstrained English are retained, but also its complexity. — $P^2E^5N^5S^1$, C T W D I

Standard Language (SLANG) (Rychtyckyj 2002, 2005) is a language developed at Ford Motor Company starting from 1990. It is designed for process sheets containing build

instructions for component and vehicle assembly plants. It is still used at Ford and has been continually extended and updated to reflect technical and business-related advances. With SLANG, engineers can write instructions that are clear and concise and at the same time machine-readable. Based on these instructions, the system can, among other things, automatically generate a list of required elements and calculate labor times. In addition, the restricted nature of the language is exploited to translate such instructions with the help of machine translation for their use in assembly plants in different countries. All SLANG sentences are in imperative mood and follow a certain general pattern starting with a main verb and followed by a noun phrase. There are additional restrictions on vocabulary and semantics. These are two exemplary sentences:

> OBTAIN ENGINE BLOCK HEATER ASSEMBLY FROM STOCK
>
> APPLY GREASE TO RUBBER O-RING AND CORE OPENING

A parser is used to check for compliance with the restrictions. English grammar is followed with some minor deviations: For example, articles can be dropped and some kinds of modifiers can be used in unnatural ways. — $P^3E^1N^4S^2$, C F W D I

SBVR Structured English (OMG 2008) is a CNL for business rules first presented around 2005. It is part of the Semantics of Business Vocabulary and Business Rules (SBVR) standard. It was probably influenced by a language called RuleSpeak that is very similar and was first presented in 1994. The vocabulary is extensible and consists of four types of sentence constituents: terms (i.e., concepts), names (i.e., individuals), verbs (i.e., relations), and keywords (e.g., fixed phrases, quantifiers, and determiners). Each of these has its own color and style, as the following examples show:

> A rental must *be guaranteed by* a credit card *before* a car *is assigned to* the rental.
>
> **Rentals by Booking Mode** *contains* the categories 'advance rental' and 'walk-in rental.'

The SBVR standard provides formal semantics based on second-order logic with Henkin semantics. The second of the examples makes use of the second-order features. Some keywords have a precise meaning, such as *or* meaning inclusive logical disjunction (unless followed by *but not both*). Other keywords, however, are less precise, such as the determiner *a* being defined as "universal or existential quantification, depending on context based on English rules." The language strictly defines the permissible sentence constituents, but is much less strict in defining the order in which these constituents can be put. The syntax structure can be ambiguous (e.g., when using *and* and *or* in the same sentence), and so can be quantifier scopes and anaphoric references. There is no formal grammar of the language, and its definition depends to some degree on the linguistic understanding of a human reader. — $P^3E^4N^4S^2$, C F W I

Attempto Controlled English (ACE) (Fuchs, Kaljurand, and Kuhn 2008) is a CNL with an automatic and unambiguous translation into first-order logic. ACE was first presented in 1996 as a language for software specifications. Later, the focus shifted towards knowledge representation and the Semantic Web. The language has been extended over the years in various ways. The most notable features of ACE include complex noun phrases, plurals, anaphoric references, subordinated clauses, modality, and questions. These are two exemplary ACE sentences:

> A customer owns a card that is invalid or that is damaged.
>
> Every continent that is not Antarctica contains at least 2 countries.

ACE sentences are deterministically mapped to discourse representation structures, a notational variant of first-order logic. These expressions, however, are underspecified in the sense that many deductions (e.g., when involving plurals or modal verbs) require external background axioms that are not fixed by the ACE definition (these axioms are external in the sense that they are not necessarily expressible in ACE). This makes it possible to use ACE in different areas such as ontology editors, rule systems, and general reasoners with semantics that are not fully compatible. ACE is, with a few minor exceptions, fully natural on the sentence level, but longer texts do not have a natural text flow. Recently, ACE has also been used in the context of rule-based machine translation (Kaljurand and Kuhn 2013), but translation was not a stated goal during the design of the language. — $P^4E^3N^4S^3$, F W A

"Drafter Language" (Power and Scott 1998) is a CNL used in a system called Drafter-II. The language is used for instructions to word processors and diary managers. The system utilizes a conceptual authoring approach: Users cannot directly edit the CNL text, but they can only trigger modification actions starting from a small stub sentence. In this way, incomplete statements are gradually completed by the user. The following example is a sequence of two incomplete statements showing one such completion step:

> Schedule **this event** by applying *this method*.
>
> Schedule the appointment by applying *this method*.

The first sentence has two missing parts: *this event* and *this method*. At this point, the user can choose, for example, *the appointment* to fill in the first missing part, which leads to the second statement, which is still incomplete but has only one missing part left. Once a statement is completed, Drafter-II internally maps it to Prolog expressions, which are then automatically executed. As structural ambiguity can be resolved based on the given sequence of modification actions, languages following the conceptual authoring approach do typically not attempt to fully eliminate structural ambiguity. A given text can have multiple parse trees, only one of which corresponds to the way it was created. — $P^4E^1N^4S^3$, F W D A

E2V (Pratt-Hartmann 2003) is a controlled language that was introduced in 2001 and corresponds to the language $\mathcal{E}_3$ studied in later work (Pratt-Hartmann 2004). The ultimate goal is "to provide useable tools for natural language system specification." E2V deterministically maps to the two-variable fragment of first-order logic. Because of this, satisfiability of E2V sentences and texts is decidable and computation is NEXPTIME complete. Two examples of E2V sentences are shown here:

> Some artist does not despise every beekeeper.
>
> Every artist who employs a carpenter despises every beekeeper who admires him.

The language is defined by 15 simple grammar rules plus nine predefined lexical rules for general words such as *every* and *does not*. A separate, user-defined lexicon contains the domain-specific words such as *artist* and *admires*. Altogether, E2V is a precise, natural, simple, but relatively inexpressive controlled language. — $P^5E^2N^4S^4$, F W A

Formalized-English (FE) (Martin 2002) is a CNL for knowledge representation. It is based on Conceptual Graphs and the Knowledge Interchange Format, and focuses on expressiveness. It covers a wide range of features including general universal quantification, negation, contexts (statements about statements), lambda abstractions,

possibility, collections, intervals, and higher-order statements (reducible to first-order logic). Two examples of statements in FE are shown here (the second one is higher-order):

> At least 93% of [bird with chrc a good health] can be agent of a flight.
>
> If 'a binaryRelationType *rt has for chrc the transitivity' then 'if '^x has for *rt ^y that has for *rt ^z' then '^x has for *rt ^z' '.

FE looks natural for simple statements, but becomes quite unnatural for more complex ones. This is due to unnatural use of parentheses, quotation marks, variables, and keywords such as *chrc*. The syntax of the language is defined by about 50 rules in a parser generator language. — $P^5E^4N^3S^3$, F W A

4.2 Languages for Comparison

For the analysis to be described in the next section, we will use the following languages for comparison, which are *not* CNLs according to our definition:

English is our representative of a natural language. — $P^1E^5N^5S^1$, C W S

Propositional logic is a very basic logic language. — $P^5E^1N^1S^5$, F W A

First-order logic can be considered an extension of propositional logic. It is more expressive, but also more complex. — $P^5E^3N^1S^4$, F W A

COBOL is one of the oldest programming languages, which some call a controlled natural language (Sowa 2000a). This is an exemplary COBOL statement:

> PERFORM P WITH TEST BEFORE VARYING C FROM 1 BY 2 UNTIL C GREATER THAN 10.

Although COBOL uses natural phrases where other programming languages use symbols or short keywords, the statement structure does not really follow natural grammar. For that reason, we do not consider it a CNL. — $P^5E^2N^2S^3$, F W A I G

Manchester OWL syntax (Horridge et al. 2006) is a user-friendly syntax for the ontology language OWL. This is an exemplary expression:

> Pizza and not (hasTopping some FishTopping) and not (hasTopping some MeatTopping)

Instead of logical symbols, natural words such as *not* and *some* are used. The general structure, however, resembles formal and not natural languages, which is why we do not consider it a CNL. — $P^5E^2N^2S^4$, F W A

Naturally, there are many more languages that could be used for comparison, but this list seems to be a good sample.

5. Analysis

The data presented in the previous section and in the appendix allow for different kinds of aggregations and analyses. In particular, the classes and properties of the observed languages and the timeline of their evolution are interesting.

5.1 PENS Classes

Table 2 summarizes the PENS classes and properties of the discussed CNLs. Some interesting patterns can be found in these data. Theoretically, there are $5^4 = 625$ possible PENS classes, but not all of them are observed "in the wild." Some are even practically impossible, as far as we can tell, such as the perfect class $P^5E^5N^5S^5$. The CNLs introduced previously cover 25 distinct classes, which might seem a small number with

Table 2
Observed PENS classes and properties of CNLs (sorted by PENS class).

class	properties	languages
$P^1E^5N^5S^1$	C T W I	IBM's EasyEnglish
	C W S G	Special English
	C W A	E-Prime
	C W G	Plain Language
$P^2E^1N^3S^2$	C S D G	CAA Phraseology, FAA Phraseology, ICAO Phraseology, PoliceSpeak, SEASPEAK
$P^2E^1N^3S^3$	C W D I	Airbus Warning Language
$P^2E^5N^4S^1$	F W A	AIDA
$P^2E^5N^5S^1$	C T W D A I	ALCOGRAM, COGRAM
	C T W D A	CLCM
	C T W D I	ASD-STE, Avaya CE, Bull GE, CTE, CASL, CE at Douglas, DCE, General Motors GE, PACE, Sun Proof
	C T W D	Wycliffe Associates' EasyEnglish
	C T W I	iCE, SMART Controlled English
	C W D I	AECMA-SE, CFE, CASE, CE at Clark, CE at IBM, CE at Rockwell, EE, HELP, ILSAM, KISL, NCR FE
	C W D G	Massachusetts Legislative Drafting Language
	C W I	Boeing Technical English, NSE, SMART Plain English
	C W	Basic English
	T W D I	MCE, Océ Controlled English
	T W A	KCE
	T W I	CLOUT
$P^3E^1N^4S^2$	C F W D I	SLANG
	F S D I	Voice Actions
$P^3E^2N^4S^3$	F W D A	RNLS
$P^3E^3N^3S^3$	F W A	ClearTalk
	F W I	ITA CE
$P^3E^3N^4S^2$	F W I	CPL
$P^3E^4N^4S^2$	C F W I	RuleSpeak, SBVR-SE
$P^4E^1N^4S^3$	F W D A	Drafter Language, MILE Query Language
$P^4E^1N^4S^4$	F W A	Quelo Controlled English
$P^4E^1N^5S^3$	T F D A	PILLS Language
$P^4E^2N^4S^3$	F W D A	Atomate Language
	F W A I	Gellish English
	F W A	GINO's Guided English
	F W I	CELT
$P^4E^3N^4S^3$	F W D A	PROSPER CE
	F W A	ACE
$P^4E^3N^5S^3$	F W D A	ICONOCLAST Language
$P^5E^1N^4S^3$	F W D A	CLEF Query Language
	F W A	Ginseng's Guided English
$P^5E^1N^4S^4$	F W D A	Coral's Controlled English
	F W A	PathOnt CNL
$P^5E^1N^4S^5$	F W A	Sowa's syllogisms
$P^5E^2N^3S^4$	F W D A I	TBNLS
	F W A	OWLPath's Guided English, SQUALL
$P^5E^2N^4S^3$	F W A	CPE, CLIP, OWL ACE, SOS
$P^5E^2N^4S^4$	F W D A	BioQuery-CNL, PERMIS CNL, ucsCNL
	F W A	CLOnE, DL-English, E2V, Lite Natural Language, OSE
	F W G	Rabbit
$P^5E^3N^3S^3$	F W D A	CLM, ForTheL, Naproche CNL
	F W A	CLCE, PNL
$P^5E^3N^4S^3$	F W D A	Gherkin
	F W A G	RECON
	F W A	First Order English, PENG, PENG-D, PENG Light
	F W I	iLastic Controlled English
$P^5E^4N^3S^3$	F W A	FE

respect to the entire PENS space, but they are, as we will see, widely scattered. Even though some hotspots of classes and properties can be identified, the languages exhibit a broad variety.

Visualization of the languages in the conceptual space can give us a better picture of the data. Because the PENS scheme is four-dimensional, it is difficult to visualize all dimensions in a single diagram. Figure 1 shows a diagram for each of the six possible dimension pairs: The dots represent CNLs in comparison with natural languages such as English (white dot) and common formal languages (black dots). Note that the dots represent PENS classes and not individual languages.

It is evident that the CNLs are widely scattered between the two extreme cases of natural English (white dot) and propositional logic (black dot in the corner). Seen from any angle, the set of existing CNLs exhibits wide variation. Except for the subspace with a naturalness level of less than 3, where there can be no CNLs by our definition, they cover a large part of the conceptual space. This indicates that PENS is a powerful scheme for distinguishing different CNLs.

The diagrams also show that the CNL classes form one single cloud, from any perspective, and not two or more disconnected clouds. This means that it would be difficult to come up with a clean categorization scheme that would subdivide the large and diverse set of existing CNLs. This seems to justify the decision of using the term CNL in a broad sense and not replacing it by more specific terms.

For several dimension pairs, strong correlations are observed. Precision and simplicity are positively correlated: More precise languages tend to be simpler (Spearman's rank correlation coefficient $\rho = 0.90$, using individual languages as data points and excluding the languages for comparison). Expressiveness and simplicity exhibit a strong

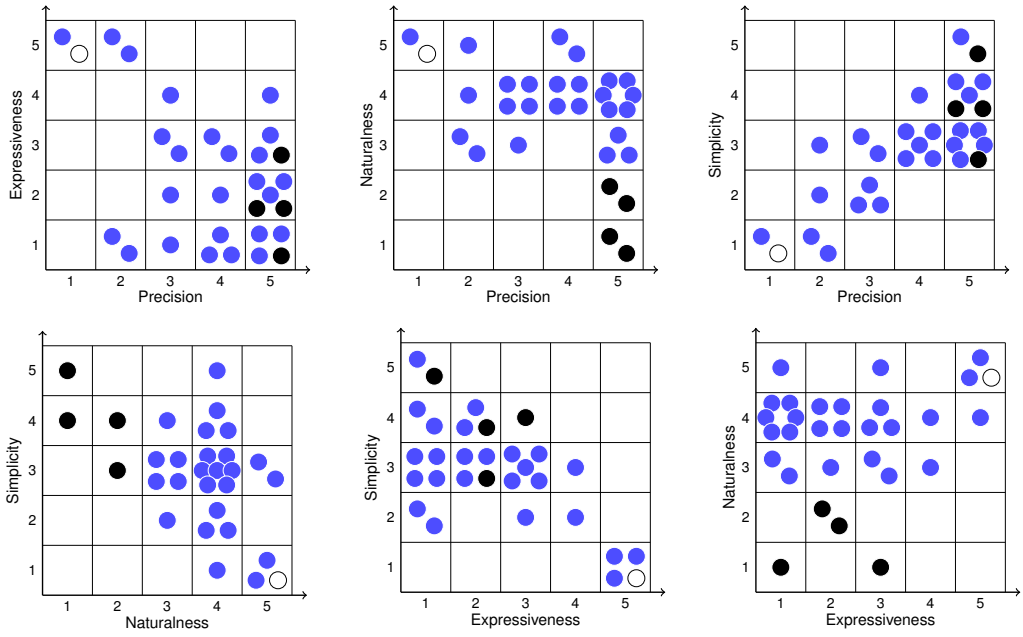

Figure 1
Visualization of the PENS dimensions of existing CNLs, as compared with natural languages (white dot) and common formal languages (black dots). Each dot represents a PENS class containing one or more languages.

negative correlation: More expressive languages tend to be more complex ($\rho = -0.82$). In addition, naturalness/expressiveness are strongly positively ($\rho = 0.77$) and naturalness/simplicity strongly negatively correlated ($\rho = -0.76$). At a slightly lesser degree, negative correlation values are obtained for the pairs precision/naturalness ($\rho = -0.67$) and precision/expressiveness ($\rho = -0.66$). These observations seem to be in line with what one would intuitively expect.

5.2 Properties

Let us turn to the properties. Table 3 shows the number of CNLs for each of the properties we considered and their combinations. As some languages have been used more extensively and over longer periods of time than others, these numbers do not necessarily reflect the actual importance or popularity of the different language types. The table also shows the average PENS values for each type. Again, we should be careful when interpreting these numbers, as all languages were equally weighted, which does not take into consideration that some languages are much more mature and widespread than others. Nevertheless, these numbers reveal some interesting facts.

For a bit less than half of the languages, the goal is to increase comprehensibility. Formal representation is the goal of another, only slightly overlapping, half. About 22% of all languages have translatability as their goal. There is a large overlap of the types C and T, whereas these two barely overlap with F. Existing CNL approaches can therefore be roughly subdivided into two groups of similar size: One consisting of languages for improved comprehensibility and translatability, and the other made up of languages that have formal representation as their goal. Mostly, languages of the types C and T are domain-specific, originated from industry, and focus more on expressiveness and naturalness than on precision or simplicity. Languages of type F, in contrast, mostly have an academic origin and tend to have a much stronger focus on precision and simplicity at the cost of expressiveness and naturalness.

When it comes to the distinction between written and spoken languages, we see a very one-sided picture: More than 90% of all languages are intended to be written; we found only seven languages that are intended to be spoken (one of which is intended to be spoken *and* written). The reason for this might be that controlling a spoken language is much more difficult in practice. Written texts can be revised and given to a language checker before publication, whereas spoken language typically lacks this two-stage process. It is an interesting fact that six out of the seven spoken languages originated

Table 3
Properties of existing CNLs with average PENS values.

	property	total	combined with property									PENS average			
			C	T	F	W	S	D	A	I	G	P	E	N	S
C	comprehensibility	45	–	17	3	40	6	33	4	33	8	2.0	4.3	4.7	1.2
T	translation	22	17	–	1	21	0	17	5	18	0	2.0	4.8	5.0	1.1
F	formal representation	54	3	1	–	52	1	19	45	10	2	4.4	2.3	3.8	3.2
W	written	93	40	21	52	–	1	46	49	42	5	3.3	3.5	4.3	2.3
S	spoken	7	6	0	1	1	–	6	0	1	6	2.0	1.6	3.4	1.9
D	domain-specific	53	33	17	19	46	6	–	20	29	6	2.8	3.5	4.4	1.9
A	academia	50	4	5	45	49	0	20	–	4	1	4.3	2.5	3.9	3.1
I	industry	43	33	18	10	42	1	29	4	–	0	2.3	4.3	4.7	1.4
G	government	10	8	0	2	5	6	6	1	0	–	2.4	2.5	3.8	2.0

from a governmental environment. On average, written languages have higher PENS values in all four dimensions.

Concerning domain-specificity, the data are balanced. About half of the languages are designed for a specific and narrow domain. The other half follow a more general-purpose approach. Comprehensibility is the prevalent goal for domain-specific languages, and they mostly originated from industry. No clear tendencies can be identified with respect to the PENS dimensions.

Concerning the last three properties, the data show similar language counts for academic and industrial CNLs: 50 and 43 languages, respectively. On the other hand, only ten CNLs were found that originated from a governmental environment. It has to be noted, however, that information about CNLs from industry is typically much scarcer than about languages from academia or governments. It is therefore likely that most of the languages that escaped this survey (because of missing or hard-to-find information) are industrial ones. Such a bias might also be present in the case of some of the other properties as well. In any case, academia apparently focuses much more on languages for formal representation than for comprehensibility or translation, whereas industry seems to have an opposite focus.

5.3 Design Decisions

Apart from being a description of the current state of the art, Table 3 can be a valuable tool for making design decisions when creating a new CNL. In such a situation, the application environment of the language to be defined is typically fixed, but not yet the inherent properties of the language itself. Those inherent language properties are supposed to be fixed only during the design process. At the early design stage, Table 3 can be used to check the level of previous work on CNLs for a given combination of environment properties. It also delivers the PENS classes of a typical CNL in this environment, which can be used to guide the design process.

For example, if you intend to create a domain-specific, industrial CNL to enhance comprehensibility, the table tells you that the combination of these properties is not unusual at all (at least pairwise combinations). Furthermore, the table indicates that such a language typically has a PENS class somewhere between $P^2E^3N^4S^1$ and $P^3E^5N^5S^2$. As a second example, somebody might want to design a CNL for speech translation. A quick look at the table reveals that no such CNL has been reported so far, which indicates that a significant amount of original work is needed for the design of such a language. We also see that a typical spoken CNL is very different from a typical language for translation in terms of expressiveness and naturalness. This suggests two important design decisions: How expressive should the resulting language be, and how natural?

The table can reveal such questions about design decisions, but of course it cannot answer them. Nevertheless, such information about existing approaches in similar problem domains and environments can be very valuable to focus the design effort to the crucial aspects.

5.4 Timeline

Because CNLs have been defined and used over many decades and have influenced each other, it is interesting to draw the evolution of these languages on a timeline, as Figure 2 does. Each bar represents the "life" of a language, that is, the period when the language was studied or used. For some languages, the year of "birth" or "death" is unknown, which is indicated by dashed bars fading in and out. The vertical lines

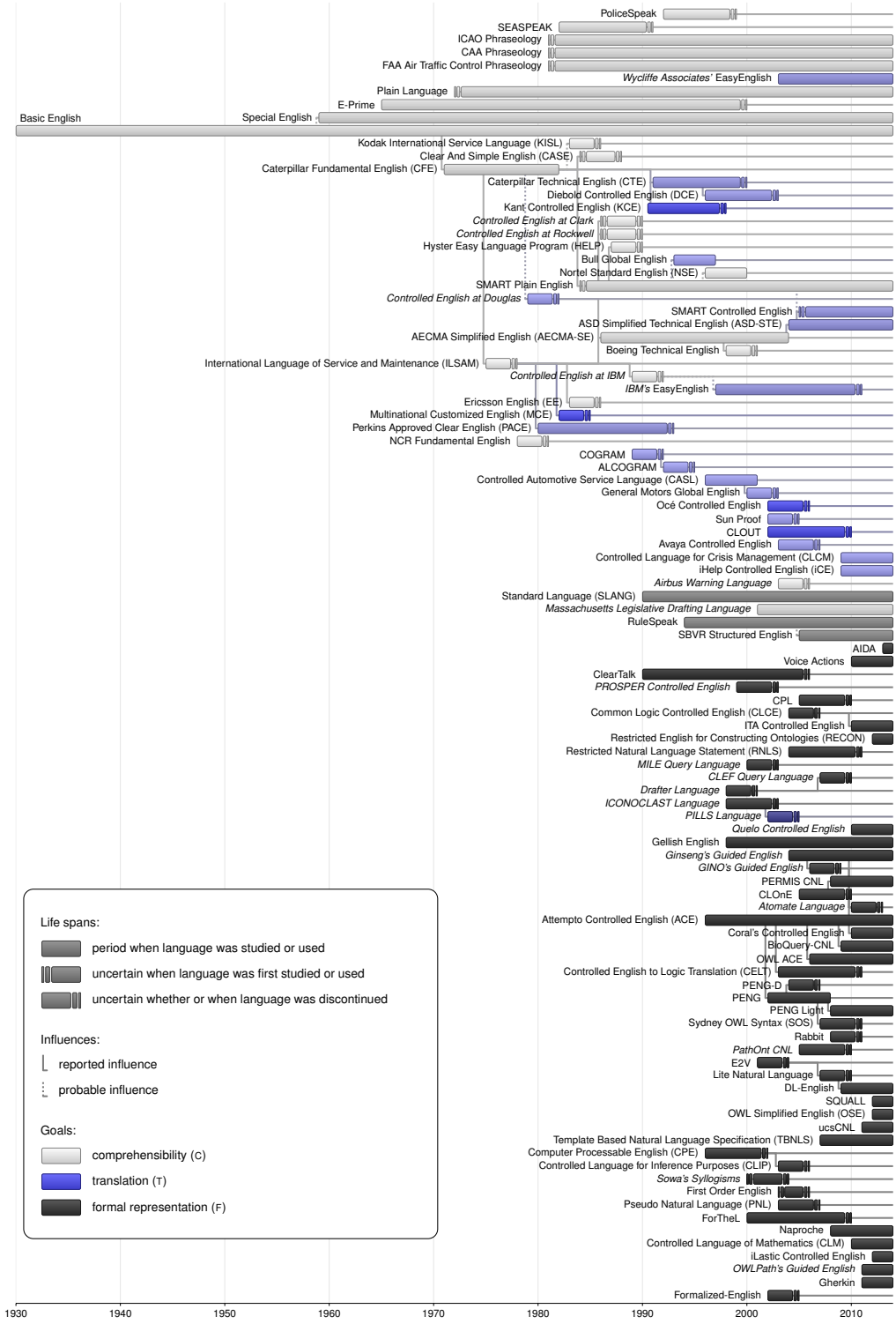

Figure 2
The timeline of the evolution of controlled English.

show influences from other languages at the time of birth (solid for reported influences; dashed for influences that are not reported but seem probable). The colors of the bars represent the goals of the languages, as indicated in the legend.

The oldest CNL, Basic English, is also the most influential one. It influenced CFE, and indirectly ILSAM, both very influential languages in their own right. Altogether, more than 20 languages were directly or indirectly inspired by Basic English. Among the more recent languages, ACE is the most influential in terms of offspring languages.

Looking for an overall theme in the evolution of CNL, one can identify something that could be called three "eras": the general, technical, and logical eras. The **general era** lasted until the late 1960s or early 1970s. Only a few languages were defined and used during this time, all of which were designed to improve human comprehension and to serve as general languages with no specific application domain or narrow community in mind. These languages survived in their small niche, when, during the subsequent **technical era** that began in the early 1970s, CNLs were applied to technical documentation for improved human comprehension as well as improved machine translation. Again, this branch of languages did not disappear at the end of the era and continues to be used today, but a new type of CNL emerged. During the **logical era** that began in the mid 1990s, many CNLs were created with some sort of mapping to formal logic, which enabled not only automatic processing but actual automatic interpretation. These three eras partly correspond to the three goals introduced in Section 2.3: The first CNLs were of type C, type T emerged in the technical era, and type F in the logical era.

5.5 Evaluations

Finally, we can turn to a crucial aspect that we have not yet discussed: Do CNLs actually achieve the goals they were designed for? A number of studies have been reported that evaluate the supposed advantages of these languages. The relevant research question obviously depends on the goal the language is supposed to achieve. In their most general forms, the research questions for the types C, T, and F can be stated as follows:

- C Does a CNL make communication among humans more precise and more effective?
- T Does a CNL reduce overall translation costs at a given level of quality?
- F Does a CNL make it easier for people to use and understand logic formalisms?

Each of these general research questions can be broken down, and most studies target more specific questions.

For type C, two studies on AECMA-SE showed that the use of controlled English significantly improves text comprehension, with a particularly large effect for complex texts and non-native speakers (Shubert et al. 1995; Chervak, Drury, and Ouellette 1996). The results of other studies were similar but not significant (Stewart 1998). The language CLCM has been found to have a positive effect on reading comprehension for most groups of readers under certain circumstances such as stress situations (Temnikova 2012).

Concerning type T, it has been reported that the use of the controlled language MCE for machine-assisted translation leads to a "five-to-one gain in translation time" (Ruffino 1982). Similar results have been presented for the language PACE, with which post-editing of machine-assisted translation is "three or four times faster" than without

(Pym 1990). It has been shown that the adherence to typical CNL rules improves post editing productivity and machine translation quality (Aikawa et al. 2007; O'Brien and Roturier 2007). For the language CLCM, it has been reported that CNL texts are easier to translate than uncontrolled ones (Temnikova and Orasan 2009; Temnikova 2012) and that the time needed for post-editing is reduced on average by 20% (Temnikova 2010, 2012).

Studies on type F can be subdivided into those that test the general usability of CNL tools and those that specifically evaluate the comprehensibility of the actual languages. Starting with the usability studies, it has been shown for the language CLOnE that its interface is more usable than a common ontology editor (Funk et al. 2007). Similarly, Coral's controlled English has been shown to be easier to use than a comparable common query interface (Kuhn and Höfler 2012). Positive usability results for CNL tools have also been reported for GINO (Bernstein and Kaufmann 2006), CLEF (Hallett, Scott, and Power 2007), CPL (Clark et al. 2007), PERMIS (Inglesant et al. 2008), Rabbit (Dimitrova et al. 2008), and ACE (Kuhn 2009). Turning to the comprehensibility studies, it has been shown for the CLEF query language that common users are able to correctly interpret given statements (Hallett, Scott, and Power 2007). ACE has been shown to be easier and faster to understand than a common ontology notation (Kuhn 2013), whereas experiments on the Rabbit language gave mixed results (Hart, Johnson, and Dolbear 2008).

In addition to these high-level evaluations, more specific tests have been reported such as evaluations on coverage (Bernstein et al. 2006; Kaljurand 2007), performance, convergence (Adriaens and Macken 1995), parseability (Wojcik, Harrison, and Bremer 1993), computational complexity (Pratt-Hartmann 2003; Thorne and Calvanese 2010), text complexity, and text length (Temnikova 2012).

In general, there seems to be good evidence for each of the language types that the use of CNL can be advantageous. This does not mean, of course, that CNL approaches always perform better. This depends heavily on the precise problem domain, the background of the users, and—perhaps most importantly—the quality of the design of the language and its supporting tools.

6. Conclusions

To conclude, we can come back to the aims set out in the Introduction of this article. The first goal was to get a better theoretical understanding of the nature of controlled languages. First of all, this article shows that despite the wide variety of existing CNLs, they can be covered by a single definition. The criteria of the proposed definition include virtually all languages that have been called CNLs in the literature. We could show that these languages form a widely scattered but connected cloud in the conceptual space between natural languages on the one end and formal languages on the other. The informal statement that CNLs are more formal than natural languages but more natural than formal ones is substantiated and verified.

The next goal was to establish a common terminology and a common model. We emphasized the difference between characteristics of the *environments* of languages on the one hand and the properties of the *languages themselves* on the other. Both aspects are important, but the second is more difficult to capture in a quantitative way. Nine general properties have been collected to describe the application environments of CNLs. As a novel addition to this model, we proposed the four-dimensional PENS scheme to describe inherent language properties. This scheme allows for classification of CNLs on a discrete scale on the dimensions of precision, expressiveness, naturalness,

and simplicity. Together, this allows us to formally model the important properties of languages and their environments in a simple way, and to put order and structure to a previously fuzzy and disconnected field.

The third goal was to provide a starting point for researchers interested in CNL. The most important conclusion in this respect is the fact that many more CNLs exist than have been found in any previous survey. Previously, the most comprehensive overview counted 41 CNLs (Pool 2006) based on various natural languages, whereas this survey covers 100 languages for English alone. The diversity of languages and the different environments in which they were studied and used apparently had the consequence that many CNL researchers and developers were not aware of a large number of relevant languages. As a starting point for researchers, this work presents a diverse sample of twelve important and influential languages, along with a long list of all CNLs collected. The introduced model of languages and environments can also facilitate the identification of a particular research focus and the collection of relevant prior work.

The fourth goal was to help CNL developers make design decisions. To that aim, the data of this survey can be used to direct developers to existing CNL approaches in a given environment and problem domain. The data can reveal whether a certain kind of CNL usage is common, rare, or inexistent until now, which can be used as an indication of the amount of original work required. Furthermore, the typical language properties of CNLs in terms of precision, expressiveness, naturalness, and simplicity can be retrieved for a given usage scenario. This information might be very useful to identify important design decisions and to find existing approaches to build upon.

I would like to conclude with the observation that the study of controlled languages is a very dynamic and highly interdisciplinary field, for the most part occupying small niches in the academic, industrial, and governmental worlds. However, adding all these niches together gives us a large body of past and ongoing work. Assuming that people will have to interact even more closely with computers and across language borders in the future, I am convinced that we will see even more work in this area.

Appendix A: Full List of English-Based Controlled Natural Languages

This is the full list of 100 English-based CNLs in alphabetical order. See Section 4 for the details of this collection.

AECMA Simplified English (AECMA-SE) (AECMA 1986) was the predecessor of ASD Simplified Technical English. *See Section 4.1.* — $P^2E^5N^5S^1$, C W D I

AIDA (Kuhn et al. 2013) is a CNL to allow for informal and underspecified representations of scientific assertions in an approach for semantic publishing called "nanopublications." Single English sentences are used as a scaffold for underspecified representations and for the inclusion of informal statements in formal RDF-based structures. These sentences are Atomic, Independent, Declarative, and Absolute (hence the name AIDA). This is an example:

> The degree of hepatic reticuloendothelial function impairment does not differ between cirrhotic patients with and without previous history of SBP.

— $P^2E^5N^4S^1$, F W A

"Airbus Warning Language" (Spaggiari, Beaujard, and Cannesson 2003) is a language for short industrial warnings, focusing on abbreviations and restricting the word order. This is an exemplary statement:

> ENG1 REV NOT LOCKED

— $P^2E^1N^3S^3$, C W D I

ALCOGRAM (Adriaens and Schreors 1992) is a CNL developed at Alcatel. It originated from COGRAM as an "algorithmic variant," focusing on the use within a computer-aided language learning tool. In contrast to COGRAM, which consists of three components that declaratively define the language, ALCOGRAM is defined based on a four-staged algorithm. Each of these four stages checks certain aspects: preparatory textual control (e.g., "Define technical terms and acronyms in advance"), syntactic control (e.g., "Write one instruction per sentence for single actions"), lexical control (e.g., "Avoid gender-specific language"), and micro control (e.g., "Use words for a number when it is the first word in the sentence"). These are two examples of ALCOGRAM sentences:

> Set the switch to the middle. Press the button on your right.
>
> When the test circuit is called, a test tone with the proper transmit level is returned.

— $P^2E^5N^5S^1$, C T W D A I

ASD Simplified Technical English (ASD-STE). *See Section 4.1.* — $P^2E^5N^5S^1$, C T W D I

"Atomate Language" (Van Kleek et al. 2010) is part of the Atomate interface, which lets users define simple automatic tasks and reminders taking context and current activity into account. The language was inspired by CLOnE, ACE, and the GINO and Ginseng systems. This is an example of such a task definition:

> Alert me when my location is home on/after Tuesdays at 5pm with the message:
> Trash day!

A special editor supports users in writing such sentences, using a mixture of predictive editing and conceptual authoring. The sentences are mapped to RDF and automatically triggered when the preconditions are met. — $P^4E^2N^4S^3$, F W D A

Attempto Controlled English (ACE). *See Section 4.1.* — $P^4E^3N^4S^3$, F W A

Avaya Controlled English (Avaya 2004) is a language for technical publications in the telecommunication and computing industry. Its use should reduce translation costs and should make texts easier to understand for human readers. It puts restrictions on the lexicon (e.g., "Do not use *abort*"), grammar (e.g., "Use active voice"), semantics (e.g., "Use *may* only to grant permission"), and style (e.g., "Put command names in bold monospaced type"). An open list of about 250 words defines preferred terminology for the given computer and telephony domain, and clarifies usage and meaning of these words. These are two examples of sentences:

> This procedure describes how to connect a dual ACD link to the server.
>
> If the primary server fails, you can use the secondary server.

— $P^2E^5N^5S^1$, C T W D I

Basic English. *See Section 4.1.* — $P^2E^5N^5S^1$, C W

BioQuery-CNL (Erdem and Yeniterzi 2009) is a language for biomedical queries. It serves as an interface language for a query engine based on answer set programming. BioQuery-CNL was initially designed as a subset of ACE with some small modifications handled in a preprocessing step. The ACE parser was used for processing the language. In later versions, however, the language diverged from ACE and evolved into an independent language with its own parser. This is an exemplary query:

> What are the genes that are targeted by all the drugs that belong to the category Hmg-coa reductase?

— $P^5E^2N^4S^4$, F W D A

Boeing Technical English (Wojcik, Holmback, and Hoard 1998) was an extension of AECMA Simplified English to improve readability and consistency of documents, with the specific goal to broaden the scope beyond the aviation domain. The language seems to have been discontinued and apparently was never deployed at Boeing. — $P^2E^5N^5S^1$, C W I

Bull Global English (Smart Communications Inc. 1994) or **Bull Controlled English** is a language developed at Groupe Bull, a French computer company.

It was probably influenced by SMART Plain English. Bull Global English can be summarized by the following ten rules (Karkaletsis and Spyropoulos 1997), which have a considerable overlap with the rules of Caterpillar Fundamental English:

1. Make positive statements.
2. Keep sentence length 21 words.
3. Avoid false nomenclature.
4. One thought per sentence.
5. Use simple sentence structures.
6. Use active voice and parallel construction.
7. Avoid conditional tenses.
8. Avoid abbreviations and colloquialisms.
9. Use correct punctuation.
10. Use standardized nomenclature.

— $P^2E^5N^5S^1$, C T W D I

CAA Phraseology (CAA 2011) is a language for air traffic control introduced by the Civil Aviation Authority (CAA) in the 1980s or possibly earlier. It is very similar to the phraseologies by FAA and ICAO. — $P^2E^1N^3S^2$, C S D G

Caterpillar Fundamental English (CFE). *See Section 4.1.* — $P^2E^5N^5S^1$, C W D I

Caterpillar Technical English (CTE) (Hayes, Maxwell, and Schmandt 1996; Kamprath et al. 1998) is the second CNL developed at Caterpillar. Its development started in 1991, that is, almost a decade after the discontinuation of CFE. Apart from improving consistency and reducing ambiguity of technical documentation, the goal of CTE was to improve translation quality and reduce translation costs with the help of machine translation. This is an example of a CTE text:

> This category indicates that an alternator is malfunctioning. If the indicator comes on, drive the machine to a convenient stopping place. Investigate the cause and determine the solution.

In contrast to CFE, texts in CTE are supposed to be translated before given to personnel in non-English speaking countries. As a further difference, CTE was designed to be an "enforceable controlled English" that comes with an authoring tool that enforces the compliance with the restrictions. The CTE lexicon consists of about 70,000 terms with a "narrow semantic scope" (compared with CFE's less than 1,000 terms with a broader semantic scope). The syntax is restricted, too, including restrictions on the use of conjunctions, pronouns, and subordinate clauses. CTE comes with a language checker that

allows for interactive disambiguation on the lexical level, enriches the technical texts with SGML annotations, and uses the syntax analyzer of the KANT system (see KANT Controlled English). — $P^2E^5N^5S^1$, C T W D I

Clear And Simple English (CASE) (Pym 1990) was a controlled English introduced in the 1980s at the J. I. Case Company, a manufacturer of construction and agricultural equipment. It descended from CFE. — $P^2E^5N^5S^1$, C W D I

ClearTalk (Skuce 2003) is a CNL for the Semantic Web first presented in the 1990s. Its creator claims that documents in ClearTalk can be "almost automatically" translated into a formal logic notation and into other natural languages. It "offers a flexible degree of formality" that lets an author choose to "leave or remove ambiguity." It has been used to encode more than 25,000 facts in different technical domains. ClearTalk is heavily restricted on the syntactic level (e.g., basic sentences have the general form *subject predicate complement modifier-phrases*) as well as on the semantic one (e.g., the determiner *a* at subject position represents universal quantification). These restrictions are expressed in a large number of rules. Two examples of sentences are shown here:

> Any adverb that modifies a verb must be adjacent to (that verb or another adverb).
>
> Mary hopes that [- Bill loves her -].

ClearTalk can itself be described in ClearTalk; the first example is from this self-description. Different forms of parentheses are used to disambiguate different kinds of scopes. — $P^3E^3N^3S^3$, F W A

"CLEF Query Language" (Hallett, Scott, and Power 2007) is a language used within a system called CLEF (Clinical E-Science Framework), which should help clinicians, medical researchers, and hospital administrators to query electronic health records. The language was influenced by the Drafter language. Basic queries are composed of three elements: the set of relevant patients, the received treatments, and the outcomes. This is an example:

> For all patients with cancer of the pancreas, what is the percentage alive at five years for those who had a course of gemcitabine?

Complex queries can have multiple elements of the same type. The system uses a conceptual authoring approach for writing queries, which are then translated in several steps to SQL and given to a database engine. — $P^5E^1N^4S^3$, F W D A

COGRAM (Adriaens and Schreors 1992) was a controlled language developed in the late 1980s for the telecommunication domain (at Alcatel). It was developed as a response to the finding that the existing controlled languages AECMA Simplified English, Ericsson English, and IBM's controlled English were "incomplete and defective in many ways." COGRAM consists of a vocabulary of approximately 5,000 words plus another 1,000 technical terms, and a grammar with about 150 rules. These rules fall into three categories: "Do not use X," "Use only X," and "Avoid (try not to use) X." Grammar rules of the last type can be seen as style-guides that do not restrict the coverage of the language. The language definition is divided into three components: lexical (e.g., "Use short infinitives of regular action verbs"), syntactic (e.g., "Do not use a participle to introduce an adverbial clause"), and stylistic (e.g., "Expound major topics, restrict minor topics"). The definition of COGRAM was found to be "not the most motivating of texts for

technical writers to use in the writing process," which led to the development of ALCOGRAM. — $P^2E^5N^5S^1$, C T W D A I

Common Logic Controlled English (CLCE) (Sowa 2004) is a language that can be translated into first-order logic with equality in the form of the Conceptual Graph Interchange Format. It is defined by a grammar in Backus-Naur form "that allows every ambiguity to be resolved when a sentence is parsed." Some of the most important syntax restrictions are: no plural nouns, only present tense, and variables instead of pronouns. For an unambiguous mapping to logic, a number of interpretation rules are applied and parentheses are used to determine the structure of deeply nested sentences. Sentences in this language should be similar to those found in software documentation and textbooks of mathematics, for example:

> If some person x is the mother of a person y, then the person y is a child of the person x.

> Declare give as verb (agent gives recipient theme) (agent gives theme to recipient) (theme is given recipient by agent) (theme is given to recipient by agent) (recipient is given theme by agent).

Imperative sentences, as the second example, are used to import or declare words. Names, nouns, verbs, adjectives, adverbs, and prepositions can be declared in this way. — $P^5E^3N^3S^3$, F W A

Computer Processable English (CPE) (Pulman 1996; Sukkarieh and Pulman 1999) is a controlled language that can be "completely syntactically and semantically analyzed." An early version of the language used KIF as its logic formalism, whereas McLogic was used later on. The language comes with a bidirectional grammar implemented as a Prolog unification grammar. Two examples are shown here:

> Every animal X eats some animal that is smaller than X.

> Every registered user who has borrowed less than ten copies can borrow every available copy.

The mapping to logic seems to be deterministic, even though the available literature is not explicit about this. — $P^5E^2N^4S^3$, F W A

Computer Processable Language (CPL) (Clark et al. 2005) is a controlled variant of English developed at Boeing. It is very different from earlier CNL approaches Boeing was involved in, such as ASD-STE and Boeing Technical English. CPL is much more restricted than these earlier approaches and sacrifices to some degree expressiveness and naturalness for the sake of automated reasoning support. Basic CPL sentences are restricted to the pattern *subject + verb + complements + adjuncts*. There are further restrictions on the syntax, for example, that definite references have to be used instead of pronouns. Statements involving universal quantification are constructed from seven templates such as "If sentence1 then typically sentence2," where *sentence1* and *sentence2* are basic CPL sentences of the structure introduced above and where *typically* is a reliability degree: one of *(almost) always*, *usually*, *sometimes*, and *never*. These are two examples of CPL sentences:

> IF a person is carrying an entity that is inside a room THEN (almost) always the person is in the room.

> AFTER a person closes a barrier, (almost) always the barrier is shut.

A parser translates CPL sentences into a frame-based language with well-defined semantics. In contrast to most other logic-based CNL approaches with custom-built parsers, the parsing process of CPL involves different external tools and resources. An existing parser for unrestricted English is used to generate an intermediary logical form. Then, WordNet and other resources are used to make a "best guess." The resulting logical representation is then paraphrased and shown to the user for verification or correction. — $P^3E^3N^4S^2$, F W I

Controlled Automotive Service Language (CASL) (Means and Godden 1996; Means, Chapman, and Liu 2000) is a controlled language for writing service manuals and bulletins at General Motors developed in the 1990s. The goal was to improve translatability, as well as consistency and readability. The approach moved from an "author-centric model" towards a "hybrid model" that included the role of an editor, before it went to full production in 2000 (Godden 2000). The CASL restrictions are defined by 62 rules, including restrictions on sentence structure, word order, vocabulary, and punctuation. This is an exemplary sentence:

> Several diseases result from asbestos exposure, with latency periods of 10 to 40 years or longer.

Writers are supported by a software tool called CASLChecker. — $P^2E^5N^5S^1$, C T W D I

"Controlled English at Clark" (Adriaens and Schreors 1992) was a language used at the Clark Material Handling Company. It was developed around the late 1980s and was influenced by SMART Plain English. — $P^2E^5N^5S^1$, C W D I

"Controlled English at Douglas" (Kleinman 1982) was a language developed in 1979 by the McDonnell Douglas aerospace company for their technical manuals. It was based on a dictionary of about 2,000 words (most of them verbs), favoring short and simple words and aiming at a single word per meaning and a single meaning per word. In addition to the words of the dictionary, "nomenclature words" can be introduced. The goal was to improve readability, translatability, and standardization. It was probably influenced by CFE and had itself an influence on AECMA SE. — $P^2E^5N^5S^1$, C T W D I

"Controlled English at IBM" (Adriaens and Schreors 1992) was a language developed and used at IBM in the late 1980s. It was influenced by ILSAM and might have influenced EasyEnglish, which was also developed at IBM several years later. It relied on a closed list of words, and writers were assisted by different instruction programs. — $P^2E^5N^5S^1$, C W D I

"Controlled English at Rockwell" (Adriaens and Schreors 1992) was a language used at the company Rockwell International. It was developed around the late 1980s and was influenced by SMART Plain English. — $P^2E^5N^5S^1$, C W D I

Controlled English to Logic Translation (CELT) (Pease and Li 2010) is a controlled natural language presented in 2003. It is a domain-independent language inspired by ACE. In contrast to ACE, it uses existing linguistic and ontological resources, concretely the SUMO ontology and WordNet. These are two exemplary sentences:

> Dickens writes Oliver Twist in 1837.

> Every boy likes fudge.

The syntax structure of CELT sentences is deterministically parsed. Heuristics are applied only afterwards to map the words to SUMO and WordNet. The language is implemented as a unification grammar in Prolog. — $P^4E^2N^4S^3$, F W I

Controlled Language for Crisis Management (CLCM) (Temnikova 2010, 2011, 2012) is a language for writing instructions about how to deal with crisis situations. The language is defined by about 80 simplification rules. These simplification rules include restrictions on text structure (e.g., "Write a title for every specific situation"), formatting (e.g., "Separate with a new line each block of instructions"), lexicon (e.g., "avoid technical terms"), syntax (e.g., "Avoid passive voice"), semantics (e.g., "Use only literal meaning"), and pragmatics (e.g., "Remove unimportant information"). — $P^2E^5N^5S^1$, C T W D A

Controlled Language for Inference Purposes (CLIP) (Sukkarieh 2003) is a language based on the logic notation McLogic and influenced by CPE. It is "semantically driven," meaning that it was designed around the given logic formalism and not vice versa. Two examples are shown here:

> Every student who laughs succeeds.
>
> Smith and Jones sign five contracts.

— $P^5E^2N^4S^3$, F W A

Controlled Language for Ontology Editing (CLOnE) (Funk et al. 2007), previously called **CLIE Controlled Language**, is a CNL designed as a front-end language for OWL, covering only a small subset of it. It is defined by ten basic sentence patterns. It adds procedural semantics on top of OWL to introduce and remove entities and axioms. These are two examples of CLOnE sentences:

> Persons are authors of documents.
>
> Forget everything.

— $P^5E^2N^4S^4$, F W A

Controlled Language Optimized for Uniform Translation (CLOUT) (Muegge 2007) is a CNL to improve machine translation. It puts restrictions on the vocabulary and prohibits structures such as passive voice and pronouns. — $P^2E^5N^5S^1$, T W I

Controlled Language of Mathematics (CLM) (Humayoun and Raffalli 2010) is a language for expressing mathematical texts, as found in textbooks. The language is similar to Naproche CNL and ForTheL. The grammar of CLM is implemented in Grammatical Framework and allows for deterministic translation into first-order logic. The goal is to automatically verify mathematical proofs. — $P^5E^3N^3S^3$, F W D A

Coral's Controlled English (Kuhn and Höfler 2012) is a controlled language for expressing formal queries to annotated text corpora. It is influenced by ACE, but is much less expressive, simpler, and more domain-specific. It is embedded into a query interface called Coral to enable users with no particular background in computer science to effectively use large corpora of annotated texts. This is an exemplary query:

> Find all passages where a noun phrase contains a verb phrase; the verb phrase precedes a prepositional phrase; the prepositional phrase contains a verb "see";

Such queries are deterministically mapped to AQL, an existing formal query language. The language is defined by 51 simple grammar rules. — $P^5E^1N^4S^4$, F W D A

Diebold Controlled English (DCE) (Hayes, Maxwell, and Schmandt 1996; Moore 2000) is a controlled language developed at Diebold with the goal to make translation faster

and less expensive by assisting human translators with specific translation tools. It was inspired by CTE, but is less strict concerning lexicon and grammar, making the approach more flexible. It consists of three main components: a lexical database, a set of grammar rules, and a checking tool. — $P^2E^5N^5S^1$, C T W D I

DL-English (Thorne and Calvanese 2010) is a Description Logic–based controlled language presented together with other similar languages to study and compare their computational complexity. It is similar to Lite Natural Language by the same research group. — $P^5E^2N^4S^4$, F W A

"Drafter Language." *See Section 4.1.* — $P^4E^1N^4S^3$, F W D A

E-Prime or **E'.** *See Section 4.1.* — $P^1E^5N^5S^1$, C W A

E2V. *See Section 4.1.* — $P^5E^2N^4S^4$, F W A

EasyEnglish (by IBM) (Bernth 1997), not to be confused with Wycliffe Associates' EasyEnglish, is a language developed at IBM, which might have been influenced by an earlier controlled English at the same company (Adriaens and Schreors 1992). The main goal of EasyEnglish was to improve machine translation. The approach is based on a sophisticated grammar checker that returns suggestions and warnings. Apart from detecting common grammar errors, the system can enforce the use of a certain controlled vocabulary and can spot ambiguities. For such ambiguities, the system can propose alternatives, but it is ultimately up to the user whether to follow the system's suggestions or not. The problems encountered in a given document are quantified in the form of a clarity index, which must be above a certain threshold value. The fact that the restrictions of the language are not enforced but just suggested does not make the language more precise or simpler than full natural English. EasyEnglish has been extended later to check not only on the sentence level but also on the document level, and this has been implemented in a tool called EasyEnglishAnalyzer (Bernth 2006). — $P^1E^5N^5S^1$, C T W I

EasyEnglish (by Wycliffe Associates) (Betts 2003), not to be confused with IBM's EasyEnglish, is a controlled language used for transcribing biblical texts. The original goal was to improve the translation process into other languages, but EasyEnglish is also directly used by readers with limited knowledge of English. The language is restricted with respect to lexicon, syntax, and semantics. There are two levels: Level A makes use of about 1,200 words, and level B has a larger lexicon of about 2,800 words. In either case, the meaning of these words is restricted. For example, *fair* can only mean *unbiased*, and *to see* cannot be used in the sense *to meet*. It is possible to use words that are not on the list, if they are explained in separate EasyEnglish sentences. The following is an excerpt of a text in EasyEnglish (*moor* is not in the lexicon and has to be explained):

> The Highlands of Scotland consist of lakes, mountains and moors. The moors are flat empty lands where no trees grow. This land is wonderful and magnificent because it is so empty.

There is a strict sentence length limit of 20 words, and paragraphs may not contain more than 150 words. Sentence structure is kept simple by allowing not more than two finite clauses and not more than two prepositional phrases per sentence. Furthermore, deep nesting and passives are restricted. In addition, texts should adhere to logical

simplicity: "EasyEnglish writers are encouraged to identify the basic idea units in a complex sentence or paragraph and arrange them in logical order." — $P^2E^5N^5S^1$, C T W D

Ericsson English (EE) (Adriaens and Schreors 1992) was a language developed at Ericsson in the early 1980s, influenced by ILSAM. It is built on a closed list of acceptable words, but other words can be introduced if accompanied by a definition in EE. — $P^2E^5N^5S^1$, C W D I

FAA Air Traffic Control Phraseology. *See Section 4.1.* — $P^2E^1N^3S^2$, C S D G

First Order English (Pool 2006) is a controlled natural language that maps to first-order logic. No detailed description of this language is available. — $P^5E^3N^4S^3$, F W A

Formalized-English (FE). *See Section 4.1.* — $P^5E^4N^3S^3$, F W A

ForTheL (Vershinin and Paskevich 2000) is a CNL for mathematical texts similar to Naproche CNL and CLM. The name stands for "Formal Theory Language." Statements in this language can be automatically translated into first-order logic with equality. The following is an exemplary text:

> Lemma 1. Each set has a subset.
> Proof. 0 is a subset of all sets. QED.

— $P^5E^3N^3S^3$, F W D A

Gellish English (van Renssen 2005) is a controlled language designed as a common data language for industry. The first version was ready in 1998. Basically, it consists of simple subject–predicate–object structures with predefined relations in the form of fixed phrases such as "is a specialization of" and "is valid in the context of." These are two examples:

> collection C each of which elements is a specialization of animal
>
> the Eiffel tower has aspect h1
> h1 is classified as a height
> h1 is qualified as 300 m

Meta-information about the context of such statements can be expressed in the form of additional "accessory facts." Gellish builds upon a fixed upper ontology with a large number of predefined concepts and relation types. Texts in Gellish can be transformed into a formal tabular representation. The semantics of the language is not fully formalized, which means that there is no mapping to an established logic formalism. Gellish support simple kinds of if–then rules (van Renssen 2011), but these rules do not allow for universal quantification over several variables in a general way. — $P^4E^2N^4S^3$, F W A I

General Motors Global English (Means, Chapman, and Liu 2000) or just **Global English** is a controlled language developed at General Motors. The goal was to improve comprehension for non-native speakers and translatability. It is defined by 15 rules based on four principles: "be brief," "be clear," "be direct," and "be culturally alert." These rules include a limit on the sentence length and grammatical restrictions such as the exclusion of passive voice. The language evolved from a reduced set of twelve of the 62 rules of the CASL language, which was developed at General Motors several years earlier. In contrast to CASL, Global English does not come with a software tool for checking the compliance with the restrictions. — $P^2E^5N^5S^1$, C T W D I

Gherkin (Nečas 2011) is a language for writing executable scenarios for software specifications. This is an excerpt of a scenario description:

> Scenario: Unsuccessful registration due to full course
> Given I am a student
> And a lecture "PA042" with limited capacity of 20 students
> But the capacity of this course is full
> [...]

The structuring words such as *Given*, *And*, and *But* are fixed. The restrictions on the remaining text such as "I am a student" are implemented in ordinary programming languages using regular expressions, and are stored in small modules called "step definitions." The concrete step definitions are not part of Gherkin, but have to be implemented for the particular task at hand. Gherkin is therefore highly customizable and extensible, and the classification given here is meant to apply to a typical concrete language that is based on Gherkin. — $P^5E^3N^4S^3$, F W D A

"GINO's Guided English" (Bernstein and Kaufmann 2006) is a language used in GINO, a system to query and edit ontologies. The language was influenced by Ginseng and supports the same kinds of queries. In addition, GINO has some limited support for procedural statements to introduce new entities, for instance:

> There is a subclass of class water area named lake.

Query statements are mapped to SPARQL and procedural statements map to OWL axioms to be added or modified. Queries can exhibit structural ambiguity, in which case the system evaluates all possible interpretations and shows to the user the union of their answers. The grammar that describes the language consists of 120 grammar rules. — $P^4E^2N^4S^3$, F W A

"Ginseng's Guided English" (Bernstein et al. 2006) is a CNL used in a system called Ginseng, which is a query interface to access knowledge bases in the form of OWL ontologies. The vocabulary for the language is loaded from the respective ontologies. These are two examples of queries:

> What are the capitals of states that border Nevada?
>
> Is there a city that is the highest point of a state?

The grammar consists of 120 static grammar rules plus additional dynamic rules generated from the ontologies. — $P^5E^1N^4S^3$, F W A

Hyster Easy Language Program (HELP) (Smart 2003) is a controlled English developed in the 1980s for maintenance manuals for lift trucks. It is based on SMART Plain English and thus indirectly on CFE (Pym 1990). — $P^2E^5N^5S^1$, C W D I

ICAO Phraseology (Eurocontrol 2009) is controlled language for air traffic control defined by the International Civil Aviation Organisation (ICAO) in the 1980s or even earlier. It is very similar to the phraseologies by FAA and CAA. — $P^2E^1N^3S^2$, C S D G

"ICONOCLAST Language" (Power 1999) is a CNL to write patient information leaflets. It is similar to the Drafter language. A conceptual authoring approach is utilized and a formal logic representation is used in the background. This is a simple example:

> If you develop a rash, you should consult your doctor.

— $P^4E^3N^5S^3$, F W D A

iHelp Controlled English (iCE)[5] is a language developed by iHelp Ltd, a documentation consultancy company. iCE consists of "a set of flexible rules and vocabularies for companies wishing to standardize and improve their information." — $P^2E^5N^5S^1$, C T W I

iLastic Controlled English (iLastic 2012) is a language to allow non-developers to write intuitive and natural scripts that automatically retrieve, transform, and combine data from the Web, databases, files, and other resources. This is an exemplary statement:

> delete all files under the tmp folder if the space of the disk is lower than 1024.

— $P^5E^3N^4S^3$, F W I

International Language of Service and Maintenance (ILSAM) (Pym 1990) is an influential language similar to Caterpillar Fundamental English, from which it was derived in the 1970s. — $P^2E^5N^5S^1$, C W D I

ITA Controlled English (ITA CE) (Mott 2010) is a controlled language defined by the International Technology Alliance, a US/UK military research program. It is inspired by CLCE, but is less strict in terms of precision: It has an "informal meaning and a semi-formal mapping to predicate logic." The following are two examples of statements of different types:

> if (the person X has the person Y as brother) and (the person Z has the person X as father) then (the person Z has the person Y as uncle) .
>
> "the plan has failed" because "there was a misunderstanding".

The first example shows a "logical rule"; the second example is a "rationale" statement. Parentheses and variables are used to disambiguate. Around 90 grammar rules define the language. — $P^3E^3N^3S^3$, F W I

KANT Controlled English (KCE) (Mitamura and Nyberg 1995) is a controlled natural language for machine translation used within the KANT translation system. The language was first presented under this name in 1995, but it had at that point already been studied and used for several years. The focus is on technical documents, and KCE was the basis for the development of Caterpillar Technical English. Lexicon, grammar, and semantics are restricted. In addition, ambiguities are resolved interactively by augmenting the input sentences with SGML tags. In the following sentence, for example, the attachment of the preposition "with twelve rivets" is ambiguous:

> Secure the gear with twelve rivets.

In KCE, this ambiguity can be resolved by augmenting the sentence with an SGML tag, for instance "Secure the gear with <attach head='secure' modi='with'> twelve rivets." For the classification of the language, the question arises whether the SGML tags are part of the language or just a method to keep track of decisions concerning ambiguities. The SGML tags positively contribute to the precision of the language but heavily impede its naturalness. Because such markup tags are usually hidden and because KCE texts are initially written without tags, which are added only afterwards, we consider them a part of the KANT methodology but not of the controlled language itself. — $P^2E^5N^5S^1$, T W A

5 http://www.lindy-hop.co.uk/iHelp/ice/.

Kodak International Service Language (KISL) is a CNL developed at Kodak in the early 1980s. Some see it as a descendant of CFE (Spaggiari, Beaujard, and Cannesson 2003). — $P^2E^5N^5S^1$, C W D I

Lite Natural Language (Bernardi, Calvanese, and Thorne 2007) is a CNL based on the language E2V and its variants. It has a deterministic mapping to DL-Lite, which is a logical formalism optimized for good computational properties and is equivalent to a subset of OWL. — $P^5E^2N^4S^4$, F W A

"Massachusetts Legislative Drafting Language" (Massachusetts Senate 2003) is a restricted language for legal texts defined by the Massachusetts Senate. Its purpose is "to promote uniformity in drafting style, and to make the resulting statutes clear, simple and easy to understand and use." The language is defined by about 100 rules that restrict syntax (e.g., "Use the present tense and the indicative mood"), semantics (e.g., "Do not use 'deem' for 'consider'"), and document structure ("Use short sections or subsections"). In addition, there are close to 90 words and phrases that must not be used, with suggested replacements for each of them (e.g., *hide* instead of *conceal*, and *rest* instead of *remainder*). — $P^2E^5N^5S^1$, C W D G

"MILE Query Language" (Piwek et al. 2000) is a language to access maritime rules and regulations. It follows the conceptual authoring approach in a very similar way as the Drafter and CLEF languages. — $P^4E^1N^4S^3$, F W D A

Multinational Customized English (MCE) (Ruffino 1982) is a controlled language developed at Xerox to improve the quality of machine-assisted translation. It was based on ILSAM (Adriaens and Schreors 1992). It uses a restricted domain-specific vocabulary and "a set of writing rules which encourage a clear, concise English and a minimization of ambiguities." — $P^2E^5N^5S^1$, T W D I

Nortel Standard English (NSE) (Smart 2006) is a language developed at Nortel, a telecommunications equipment manufacturer. The development started in 1995 with the help of SMART Communications, and the language was probably influenced by SMART Plain English. — $P^2E^5N^5S^1$, C W D I

Naproche CNL (Cramer et al. 2010) is a controlled language for mathematical texts similar to CLM and ForTheL. Texts in Naproche CNL can be deterministically mapped to first-order logic and then automatically checked for logical correctness. The following is an excerpt of a proof written in this language:

> Axiom 3: For every x, $x' \neq 1$.
> Axiom 4: If $x' = y'$, then $x = y$.
> Theorem 1: If $x \neq y$ then $x' \neq y'$.
> Proof: Assume that $x \neq y$ and $x' = y'$. Then by axiom 4, $x = y$. Qed.

According to its authors, most texts of mathematical textbooks "can be rewritten in the Naproche CNL in such a way that they resemble the original text." — $P^5E^3N^3S^3$, F W D A

NCR Fundamental English (NCR 1978) is a CNL developed at NCR Corporation. The language was used for the technical manuals of the company in order to make them "easier to read and use by NCR employees and customers around the world." These are two examples of sentences:

> While repairing the unit, the field engineer also performs normal maintenance if it is needed.
>
> No maintenance can be performed until the maintenance lock has been activated.

The language consists of three parts: nomenclature, glossary, and vocabulary. Every word of the language belongs to exactly one of these categories. The nomenclature is an open set of different kinds of named individual entities, such as names of products, tools, routines, as well as named modes and conditions. The glossary is another open set of words for technical concepts, such as *audit trail*, that cannot be replaced by a phrase or brief clause using the words of the vocabulary. The vocabulary, finally, is the most interesting part. It consists of a fixed set of 1,350 words (verbs, nouns, adverbs, adjectives, pronouns, prepositions, articles, and conjunctions) plus 650 abbreviations. The content of the vocabulary ranges from fundamental words such as *a*, *not*, and *in* to domain-specific terms such as *testware*, *calibrate*, and *taxable*. The meaning of these words is restricted, and each comes with a definition in full English. The noun *medium*, for instance, is defined as "a method of payment" and must not be used in any other sense. The grammar is not explicitly restricted. — $P^2E^5N^5S^1$, C W D I

Océ Controlled English (Cucchiarini 2002) is a controlled language developed at Océ, a Dutch company in the printing and copying business. Océ Controlled English is combined with traditional machine translation techniques to improve the translation quality of the company's documentation in 17 different languages. One of the important properties of the language is that it leads to more concise texts. For example, instead of "In several windows, an icon shows the current status/activity of a printer. See the list below for a description of each status.", one would write:

> These icons show the status or activity of the copier.

The language is implemented with the help of the MAXit Checker by SMART Communications. — $P^2E^5N^5S^1$, T W D I

OWL ACE (Kaljurand and Fuchs 2006) is a controlled language for the ontology language OWL. Syntactically, it is a subset of ACE. Semantically, it is tailored towards the expressiveness of OWL and is more specific than ACE with its underspecified semantics, particularly in the case of plurals. Thus, OWL ACE is more precise but less expressive than ACE. — $P^5E^2N^4S^3$, F W A

"OWLPath's Guided English" (Valencia-García et al. 2011) is a query language for a tool called OWLPath, with which ontologies can be queried. Statements in this language start with the phrase *View any*. These are two examples:

> View any COMMODITY has_quoted_price in BMF.
>
> View any COMPANY whose STOCK_PRICE.lastTrade is_greater_than $30 and is_included_in Dow_Jones in 2009-04-24.

These statements are translated into the SPARQL query language. Even though their structure roughly follows English grammar, they cannot be considered valid English sentences. — $P^5E^2N^3S^4$, F W A

OWL Simplified English (Power 2012) is a controlled language for the Semantic Web. In contrast to most other approaches, there is no real lexicon, neither built-in nor user-defined. Only a very small number of function words are predefined, and users have to list the verbs they intend to use. All other word categories are inferred based on syntactic clues such as capitalization and adjacent words. This is an example (assuming that *governed* and *lives* are listed as verbs):

> London is capital of a country that is governed by a man that lives in Downing Street.

— $P^5E^2N^4S^4$, F W A

"PathOnt CNL" (Kim et al. 2005; Namgoong and Kim 2007) is a controlled language developed for a tool called PathOnt. The tool is multilingual, supporting English and Korean. Statements in this language are deterministically mapped to RDF triples. These are two exemplary sentences:

> Nam is a student supervised by a professor named Kim.
>
> A received specimen fixed in formalin is a soft tissue mass.

The language seems to cover only simple existential statements. — $P^5E^1N^4S^4$, F W A

PENG (Schwitter 2002) is a controlled language whose name stands for "Processable English." It is a rich but unambiguous language that can be automatically translated via discourse representation structures into first-order logic with equality. It is inspired by ACE, and the approach has a strong focus on predictive editing. These are two examples:

> Every animal A eats all plants or eats all animals B that are smaller than A and that eat some plants.
>
> While the fox sleeps, the cat chases a bird.

— $P^5E^3N^4S^3$, F W A

PENG-D (Schwitter and Tilbrook 2004) is a language derived from PENG, the main difference being that PENG-D builds upon RDF and OWL instead of discourse representation structures. — $P^5E^3N^4S^3$, F W A

PENG Light (Schwitter 2008) is another language derived from PENG. It maps to the TPTP notation for first-order logic. — $P^5E^3N^4S^3$, F W A

Perkins Approved Clear English (PACE) (Pym 1990) is a controlled language developed at Perkins, a diesel engine manufacturer and now a subsidiary of Caterpillar. The language was introduced in 1980 and was based on ILSAM. The goal was to improve machine-assisted translation. In order to avoid the use of synonyms, PACE comes with a dictionary that has been gradually extended and counted 2,500 entries in 1990, such as "passage (n): A drilling along which a fluid moves." PACE is summarized in "Ten Rules of Simplified Writing":

1. keep sentences short
2. omit redundant words
3. order the parts of the sentence logically
4. do not change constructions in mid sentence
5. take care with the logic of 'and' and 'or'
6. avoid elliptical constructions
7. do not omit conjunctions or relatives
8. adhere to the PACE dictionary
9. avoid strings of nouns
10. do not use 'ing' unless the word appears thus in the PACE dictionary

The aim of the first five rules is to make the text short and simple, and the last five rules have the somewhat opposing objective to make the text more explicit. This is an example consisting of two PACE sentences:

> Loosen the pivot fasteners of the dynamo or of the alternator. Loosen also the fasteners of the adjustment link.

— $P^2E^5N^5S^1$, C T W D I

PERMIS Controlled Natural Language (Inglesant et al. 2008) is a language for expressing access control policies for grid computing environments. It is based on CLOnE with specific extensions for authorization policies:

> Staff can print on HP Laserjet 1.
>
> I trust David to say who managers are.

Such statements are mapped to different formal target notations. Each statement follows one of only nine statement patterns. — $P^5E^2N^4S^4$, F W D A

"PILLS Language" (Bouayad-Agha, Power, and Belz 2002) is a language for medical information documents used in a system called PILLS. It follows a similar editing approach as the ICONOCLAST language, which was developed a couple of years earlier by the same research group. With the PILLS approach, different types of documents can be automatically generated from a master document and translated into different languages. — $P^4E^1N^5S^3$, T F D A

Plain Language or **Plain English** (SEC 1998; PLAIN 2011) is an initiative by the US government and other organizations. It had its origins in the 1970s with the goal to make official documents easier to understand and less bureaucratic. "Use pronouns to speak directly to readers" and "Avoid double negatives and exceptions to exceptions" are two exemplary rules. Unlike other such style guides, many of the guideline rules are strict and, with the Plain Writing Act of 2010, US governmental agencies are obliged to comply with them. With the focus being on human understandability and acceptance, documents in Plain Language do not seem to be considerably more precise or simpler from a computational point of view, when compared to full English. — $P^1E^5N^5S^1$, C W G

PoliceSpeak (Johnson 2000) is a language developed to improve police communications of English and French officers at the Channel Tunnel. The goal was to "make police communications more concise, more predictable, more stable and less ambiguous." The project was launched in 1988 and the language was ready in 1992. It has a similar goal and application area as SEASPEAK and the different air traffic control phraseologies. — $P^2E^1N^3S^2$, C S D G

"PROSPER Controlled English" (Grover et al. 2000) is a language for the specification and verification of hardware designs, developed in the late 1990s. The language is based on a restricted version of a general English grammar. Sentences of the language can be automatically mapped to a certain type of temporal logic. This is an exemplary sentence:

> If sigi is high and then is low on the next cycle, then sigo is low and after one cycle becomes high and then after one more cycle becomes low.

Ambiguity is not completely eliminated, but ambiguous sentences can be automatically spotted and reported to the user. — $P^4E^3N^4S^3$, F W D A

Pseudo Natural Language (PNL) (Marchiori 2004) is a language designed as a user-friendly language for the Semantic Web. It builds upon RDF and first-order logic, and uses Prolog to calculate inferences. These are two exemplary sentences:

> JOHN represents the person "John Smith" from the company "http:// www.example.com/staff".
>
> if IMPLY has as ARGUMENTS X and Y in this order, then X LOGICAL-IMPLY Y.

Upper-case words such as *JOHN* act as variables that can be instantiated with concrete definitions involving URIs. PNL is unambiguous and has well-defined semantics, but

unnatural capitalization mitigates the naturalness of the language. Its structure looks simple at first sight, but rather complex rules have to be applied in order to resolve ambiguous syntax trees. — $P^5E^3N^3S^3$, F W A

"Quelo Controlled English" (Franconi et al. 2011) is a language introduced in 2010 and used in a query interface called Quelo. This is an exemplary query:

> I am looking for something. It should be equipped with an automatic transmission system and sold by a car dealer. The car dealer should sell a fleet car.

Following a conceptual authoring approach, users cannot directly edit the sentences, but they can trigger modification actions on the underlying formal representation. — $P^4E^1N^4S^4$, F W A

Rabbit (Hart, Johnson, and Dolbear 2008) is a controlled language for OWL. It has been developed and used by Ordnance Survey, Great Britain's national mapping agency. Rabbit is designed for a specific scenario, in which it is used for the communication between domain experts and ontology engineers to create ontologies. Three types of statements are supported: declarations, axioms, and import statements. These are examples of the first and second type:

> Sheep is a concept, plural Sheep.

> Every River flows into exactly one of River, Lake or Sea.

The language is quite simple, being defined by a small number of sentence patterns and some modifications thereof. — $P^5E^2N^4S^4$, F W G

Restricted English for Constructing Ontologies (RECON) (Barkmeyer and Mattas 2012) is a language to represent facts and rules in an industrial environment, where these facts and rules have a deterministic mapping to first-order logic. This is an exemplary sentence:

> If any container contains part of a shipment, it contains no other shipment.

The language is defined by around 200 rules in Backus-Naur form. — $P^5E^3N^4S^3$, F W A G

Restricted Natural Language Statements (RNLS) (Breaux and Antón 2005; Breaux, Antón, and Doyle 2008) is a language for policy statements and software engineering goals introduced in 2004. The following are two exemplary RNLS statements:

> RNLS #1: The customer will select access codes.

> RNLS #2: The provider will recommend (RNLS #1) to the customer.

The second sentence refers to the first one using its identifier *RNLS #1*. There is a mapping between RNLS and Description Logic, but it is not clear whether this mapping is automated. — $P^3E^2N^4S^3$, F W D A

RuleSpeak (Ross 2003; OMG 2008; Ross 2013) is a CNL for business rules. The development of the language started in 1985 and it was first presented in 1994. It is very similar to SBVR Structured English, which emerged later. Each RuleSpeak rule belongs to one of eleven "functional categories" such as "computation rule," "inference rule," and "process trigger." For each of these categories specific templates are defined.

Computation rules, for example, contain the phrase "must be computed as" (or simply "="). The first of the following two examples is such a computation rule:

> A product's cost must be computed as the sum of the cost of all its components.

> An order may be accepted only if all of the following are true:
> - It includes at least one item.
> - It indicates the customer who is placing it.

Sometimes the color codes of SBVR Structured English are adopted to emphasize the different types of the sentence constituents. Like SBVR Structured English, RuleSpeak is linked to the SBVR standard, which provides formal semantics based on second-order logic with Henkin semantics. However, the mapping from RuleSpeak texts to the logical representation is only defined in an informal way. The strict templates considerably simplify the language, but there is no formal grammar that would fully define the language. — $P^3E^4N^4S^2$, C F W I

SBVR Structured English *See Section 4.1.* — $P^3E^4N^4S^2$, C F W I

SEASPEAK (Strevens and Johnson 1983) is an "International Maritime English" designed for clear communication among ships and harbors. Its development started in 1981. It is a controlled phraseology similar to PoliceSpeak and the different air traffic control phraseologies. — $P^2E^1N^3S^2$, C S D G

SMART Controlled English (Smart 2006) is a "more advanced version" of ASD Simplified Technical English, developed by the company SMART Communications. It was probably influenced by SMART Plain English, and has been applied to different areas. This is an excerpt of a document in SMART Controlled English:

> When the Quaternary Pump starts operation, the plunger moves inside the chamber. This movement lets the computer calculate and store a position called "Top Dead Center" (TDC).

The language is implemented in a tool called MAXit Checker, which is able to spot violations of the restrictions of the language. — $P^2E^5N^5S^1$, C T W I

SMART Plain English, sometimes called **Plain English Program (PEP)**, is a controlled language developed and used at SMART Communications since the mid 1980s.[6] It is based on CFE and was the basis for HELP and the controlled languages at Clark and Rockwell (Adriaens and Schreors 1992). As for SMART Controlled English, the tool MAXit Checker can be used to create compliant documents. — $P^2E^5N^5S^1$, C W I

"Sowa's syllogisms." *See Section 4.1.* — $P^5E^1N^4S^5$, F W A

Special English (Voice of America 2009) is a simplified English developed and used by the Voice of America, the official external broadcast institution of the US government. The language has been used since 1959 and is still used today for news on radio, television, and the Web. This makes it the second oldest English-based CNL (after Basic English) and the only one that has been in use for such a long period by the same organization. At the time of its creation, Special English was probably influenced by Basic English. The vocabulary is restricted to about 1,500 words, which have changed over time. Sentences should be short and should be spoken at a slower speed. There are no explicit restrictions on grammar or semantics. — $P^1E^5N^5S^1$, C W S G

6 http://www.smartny.com/plainEnglish.htm.

SQUALL (Ferré 2012) is a controlled natural language in the area of the Semantic Web to query and update RDF graphs. Sentences in this language are translated into the query language SPARQL, whereby structural ambiguity is resolved based on a few syntactic rules. This is an example:

> for every publication ?X, ?X has an author ?A and ?A cite-s ?X

The language is defined by about 50 simple grammar rules. — $P^5E^2N^3S^4$, F W A

Standard Language (SLANG). *See Section 4.1.* — $P^3E^1N^4S^2$, C F W D I

Sun Proof (Wells Akis and Sisson 2002) is a controlled language introduced at Sun for their technical documentation. The initial development of the language lasted from 1999 until 2002. The general objective was to write texts that are "easier to understand and to translate for humans as well as machines" but with a clear focus on translatability. Sun Proof is restricted by three sets of guidelines: style guidelines, grammar rules, and terminology. One of the most important rules is the limitation of the sentence length to 25 words. Other rules include semantic restrictions such as using *may* only for granting permission. This is an exemplary sentence:

> This chapter provides an overview of the standardized solutions that are required to make the transition from IPv4 to IPv6.

— $P^2E^5N^5S^1$, C T W D I

Sydney OWL Syntax (SOS) (Cregan, Schwitter, and Meyer 2007) is a controlled language introduced in the context of the Semantic Web. It is based on PENG and provides a bidirectional and complete mapping to the ontology language OWL. These are two exemplary sentences:

> The class adult is fully defined as any person that has at least 20 as an age.
>
> If X has Y as a father then Y is the only father of X.

— $P^5E^2N^4S^3$, F W A

Template Based Natural Language Specification (TBNLS) (Esser and Struss 2007) is a CNL approach for functional tests of control software for passenger vehicles. The language is defined by 15 templates that provide a mapping to propositional logic with temporal relations. This is an exemplary sentence:

> If $\boxed{\text{Button } B_4 \text{ is down} \quad P_1}$ occurs, then $\boxed{\text{Lamp } L_3 \text{ is red} \quad P_2}$ hold immediately, until $\boxed{\text{10 seconds} \quad T_1}$ elapsed.

P_1 and P_2 represent the propositional variables for the respective boxes, and T_1 is a time variable. — $P^5E^2N^3S^4$, F W D A I

ucsCNL (Barros et al. 2011) is a controlled natural language for use case specifications in the area of automated software testing. The language is intended to be unambiguous and is defined by a small number of simple grammar rules. There are imperative sentences to describe user actions, as well as declarative statements to describe the system state before and after user actions:

> After creating a message with 100 characters, go to the drafts folder
>
> The imported media file is a music file

— $P^5E^2N^4S^4$, F W D A

Voice Actions[7] are a CNL for spoken action commands on the Android mobile phone platform. Currently, the language covers twelve informally defined command patterns such as "map of," "note to self," and "create a calendar event." The following is an example:

> Create a calendar event: Dinner in San Francisco, Saturday at 7:00PM

These spoken commands can be automatically interpreted and executed by the system.
— $P^3E^1N^4S^2$, F S D I

Acknowledgments

I would like to thank Norbert E. Fuchs, Stefan Höfler, Kaarel Kaljurand, Rich Morin, Rolf Schwitter, Simon Spero, and David Whitten for comments on the article and general discussions on the topic. I am also thankful for the responses from Orlando Chiarello, Esra Erdem, Richard Power, Ronald G. Ross, Nestor Rychtyckyj, Donia Scott, Irina Temnikova, and Andries van Renssen to questions about specific languages. In addition, the feedback from Robert Dale, editor-in-chief of the *Computational Linguistics* journal, anonymous comments from its editorial board, and the anonymous reviews were very helpful to further improve the article. Lastly, I am extremely thankful to James Tierney for working with me on the manuscript to improve grammar and style.

References

Adriaens, Geert and Lieve Macken. 1995. Technological evaluation of a controlled language application: Precision, recall, and convergence tests for SECC. In *Proceedings of TMI95*, pages 123–141, Leuven.

Adriaens, Geert and Dirk Schreors. 1992. From COGRAM to ALCOGRAM: Toward a controlled English grammar checker. In *Proceedings of COLING '92*, pages 595–601, Nantes.

AECMA (Association Européenne des Constructeurs de Matériel Aérospatial). 1986. *AECMA Simplified English*. PSC-8S-16598.

Aikawa, Takako, Lee Schwartz, Ronit King, Monica Corston-Oliver, and Carmen Lozano. 2007. Impact of controlled language on translation quality and post-editing in a statistical machine translation environment. In *Proceedings of the MT Summit XI*, pages 1–7, Copenhagen.

ANSI/NISO (American National Standards Institute and National Information Standards Organization). 2005. *Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. Z39.19-2005.

Aristotle. ca. 350 BC. Prior analytics. `http://classics.mit.edu/Aristotle/prior.html`.

ASD (AeroSpace and Defence Industries Association of Europe). 2013. *Simplified Technical English*. Specification ASD-STE100, Issue 6.

Avaya Inc., 2004. *Avaya Style Guide*. Issue 1, Basking Ridge, NJ.

Barkmeyer, Edward and Andreas Mattas. 2012. A restricted English for constructing ontologies (RECON). NIST Interagency/Internal Report 7868, National Institute of Standards and Technology (NIST).

Barros, Flávia A., Laís Neves, Érica Hori, and Dante Torres. 2011. The ucsCNL: A controlled natural language for use case specifications. In *Proceedings of SEKE'2011*, pages 250–253, Miami Beach, FL.

Bernardi, Raffaella, Diego Calvanese, and Camilo Thorne. 2007. Lite natural language. In *Proceedings of IWCS-7*, 12 pages, Tilburg.

Bernstein, Abraham and Esther Kaufmann. 2006. GINO—a guided input natural language ontology editor. In *Proceedings of ISWC 2006*, pages 144–157, Athens, GA.

Bernstein, Abraham, Esther Kaufmann, Christian Kaiser, and Christoph Kiefer. 2006. Ginseng: A guided input natural language search engine for querying ontologies. In *2006 Jena User Conference*, 3 pages.

Bernth, Arendse. 1997. EasyEnglish: A tool for improving document quality. In *Proceedings of ANLC '97*, pages 159–165, Washington, DC.

Bernth, Arendse. 2006. EasyEnglishAnalyzer: Taking controlled language from sentence

7 `http://support.google.com/android/bin/answer.py?hl=en&answer=1715292`.

to discourse level. In *Proceedings of CLAW 2006*, Cambridge, MA.
Betts, Robert. 2003. EasyEnglish: Challenges in cross-cultural communication. In *Proceedings of EAMT-CLAW03*, 8 pages, Dublin.
Bouayad-Agha, Nadjet, Richard Power, and Anja Belz. 2002. PILLS: Multilingual generation of medical information documents with overlapping content. In *Proceedings of LREC 2002*, pages 2,111–2,114, Las Palmas.
Bourland, D. David. 1965. A linguistic note: Writing in E-prime. *General Semantics Bulletin*, 32(3):111–114.
Boyd, Stephen, Didar Zowghi, and Alia Farroukh. 2005. Measuring the expressiveness of a constrained natural language: An empirical study. In *Proceedings of RE 2005*, pages 339–352, Paris.
Breaux, Travis D. and Annie I. Antón. 2005. Deriving semantic models from privacy policies. In *Proceedings of POLICY 2005*, pages 67–76, Stockholm.
Breaux, Travis D., Annie I. Antón, and Jon Doyle. 2008. Semantic parameterization: A process for modeling domain descriptions. *ACM Transactions on Software Engineering and Methodology*, 18(2):5:1–5:27.
CAA (Civil Aviation Authority). 2011. *Radiotelephony Manual*, 20th edition, West Sussex, UK
Chervak, Steve, Colin G. Drury, and James P. Ouellette. 1996. Field evaluation of simplified English for aircraft workcards. In *Proceedings of the Tenth Meeting on Human Factors Issues in Aircraft Maintenance and Inspection*, pages 123–136, Washington, DC.
Clark, Peter, Shaw-Yi Chaw, Ken Barker, Vinay Chaudhri, Philip Harrison, James Fan, Bonnie John, Bruce Porter, Aaron Spaulding, John Thompson, and Peter Yeh. 2007. Capturing and answering questions posed to a knowledge-based system. In *Proceedings of K-CAP '07*, pages 63–70, Whistler.
Clark, Peter, Phil Harrison, Thomas Jenkins, John Thompson, and Richard H. Wojcik. 2005. Acquiring and using world knowledge using a restricted subset of English. In *Proceedings of FLAIRS 2005*, pages 506–511, Clearwater Beach, FL.
Clark, Peter, Phil Harrison, William R. Murray, and John Thompson. 2010. Naturalness vs. predictability: A key debate in controlled languages. In *Proceedings of CNL 2009*, pages 65–81, Marettimo Island.
Crabbe, Stephen. 2009. Controlled languages for technical writing and translation. In *Proceedings of the 9th Portsmouth Translation Conference*, pages 48–62, Portsmouth.
Cramer, Marcos, Bernhard Fisseni, Peter Koepke, Daniel Kühlwein, Bernhard Schröder, and Jip Veldman. 2010. The Naproche Project controlled natural language proof checking of mathematical texts. In *Proceedings of CNL 2009*, pages 170–186, Marettimo Island.
Cregan, Anne, Rolf Schwitter, and Thomas Meyer. 2007. Sydney OWL Syntax—towards a controlled natural language syntax for OWL 1.1. In *Proceedings OWLED 2007*, 10 pages, Innsbruck.
Cucchiarini, Catia. 2002. Euromap HLT case study: How HLT applications can lead to higher quality translations at lower costs: The experience of Océ Technologies.
Dimitrova, Vania, Ronald Denaux, Glen Hart, Catherine Dolbear, Ian Holt, and Anthony G. Cohn. 2008. Involving domain experts in authoring OWL ontologies. In *Proceedings of ISWC 2008*, pages 1–16, Karlsruhe.
Erdem, Esra and Reyyan Yeniterzi. 2009. Transforming controlled natural language biomedical queries into answer set programs. In *Proceedings of BioNLP '09*, pages 117–124, Boulder, CO.
Esser, M. W. and P. Struss. 2007. Obtaining models for test generation from natural-language-like functional specifications. In *Proceedings of DX-07*, pages 75–82, Nashville, TN.
Eurocontrol. 2009. *ICAO Standard Phraseology—A Quick Reference Guide for Commercial Air Transport Pilots*, Brussels.
European Commission. 2011. *How to write clearly*, Brussels.
FAA (Federal Aviation Administration). 2010. *Air Traffic Control*. Order JO 7110.65T.
Felleisen, Matthias. 1991. On the expressive power of programming languages. *Science of Computer Programming*, 17(1-3):35–75.
Ferré, Sébastien. 2012. SQUALL: A controlled natural language for querying and updating RDF graphs. *LNCS*, 7427:11–25.
Franconi, Enrico, Paolo Guagliardo, Sergio Tessaris, and Marco Trevisan. 2011. Quelo: An ontology-driven query interface. In *Proceedings of DL 2011*, pages 488–498, Barcelona.
Fuchs, Norbert E., Kaarel Kaljurand, and Tobias Kuhn. 2008. Attempto Controlled

English for knowledge representation. In *Reasoning Web—4th International Summer School 2008*, pages 104–124.

Fuchs, Norbert E. and Rolf Schwitter. 1995. Specifying logic programs in controlled natural language. In *Proceedings of CLNLP 95*, 16 pages, Edinburgh.

Funk, Adam, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, Brian Davis, and Siegfried Handschuh. 2007. CLOnE: Controlled language for ontology editing. In *Proceedings of ISWC 2007 + ASWC 2007*, pages 142–155, Busan.

Godden, Kurt. 2000. The evolution of CASL controlled authoring at General Motors. In *Proceedings of CLAW 2000*, pages 14–19, Seattle, WA.

Grover, Claire, Alexander Holt, Ewan Klein, and Marc Moens. 2000. Designing a controlled language for interactive model checking. In *Proceedings of CLAW 2000*, pages 29–30, Seattle, WA.

Hallett, Catalina, Donia Scott, and Richard Power. 2007. Composing questions through conceptual authoring. *Computational Linguistics*, 33(1):105–133.

Halpin, Terry A. 2004. Business rule verbalization. In *Proceedings of ISTA 2004*, pages 39–52, Salt Lake City, UT.

Hart, Glen, Martina Johnson, and Catherine Dolbear. 2008. Rabbit: Developing a controlled natural language for authoring ontologies. *LNCS*, 5021:348–360.

Hayes, Phil, Steve Maxwell, and Linda Schmandt. 1996. Controlled English advantages for translated and original English documents. In *Proceedings of CLAW 1996*, pages 84–92, Leuven.

Horridge, Matthew, Nick Drummond, John Goodwin, Alan L. Rector, Robert Stevens, and Hai Wang. 2006. The Manchester OWL syntax. In *Proceedings of OWLED '06*, 10 pages, Athens, GA.

Houghton Mifflin Harcourt. 2000. *The American Heritage Dictionary of the English Language*, fourth edition.

Huijsen, Willem-Olaf. 1998. Controlled language—an introduction. In *Proceedings of CLAW '98*, pages 1–15, Pittsburgh, PA.

Humayoun, Muhammad and Christophe Raffalli. 2010. MathNat—mathematical text in a controlled natural language. *Journal on Research in Computing Science—Special issue: Natural Language Processing and its Applications*, 46:293–307.

iLastic. 2012. *iLastic — Documentation*. `http://www.ilastic.com/docs`.

Inglesant, Philip, M. Angela Sasse, David Chadwick, and Lei Lei Shi. 2008. Expressions of expertness: The virtuous circle of natural language for access control policy specification. In *Proceedings of SOUPS '08*, pages 77–88, Pittsburgh, PA.

Jarrar, Mustafa, C. Maria Keet, and Paolo Dongilli. 2006. Multilingual verbalization of ORM conceptual models and axiomatized ontologies. Technical report, Vrije Universiteit, Brussels.

Johnson, Edward. 2000. Talking across frontiers. In *Proceedings of the International Conference on European Cross Border Cooperation: Lessons for and from Ireland*, 23 pages, Belfast.

Kaljurand, Kaarel. 2007. *Attempto Controlled English as a Semantic Web Language*. Ph.D. thesis, Faculty of Mathematics and Computer Science, University of Tartu, Estonia.

Kaljurand, Kaarel and Norbert E. Fuchs. 2006. Bidirectional mapping between OWL DL and Attempto Controlled English. In *Proceedings of PPSWR'06*, pages 179–189, Budva.

Kaljurand, Kaarel and Tobias Kuhn. 2013. A multilingual semantic wiki based on Attempto Controlled English and Grammatical Framework. In *Proceedings of ESWC 2013*, pages 427–441, Moutpellier.

Kamprath, Christine, Eric Adolphson, Teruko Mitamura, and Eric Nyberg. 1998. Controlled language for multilingual document production: Experience with Caterpillar technical English. In *Proceedings of CLAW '98*, pages 51–61, Pittsburgh, PA.

Karkaletsis, Vangelis and Constantine D. Spyropoulos. 1997. A knowledge-based organization of lexical resources to support multilingual information retrieval in software localization. In *Proceedings of the 1997 AAAI Symposium on Cross-Language and Speech Retrieval*, pages 120–126, Stanford, CA.

Kim, Hong-Gee, Byung-Hyun Ha, Jae-Il Lee, and Myeng-Ki Kim. 2005. A multi-layered application for the gross description using Semantic Web technology. *International Journal of Medical Informatics*, 74(5):399–407.

Kittredge, Richard I. 2003. Sublanguages and controlled languages. In Ruslan Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, pages 430–447.

Kleinman, Joseph M. 1982. A limited-word technical dictionary for technical manuals. *Technical Communication*, Q1:16–19.

Kuhn, Tobias. 2009. How controlled English can improve semantic wikis. In *Proceedings of SemWiki 2009*, pages 1–15, Hersonissos.
Kuhn, Tobias. 2010. *Controlled English for Knowledge Representation*. Ph.D. thesis, Faculty of Economics, Business Administration and Information Technology of the University of Zurich, Switzerland.
Kuhn, Tobias. 2013. The understandability of OWL statements in controlled English. *Semantic Web*, 4(1):101–115.
Kuhn, Tobias, Paolo Emilio Barbano, Mate Levente Nagy, and Michael Krauthammer. 2013. Broadening the scope of nanopublications. In *Proceedings of ESWC 2013*, pages 487–501, Montpellier.
Kuhn, Tobias and Stefan Höfler. 2012. Coral: Corpus access in controlled language. *Corpora*, 7(2):187–206.
Lukichev, Sergey and Gerd Wagner. 2006. Verbalization of the REWERSE I1 rule markup language. Deliverable I1-D6, REWERSE, Munich.
Marchiori, Massimo. 2004. Towards a people's Web: Metalog. In *Proceedings of WI 2004*, pages 320–326, Beijing.
Martin, Philippe. 2002. Knowledge representation in CGLF, CGIF, KIF, Frame-CG and Formalized-English. *LNAI*, 2393:77–91.
Massachusetts Senate. 2003. *Legislative Drafting and Legal Manual*, third edition, Boston.
Means, Linda and Kurt Godden. 1996. The Controlled Automotive Service Language (CASL) project. In *Proceedings of CLAW 1996*, pages 106–114, Leuven.
Means, Linda G., Patricia Chapman, and Aulsan Liu. 2000. Training for controlled language processes. In *Proceedings of CLAW 2000*, pages 1–13, Seattle, WA.
Mitamura, Teruko and Eric H. Nyberg. 1995. Controlled English for knowledge-based MT: Experience with the KANT system. In *Proceedings of TMI95*, pages 158–172, Louvain.
Moore, Corinne. 2000. Controlled language at Diebold, Inc. In *Proceedings of CLAW 2000*, pages 51–61, Seattle, WA.
Mott, David. 2010. Summary of ITA controlled English. Technical report, International Technology Alliance (ITA).
Muegge, Uwe. 2007. Controlled language: The next big thing in translation? *ClientSide News Magazine*, 7:21–24.
Namgoong, Hyun and Hong-Gee Kim. 2007. Ontology-based controlled natural language editor using CFG with lexical dependency. In *Proceedings of ISWC 2007 + ASWC 2007*, pages 353–366, Busan.
NCR Corporation. 1978. *NCR Fundamental English Dictionary*, Dayton, OH.
Nečas, Ivan. 2011. BDD as a specification and QA instrument. Master's thesis, Masaryk University, Brno, Czech Republic.
Nyberg, Eric, Teruko Mitamura, and Willem-Olaf Huijsen. 2003. Controlled language for authoring and translation. In Harold Somers, editor, *Computers and Translation: A Translator's Guide*. John Benjamins Publishing Company, pages 245–281.
O'Brien, Sharon. 2003. Controlling controlled English—an analysis of several controlled language rule sets. In *Proceedings of EAMT-CLAW03*, pages 105–114, Dublin.
O'Brien, Sharon and Johann Roturier. 2007. How portable are controlled language rules? A comparison of two empirical MT studies. In *Proceedings of MT Summit XI*, pages 345–352, Dublin.
Ogden, Charles K. 1930. *Basic English: A general introduction with rules and grammar*. Paul Treber & Co., London.
OMG (Object Management Group). 2008. *Semantics of Business Vocabulary and Business Rules (SBVR), v1.0*. `http://www.omg.org/spec/SBVR/1.0/PDF`.
Pease, Adam and John Li. 2010. Controlled English to logic translation. In Roberto Poli, Michael Healy, and Achilles Kameas, editors, *Theory and Applications of Ontology: Computer Applications*. Springer Netherlands, pages 245–258.
Piwek, Paul, Roger Evans, Lynne Cahill, and Neil Tipper. 2000. Natural language generation in the mile system. In *Proceedings of the IMPACTS in NLG Workshop*, pages 33–42, Dagstuhl.
PLAIN (Plain Language Action and Information Network). 2011. *Federal Plain Language Guidelines*. `plainlanguage.gov`.
Pool, Jonathan. 2006. Can controlled languages scale to the Web? In *Proceedings of CLAW 2006*, Cambridge, MA.
Power, Richard. 1999. Controlling logical scope in text generation. In *Proceedings of EWNLG 1999*, pages 1–9, Toulouse.
Power, Richard. 2012. OWL Simplified English: A finite-state language for ontology editing. In *Proceedings of CNL*, pages 44–60, Zurich.
Power, Richard and Donia Scott. 1998. Multilingual authoring using feedback texts. In *Proceedings of COLING-ACL '98*, pages 1,053–1,059, Montreal.

Pratt-Hartmann, Ian. 2003. A two-variable fragment of English. *Journal of Logic, Language and Information*, 12(1):13–45.

Pratt-Hartmann, Ian. 2004. Fragments of language. *Journal of Logic, Language and Information*, 13(2):207–223.

Pratt-Hartmann, Ian. 2009. Computational complexity of controlled natural languages. In *Pre-Proceedings of CNL 2009*, 5 pages, Marettimo Island.

Pratt-Hartmann, Ian and Allan Third. 2006. More fragments of language. *Notre Dame Journal of Formal Logic*, 47(2):151–177.

Pulman, Stephen G. 1996. Controlled language for knowledge representation. In *Proceedings of CLAW 1996*, pages 233–242, Leuven.

Pym, Peter J. 1990. Pre-editing and the use of simplified writing for MT: An engineer's experience of operating an MT system. In *Translating and the Computer 10: The Translation Environment 10 Years On*, number 10, pages 80–96, Aslib.

Reuther, Ursula. 2003. Two in one: Can it work? Readability and translatability by means of controlled language. In *Proceedings of EAMT-CLAW03*, pages 124–132, Dublin.

Robertson, F. A. 1987. *AIRSPEAK—Radiotelephony Communication for Pilots*. Prentice Hall.

Ross, Ronald G. 2003. *Principles of the Business Rule Approach*. Information Technology Series. Addison-Wesley.

Ross, Ronald G. 2013. Tabulation of lists in RuleSpeak—using "the following" clause. *Business Rules Journal*, 14(4):1–16.

Ruffino, J. Richard. 1982. Coping with machine translation. In V. Lawson, editor, *Practical Experience of Machine Translation*. North-Holland Publishing Company, pages 57–60.

Rychtyckyj, Nestor. 2002. An assessment of machine translation for vehicle assembly process planning at Ford motor company. In *Proceedings of AMTA2002*, pages 207–215, Tiburon, CA.

Rychtyckyj, Nestor. 2005. Ergonomics analysis for vehicle assembly using artificial intelligence. *AI Magazine*, 26(3):41–50.

Schwitter, Rolf. 2002. English as a formal specification language. In *Proceedings of DEXA '02*, pages 228–232, Aix-en-Provence.

Schwitter, Rolf. 2008. Working for two: A bidirectional grammar for a controlled natural language. In *Proceedings of AI 2008*, pages 168–179, Auckland.

Schwitter, Rolf and Marc Tilbrook. 2004. Controlled natural language meets the semantic Web. In *Proceedings of ALTW2004*, pages 55–62, Dublin.

SEC (U.S. Securities and Exchange Commission). 1998. *A Plain English Handbook—How to Create Clear SEC Disclosure Documents*, New York.

Shubert, Serena K., Jan H. Spyridakis, Heather K. Holmback, and Mary B. Coney. 1995. The comprehensibility of simplified English in procedures. *Journal of Technical Writing and Communication*, 25(4):347–369.

Skuce, Doug. 2003. A controlled language for knowledge formulation on the semantic Web. `http://www.site.uottawa.ca:4321/factguru2.pdf`.

Smart, John M. 2003. Controlled English for global business. *Writing for Translation—The Guide from MultiLingual Computing & Technology*, 59:19–21.

Smart, John M. 2006. SMART controlled English. In *Proceedings of CLAW 2006*, 9 pages, Cambridge, MA.

Smart Communications Inc. 1994. News from Smart Communications, Inc. In MT News International—Newsletter of the International Association for Machine Translation, Issue no. 7.

Sowa, John F. 2000a. Controlled English. `http://users.bestweb.net/~sowa/misc/ace.htm`.

Sowa, John F. 2000b. Ontology, metadata, and semiotics. In *Proceedings of ICCS 2000*, pages 55–81, Darmstadt.

Sowa, John F. 2004. Common logic controlled English (draft). `http://www.jfsowa.com/clce/specs.htm`.

Spaggiari, Laurent, Florence Beaujard, and Emmanuelle Cannesson. 2003. A controlled language at Airbus. In *Proceedings of EAMT-CLAW03*, pages 151–159, Dublin.

Stewart, Kathleen M. 1998. Effect of AECMA simplified English on the comprehension of aircraft maintenance procedures by non-native English speakers. Master's thesis, University of British Columbia.

Strevens, Peter and Edward Johnson. 1983. SEASPEAK: A project in applied linguistics, language engineering, and eventually ESP for sailors. *ESP Journal*, 2(2):123–129.

Sukkarieh, Jana Z. 2003. Mind your language! Controlled language for inference purposes. In *Proceedings of EAMT-CLAW03*, pages 160–169, Dublin.

Sukkarieh, Jana Z. and Stephen G. Pulman. 1999. Computer processable English and

mclogic. In *Proceedings of IWCS-3*, pages 367–380, Tilburg.
Temnikova, Irina. 2010. Cognitive evaluation approach for a controlled language post-editing experiment. In *Proceedings of LREC'10*, pages 3,485–3,490, Hissar.
Temnikova, Irina. 2011. Establishing implementation priorities in aiding writers of controlled crisis management texts. In *Proceedings of RANLP 2011*, pages 654–659.
Temnikova, Irina. 2012. *Text Complexity and Text Simplification in the Crisis Management Domain*. Ph.D. thesis, University of Wolverhampton.
Temnikova, Irina and Constantin Orasan. 2009. Post-editing experiments with mt for a controlled language. In *Proceedings of ISMTCL*, pages 244–248, Besançon.
Thorne, Camilo and Diego Calvanese. 2010. Controlled English ontology-based data access. In *Proceedings of CNL 2009*, pages 135–154, Marettimo Island.
Valencia-García, Rafael, Francisco García-Sánchez, Dagoberto Castellanos-Nieves, and Jesualdo Tomás Fernández-Breis. 2011. OWLPath: An OWL ontology-guided query editor. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 41(1):121–136.
Van Kleek, Max, Brennan Moore, David Karger, Paul André, and M. C. Schraefel. 2010. Atomate it! End-user context-sensitive automation using heterogeneous information sources on the Web. In *Proceedings of WWW 2010*, pages 951–960, Raleigh, NC.
van Renssen, Andries. 2005. *Gellish: A Generic Extensible Ontological Language*. Ph.D. thesis, Delft University of Technology.
van Renssen, Andries. 2011. Modeling of textual requirements in a Gellish universal database. In *Proceedings of FOMI 2011*, pages 102–115, Vincenza.
Verbeke, Charles A. 1973. Caterpillar fundamental English. *Training & Development Journal*, 27(2):36–40.
Vershinin, Konstantin and Andrey Paskevich. 2000. Forthel—the language of formal theories. *International Journal of Information Theories and Applications*, 7(3):120–126.
Voice of America. 2009. *VOA Special English Word Book: A List of Words Used in Special English Programs on Radio, Television, and the Internet*, Washington, DC.
Warren, David H. D. and Fernando C. N. Pereira. 1982. An efficient easily adaptable system for interpreting natural language queries. *American Journal of Computational Linguistics*, 8(3-4):110–122.
Wells Akis, Jennifer and William R. Sisson. 2002. Improving translatability: A case study at Sun Microsystems, Inc. *The Globalization Insider*, (4.2).
Wojcik, Richard H., Philip Harrison, and John Bremer. 1993. Using bracketed parses to evaluate a grammar checking application. In *Proceedings of ACL '93*, pages 38–45, Columbus, OH.
Wojcik, Richard H. and James E. Hoard. 1997. Controlled languages in industry. In R. A. Cole et al., editors. *Survey of the State of the Art in Human Language Technology*. Cambridge University Press, pages 238–239.
Wojcik, Richard H., Heather Holmback, and James E. Hoard. 1998. Boeing Technical English: An extension of AECMA SE beyond the aircraft maintenance domain. In *Proceedings of CLAW '98*, pages 114–123, Pittsburgh, PA.
Wyner, Adam, Krasimir Angelov, Guntis Barzdins, Danica Damljanovic, Brian Davis, Norbert Fuchs, Stefan Hoefler, Ken Jones, Kaarel Kaljurand, Tobias Kuhn, Martin Luts, Jonathan Pool, Mike Rosner, Rolf Schwitter, and John Sowa. 2010. On controlled natural languages: Properties and prospects. In *Proceedings of CNL 2009*, pages 281–289, Marettimo Island.

This article has been cited by:

1. Tobias Kuhn, Alexandre Bergel. 2014. Verifiable source code documentation in controlled natural language. *Science of Computer Programming* . [CrossRef]