A Framework for Comparing the use of a Linguistic Ontology in an Application

Julieanne van Zyl[1] and **Dan Corbett**[2]

Abstract. A framework recently developed for understanding and classifying ontology applications provides opportunities to review the state of the art, and to provide guidelines for application developers from different communities [1]. The framework identifies four main categories of ontology applications: neutral authoring, ontology as specification, common access to information, and ontology-based search. Specific scenarios are described for each category and a number of features have been identified to highlight the similarities and differences between them. In this paper we identify additional scenarios, describing the use of a linguistic ontology in an application. We populate the scenarios with a number of prominent research and industrial applications from diverse communities such as knowledge-based machine translation, medical databases, heterogenous information systems integration, multilingual text generation, Web-based applications, and information retrieval. Population of the framework should allow different communities to discover applications that fulfil specific purposes and benefits, to discover what roles the ontology plays, who the principle actors are and what they do, and what supporting technologies are used for these applications. Potential application developers can examine the descriptions in the populated framework to inspire them to use existing linguistic ontologies for their specific applications.

1 INTRODUCTION

Over the past few years, there have been a number of reports of applications using a linguistic ontology. In addition to the more traditional use of these ontologies for natural language generation and for machine translation, these applications are being used for specifying the meaning of text in a specific domain, intelligent information retrieval, and integration of heterogeneous information systems. The ontology serves as a neutral format so that other applications can access it, or so that one or more persons can understand it in another natural language. Some applications rely on *one* natural language and others are multilingual and allow translation between *more than one* natural language. Linguistic ontologies usually have the purpose of solving problems such as: how knowledge of the world is to be represented, and how such organisations of knowledge are to be related to natural language generation (NLG) system levels of organisation such as grammar and lexicons. Applications using a linguistic ontology often allow the user to express their queries in a natural language, and the information provided to the application user might also be in a natural language. Other benefits of this approach include inter-operability of software tools, and more effective use and reuse of knowledge resources. Mahesh claims that an ontology for NLG purposes is:

> *"a body of knowledge about the world (or a domain) that a) is a repository of primitive symbols used in meaning*
> *b) organises these symbols in a tangled subsumption hierarchy; and*
> *c) further interconnects these symbols using a rich system of semantic and discourse-pragmatic relations defined among the concepts"* [2].

There has been some work on comparing ontology applications, which refer to "the application that makes use of or benefits from the ontology (possibly directly)" [1]. A framework [3] was developed for comparing various projects in ontology design, which considered the differences and similarities in the way the projects treat some basic knowledge representation aspects. The projects compared include three natural language ontologies, two of which are ontologies, and not ontology applications. Jasper and Uschold [1] developed a framework (henceforth JUFramework) for understanding and classifying ontology *applications* into different categories, and identified a number of scenarios for each. It proposed a common nomenclature for different communities (ontology research groups, software developers and standards organisations who are addressing similar problems) to overcome terminological confusion.

The Reference Ontology [4] was constructed to enable users to compare and search for existing ontologies on the World Wide Web (WWW). The most significant difference between this ontology and the JUFramework is the *audience* that each of them addresses. Potential *application developers* from different communities are addressed in the JUFramework, whereas the Reference Ontology addresses potential *users* of existing Web-based ontologies. The Reference Ontology is ideal for helping users to *select* and *retrieve* the most adequate and suitable ontology for the application they have in mind. In contrast, the JUFramework should eventually provide *guidelines for potential application developers* from diverse communities to be guided in an approach to use in developing ontology applications under their specific circumstances. A number of examples have been identified for each scenario, but they have not been described in detail within the framework.

To our knowledge there is no framework for comparing *applications* that use a linguistic ontology, so the significance of our paper is the extension of the JUFramework to include scenarios for these kinds of applications. In addition, we have populated the framework with some prominent and existing applications, to demonstrate that a linguistic ontology can be used in a number of different kinds of applications. Population of the framework also provides potential ontology developers with the ability to examine a particular application, and find its purposes and benefits, the role of the ontology, the supporting technologies, who the principle actors are and what they do. If a developer wants to construct a new ontology application, the populated framework can be searched to identify and compare existing techniques for using a linguistic ontology. Developers can decide which scenario their new application fits into and obtain ideas for constructing it. This can provide substantial benefits in sharing and reusing the techniques used to construct a specific ontology application, and

[1] School of Computer and Information Science, University of South Australia, The Levels Campus, Mawson Lakes, 5095, Australia, email: zyl@cs.unisa.edu.au
[2] School of Computer and Information Science, University of South Australia, The Levels Campus, Mawson Lakes, 5095, Australia email: corbett@cs.unisa.edu.au

also the technologies that are used in specific ontology applications. Within the JUFramework, the meaning of ontology is interpreted very broadly, so that similarities of both goals and technologies developed to achieve them can be demonstrated across different communities. Rather than define the word ontology, the following characterisation is adopted:

> *An ontology may take a variety of forms, but necessarily it will include a vocabulary of terms and some specification of their meaning. This includes definitions and an indication of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms.* [1]

The next section in this paper outlines the framework dimensions used to describe each scenario. The main section populates the framework with a description of those applications that are more prominent and well documented within the research literature. Next, we compare the applications described and conclude with some future research in this area.

2 FRAMEWORK DIMENSIONS

To achieve the JUFramework's goal of easily comparing scenarios for ontology applications, each is presented in a uniform way using the same features (called framework dimensions and distinctions), and we describe these in the remainder of this section. To allow specific comparison of NLG applications, we have extended the 'purposes and benefits' dimensions for NLG applications, and included additional dimensions called 'ontology acquisition' and 'language dependency'.

1. The JUFramework groups the purposes and benefits of ontologies into the three main areas. Bateman [5] describes some functions that linguistic ontologies fulfil, and we specify these along with the purposes and benefits from the JUFramework:
   - *Communication* between people, where a linguistic ontology organises the 'semantics' of natural language expressions.
   - *Inter-operability* among computer systems, where the ontology may: a) be used as an interchange format, b) provide an interface between system external components, domain models, and NLG components, c) ensure expressability of input expressions, d) offer an interlingua for machine translation, and/or e) organise 'semantics' of natural language expressions.
   - *Systems Engineering*, including *reusability*, *search*, r*eliability*, s*pecification*, *maintenance*, and *knowledge acquisition.* A linguistic ontology may be used as a (partial) specification for an application, where it organises 'world knowledge', or attempts to organise the world itself, and/or supports the construction of 'conceptual dictionaries'.

2. The information within a particular application can play different ***roles***, which can be thought of as the following information levels:
   - L_0: *Operational Data* is a role that information plays, where it is consumed and produced by applications during run-time. Information at this level is written using terms from a vocabulary defined at L_1.
   - L_1: *Ontology* is a role that information plays, where it specifies terms and definitions for important concepts in some domain. The information at this level provides a vocabulary for the language used to author information at L_0.
   - L_2: An *Ontology Representation Language* is a role that information plays when the ontology authors or application developers use it to write an ontology at L_1 during the development process.

3. Each scenario in the framework involves a set of ***actors***. Each actor represents a role that a person or application may play:
   Ontology Author (OA) is the author of the ontology.
   (Operational) Data Author (DA) is the author of operational data in the language that uses and/or is defined in terms of the vocabulary of the ontology.
   Application Developer (AD) is the developer of the Application.
   Application User (AU) is the user of the Application.
   Knowledge Worker (KW) is the person who makes use of the knowledge [1].

4. There are a number of ***supporting technologies*** for the ontology applications, including: the languages that the ontologies are represented in, the languages that they are implemented into, the knowledge interchange languages, translation tools, and distributed objects that communicate information between applications.

5. Applications and their supporting technologies have ***different levels of maturity***: An ontology application may be an untested idea, or a specification for a class of potential applications. Implemented applications can vary from tiny scale demonstrations of feasibility in a research environment to applications in a commercial environment.

6. An ontology can ***represent its meaning*** in a number of different ways. Four notional points are identified along a continuum of formality: 1) highly informal: expressed loosely in natural language; 2) structured-informal: expressed in a restricted and structured form of natural language; 3) semi-formal: expressed in an artificial formally defined language; 4) rigorously formal with meticulously defined terms.

7. ***Different architectures*** are appropriate for accessing information resources. Some applications provide for *sharing*, where multiple agents reference a common piece of information. Other applications *exchange* by copying the data between them. This distinction is not always discussed because it is often difficult to know which architecture is used in the applications.

8. ***Ontology acquisition*** explains the different goals that applications have for acquiring concepts for the ontology. A decision has to be made concerning "what knowledge to acquire, formulating the knowledge according to the principles and guidelines behind the ontology, and then actually representing the knowledge according to the structure and axiomatic semantics of the ontology " [2]. The majority of NLG applications use the lexicon driven approach. For each application, we state one of the following approaches to ontology acquisition:
   - *Encyclopedia driven*, where the goal is to cover an entire encyclopedia rather than the particular conceptual knowledge that may be needed for a domain or a task.
   - *Domain analysis driven*, where a small domain is chosen and the resulting ontology is useful only for that domain.
   - *Task-driven*, where the focus is on the needs of a task such

as database merging or integration of business enterprise models. The resulting ontology may contain only knowledge needed for performing a task and hence may not cover a domain fully.

- *Lexicon-driven*, where the goal is to include all concepts that are necessary to represent the meanings of words in a lexicon. The resulting ontology may incorporate specific nuances in word meanings from a particular language, especially when a lexicon for a single natural language is being developed.
- *Formal* approach, where the goal is to adhere to a formalism and to maintain the formal cleanliness of the ontology at all costs. The resulting ontology may not be large enough to serve any domain or task.
- *Situated Development* where concepts in the ontology are acquired incrementally in the domain of the input texts, relying on continuous interactions with lexicon and semantic analyser teams.
- *Grammatically-driven*, where the goal is to include all concepts necessary to represent the grammar of a language. The concepts are "accounts of the semantics that may be expressed in grammatical units such as clauses, nominal groups, circumstantial phrases, etc." *[6]*.

9. ***Language Dependency****:* Some NLG applications rely on *one language* to interpret meanings of concepts, and others are quite *independent* of the language used and are usually multilingual.

3 NLG ONTOLOGY APPLICATION SCENARIOS

The JUFramework classifies applications that use one or more ontologies into four different categories and identifies a number of scenarios for each. We have extended the framework by including an additional scenario within each category for NLG applications, and we populate these scenarios with descriptions of applications. Each example is illustrated with a diagram similar to those used to illustrate the abstract scenarios, except that our diagrams are specific to the application described. Our diagrams are highly simplified and illustrate only the general features of an application, excluding the details, so they can be compared with each other. The rest of this section describes the additional scenario for each of the JUFramework categories, and the applications that we have chosen as examples.

3.1. Neutral Authoring to Support Machine Translation

This category is different to 'neutral authoring' scenario within the JUFramework. The JUFramework describes applications of ontologies where the ontology is authored in a neutral representation so that it can be translated and used in multiple target systems. An NLG application supports machine translation by providing *language-neutral terms* to which lexical terms of different languages can be attached. These applications are often Knowledge-Based Machine Translation (KBMT) systems, which translate the meanings (semantics) of text in one natural language to another natural language. The representations of meanings are captured by the language independent ontology, which acts as an interlingua. The principal reasons for using an ontology are: to provide a grounding for representing text meaning in an interlingua; to enable lexicons for different languages to share knowledge; and to enable source language analysers and target language generators to share knowledge [2]. Table 1 describes two KBMT applications to instantiate this scenario.

PANGLOSS [7, 8] translates unrestricted ASCII text files from Spanish to English. The PANGLOSS ontology is SENSUS, which is a large-scale conceptual network for supporting semantic processing in other PANGLOSS modules. The application is illustrated in Figure 1.

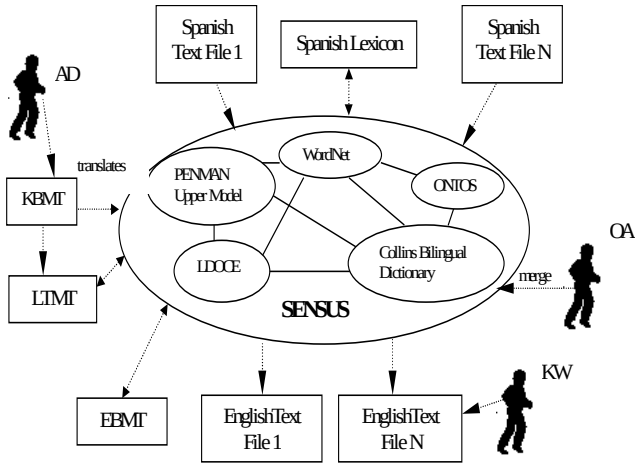

Figure 1: Pangloss

Mikrokosmos (μK) [9] translates Spanish and Chinese news articles into English. The main goal of μK is to develop a system that produces a comprehensive Text Meaning Representation (TMR) for an input text in a number of source languages. This is illustrated in Figure 2.

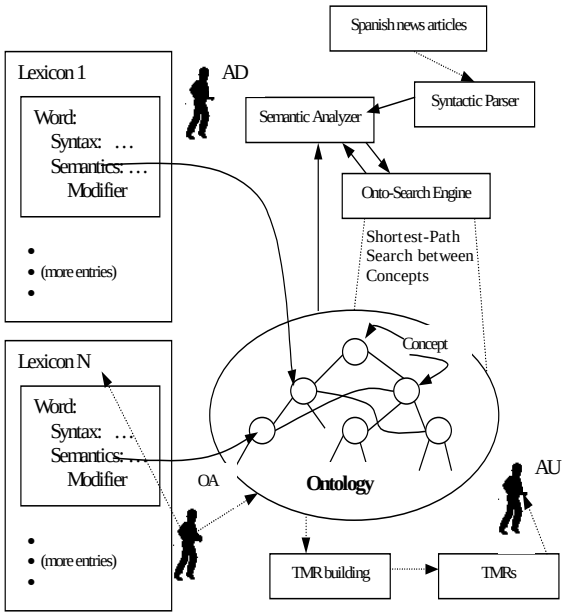

Figure 2: Mikrokosmos

Table 1: Framework Dimensions for Pangloss and MIKROKOSMOS

	PANGLOSS	MIKROKOSMOS
Purpose and benefits of the NLG ontology	Serves as an *inventory of the symbols* that appear in interlingua expressions, and *encodes semantic preferences* that allow selection of one interlingua expression over another. The ontology is therefore both a lexicon and a set of *grammar preferences* for the interlingua language.	Facilitates *natural language interpretation* and *generation*. Provides concepts to *represent word meanings* in a lexicon for a source or target language. Provides the *search space* for a powerful search mechanism. Concepts in the ontology are the *building blocks* used by the lexicon and the analyser to construct Text Meaning Representations (TMR).
Role of information in NLG Ontology	Terms within SENSUS and lexical terms within the lexicons play the role of *ontology*, so that KWs can learn the semantics of a domain. They also play the role of *operational data*, where they are used as tokens in interlingua expressions for machine translation.	Terms within the μK ontology play the role of *ontology*, as interlingual representation of meanings fed to target language generators. A combination of lexical terms and ontology terms play the role of *operational data* during machine translation.
Actors	The Ontology Authors: 1. merged PENMAN Upper Model and ONTOS by hand; 2. merged word senses from an English dictionary and a Spanish-English bilingual dictionary, with WordNet; and 3. built the Spanish lexicon. ADs wrote translators to convert the Spanish input files to English. KWs make use of the translations.	OAs developed the ontology and the lexicon. ADs were the lexicographers who built the lexicons in different languages, the system builders, and testing and evaluation experts. KWs make use of the translations.
Supporting Technologies for the ontology and application	KB Machine Translator is the mainline PANGLOSS engine, which consists of an analyser, (PANGLYZER) and a generation module centred on the English generator called PENMAN. An Example-Based MT (EBMT) system assists with translations between languages. A Lexical Transfer Machine Translator (LTMT) system, relies on a machine-readable dictionary.	A syntactic parser parses input text. The Mikrokarat tool supports graphical editing of ontology. A semantic analyser supports mappings to lexicon. An ontological graph search mechanism, (Onto-Search) checks the constraints found by semantic analyser. A lexicon primarily connects ontology and onomasticon (a special-purpose lexicon of named entities such as cities, corporations, or product's names).
Maturity	Research prototype.	Research prototype.
Meaning of Representation of NLG Ontology	SENSUS was represented in the *FrameKit knowledge representation language*, but is now re-engineered in *C++*. Contains representations for about 70,000 commonly encountered objects, entities, qualities, and relations. These objects are mapped to lexical items of different languages. SENSUS includes both high-level terms (such as "inanimate object") as well as specific terms (such as "submarine"). Terms are organised into a subsumption lattice, with each concept in the lattice corresponding to a word sense.	Ontology is represented as a *frame hierarchy* with multiple inheritance, so it allows for numerous links among the concepts. *Language-specific* word meanings are represented in a lexicon, with *language neutral* meanings in the ontology. Concepts have a many-to-many mapping to word senses in natural languages. The 6000-8000 primitives are organised in a highly interconnected ontological network, to allow for a lexicon for each language to use the primitives in their meaning representations. There is a *limited expressiveness* in the representation, to allow for acquisition of large-scale lexicons that conform to the ontology.
Ontology Acquisition	*Lexicon-driven*, because the goal is to include all concepts that are necessary to represent the meanings of words in a lexicon.	*Situated Development*. It is not lexicon driven, because ontology development and lexicography are processes that both assist each other and at the same time constrain each other.
Language Dependency	Language *independent*. Most of the ontology concepts have a one-to-one mapping with the lexicon items.	*Independent*: not specific to any particular language, but concepts have English names for convenience.

3.2. Ontology as Specification

In this scenario, an ontology within the JUFramework "models the application domain and provides a vocabulary for specifying the requirements for one or more target applications" [1]. In an NLG application there is also a linguistic ontology used in combination with the domain ontology, to assist in interpreting the *meaning* of the text within that domain. The linguistic ontology's role in interpreting the meaning of the text is secondary to the domain ontology's role of specifying the requirements, but the two working together significantly improves information retrieval and the understanding of the information retrieved. We describe two ontology applications for this scenario in Table 2.

The **Unified Medical Language System (UMLS)** project [10] develops machine-readable 'Knowledge Sources' that can be used by a variety of medical application programs to overcome retrieval problems caused by differences in terminology, and the scattering of relevant information across many medical databases. The Semantic Network (SM) (shown in Figure 3) is the ontology, and contains information about the types or categories to which all concepts in a Metathesaurus (MT) have been assigned, and the permissible relationships among these types. The Specialist Lexicon (SL) contains *syntactic* information for Metathesaurus terms. The Information Sources Map (ISM) is mapped to all kinds of medical databases.

Plinius [11] is used to capture the contents of natural language textbooks that describe the mechanical properties of ceramic materials. The textbooks cover a wide range of subjects, so a set of integrated ontologies covers concepts such as materials and their properties, processes to make these materials, and flaws of materials such as cracks and pores. A lexicon maps natural-language tokens from the text (illustrated in Figure 4), onto formal expressions in the domain ontology. The same ontology is used for the semantic part of the lexicon and the background knowledge base.

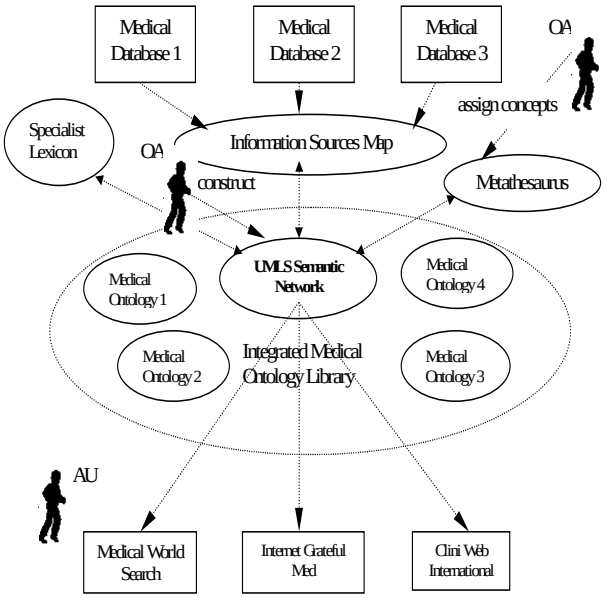

Figure 3: UMLS

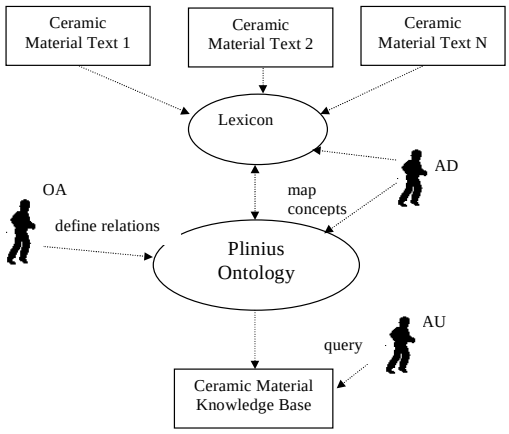

Figure 4: Plinius Ontology

3.3. Common Access to Information

Applications use *one or more shared ontologies* to integrate heterogeneous information systems and allow common access for humans or computers. This enforces the shared ontology as the standard ontology for all participating systems, which removes the heterogeneity from the information system. The heterogeneity is a problem because the systems to be integrated are already operational and it is too costly to redevelop them. A linguistic ontology is sometimes used to assist in the generation of the shared ontology, or is used as a top-level ontology[3] for the shared ontologies to inherit from it. Benefits are the integration of heterogeneous information sources, which can improve interoperability, and more effective use and reuse of knowledge resources.

The **Mediator envirOnment for Multiple Information Sources (MOMIS)** system [13] semi-automatically integrates the schemas of heterogeneous information systems into one shared ontology (illustrated in Figure 5), called a Common Thesaurus (CT) permitting the information systems to be queried from one place. A linguistic ontology (WordNet) is used to assist in the generation of relationships in the CT.

In the **Knowledge Reuse and Fusion Transformation (KRAFT)** project [14], several smaller but heterogeneous shared ontologies (shown in Figure 6) are used to solve the integration of heterogeneous information systems. A top-level ontology (WordNet) is linked to a number of shared ontologies to reduce the problems of different people interpreting the terms in the shared ontologies differently. The hierarchical structure implies that all shared ontologies always have at least one ontology in common, and all share WordNet.

3 "Top-Level ontologies describe very general concepts like space, time, matter, object, event, action, etc., which are independent of a particular problem or domain" [12].

Table 2: Framework Dimensions for UMLS and Plinius

	UMLS	**Plinius**
Purpose and benefits of the NLG ontology	The Semantic Network provides a *consistent categorisation* of all concepts represented in the Metathesaurus. It *improves consistency* of medical terms and determines what *sentences mean* in a corpus of medical documents.	Provides concepts that serve as semantic translations of natural–language words and phrases in the lexicon. Enables co-operation between the knowledge bases used as resources in the knowledge extraction process. Implicitly specifies the desired output of the language-dependent process.
Role of the information	Information within the Semantic Network plays the role of *Ontology*, where it specifies the meaning of the Metathesaurus concepts. The information within the SN, SL, MT, and ISM play the role of *Operational data.*	The information about ceramic materials plays the role of *ontology*, where it specifies information for the knowledge base. The lexical terms and ontology concepts play the role of *Operational Data* where they provide a neutral format for translation during querying.
Actors	OAs constructed the semantic network, KWs assigned MT concepts to the ontology, and AUs retrieve information from the applications of UMLS.	OAs defined relations between the complex concepts, and the ADs mapped concepts in the domain ontology to the lexicon. AUs query the application to obtain knowledge about ceramic materials.
Supporting Technologies for the ontology and application	The MT provides a uniform, integrated distribution format for many biomedical vocabularies and classifications. The Specialist Lexicon contains syntactic information for many MT terms. Software called Metamorphosys is useful in producing customised versions of the MT.	A lexicon maps natural-language tokens onto formal expressions in the knowledge representation language.
Maturity Level	A commercial product, whose research began in 1986. Has numerous applications in the medical domain.	A research prototype.
Meaning of Representation of Ontology	Semantic Network and specialist lexicon is represented in three formats: a *relational table*, a *unit record*, and an *Abstract Syntax Notation One format*. There are major groupings of semantic types for organisms, anatomical structures, biologic function, chemicals, events, physical objects, and concepts or ideas. Primary link is the `is-a' link, which establishes the hierarchy of types within the Network and is used for deciding on the most specific semantic type available for assignment to a Metathesaurus concept. A set of non-hierarchical relations between types is grouped into five major categories.	The ontology is *represented in Prolog*, and is highly expressive. The ontology was constructed using a *bottom-up approach*: Chemical elements, which are kinds of atoms (called atomic concepts) were identified initially, and construction rules constructed for more complex concepts. Representation of the ontology is a combination of formal and informal approaches. Atomic concepts are informal (in natural language), and unambiguous and complex definitions are formally defined in the language of sets and tuples. Mappings between the lexical constituents and ontology concepts is many-to-many.
Ontology Acquisition	*Domain analysis* driven, because it concentrates on the medical domain. It is also *lexicon-driven*, because a lexicon and a thesaurus are used to assist in providing meanings to medical terms.	*Domain analysis*, where the domain of ceramic materials has been chosen. It is also *lexicon driven*, where there is a link between natural language terms in the lexicon and concepts in the ontology.
Dependency	Language (English) *dependent*	Language (English) *dependent.*

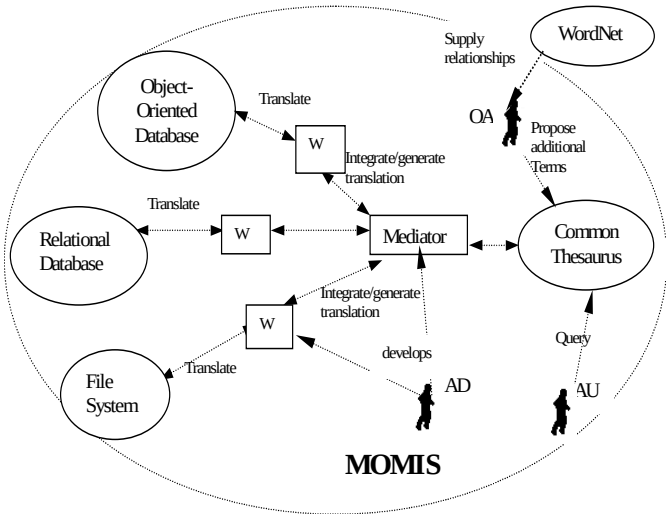

Figure 5: MOMIS

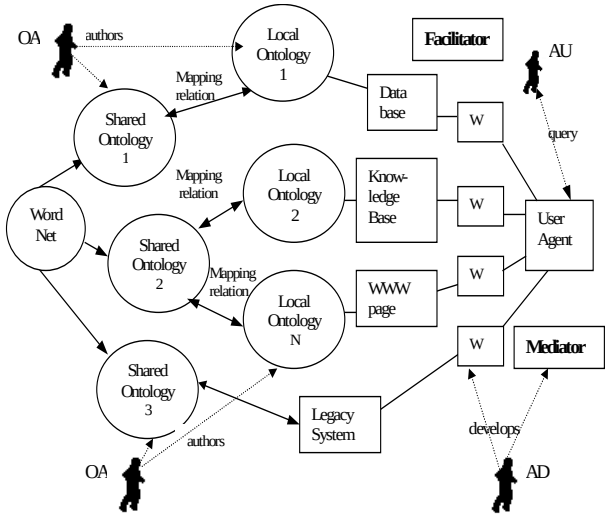

Figure 6: KRAFT

Table 3: Framework Dimensions for MOMIS and KRAFT

	MOMIS	**KRAFT**
Purpose and benefits of WordNet	Provide *semantics for relationships* in the Common Thesaurus.	Specify the *meaning of terms* for communication to other ontologies. Assists the lower level ontologies to prescribe types of knowledge that need to be acquired in the domain.
Role of the information	Terms within the CT play the role of *operational data* when they are queried during run-time. CT also plays role of *ontology* when it specifies terms in information sources. Information in WordNet plays the role of *Ontology Representation Language*, where it is used by OAs, during the development process of the CT.	KRAFT agents and the shared ontologies play the role of operational data. Concepts within the local ontologies play the role of *ontology* where they specify the schemas of the information sources. Information in WordNet plays the role of *Ontology Representation Language*, where it specifies top-level terms for the shared ontologies in the hierarchy.
Actors	OA interacts with an ODB-Tool to supply additional relationships to those inferred automatically between classes in source schemas. These form the CT. OAs use WordNet to derive additional relationships. AUs query the CT and the query is optimised to a local query for the correct information source.	OAs author local ontologies based on each resource's schema. OAs author shared ontologies in hierarchy below WordNet. ADs map each information system onto one of the shared ontologies, using mapping relations. AUs query the information sources through a User Agent.
Supporting Technologies for the ontology and application	*Wrappers* (W) translate source schema. *Mediator* processes and integrates descriptions received from wrappers, and automatically generates translation of global queries from the AU into different sub-queries. *ODB-Tools engine* based on Description Logics, performs schema validation for generation of CT; *ARTEMIS Tool Environment* evaluates and clusters classes in ontology.	KRAFT *agents*: wrappers (W in Figurre 6) link resources onto other agents; facilitator recommends services, and a mediator assists with query answering. Database resources are managed semi-formally by independent instances of the *Prolog/Functional Data Model Database Management System.* The *Common Command* and *Query Language* communicates between agents. *KRAFT Constraint Interchange Format* expresses actual information to be extracted, reused and fused.

Table 3: Framework Dimensions for MOMIS and KRAFT

	MOMIS	**KRAFT**
Maturity Level	An *advanced prototype*. Wrappers and mediators are quite mature, and are used commercially in data warehouses. *Description Logics* is used in commercial applications, so is quite mature.	An *advanced prototype* that has been partially implemented in the domain of telecommunications network data services design.
Meaning of Representation of WordNet	Similar to a thesaurus, is structured around groups of synonymous words in four categories: nouns, verbs, adjectives, and adverbs. The groups are called synsets, and there are approximately 48,800. Also like a dictionary, it provides definitions and usage examples. In addition to the thesaurus and dictionary features, it provides explicit relationships among the synsets ('is-a' and 'has-a'). The WordNet database consists of two files in ASCII format that is human and machine-readable.	
Sharing vs. Exchange	Information in CT is shared to allow for optimisation of queries.	Data is exchanged between information resources, with agents making decisions
Ontology acquisition	*Lexicon* driven, where the goal is to cover every word in the English dictionary.	
Language Dependency	Language *Dependent* (English)	

3.4. Ontology Based Search

An ontology is used in many applications for searching an information repository or a number of repositories for desired resources. [1]. The repository for these applications is often the World Wide Web (WWW) or an intranet and the resources may be documents, web pages or names of experts. It is uncommon for a linguistic ontology to be used in this kind of application. If a linguistic ontology is used, it is usually in addition to another domain ontology. The dimensions for these applications are described in Table 4.

Ontogeneration [15] is an interactive information-retrieval system (illustrated in Figure 7) that uses a linguistic ontology called Generalized Upper Model (GUM) to generate Spanish text descriptions in response to queries in the chemistry domain. GUM has been used in other NLG projects for English, German and Italian.

OntoSeek [16] uses a large linguistic ontology (called SENSUS) for Internet-based retrieval of object-oriented (OO) components, and for retrieval of yellow pages and product catalogues. This application is illustrated in Figure 8.

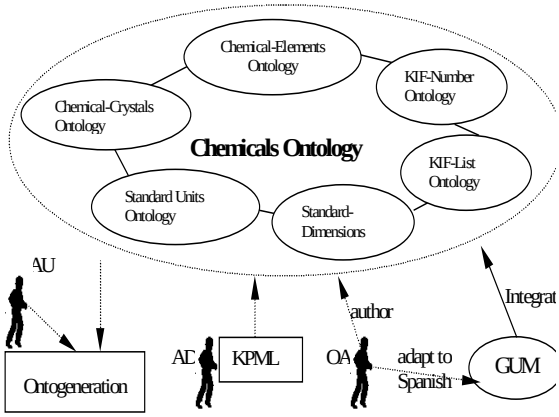

Figure 7: Ontogeneration

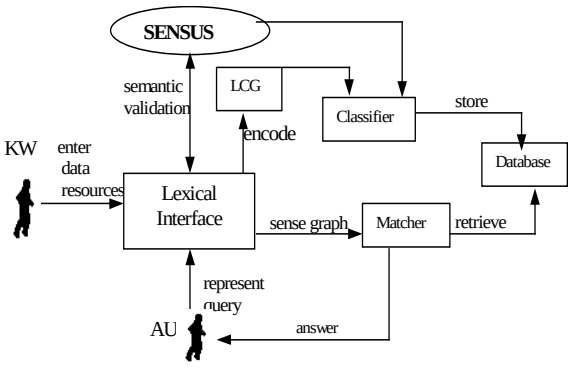

Figure 8: OntoSeek

Table 4: Framework Dimensions for Ontogeneration and OntoSeek

	Ontogeneration	**OntoSeek**
Purpose and benefits of the NLG ontology	GUM provides a level of organisation for *interfacing* between the domain model and NLG components. Assists in *multilingual text generation*.	An *interface* between the AUs and the application to assist in semantic encoding of information and to assist efficient and effective retrieval of information. Provides a *vocabulary* to describe various senses (meanings) of words, and to describe the semantic relationships among senses. Promotes *reuse* of an existing ontology.
Role of the information	The grammatical semantic information in GUM plays the role of *operational data* during text generation, and ontology when it specifies the grammatical semantics. The concepts within the Chemicals Ontology play the role of *ontology* when they specify terms and definitions within the Chemicals domain.	LCGs play the role of *operational data* when AUs query the system. They play the role of *ontology* when they are used to encode the information resources. Information in SENSUS plays the role of *ontology*, where it describes the various senses of words. SENSUS also plays the role of *operational data*, when it is used to find synonyms for terms used by the AU during querying.
Actors	OAs authored the Chemicals domain ontology. OAs adapted GUM to the Spanish language so that its concepts and relations can be reused. ADs use an existing multilingual environment to develop the Spanish grammar. AUs learn about Chemistry in Spanish, from the Chemical applications.	KWs enter data resources through the lexical interface. A 'Lexical Conceptual Graph' (LCG) is used to encode resources, and SENSUS validates the semantics of words entered by the KW. LCGs are classified according to a subsumption relationship and entered into a database. AUs query the database via the lexical interface.
Supporting Technologies for the ontology and application	Ontology Design Environment translators assist in transforming ontologies into different languages, such as Ontolingua or SQL. A relational database stores the Chemicals ontology. User interface is in Java. The Komet-Penman Multilingual technology builds resources for Spanish text generation, using Common Lisp and the Loom Description Logic. A broker: (Onto)2Agent is used to browse and retrieve concepts.	*Ontolingua* describes SENSUS content. *Conceptual Graphs* represent resource information and assist in querying. There is a *Java* user interface, with a *C++ internal structure*. An *Object-Oriented DBMS* stores the LCGs.
Maturity Level	A research prototype.	A research prototype, tested in two domains: retrieval of object-oriented components and retrieval of yellow pages and product catalogues.
Meaning of Representation of the Linguistic Ontology	GUM is implemented in the *Loom* Description Logic language. GUM is organised into two hierarchies: 1) concepts, representing the basic semantic entities entailed by natural language grammars. 2) relations represent the participants and circumstances involved in the processes and the logical combinations among them [15].	The meaning of the SENSUS ontology is described for the application PANGLOSS in Table 1.
Ontology Acquisition	*Grammatically driven*, where the concepts are accounts of the semantics that may be expressed in *grammatical* units, rather than the semantics of words.	*Lexicon-driven*, where the goal is to include all concepts necessary to represent the semantics of words.
Language Dependency	GUM is neither *language dependent* nor *independent*, because semantic distinctions relevant to different languages are recognised.	Language *dependent* (English)

Table 5: Summary of the Reuse of Linguistic Ontologies in Different Applications

Pangloss	**Mikrokosmos**	**MOMIS**	**KRAFT**	**Ontogeneration**	**OntoSeek**
SENSUS (merged WordNet and Penman Upper Model)	Began as an extension of Pangloss (SENSUS ontology)	WordNet	WordNet	GUM (from Penman Upper Model)	SENSUS (merged WordNet and Penman Upper Model)

4 COMPARISON OF APPLICATIONS USING A LINGUISTIC ONTOLOGY

This section compares and contrasts the ontology applications using the framework dimensions. Many of these applications use linguistic ontologies with a similar terminology and structure, because WordNet and the Penman Upper Model have been merged into other linguistic ontologies, and subsequently reused in different applications. Table 5 summarises the ontologies that have been reused in the applications presented in this paper. SENSUS is used in both PANGLOSS and OntoSeek, but its application is significantly different. The application of WordNet in MOMIS and KRAFT is also different to its use in PANGLOSS and OntoSeek. GUM is a result of a continuing evolution beginning with the Penman Upper Model, but the use of GUM in Ontogeneration is different from the other applications that use the Penman Upper Model. In 1994, the Mikrokosmos project began as an extension to the PANGLOSS system. Our research highlights the different *applications* of these linguistic ontologies by describing each of the framework dimensions.

The main ***purpose*** of the linguistic ontology in the 'neutral authoring' scenario is to support machine translation by providing language-neutral terms to which lexical terms of different languages can be attached. This provides ***benefits*** of reuse, where a number of lexicons in different languages can be used with the same ontology for multilingual translation. In 'common access to information' WordNet specifies the semantics of terms for one or more other ontologies. The linguistic ontology in this case assists in the generation of (in MOMIS), or use of (in KRAFT) a shared ontology or ontologies. Benefits of one or more shared ontologies include the reduced cost of multiple applications having common access to data, which may in turn facilitate inter-operability. Linguistic ontologies used to 'specify knowledge' provide an interface between models of the domain and a lexicon (or a kind of lexicon), so that users can understand the terminology of a specific domain. A linguistic ontology used in 'search' also acts an interface, although it is between the domain knowledge and the query facility, so that users can query and obtain information using a natural language (or close to it). When search applications employ a linguistic ontology, information retrieval is improved and the application is significantly easier to use.

MOMIS and KRAFT are the only applications where the ***role*** of the information in the linguistic ontology is an *Ontology Representation Language*, used during the development process of the CT (in MOMIS) and the shared ontologies (in KRAFT). WordNet is not used during execution of either of these applications, but the CT (in MOMIS) and the shared ontologies (in KRAFT) play the role of *operational data* within the application. These two applications demonstrate that WordNet can be utilised in different ways. The linguistic ontology in all the other applications described in this paper play the role of operational data, and some use an *ontology* to specify the meaning of concepts used in either a lexicon (or a kind of lexicon) or another ontology (LCGs in OntoSeek). GUM is different because it specifies the grammatical semantics, rather than the semantics of words.

Actors of an application using a linguistic ontology appear to have much more work to do than those without a linguistic component. In PANGLOSS and Mikrokosmos, much of the ontology authors' time was spent merging (mostly by hand) existing ontologies and dictionaries to create an ontology suitable for their application. Some of these linguistic ontologies will be reused, so the time is well spent. The ontology authors (or lexicographers) may also create a lexicon (or a number of lexicons) for the application. If there is a lexicon, the ontology author needs to map it to the ontology. Application developers also have more work to do in building translators, syntactic parsers, semantic analysers and text generators. Some of these tools are also reused. Wrappers and mediators also need to be developed, along with tools to validate information schemas of heterogeneous sources. The linguistic ontologies in the 'search' applications were reused and from the documentation, it appeared that this reuse improved the ease of development of the application. Both the 'search' applications also reused existing *technologies*. These applications demonstrate that it is possible to reuse linguistic ontologies to improve development efficiency and to save developing specific tools for an application.

All the applications in this survey are research prototypes, except for UMLS, which is a commercial product with many medical applications. We could not find recent documentation on PANGLOSS and Plinius, so it appears that their work is not being continued. The Mikrokosmos project may have taken over from PANGLOSS. Some of the technologies used within the applications that we described are used in other commercial applications: for example object-oriented databases, relational databases, wrappers, mediators, Common Lisp, Java, the C++ programming language, and description logic languages. This demonstrates the ***maturity*** of the technologies.

The ***meaning of the representation*** is similar in all but the 'ontology as specification' applications. WordNet is organised around structures of synonymous words in categories, and has 'is-a' and 'has-a' relationships. SENSUS was merged with WordNet, the Penman Upper Model and dictionaries. Therefore, portions of SENSUS are similar to WordNet, where it organises terms into a subsumption lattice and provides concepts for each word sense. Although GUM is different from the other linguistic ontologies because it describes a grammatical semantics, it is also rather similar to SENSUS because it originates from the Penman Upper Model. The μK ontology represents the concepts as language neutral meanings in a frame hierarchy, mapped to a lexicon with language specific meanings. In UMLS, the semantic network has mainly 'is-a' links to establish a hierarchy, with mappings to concepts in the Metathesaurus. Plinius employed an uncommon method (bottom-up) for designing their ontology. Atomic concepts (chemical elements) in the domain of ceramic materials were designed first, and more complex concepts are based on the atomic ones. The Plinius ontology is a combination of both semantic meanings and domain knowledge, so is quite different from a purely linguistic ontology such as WordNet.

Most linguistic ontologies acquire their concepts with a lexicon-

driven ***goal***. The exceptions for the applications described here are: Mikrokosmos, where the ontology development and lexicography are processes that both assist and constrain each other; and GUM, which has a grammatical semantic driven goal. UMLS and Plinius have a domain-driven goal as well as lexicon-driven.

PANGLOSS and Mikrokosmos are ***language independent*** because they are used for multilingual translation. GUM is mostly language independent, but does recognise the semantic distinctions of different languages. All the other linguistic ontologies described here are language dependent, where the language is English.

5 CONCLUSION

This paper has demonstrated that a linguistic ontology can be used (and also reused) in a number of different kinds of applications. We have described an additional scenario for each of the JUFramework's categories of ontology applications, and populated it with some prominent research and industrial applications. Potential application developers can now examine the descriptions in the populated framework to inspire them to use linguistic ontologies for their specific applications. The population of the framework was limited to only a few applications, and could be extended to include more.

The framework dimensions could be extended so that a specific technique for using a linguistic ontology is studied in more detail. Additional dimensions could describe: a reusable linguistic ontology in more detail, techniques (especially automatic) for integrating or merging the linguistic ontology with a domain ontology, techniques for using a lexicon with an ontology in an application, and the methodologies employed in developing applications that use a linguistic ontology. Another dimension could also describe how the linguistic ontology has been changed to specifically suit the function of a particular application. This kind of dimension would highlight the differences between applications using similar linguistic ontologies such as WordNet, SENSUS and the Penman Upper Model.

We would like to implement the JUFramework as an ontology, which could be integrated with the Reference Ontology, so that both potential users and application developers could search for appropriate ontologies (including linguistic) and obtain guidelines for developing an ontology application specific to their own needs. The JUFramework ontology could have five upper level concepts (one for each category), so that a potential application developer could place their proposed ontology application within a category, according to the purposes and benefits required of the new application. From the classification of concepts within the JUFramework ontology, the developer could find out what kind of linguistic ontologies are used in different kinds of applications, what technologies to use with the linguistic ontology, and whether these technologies are mature enough to use commercially.

ACKNOWLEDGEMENTS

We would like to thank the referees for their comments, which helped improve this paper.

REFERENCES

[1] Jasper, R. and M. Uschold. *A Framework for Understanding and Classifying Ontology Applications.* in *KAW'99 Twelfth Workshop on Knowledge Acquisition, Modeling and Management*. 1999. Voyager Inn, Banff, Alberta, Canada.

[2] Mahesh, K., *Ontology Development for Machine Translation: Ideology and Methodology.*, . 1996.

[3] Noy, N.F. and C.D. Hafner, *The State of the Art in Ontology Design. A Survey and Comparative Review.* AI Magazine, 1997. **18**(3): p. 53-74.

[4] Arpírez-Vega, J.C., *et al. (ONTO)2Agent: An ontology-based www broker to select ontologies*. In *13th European Conference on Artificial Intelligence ECAI'98*. 1998. Brighton, England.

[5] Bateman, J.A. *The Theoretical Status of Ontologies in Natural Language Processing*. in *Proceedings of the workshop on 'Text Representation and Domain Modelling --Ideas from Linguistics and AI'*. 1992. Technical University Berlin.

[6] Bateman, J.A., R. Henschel, and F. Rinaldi, *The Generalized Upper Model 2.0*, . 1995.

[7] Knight, K. and S. Luk. *Building a Large-Scale Knowledge Base for Machine Translation*. in *Proc. of the National Conference on Artificial Intelligence (AAAI)*. 1994.

[8] Swartout, B., *et al. Toward Distributed Use of Large-Scale Ontologies.* in *Symposium on Ontological Engineering of AAAI*. 1996. Stanford (Calfornia): Mars.

[9] Viegas, E. *An Overt Semantics with a Machine-guided Approach for Robust LKBs*. in *In the Proceedings of SIGLEX99 Standardizing Lexical Resources, as part of ACL99,*. 1999. University of Maryland.

[10] National Library of Medicine., Unified Medical Language System. 1999. http://www.nlm.nih.gov/research/umls/

[11] van der Vet, P.E. and N.J.I. Mars, *Bottom-Up Construction of Ontologies.* IEEE Transactions on Knowledge and Data Engineering, 1998. **10**(4): p. 513-526.

[12] Guarino, N. *Formal Ontology and Information Systems*. in *Proc. of the 1st International Conference on Formal Ontology in Information Systems.* 1998. Trento, Italy: IOS Press.

[13] Bergamaschi, S. and M.V. S Castano, D Beneventano. *Intelligent Techniques for the Extraction and Integration of Heterogeneous Information,*. in *IJCAI-99 Workshop on Intelligent Information Integration*. 1999. Stockholm.

[14] Preece, A., *et al. The KRAFT Architecture for Knowledge Fusion and Transformation*. in *Expert Systems conference*. 1999.

[15] Aguado, G., *et al. ONTOGENERATION: Reusing domain and linguistic ontologies for Spanish text generation*. in *13th European Conference on Artificial Intelligence ECAI'98, Workshop on Applications of Ontologies and Problem solving Methods*. 1998. Brighton, England.

[16] Guarino, N., C. Masolo, and G. Vetere, *OntoSeek: Using Large Linguistic Ontologies for Accessing On-Line Yellow Pages and Product Catalogs*, . 1999, National Research Council, LADSEB-CNR: Padavo, Italy.