Practical Ontologies: Requirements, Design and Applications to Semantic Integration and Knowledge Discovery

by

Daniela Rosu

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

© Copyright by Daniela Rosu 2013
Abstract

Due to their role in describing the semantics of information, knowledge representations, from formal ontologies to informal representations such as folksonomies, are becoming increasingly important in facilitating the exchange of information, as well as the semantic integration of information systems and knowledge discovery in a large number of areas, from e-commerce to bioinformatics.

In this thesis we present several studies related to the development and application of knowledge representations, in the form of practical ontologies. In Chapter 2, we examine representational challenges and requirements for describing practical domain knowledge. We present representational requirements we collected by surveying current and potential ontology users, discuss current approaches to codifying knowledge and give formal representation solutions to some of the issues we identified.

In Chapter 3 we introduce a practical ontology for representing data exchanges, discuss its relationship with existing standards and its role in facilitating the interoperability between information producing and consuming services.
We also consider the problem of assessing similarity between ontological concepts and propose three novel measures of similarity, detailed in Chapter 4. Two of our proposals estimate semantic similarity between concepts in the same ontology, while the third measures similarity between concepts belonging to different ontologies.

In Chapters 5 and 6 we discuss applications of our similarity measures. Specifically, in Chapter 5 we present a method for discovering mappings between different bio-ontologies and discuss the results of a quality assessment performed by two expert biologists on the top mappings identified by our method. In Chapter 6 we detail an integrative method for discovering gene associations: the Gene Association Predictor (GAP) and present the results of assessing the predictions made by GAP using a multi-pronged approach: (1) overlap with datasets of known, experimentally derived, gene associations, (2) comparison with other prediction services, and (3) manual curation of scientific literature.
Acknowledgments

Completing this thesis has been a long and arduous process and it is my pleasure to thank the many people who helped me along the way.

Foremost, I would like to thank my supervisor, Dr. Igor Jurišica, who has been a steady influence during my PhD. Igor’s ability to balance research and personal pursuits inspired me; his advice was consistently timely and useful, his support and guidance invaluable, and his patience with me … endless. I have learned a lot from him and I am deeply grateful for his help in realizing my scholarly dreams.

I owe an immense debt of gratitude to Dr. Michael Grüninger for his generous advice and constant encouragement. His passion for ontology engineering and patient teaching inspired me to want to learn more about representing knowledge and do research in this field. I benefited enormously from his suggestions, lectures, and the Torontology meetings he organizes. I am also very grateful to Dr. Ryan Lilien and Dr. Kelly Lyons who gave me detailed and constructive feedback on my thesis work and a welcome diversity of perspectives. Their encouragement and suggestions have been tremendously helpful. Dr. Periklis Andritsos, Dr. Lawrence Hunter, and Dr. Eric Yu graciously agreed to be my external examiners and their feedback was invaluable.

This work could have not been completed without the financial support I received from the Centre for Advanced Studies at the IBM Toronto Lab and I am indebted to Joanna Ng and Alex Lau for their assistance and collaboration during my IBM fellowship.

To all members of the Jurišica group: thank you for making me look forward to coming to the lab everyday. Chiara, Fiona, Mahima, Dan, and Kevin: many thanks for answering with patience my (many) biology related questions. Kristen and Max: thank you for our many long late night discussions and for the patience with which you read and commented on the early drafts of my thesis.

Cristiana and Sara: thanks for being awesome officemates and such dear friends, and for always being there to listen, console (most of the time), but also celebrate with me. I could have not
done it without you! Marc: I will always remember our many lively discussions. Thanks for constantly encouraging me to look forward to a life after grad school. Christian: thank you for always making me smile (and for listening with unwavering patience to my random ramblings).

A big thank you to my many CS friends (Yannis, Periklis, Mihaela, Matei, Anastasia, Sam, Tassos, Sotirios, Adrian, Natasha, Stavros, Themis, Fanis, Cosmin, Antonina, Apostolis, George², Kleoni, Stratis, Grace, Rui, Abraham) who shared with me over the years the ups and downs of grad school and who also taught me how to celebrate, … really celebrate!

This thesis would have not been completed without the love and support of my friends outside school, who kept me sane and constantly reminded me that there is life outside the lab. Raluca, Valentina, Charin, Rani, Yuki: thank you for being there for me!

Răzvan: I am very sorry you had to bear the brunt of the frustrations I have felt during these long years in grad school. We shared many happy moments, too, and I have learned a lot from you. I hope you can also share in the joy of the (late) successes.

Lastly, and most importantly, I would not have chosen this path if not for my parents, who instilled within me a passion for science and research. To them I dedicate this thesis. Mulțumesc pentru tot!
Table of Contents

Acknowledgments.......................................................................................................................... iv
Table of Contents.......................................................................................................................... vii
List of Tables ................................................................................................................................. ix
List of Figures ............................................................................................................................... x
List of Appendices ........................................................................................................................ xii

1 Introduction .............................................................................................................................. 1
   1.1 What is “knowledge”? ........................................................................................................ 1
   1.2 What is a knowledge representation? ................................................................................ 3
   1.3 What is a (practical) ontology? ....................................................................................... 11
   1.4 Applications of ontologies ............................................................................................. 15
   1.5 Research contributions of this thesis ............................................................................ 17
   1.6 Thesis Outline ................................................................................................................. 24

2 Knowledge Representation Requirements for Practical Ontologies ........................................ 25
   2.1 Requirements for practical ontology frameworks – users’ perspective ....................... 26
   2.2 Practical representations of knowledge ......................................................................... 36

3 From Information Models to Knowledge Models – Insights into Practical Ontology Design: towards a Core Business Data Interchange Ontology .......................................................................................... 82
   3.1 Related work .................................................................................................................... 85
   3.2 Ontology engineering strategy ....................................................................................... 86
   3.3 A short overview of the Process Specification Language ............................................. 90
   3.4 A short overview of the Open Applications Group Integration Specification (OAGIS).... 93
   3.5 Overview of the Core Business Data Interchange Ontology ......................................... 98
   3.6 Discussion and future work ......................................................................................... 111

4 Measuring the Similarity between Ontological Concepts ..................................................... 117
   4.1 Feature-based similarity measures .............................................................................. 119
List of Tables

Table 3.1 Scope of OAGIS 9.0 .................................................................................................................. 94

Table 3.2 OAGIS Verb request-reply pairings ..................................................................................... 97

Table 4.1 Discriminating power of different Information Content Measures ................................. 138

Table 6.1 Information on Selected features. Last column shows the coverage of individual features, i.e. 100% coverage gives the fraction of the 19K protein-coding human genes for whom the corresponding feature set is empty. ................................................................................................. 158

Table 6.2 Number of protein and total number of related protein pairs contained in each data source ......................................................................................................................................... 161

Table 6.3 Configuration Table: for ease of reference, at any threshold setting, different combinations of similarity measures at different levels of term, per-feature, and gene similarity are given specific configuration numbers ................................................................................................................................. 166

Table 6.4 List of the confirmed PPI interactions, types of the interactions and the PubMed IDs of the articles confirming the predicted interactions experimentally. .............................................. 176

Table B.1 OBOL representation of a GO term ..................................................................................... 220

Table D.1 Web Ontology Language elements ...................................................................................... 228

Table 0.1 Messages types in CBDIO ..................................................................................................... 242
List of Figures

Figure 2.1 The 8 basic relations of Region Calculus................................................................. 43

Figure 2.2 Chromosomes capped by telomers................................................................. 51

Figure 2.3 General scheme of kinase function........................................................ 64

Figure 2.4 Human Dihydrofolate reductase, DHFR, with dihydrofolate (left) and NADPH (right) bound............................................................................................................................................. 65

Figure 3.1 The architecture of a BOD .................................................................................. 94

Figure 3.2 Get Purchase Order BOD................................................................................. 96

Figure 3.3 Verb and Noun components of the GetPurchaseOrder Business Object .......... 97

Figure 5.1 Isoleucine biosynthesis pathway feedback inhibition ........................................ 143

Figure 5.2 Correctness of the alignments between MF and BP terms, relatedness$^s_l$ ............. 146

Figure 5.3 Correctness of the alignments between MF and BP terms, relatedness$^d_l$ ............. 146

Figure 5.4 Correctness of the alignments between MF and BP terms, average relatedness...... 147

Figure 5.5 Example of structural inconsistency in GO....................................................... 151

Figure 6.1 Overlap between the proteins (a) and interacting protein pairs (b) in the gold datasets ..................................................................................................................................................... 162

Figure 6.2 F1-scores for different configuration settings. Error bars show the standard error of the mean................................................................. 167

Figure 6.3 Precision versus recall curves for different term similarity measures............... 168

Figure 6.4 AUC score comparison at different term similarity measures. The average of AUC values are shown as bars, and for each bar, the standard error of the mean is shown as a thin error bar. .............................................................................................................................................. 168
Figure 6.5 Precision vs. recall curves using every single feature as the only predictive feature of GAP.............................................................................................................................................169

Figure 6.6 AUC score comparison when individual features are set to be the only predictive feature of GAP. The average of AUC values are shown as bars; the standard error of the mean is shown as a thin error bar at the top of each bar. ...........................................................................................................170

Figure 6.7 Precision vs. recall curves of GAP as compared to the selected benchmark PPI predictors.............................................................................................................................................174

Figure 6.8 AUC score comparison between GAP and the five benchmark PPI prediction tools. The average of AUC values are shown as bars; error bars correspond to the standard error of the mean............................................................................................................................................174

Figure 6.9 Functional intra-connectivity of autism genes predicted by GAP (a), predicted by STRING (b), and drawn from databases of known direct and indirect interactions (c). The transparency of the edges is inversely proportional to the rank score/score of the predicted associations. The size of each node and the node highlight size are, respectively, proportional to the node degree and the SFARI score. The node highlight colors change from red (the lowest SFARI score) to blue (highest SFARI score). ............................................................................178

Figure 6.10 Novel autism genes predicted by GAP, and the evidences from the related diseases, drugs, GO terms, and pathways. ......................................................................................................................180

Figure 6.11 Histogram of the association-degree of 11,215 genes predicted to be functionally associated to SFARI-listed autism genes. ...........................................................................................................182

Figure 6.12 Network of functional associations among SFARI-listed autism-related genes (yellow) and GAP-predicted autism-related genes (black)..................................................................................................183

Figure A.1 ISO Terminology-related Standards .................................................................219

Figure D.1 The structure of OWL 2.0.................................................................................225

Figure G.1 All OAGIS 9.0 Nouns ...................................................................................231
List of Appendices

Appendix A ISO Terminology– related standards................................................................. 218
Appendix B OBOL .............................................................................................................. 220
Appendix C Representations of Spatial Knowledge............................................................ 223
Appendix D Ontology Web Language (OWL)................................................................. 225
Appendix E Logic definability.......................................................................................... 229
Appendix F Computability and Complexity...................................................................... 230
Appendix G OAGIS............................................................................................................. 231
Appendix H The Core Business Data Interchange Ontology - Terminology................. 232
Appendix I Benefits of Formal Representations of Knowledge........................................ 243
1 Introduction

How to best represent diverse domain knowledge and how to discover new knowledge with the help of these representations are central questions in the field of Semantic Knowledge Management.

1.1 What is “knowledge”?

The answer, not surprisingly, depends on the respondent, as researchers in various fields approach this question from different practical and theoretical angles. Disciplines such as sociology, education and management emphasize the social aspects of acquiring, developing and transferring knowledge between individuals and groups of people. In contrast, technical fields such as engineering and computer science, artificial intelligence, in particular, focus on reproducing and enhancing human reasoning with the help of formal models of knowledge and give primacy to aspects related to the process of formalizing human knowledge.

According to philosopher Michael Oakeshott, practical knowledge is imparted and acquired through direct experience and becomes an intrinsic constituent of the very fabric of our thoughts remaining, for the most part, unconscious, inarticulable. It stands in direct contrast to technical knowledge, which is the sort of thing that can be consciously articulated, shared, taught, written down and also learnt from books [Oakeshott1962]. This distinction is akin to the differentiation between tacit and explicit knowledge, first put forward by the mathematician and philosopher Michael Polanyi [Polanyi1967]. Polanyi believed that 'we can know more than we can tell' and that the tacit, personal, knowledge that each of us posses is difficult to translate into rational language, while explicit knowledge lends itself to formalization. Both perspectives implicitly assert that knowledge is more than just information and is intimately associated to the individual or group who possess it, but the philosophical debates related to process through which the information becomes knowledge are far from settled. The literature is, however, mostly
consistent in distinguishing between data, information and knowledge. Data are generally considered to be objective, i.e. devoid of a specific intention, representations of concrete facts such as qualitative and quantitative measurements and observations of environmental variables [Prax2000]. Information is thought of as data that has been organized in order to reflect a meaning, and it is, thus, manifesting the intention of its creators [Albino2001]. Knowledge, on the other hand, is information that has been acquired and assimilated by a person or group [Durant2000].

Researchers in Knowledge Management divide knowledge into three categories: (1) unformalizable, (tacit) knowledge that can only be transferred through methods such as observation, training, mentoring and imitation, (2) formalizable, in principle, but not yet formalized because of its complexity, e.g., scientific knowledge, social and organizational barriers, e.g., knowledge about the power hierarchy in an enterprise, etc., (3) formalized\(^1\), in the sense that it has been made explicit, codified and recorded in books, reports, regulations, etc. [Lejeune2011]. We are concerned in this research with formalizable and formalized knowledge.

Knowledge Management research also partitions knowledge according to the type of human learning involved in developing it into three interacting types: (1) perceptual, acquired through direct experience and involvement, i.e. corresponding to psychomotor learning, and largely identified with the development of skills, (2) affectual, exemplified by attitudes and sentiments attached to objects, situations, etc., and evolved through affective learning, and (3) conceptual, resulting in the construction of mental representations such as semantic networks of abstract concepts and developed through cognitive learning [Yang2009]. Research in Knowledge Representation, of which the research undertaken in this thesis is a part, is largely concerned with formalizing conceptual knowledge, but recent research efforts have also been directed towards formalizing various aspects of affective and perceptual knowledge, as well as the mental states and processes involved in reasoning and behavioural responses [Hastings2011, Arkoudas2008].

_____________________________

\(^1\) In this thesis we used formalized knowledge
There is also a distinction being made in the literature between (1) declarative (conceptual) knowledge, *knowing what*, (2) procedural (operational) knowledge, *knowing how*, and (3) problem solving knowledge, which builds on both declarative and procedural knowledge. We do not treat them separately in this thesis. Also outside the scope of our work is debating philosophical aspects of knowledge, such as whether knowledge is (1) *justified true belief*, i.e., we know something because it is true, and we are justified in believing that it is true, or (2) *reliable true belief*, i.e., we know something because it is true, and we formed the belief that it is true via a reliable cognitive process.\(^2\)

We are considering in this work issues not only related to people’s ability to find and use the available knowledge, but also to the machine-based intelligent\(^3\) processing of the world’s digitally available information which is, for the most part, still stored as unstructured text files or images, effectively only understandable by humans.

Our own ability to make sense of the available knowledge as well as the machine-based comprehension can be severely impaired by the ambiguities and differences in the terminology used by various content producers, rendering the process of integrating disparate pieces of knowledge difficult or impossible. Knowledge representations, from the very informal folksonomies to formal\(^4\) ontologies, can be of tremendous help in this regard.

## 1.2 What is a knowledge representation?

We discuss in this section several notions central to knowledge representation.

**Concepts, terms and terminologies.** The International Standards Organization (ISO), via its technical committee on *Terminology and other language and content resources*, coordinates the efforts to regulate terminologies and the terminology development process and has produced over the years several important standards, listed in Appendix B. Of particular interest to


\(^3\) Machine-based comprehension, e.g., [Bookman1994], is a complex framework at the intersection of research themes from cognitive psychology, cognitive science, corpus linguistics and artificial intelligence.

\(^4\) Formal means that the specification is encoded in a language with a precise semantics, usually logic-based.
knowledge representation development are the guidelines referring to concepts, terms, and definitions.

According to ISO 1087-1:2000: a **concept** is a unit of knowledge created by a unique combination of characteristics. However, ISO TR 12310, *Health informatics — Principles and guidelines for the measurement of conformance in the implementation of terminological systems*, stipulates that a **concept** is a “single mental representation of some real or abstract thing”. This formulation is problematic because there is no clear consensus on what constitutes a single (basic) unit of mental representation. Current standards that organize knowledge do not follow a clear strategy when defining concepts: e.g., SNOMED CT lists as concepts both “myocardial infarct” as well as the rather convoluted “Family history of myocardial infarct in first degree female relative less than 65 years of age (situation)”.

A **term** identifies a concept, i.e. it is the verbal designation of a concept. All terms in a standardized terminology are required to be monosemic, and relate to similar terms via **synonymy**, which is defined as a “relation between or among terms in a given language, representing the same concept”.

A **definition** is the “representation of a concept by a descriptive statement which serves to differentiate it from related concepts”. A definition should describe the concept, not the words its designation comprises, and should describe exactly one concept. In addition, a definition should describe the concept precisely, explaining what the concept is, not what it is not and should not contain features that belong to superordinate or subordinate concepts.

There is no general characterization of the requirements for representational frameworks and individual standards spell these out only according to their particular needs. For example ISO 13584-32:2010 states that the modeling infrastructure should provide capabilities:

- “To model constraints on properties by restricting their domain of values”;

---


6 ISO 13584-32:2010 Industrial automation systems and integration - Parts library.

7 from part 42, Methodology for structuring part families, of ISO 13584-32:2010
• “To model and distinguish characterization classes and categorization classes”;

• “To model aggregation and composition using a single resource mechanism;”

• “To describe strings that carry external references;”

• “To connect classes that belong to different class hierarchies”.

In the knowledge representation literature, at large, there are several alternative philosophical principles underpinning current knowledge engineering methodologies. In one alternative, the entities in the real world are partitioned into (1) sets, (2) individuals, i.e. concrete entities, and (3) universals, entities that can be instantiated simultaneously by different individuals, such as fractures, molecular functions and drug effects [Degen2001, Smith2006]. The individuals are further divided into continuants, things that keep their form over time, such as individual patients, and occurrents, things that take place during time, such as a particular disease afflicting a given person or an occurrence of meiosis happening right now in a specific cell. The connection between continuants and occurrents can be summarized as follows: continuants participate in occurrents, e.g proteins catalyze reactions. The associations between entities are established via relations which can be classified, according to the types of things they relate, as relations between sets, between universals, between individuals, and cross-categorical.

Several other distinctions between entities are advocated in the literature, including:

(1) **physical** versus **non-physical** entities: non-physical entities are manifested/defined by physical entities, e.g., a semi-folded sheet of paper defines an angle, a cell membrane defines the interior of the cell. Sometimes, however, non physical entities are manifested by non-physical entities as in depression is manifested by sad thoughts.

(2) **discrete** versus **mass entities**: discrete entities are constituted/made of mass entities, e.g., heart is made of heart tissue. Intuitively, discrete entities can be counted while mass entities can only be measured

(3) **collectives** versus **constituent (discrete) entities**, e.g. liver cell is a constituent of liver, a wheat ear is made of wheat florets. Loss or gain of constituent discrete entities does not, in principle, affect the identity of the collective.
In this research, a *knowledge representation* is an abstract description which refers to some entity or entities and a *representational artifact* is a representation that is materialized in some medium, e.g. paper-based diagrams, a vocabulary stored as text in a digital file, etc. However, in what follows, for ease of reading, we will use *knowledge representation* instead of *representational artifact* when the distinction between them is not important.

Although pertinent, to some degree, to our discussion of knowledge representation requirements in Chapter 2, we will not address the many philosophical debates currently raging in the field, e.g., (1) about mass and discrete entities, as illustrated by the question if a liver tissue “exists” as a distinct entity separately from the liver it was extracted from, or (2) about the *cognitivist* and the *realist* approaches to modeling knowledge (in brief, in the cognitivist approach, knowledge is in the mind and not all concepts are required to have instantiations in reality, while realism, as expounded in [Lord2010, Smith2004-b, Smith2006], requires that all classes/terms/concepts in a representation have instances in reality).

**Desiderata for knowledge representations.** Various requirements for representing knowledge have been mentioned in the literature. Some were put forward as a means of comparing various representational languages [Golbreich2003, Stevens2006] others simply to express general principles about the structure and purpose of representations. We quote here a list of desiderata that has been assembled by researchers working on medical terminologies, noting that, although proposed in the biomedical context, they have a general applicability. Cimino was the first to contribute to this, as yet, evolving list by articulating the following main themes: *vocabulary content*, i.e., good domain coverage; *concept orientation*, i.e., nonvagueness, nonambiguity and nonredundancy; *concept permanence*, i.e., concepts cannot be made obsolete once created, although their names can change; *polyhierarchy*, i.e., multiple inheritance; *formal definitions*,

8 We note, however, that some authors, e.g., [Dumontier2010], have pointed out that *realism* is incompatible with representing, among other things, scientific hypotheses and objects that are conjectured, but not yet proven, to exist, e.g., a term representing the ability of boron to form triple bonds would not have been allowed in a realist ontology for chemistry, as this fact, although predicted theoretically at the beginning of last century, was only proven empirically very recently (June 2012). http://www.nature.com/news/boron-finally-gets-a-triple-bond-1.10836

9 Concept - an abstract idea representing specific instances - is used here as the meaning of symbols as understood by humans [Cimino2006].
i.e., definitions that can be manipulated by computers; *no residual categories*, i.e., “not elsewhere classified”, or “not otherwise specified”, or other catch-all terms as a solution for the incompleteness of the representation; *multiple granularities; context representation; graceful evolution; nonsemantic; concept identifiers; and multiple consistent views* [Cimino1998].

In the face of criticism from researchers who advocate for terminologies to be composed only of *universals* with corresponding instances in reality, e.g. [Smith2008], Cimino later defended the above desiderata, and the concept-oriented approach to modeling knowledge by stating that the coexistence of concepts and universals in the same representation is not only possible but also desirable [Cimino2006]. He also extended the original list with several, to some extent overlapping, desiderata that address the purpose of terminologies. According to him, engineered terminologies should provide support for (1) *capturing what is known*, i.e., the objective observations, not just our interpretation of them, (2) *information retrieval*, i.e., querying (3) *lossless information storage, retrieval and transfer*, i.e., graceful evolution (4) *data aggregation*, (5) *data reuse*, and also (6) *inferencing*.

**On the use of logic for representing knowledge.** More often than not, logic, in one of its many incarnations, is the formalism employed for representing knowledge, as some of the above desiderata can only be fully met by formal knowledge representations. As an added bonus, logic-based representations are also, generally, amenable to automated computation. In the absence of a universal logic, however, we are left with the task of investigating the representational strengths and weaknesses of the various available logics. We briefly recall here the most widely used logic-based languages, with an emphasis on aspects related to expressiveness and computational complexity.

Frame-based languages, classes or frames are the fundamental building blocks. They represent collections of instances and have a set of slots, or attributes, associated with each of them. These slots can be filled with values from other frames. A taxonomy in a frame-based system is specified with the help of a special type of slot names *kind-of*, which allows the specification of subclass relationships. Particular instances are represented using *instance frames*, and an inheritance mechanism allows subclasses to inherit the slots of their superclass. Frame-based systems have a long history in knowledge representation, Ontolingua [Farquhar1997], being a representative example.
Description Logics (DL) provide a richer alternative to frame-based systems. They allow the formalization of knowledge into classes, related via subsumption and equivalence relations that can be used to automatically infer an induced taxonomy, in contrast to frame-based systems where the backbone taxonomy has to be manually defined. In a DL framework, concepts can be defined in terms of other classes using binary relationships called roles. DL frameworks also benefit from the availability of reasoning services that can check the satisfiability and consistency of the descriptions, in addition performing automatic construction of class hierarchies and the classification of instances.

More expressive frameworks are grounded in First Order and Second Order Logic, which offer knowledge engineers almost unlimited expressive power. However, the use of first-order logic constructs has to be judicious, as there are subtle tradeoffs between expressive power and computational complexity, which can range from tractable to undecidable, e.g., [Doyle1991, Levesque1987, Stevens2000].

Identifying a formalism that is sufficiently expressive to allow us to describe information in a convenient and shareable form covers, however, only one side of the modeling process, as constructing a (formal) representation, is rarely an end goal per se. Knowledge representations, although useful by themselves, are often part of complex software systems, such as knowledge bases that act as complex repositories of information and are equipped with (1) reasoning services, mainly geared towards supporting the conceptual representation, as well as (2) database-like querying facilities that provide access to the data. Consequently, knowledge engineers need to concern themselves also with whether efficient algorithms exist for (1) solving the practical problem whose representation they constructed and/or (2) extracting and inferring information from the described knowledge, e.g., via automated reasoning and querying services.

We are interested, thus, not only in the descriptive complexity of a problem, i.e., how difficult it is to define it, but also in its algorithmic complexity, i.e., how difficult it is to solve/answer it. The connection between the two permeates computer science research, which has produced both

---

10 We are focusing here on logic-based, non-procedural query languages, e.g., relational calculus, Datalog, etc.
generic algorithms for solving problems definable in certain logics and also characterizations of
complexity classes in terms of the logical definability of the problems they solve.

Problems expressible in a certain logic, on a specific class of structures, are known to have
specific computational upper bounds, the direct practical implication of which being that
defining a property/question of interest in a logic gives immediate upper bounds on the
computational complexity of the algorithms that can solve it. It should be noted, however, that
the converse argument does not hold, i.e., the undefinability of a property in a logic does not
imply that no efficient algorithms exist for solving it.

In Appendix F we list many of the well known descriptive complexity results, such as the fact
that properties definable in first order logic, with some additional constructs, are computable in
polynomial time. As an example, the question whether a graph $G(V, E)$, where $V$ is the set of
nodes and $E$ is the set of edges, contains a triangle is definable in first order logic as $\exists x, y, z \ E(x, y) \land E(y, z) \land E(z, x)$ and various algorithms can be implemented that compute the answer in at
most polynomial time in the size of the input graph\(^{11}\). Asking whether the same graph is 3-
colorable or admits a Hamiltonian path, however, cannot be done with the help of a first order
logic formula, as we need the ability to quantify over sets, only possible in higher order logics,
existential second order logic (ESO), in this particular case. Both problems are known to be NP-
complete\(^{12}\).

Of particular importance for practical applications of knowledge representation and reasoning is
the computational complexity of query answering over knowledge bases. This question has been
actively investigated by researchers in the database community, but although many of the results
established for database queries are also valid for queries over knowledge bases, not all of them
are, as classical database systems (1) are not “aware” of the conceptual information, i.e., the

\(^{11}\) In fact, every graph property definable in first order logic can be computed in linear time (Seeses’ theorem).

\(^{12}\) In fact, it is known that NP is exactly the class of problems definable in ESO and immediate NP-completeness
results can, thus, be obtained for a variety of problems that are ESO-definable.
intensional part, or TBox\textsuperscript{13}, in knowledge bases, and (2) operate under closed-world semantics, while knowledge bases function under open-world semantics.

The complexity of evaluating queries is traditionally divided into (1) data complexity, where the query is assumed fixed and only the size of the database/knowledgebase is taken into consideration, and (2) combined complexity, in which both the query and the database are considered variables, i.e., part of the input. A third class, expression complexity, in which only the database is fixed, is considered very close to combined complexity and rarely discussed separately. As pointed out by Vardi in his seminal paper on relational query languages [Vardi1982], data complexity is a measure for the expressive power of the language in which the query is expressed, as it essentially characterizes the difficulty of the questions that can be asked in that language. On the other hand, expression complexity measures “the succinctness of the [query] language”. As observed by Vardi, the characteristics of a query matter much more than the size of the database, and for Boolean queries\textsuperscript{14} the expression complexity is (generally) exponentially bigger than the data complexity.

There is a direct connection between query answering on a database/knowledgebase and the entailment of logical formulas, which is a decision problem that can be studied with the help of complexity theory. More specifically, given a knowledgebase $B$ and a Boolean conjunctive query $q$, answering $q$, i.e. returning true/yes or false/no, amounts to checking whether $B$ entails the logical formula corresponding to $q$. If $q$ contains distinguished variables, i.e., the answer set for $q$ contains tuples of values instead of just true/yes or false/no, answering $q$ can be reduced to checking query entailment by rewriting $q$ into a set of conjunctive queries such that each query can be expressed as a concept. The answer set for these queries can be checked in time polynomial in the number of distinct individual names used in the ABox [Glimm2008].

\textit{Caveats:} many of the algorithms developed for answering conjunctive queries over knowledge bases assume that the relations used in the queries are not transitive, which can be a severe

\textsuperscript{13} In a knowledge base (KB) the TBox holds the definitions of the concepts and relations (roles) as well as the axioms that together define the terminology employed by the KB. The second component of a KB is the extensional part, or ABox.

\textsuperscript{14} Queries with no distinguished variables, for which the answer is yes or no.
limitation. Some algorithms also impose the restriction that only individuals named in the ABox are assigned to query variables and under this condition, the semantic interpretation of the queries is no longer first-order, i.e., open world. Some current tools, such as Pellet\textsuperscript{15}, do allow for the evaluation of queries under the standard first-order semantics, but do that for very restricted classes of queries. It is also worth noting that query answering over knowledge bases, under standard first-order semantics, corresponds to answering queries over database with incomplete information, and many complexity results from that area of research transfer over directly.

*Increasing the expressive power of a language may lead to undecidability.* Even in fragments of first order logic for which answering conjunctive queries is decidable, relatively minor augmentations in the expressive power of the language, such as adding negation, or general comparison operators, even standard inequality, or rules which allow relational, i.e. Datalog-like queries, can lead to undecidability [Rosati2007]. All is not lost, however, as in a very recent line of inquiry, De Giacomo and co-workers, have also shown that extending certain description logics\textsuperscript{16} with higher-order logic capabilities, based on Henkin-style semantics, has, under certain suitable restrictions, no impact on the computational complexity of various reasoning tasks [DeGiacomo2011]. Further complexity and undecidability results can be found in Appendix F.

Since they expose the “critical” combinations between representational and query constructs with respect to the complexity and decidability of basic reasoning and query answering, the results presented in this section have important implications for the development of practical knowledge-based systems, especially for the design of the knowledge representations we refer to, in this thesis, as *practical ontologies*.

### 1.3 What is a (practical) ontology?

The word *ontology* has had a long history in philosophy, where it is a branch of metaphysics in which it functions as a theory of the nature of existence and deals with topics such as: what is

\textsuperscript{15} http://clarkparsia.com/pellet/

\textsuperscript{16} Description logics are, in fact, syntactic variants of first order logic.
existence, what entities can be said to exist, and how can entities be grouped, or subdivided based on notions of similarity and difference.

In information science, however, an ontology denotes an artifact produced with the purpose of representing knowledge about a particular domain\(^\text{17}\) and it is defined rather broadly, as *an explicit specification of a conceptualization* [Gruber1992], i.e. a description of the concepts and relationships that are presumed to *exist* in a domain of interest, as envisioned by an agent or a community of agents. Although the usage of the terms specification and conceptualization, as well as the intended realism\(^\text{18}\) of the model are still debated, in essence, according to this definition, an ontology specifies the concepts and relationships, that are relevant for modeling a domain, in the form of definitions of representational vocabulary, e.g. classes and relations, and formal constraints (axioms).

Some authors object that the above definition is overly broad and admits too wide a spectrum of specifications, from simple folksonomies, glossaries, and thesauri to complex logical theories expressed in first or second order logics [Smith2001] and the World Wide Consortium (W3C)\(^\text{19}\) has tried to bring more clarity to the matter by defining an ontology as “a set of *precise* descriptive statements about some part of the world (usually referred to as the domain of interest or the subject matter of the ontology)”\(^\text{20}\). Computer science research, aiming to take advantage of the developments in mathematical logic made during the 20th century, e.g., [McCarthy1980], further narrowed the definition of an ontology to a formal computational model. Ontologies, in the computer science sense, facilitate automated reasoning and are essential components of modern knowledge based systems [Gruber1992, Guarino1995, Neches1991]. As Calvanese and co-workers, among others, point out, there is no precise agreement on the definition of an

---

\(^{17}\) This departure from the original philosophical roots is still a hotly contested aspect in ontology development. For more information on this on-going debate: [http://www.applied-ontology.org/ontologicalrealism/](http://www.applied-ontology.org/ontologicalrealism/)

\(^{18}\) The realist perspective, an ontology is a representational artifact whose representational components designate universals that exist in reality, and also the relationships between them [Smith2006].

\(^{19}\) [http://www.w3.org/Consortium/](http://www.w3.org/Consortium/)

ontology [Calvanese2009]. However, a common core underlies virtually all approaches, identified by Uschold and Grüninger [Uschold2004] as:

- a vocabulary of terms denoting the things of interest in the domain being modeled;
- a specification of the meaning (semantics) of the terms in the vocabulary, preferably stated in a logic-based formalism.

Several ISO standards also touch on ontology engineering, albeit, for now, in a less extensive and methodical fashion than the standards dealing with the development and management of terminologies. For instance, the ISO/TC 37/SC 3 committee is working on standardizing “Ontology integration and interoperability (OntoIOp)”22. This standardization effort covers the following aspects:

- Representation of information in distributed ontologies and knowledge bases;
- Mappings between ontologies, services and devices, as objects definable in the language;
- Translations between ontology languages, as objects available in the language;
- Heterogeneous ontologies combining parts written in different languages.

As of now, there is no ISO-standardized definition of a basic ontology and for the moment many of the ISO standards that deal with topics related to ontology creation and management are in the development phase. Furthermore, since there is no systematized treatment of ontology issues and relevant subjects are discussed within separate standards, it is difficult, for the moment, to construct a cohesive picture of a (ISO) standardized ontology development framework.

**Practical ontologies.** The use of the term *practical* is rather informal in the literature. It is at times used as a replacement for the term *common sense*, as in *practical reasoning*, e.g., [Atkinson2006], but it is most often employed in reference to the feasibility of various

---

21 *Ontological engineering* is the information science field that studies methods and methodologies for the identification and formalization of knowledge into ontologies.

computational tasks such as reasoning over knowledge bases [Pease2000, Horrocks2005], query rewriting [Perez-Urbina2009], and language parsing [Dzikovska2004]. It is also used in direct connection to knowledge representations with the meaning of real-life machine-understandable representations, e.g., practical knowledge representations in [vanHarmelen1999] and practical ontologies in [Horrocks2005]). In this research, we subscribe to the latter meaning and refer to the ontologies that model as machine-understandable representations the knowledge used for practical computational tasks as practical ontologies.

Desiderata for knowledge representations revisited. As research has moved from work on controlled vocabularies towards formal ontologies, the set of desiderata has evolved to suit this more complex form of representation. For example, Burgun lists the following desirable characteristics for reference ontologies: good lexical coverage, i.e., good domain coverage (in terms of lexical entities and the inclusion of many synonyms), good relational coverage, compatibility with standards, modularity, and ability to represent variation, i.e., capabilities for modeling non-canonical situations [Burgun2006]. Although meant as desiderata for ontologies to be used for annotating biomedical documents, the empirically derived representational requirements proposed by Bada and Hunter [Bada2011], are generally applicable and four of the six: (1) avoid general terms with context-specific meanings, (2) avoid ambiguous definitions, (3) include terms representing noncanonical instances, (4) good relational coverage, are related in spirit to, and partially overlap with the desiderata proposed by Burgun. The two remaining desiderata concern the management of ontologies: (5) integrate overlapping terms across ontologies, (6) integrate domain-specific ontologies into mid-level ontologies. In order to accomplish these last two desiderata, Bada and Hunter argue in favour of introducing explicit relationships among terms in different ontologies, preferably via formally specified concept identity and other types of relations. We contribute to this topic by introducing, in Chapter 4, and evaluating, in Chapters 5 and 6, semantic similarity measures that can be used to establish relationships between terms in different ontologies.

The desiderata discussed above are quite general and we endeavor to present in Chapter 2 a finer grained analysis of representational requirements.
1.4 Applications of ontologies

There are numerous applications of ontologies that fall under the broad umbrella of Semantic Knowledge Management. We briefly review here two of them that are related to our research.

*Semantic integration and search.* Many web-based systems, such as the e-commerce giant Amazon and the network of bioinformatics resources that form the GO Consortium\(^{23}\), are already employing internally, and also offering to users, practical knowledge representations from simple controlled vocabularies to fully fledged ontologies, e.g. Amazon’s product categories and the Gene Ontology, which users can exploit for a variety of tasks from tagging and annotating products and documents to formulating queries and performing complex analyses that lead to the discovery of new knowledge. Search engines, from general purpose ones such as Google and Yahoo, to those focusing on scientific articles, such as Google Scholar, PubMed\(^ {24}\), CiteSeerX\(^ {25}\), and Scirus\(^ {26}\), have also been benefiting for some time from the availability of document annotations\(^ {27}\) with terms drawn from the glossaries and ontologies developed for scientific fields. Moreover, search engine giants, such as Google, Microsoft, Yahoo and Yandex are also funding, together and separately, efforts to construct machine processable representations that will improve the computational understanding of real-world entities and their relationships and lead to more relevant search results. Among these initiatives, the WebSchema\(^ {28}\) and the GoodRelations Ontology\(^ {29}\), now adopted by most popular search engines, are perhaps the most well known and provide standardized terms that content creators can embed into Web pages using microformats such as hCard, hCalendar and hProduct. Google’s Knowledge Graph\(^ {30}\), a vast collection of facts about and connections between individuals, places

\(^{25}\) [http://citeseerx.ist.psu.edu/index](http://citeseerx.ist.psu.edu/index)
\(^{26}\) [http://www.scirus.com/](http://www.scirus.com/)
\(^{27}\) The annotations associated to a scientific paper are keywords describing the research areas to which an article is relevant, as well as other important information, such as the methodology and materials used.
\(^{28}\) [http://www.w3.org/2001/sw/interest/webschema.html](http://www.w3.org/2001/sw/interest/webschema.html)
\(^{29}\) [http://www.heppnetz.de/projects/goodrelations/](http://www.heppnetz.de/projects/goodrelations/)
\(^{30}\) [http://www.google.com/insidesearch/features/search/knowledge.html](http://www.google.com/insidesearch/features/search/knowledge.html)
and things in the world, is another recent and important effort in linking and codifying knowledge.

Creating these interconnected representations of the world is a crucial step towards building the next generation of knowledge integration engines, which understand human intentions and goals and taps into the joint intelligence humanity has so far accumulated. This will not only benefit immensely the now ubiquitous and indispensable Web search engines, which are in their semantic variant one of the high profile applications of practical ontologies, but also lead to the discovery of new facts and relationships, improving our collective knowledge.

**Knowledge discovery.** As previously mentioned, semantic markups and annotations not only improve the relevance of searches, but also support integrative knowledge discovery efforts and complex query answering. Smart digital assistants, such as Siri (Apple), Evi (TrueKnowledge), and Majel (Google), use practical ontologies which represent core common sense knowledge in order to process and answer natural language queries. Similarly, more complex knowledge discovery systems, such as Cyc’s Analytic Environment[^31], allow users to pose complex questions by stating their informational objectives, and return not just a list of potentially relevant documents, but also answers accompanied by justifications in the form of chains of reasoning. For example, Cyc’s financial analytic environment exploits rich relations among publicly available and corporate data in order to provide financial experts with answers that will impact their investment and trading strategies.

Biomedical researchers also use intelligent query answering systems, such as the Semantic Research Assistant (SRA)[^32] [Lenat2010], which provide a semantic interface to biomedical data sources to get answers to complex queries. e.g., *Which patients had a heart attack less than 1 week prior to a coronary artery bypass graft between 2011 and 2012?* or *What were the post-operative mortality rates for patients who developed a bacterial infection within three months of undergoing a pericardial window procedure?* These semantic data mining frameworks can also enable researchers to take advantage of existing metadata, e.g., annotations and document

[^31]: http://cyc.com/cyc/applications
[^32]: developed by the Cleveland Clinic Foundation
markups, to solve tasks such as patient clustering for clinical trials and prioritization of hypotheses, e.g., putative genetic and protein interactions, for wet lab experiments.

1.5 Research contributions of this thesis

The projects constituting the research contributions of this thesis advance the field of Semantic Knowledge Management [SKM2009] by improving the understanding of the requirements for representing knowledge, offering insights into the design of practical representations; providing a practical ontology for data exchange and supplying new tools for semantic integration (similarity measures to help find mappings between ontologies) and knowledge discovery (a web-based service for predicting functional gene associations).

In order to create formal representations (ontologies) of knowledge one must identify and understand the informational requirements that must be captured by the representation. As pointed out in the literature, e.g. [Grüninger1997, Schlenoff1996], there are two ways in which representational requirements are linked to ontologies: (1) they can act as a foundation for the creation of ontologies, and (2) they can be used to evaluate the completeness (wrt. of a set of requirements) of existing ontologies.

Representational requirements can also help determine relationships between ontologies by identifying which subset of requirements each ontology supports and whether some ontologies are in fact expansions of others, i.e. were designed by extension.

In this thesis we touch on relationships between ontologies in two ways, by designing an ontology for data exchanges that extends an existing ontology and by providing new semantic similarity based techniques for finding mappings between related ontologies. We also present applications of ontology mappings to bioinformatics.

In what follows we give a summary of the research contributions of this thesis.
Knowledge Representation Requirements for Practical Ontologies (Chapter 2)

Research questions

(1) What are the main types of knowledge of practical importance, what representational requirements do they have and how are the existing representational languages and frameworks addressing them?

(2) what are the requirements that a framework for developing and using practical ontologies should meet, i.e., what features will allow people who are not knowledge engineers to design and use practical knowledge representations (practical ontologies).

There is no systematic overview in the literature of the types of knowledge of practical importance, their representational requirements and of the extent to which the existing representational languages meet them. We attempted to help fill this gap by identifying types of knowledge of practical interest and providing a methodical analysis of their representational requirements and guidelines for choosing suitable representational languages.

We are also not aware of any empirically derived requirements for practical ontology infrastructures in the literature and we endeavoured in this research to help fill this gap by designing and administered a survey to people who are not knowledge engineering specialists but are currently using or plan to use ontologies in the future.

Based on the information collected during the interviews we extracted (F1) a set of basic types of knowledge they are dealing with regularly, (F2) a list of practical tasks that make use of ontologies, and the reasoning requirements of these tasks, (F3) accessibility and usability issues in current ontology-enabled systems, as well as (F4) issues related to shareability and the collaborative development of ontologies.

The information collected via de user survey, i.e., (F1), was expanded with data we gathered by reviewing the literature, e.g., [Antezana2009, Bhat2011, Elahi2007, Hoehndorf2007, Horrocks2005, Kutz2011, Kutz2012, Mate2012, Leite2005, Mendonca2011, Richesson2012, Sojic2011] leading to the identification and selection of the following types of knowledge considered to be of practical importance: spatial and
dynamic knowledge, activities and causality, complex associations, transformations and mutations, hypothetical, uncertain and qualitative knowledge, and the representations of patterns for which we provided (1) an in-depth analysis of their representational requirements as well as (2) guidelines for choosing appropriate representational languages.

Informed by findings F1-F4, we also provided recommendations for practical ontology frameworks that aim at making practical knowledge acquisition and representation accessible to people who do not have a background in formal knowledge representation techniques.

From Information Models to Knowledge Models: Insights into Practical Ontology Design – Towards a Core Business Data Interchange Ontology (Chapter 3)

Research question. Formalize the canonical business data interchanges.

The main motivation for undertaking this work is that of contributing to (G1) the semantic integration of the architecture of an enterprise, its information systems and the workflow logic into a synergistic unity, as well as to (G2) the semantic interoperability between enterprises.

Several existing research efforts have been directed at defining semantic frameworks for acquiring, organizing, sharing and using the knowledge intrinsic in business tasks and data exchanges, e.g., enterprise modeling [Grüninger2000, Ohgren2005], business process representation [Atefi1997, Haller2006, Pedrinaci2008, Fiorentini2009].

We designed the Core Business Data Interchange Ontology, CBDIO, a practical ontology for representing data exchanges, as an intermediary model that can assist in relating the existing ontologies in this area and help accomplish goals G1 and G2. CBDIO has no obvious inconsistencies and captures the intuition behind the Open Applications Group Integration Specification (OAGIS)\(^{33}\), an emerging industry standard which aims to provide

\(^{33}\) http://www.oagi.org/oagis/9.0/
a canonical business language for information integration. The ontology engineering process also provided us with the opportunity to gain further insight into the representational requirements for processual knowledge as well as representational requirements for practical ontologies.

The formal ontology we are proposing could help not only human analysts but we also anticipate that our work can also be used to support tasks, such as

- the creation of enactable descriptions of business processes
- automatic composition of business processes
- monitoring, problem resolution, exception handling and verification of integration scenarios

**Measuring the Similarity between Ontological Concepts (Chapter 4)**

Assessing the similarity between ontological concepts is a task central to ontology-enabled information integration as well as to a variety of knowledge discovery tasks.

*Research question.* Estimate the similarity between ontological concepts.

Many of the existing semantic similarity measures have limitations stemming from assumptions they make, e.g., some path distance-based measures assume that the taxonomical relations are uniform, [Rada1989, Sussna1993, Wu1994], while frequency based information content measures, e.g., [Resnik1995] depend heavily on the quality of the available statistics.

We proposed several alternative, complementary measures. Two of our proposals are intrinsic approaches to estimating the shared information content of two ontological concepts and, thus, do not rely on probabilistic assumptions and the availability of frequency data. Our approaches can be used to assess the similarity of ontology concepts when corpus occurrence or annotation data are not available, and other information-content based methods that depend on them, such as [Resnik1995], cannot be applied. When occurrence information is available, we have proposed to estimate the similarity between
two concepts using the relatedness ratio, an estimation of the mutual information between the random variables representing the two terms being compared. We have also introduced a new hybrid measure, relatedness, which combines information based and feature based approaches to measure the similarity between two ontological concepts and can be particularly useful when comparing concepts in different ontologies.

We used our intra-ontology measures to estimate the similarity between all pairs of term in the Gene Ontology and compared the results to those obtained by applying to the same pairs of terms several existing methods. Our measures correlate well with existing measures, but are more discriminating (produce a larger range of scores) and are more sensitive to the topology of the ontologies they applied. Task based evaluation for all measures are reported in Chapters 5 and 6.

Detecting Relationships between Concepts from Multiple Ontologies (Chapter 5)

The performance of semantic integration systems depends to a large extent on the availability and quality of the mappings between the various knowledge representations used by the entities they integrate. Most often these mappings are not available and must be defined manually, where feasible, or discovered automatically.

Research question. Discover relationships between concepts in different ontologies, in particular between the components of the Gene Ontology.

The three constituents of the Gene Ontology (GO) are independent hierarchies and although a biological process is thought of as a series of events or molecular functions\(^\text{34}\), until recently, there have been no associative relationships between terms in these three hierarchies to indicate, for example, whether a molecular function is involved in a biological process.  

\(^{34}\) According to the Gene Ontology documentation available at: http://www.geneontology.org/
The Gene Ontology consortium started introducing *regulates* relationships between terms in the MF and BP ontologies and this initiative is ongoing. However, although relationships such as the one between *regulation of kinase activity* (BP) and *kinase activity* (MF) are now made explicit others, such as between *transcription* (BP) and *aryl hydrocarbon receptor binding* (MF), are not. Furthermore, this initiative does not include other types of relatedness relationships, such as *localization*: for example between *nucleus* (CC) and *DNA binding* (MF), or between *chromosome* (CC) and *sister chromatid biocondensation* (BP).

We used the inter-ontology measure, *relatedness*, which we developed in Chapter 4, to discover mappings between different bio-ontologies.

The top mappings detected using our methods were qualitatively assessed by two expert biologists. Our experiments showed that the precision of the *relatedness* measure degrades gracefully from 100% for the top pairs to 71% for the whole set of 250 pairs, while the precision of the *relatedness* measure goes from 100% for the top 10 pairs to 77% for the top 100 pairs and down to 71% overall precision.

Our analysis of the top scoring mappings has shown that the *relatedness* measure is able to find term relationships that are not covered by the *regulates* relationships recently introduced in the Gene Ontology (approximately 60% of the true positives in our evaluation dataset denote relations that are not regulatory in nature). The direct implications are two fold: (1) the cross-ontology relationships detected should be submitted to the Gene Ontology consortium for consideration to be included in future versions of GO, and (2) there is an opportunity to augment the portfolio of relationships types currently used for representing genomic knowledge.

Moreover, our measure is also able to detect relationships that would not be detected by the lexical methods, which are the most widely used ontology mapping strategies. Another contribution of this study is the generation of a “gold standard” of 250 manually assessed mappings.
Application to Knowledge Discovery – An Integrative Method for Predicting Gene Associations (Chapter 6)

Research question. Estimate the functional association between genes using an integrative approach based on a diverse available data.

Most existing computational tools focus on the prediction of protein-protein interactions; however, the prediction of gene functional association has broader applicability because it can incorporate all forms of interaction (e.g., protein-DNA, microRNA-protein, pathway co-membership) involved in a biological process, pathway or disease, which can be useful in the clarification of cellular pathways, and can help direct and prioritize future experimental hypotheses.

In this study we developed an integrative method for discovering gene functional associations: the Gene Association Predictor (GAP), which incorporates the introntology measures we developed in chapter 4.

GAP improves on existing integrative methods by including a novel semantic similarity component to better take advantage of the available gene annotations. The proposed semantic mechanism of inferring gene associations is capable of processing implicit association evidence in order to identify and prioritize novel predictions of genes' functional relationships.

The current version of GAP integrates pathway information from many different public online databases (e.g., Reactome, KEGG, NetPath), Gene Ontology annotations as well as drug and disease associations mined from scientific literature. GAP’s semantic similarity measure is, however, general, and can be extended to take advantage of other types of biological data sources. GAP predicts interactions among 19,027 human protein-coding genes; the gene names and symbols are those provided by the HUGO Gene Nomenclature Committee (HGNC).

GAP’s predictive power was compared to that of other existing predictive methods using a multi-pronged evaluation approach: (1) overlap with datasets of known, experimentally derived, gene associations, (2) comparison with other prediction services, and (3) manual
curation of scientific literature. Our experiments showed that GAP has a superior *positive predictive value* (i.e. precision rate), and *specificity vs. sensitivity* (i.e., true positive vs. true negative rates) in identifying known associations as compared to the existing predictors. In support of the novel potential interactions predicted by GAP, we selected a set of highly-scored novel interactions (i.e., not currently found in any known databases), and then manually searched the scientific literature for corroborating information. We also provided further supporting evidence for GAP's predictions from an external expert curated gene-disease association database.

GAP is available via a web-interface at http://ophid.utoronto.ca/gap.

### 1.6 Thesis Outline

In chapter 2 we identify several important classes of representational requirements for domain knowledge, with a special emphasis on the biomedical domain, and investigate the expressive power of the formalisms needed to express them. We also examine existing knowledge representation solutions for the representational requirements we identified and discuss shortcomings and possible solutions. In chapter 3 we present a practical ontology designed with the goal of representing data exchanges, and discuss its applications and lessons learned. In chapter 4 we investigate in depth the problem of assessing the similarity between ontological concepts and propose three novel measures of similarity. In chapters 5 and 6 we describe practical applications of the similarity measures we proposed in chapter 4: finding mappings between different ontologies in chapter 5 and advancing biomedical knowledge using gene association predictions in chapter 6.
2 Knowledge Representation Requirements for Practical Ontologies

The task of classifying all the words of language, or what's the same thing, all the ideas that seek expression, is the most stupendous of logical tasks.

Charles Sanders Peirce, letter to editor B. E. Smith of the Century Dictionary

Knowledge is one of the most important resources at our disposal. As John McCarthy observed, our “ability to use common sense knowledge depends on being able to do common sense reasoning”35. An increasing amount of effort is being devoted to developing representational strategies that will improve our ability to share and process knowledge.

To date, most knowledge representation frameworks fall roughly into two categories: (1) very informal, with minimal infrastructure, or (2) formal, i.e., logic-based and thus benefiting from a solid theoretical infrastructure. The first approach, very popular in collaborative web-based environments such as Flikr, Del.icio.us and CiteULike, requires minimal specialized knowledge to use and develop, but does not provide a mechanism for performing data validation or ensuring data consistency, which impacts negatively on the level of trust associated with any type of data analysis. The abundance of idiosyncratic definitions also limits their usability. The second approach fosters the creation of ontologies which contain machine processable formal definitions and are primarily meant to be used in computer systems.

Traditionally, knowledge representations have been created by knowledge engineers who were domain knowledge experts and were expected to make justified choices with regards to the concepts and relationships chosen to model the knowledge within a certain domain. The difference between knowledge engineers and the users of knowledge representations, however, often blurs when the latter are able to influence and directly participate in the creation of the representation. Thus, there is need for a flexible framework that would allow, when desired, the inclusive participation of all stakeholders in the creation of a knowledge representation.

35 http://www-formal.stanford.edu/jmc/someneed/node3.html
We feel that there should be a middle ground between the two approaches and we therefore investigate in this work (1) what representational constructs are necessary, or just useful, for describing domain knowledge and (2) what facilities should be included in a practical representation framework, (e.g., inference engines, intuitive mapping construction) in addition to the direct importation of concepts from other ontologies.

We present here (1) the results of a survey on representational and functional requirements we conducted with users who work with ontologies now or plan to utilize ontologies in the future, and (2) an in depth-analysis of challenges and solutions for practical representations of domain knowledge.

2.1 Requirements for practical ontology frameworks – users’ perspective

The key research question examined here is what features should a framework provide for building what this research defines as *practical ontologies*, so that people who may not be computer specialists can, and would, use it.

In order to identify the gap between existing ontology representations, infrastructures and tools and the real-life requirements of the users who work with ontologies now or plan to make use of them in the future we have conducted a user survey and we report in this section our findings. We are not aware of any empirically derived requirements for practical ontology infrastructures in the literature.

For the purposes of this study we identified two roles of the potential ontology users. First role is “end-user” – a person who uses existing ontology representation, infrastructure, tools to achieve domain specific goals – for example find certain protein information in the ontology based bio-med knowledge system, comment on the existing information in the ontological representation, etc. Second role is the “ontology administrator” – a person who is responsible for managing ontology life cycle stages, such as design, creation, deployment, maintenance, etc.[Rosu2009]
While separating these two roles, we realize that these roles often cross pollinate and merge, especially in highly collaborative, community based environments.

**Participants and procedure.** 10 participants took part in our study, recruited from a university community. They have mostly a bioinformatics background, and 6 have formal computer science training. 9 participants have very little or no knowledge engineering training and 1 participant has had exposure to knowledge engineering methods.

The survey takers were asked to answer 20 questions divided into 3 categories: *use cases and applications*, *ontology representation*, *querying and sharing and reuse*, and were also invited to provide unstructured comments on ontology-related topics of interest that were not covered by the questionnaire. The opportunity to solicit clarifications of the questions and of the terminology used in the questionnaire was available to all participants.

### 2.1.1 Survey questionnaire and results

In what follows we present the questionnaire we gave to all participants in our survey accompanied by an aggregate analysis of the answers we have collected.

**Use cases and applications**

**Q1.** *What types of ontologies do you use: e.g. top-level ontologies, such as Cyc and DOLCE, which describe very general concepts that are the same across all domains, domain ontologies, such as Gene Ontology which model a specific domain, application ontologies, such as Cell Cycle Ontology, other?*

**A:** All participants have indicated that ontologies are involved in their work. 9 survey takers listed domain ontologies: the NCBI Taxonomy, the UniProt related ontologies, the Gene Ontology, the Cell Cycle Ontology, the Kegg module ontology and pathways ontology, and the
Mammalian Phenotype Ontology. One participant mentioned having used top level ontologies (Cyc\textsuperscript{36}), and thesauri (WordNet).

**Q2.** Please describe briefly how you use ontologies? If applicable, please describe a typical “user story”:

A: Most of the uses described by the survey takers are from the field of bioinformatics: annotation of objects of interest (genes), enrichment analysis, prediction of protein interactions, gene clustering, functional analysis of genes, enhanced data visualization (e.g., visualization of protein-protein interaction graphs) and ontology-enabled search/querying. Other uses mentioned are not domain specific: extraction of information from text and finding implied information.

**Q3.** What tasks are supported/should be supported by the ontology infrastructure in your domain?

A: The following tasks have been listed as being currently supported by the infrastructure used by the interviewees: ontology-enabled visual exploration of data, annotation of data with ontology terms and (ontology-enabled) keyword search/document retrieval, data clustering (e.g., annotation-based gene clustering), ontology-enabled data analysis (e.g., gene enrichment analysis),

The survey participants indicated that they would like (improved) support for: ontology assisted interoperability with other resources/data integration, ontology mapping (e.g., between terms describing functions served by proteins and biological processes/pathways and between terms describing the functions played by gene products and their localization in various cellular components), and creating autonomous local (task-specific) ontologies.

**Q4.** What additional properties/actions/tasks would you like to have at your disposal in the ontology/infrastructure that you use now?

\textsuperscript{36} Here Cyc refers to the open source release of the top-level Cyc ontology, also referred to as Opencyc available at http://www.cyc.com/opencyc.
A: The participants in the study indicated that they would like to: (a) have access to better ontology-enabled analysis tools (e.g. enrichment analysis); (b) be provided with mappings between different ontologies, or be offered the ability to construct (task-specific) mappings; (c) have access to ontologies with a richer pool of available relations and/or be able to define new relations in the ontologies with which they work (e.g. make explicit that a biological function term is related to a cell localization term), (d) graphic representation and clustering information, (e) be given a mechanism to represent the uncertainty of ontology annotations and/or of certain relationships between ontology terms. Improved user interfaces with the ontology infrastructures, and ontology-assisted data integration (e.g. data linking and merging) were also listed among the preferred additions to current systems.

Q5. What type of reasoning capabilities do you need the ontology to support (e.g. are two concepts equivalent, does a concept/relationship subsume another, what implied relationships can be inferred)?

A: The ability to check if two concepts are equivalent was the most frequently listed reasoning capability, 6 out of 10 survey participants, followed by concept subsumption (5/10), and the ability to infer relationships between instances or concepts (5/10). Support for probabilistic reasoning was also mentioned (1/10).

Q6. For your current ontology infrastructure, how would you estimate the level of usage of all the reasoning capabilities provided by ontology? (how much you use vs. how much is available). What reasoning capabilities you use the most?

A: 5 out of the 10 respondents did not answer this question. The other half of the survey respondents have indicated that the ontology infrastructure they currently use provides very limited reasoning capabilities, i.e., concept subsumption (largely assessed according to the taxonomy portion of the ontologies).

Q7. How would you rate the reliability of the knowledge contained by the ontology system (low, medium, high). On which factors do you base your rating?

A: 5 out of 10 respondents have high confidence in the knowledge captured in the ontology infrastructure that they use, 4 out of 10 have medium/fair confidence. One participant did not
provide a rating. It should be noted, however, that none of the respondents provided any objective evaluation criteria and thus the results have to be interpreted as purely subjective assessments.

**Q8.** How often do you find the ontology you use to be incomplete or missing some important concepts.

**A:** 4 out of 10 of the survey participants mentioned issues related to the quality of the ontologies that they use: missing ontology concepts, concepts that are too broad or too granular insufficient variety of relationships between concepts, unavailable mappings between concepts in different ontologies. The other participants either did not comment on this or reported not having yet experienced issues related to the incompleteness of the ontologies they use.

**Q9.** Is there a process for reporting/fixing bugs/errors/deficiencies in the ontology? If yes, please describe it briefly.

**A:** 5 out of 10 survey takers reported being aware of the currently available options reporting issues with the ontologies they use, i.e., emailing ontology administrators, internet-based bug tracking. The other half of the participants reported not being aware of the bug reporting procedures, if any.

**Ontology representation**

**Q10.** What language is used to represent the ontologies you are working with: description logic, RDF(S), DAML+OIL, OWL, etc.?

**A:** Not surprisingly, given the background of our survey subjects, only 3 out of 10 respondents could provide details of the representational language of the ontologies they use. The following languages were mentioned in the survey responses: OWL, CycL, RDF and OBO. Due to the limited number of study participants we cannot infer which language is the most often used to represent practical ontologies.

**Q11.** How long did it take you to become proficient in this ontology system? How would you rate your level of knowledge of this system (for example “power user”, “novice user”, etc.
A: 3 participants described themselves as novice users, 6 as medium or intermediate users and 1 as an experienced user.

Querying

Q12. What type of information are you searching for?

A: We grouped the types information reported by our participants as part of the following categories of activities: ontology-enhanced search, e.g., what proteins are located in parts of the cytoplasm, what proteins participate in the same biological process as another protein, retrieving information about protein interactions and associations, advanced ontology-enabled knowledge discovery, e.g., enrichment analysis (i.e., what Gene Ontology categories is a group of genes of interest enriched for), reconstituting the biological context associated to the activity of a gene, and ontology-assisted machine learning, e.g., semantic parsing of documents, i.e., concepts/entities and relationships identification with the purpose of deriving the meaning of individual phrases and entire documents.

Q13. What tools do you use to obtain such information? Please describe the process that you use for finding the information you are interested in.

A: The participants in the study indicated that they have used, or are currently using, the following tools and methods to obtain information of interest: manual search and data processing (2 users), online search/query tools and interfaces: e.g AmiGo offered by the Gene Ontology consortium (6 users), open source software packages: e.g topGo available on the Bioconductor project website\(^{37}\) (2 users), custom programs/own code (5 users).

3 out of 10 users reported using more than one method for finding information of interest.

Q14. How would you describe the process of finding information in the ontology (very easy, easy, moderately difficult, difficult, very difficult)? Please explain your answer.

A: 1 participant assessed the process as *difficult*, 1 other participant as *moderately difficult*, 4 out of the 10 survey takers indicated that the degree of difficulty depends on the type of information being sought, and *ranges from easy to difficult*, and the remaining 4 participants have indicated that they considered the process to be *easy*.

**Q15.** *In your opinion, would the search process benefit from functionality that analyzes search requests from different users, such as the suggestions offered by Amazon.com and/or other uses of collective intelligence? If applicable, please elaborate.*

A: 6 of the 10 participants indicated that recommendations, if available, would aid in searching for the information of interest.

**Q16.** *Would you find it useful to be able to define, or have access to, partial views that contain only information relevant to your tasks? Please elaborate why.*

A: *Yes* (7/10), e.g., a more compact/restricted view of the ontology is easier to interpret, *No* (3/10), e.g., access to the full ontology is preferred or needed.

**Sharing and Reuse**

**Q17.** *Does the ontology infrastructure that you currently use support collaboration among different end users? If it does, please describe the collaboration mechanism. If it doesn’t, would you find the addition of such collaboration functionality useful?*

A: 2 of the participants indicated that the infrastructure they use provides some form of collaboration, but they did not specify the details of the collaboration mechanism. 7 participants indicated that there is no support for collaboration in the infrastructure they use. 4 of them would welcome the addition of collaboration capabilities while 3 of them were not certain how it would be useful. One participant did not answer this question.

**Q18.** *In the ontology infrastructure that you currently use, how can you share the information that you find useful (or the query to obtain such information from ontology) with other users of ontology?*

A: 4 of the respondents were not aware of any facilities for sharing this type of information being offered by the ontology infrastructure they were using. 5 respondents mentioned at least one
potential/existing means of sharing useful information, e.g., mailing lists, user forums, and wikis. 1 survey taker did not answer this question.

Q19. *Is linking to, and importing from other ontologies important in your work? If yes, please describe what type of ontology integration issues you have encountered.*

A: 7 of the participants indicated that linking and importing from other ontologies was important or somewhat important: e.g., many single organism and protein databases import annotations with terms belonging to different ontologies. 3 participants indicated that this issue was not important to their current work.

Q20. *Would you like to be able to modify the ontology directly, suggest certain modifications to ontology owner or in any other way affect the design of the ontology?*

A: 6 of the participants indicated that they would like the ontology infrastructures to provide means for the user to influence and contribute to the design of the ontologies they are going to use, but 2 of them indicated that they would rather do this indirectly by suggesting modifications to qualified knowledge engineers or community members. Being able to see the modifications submitted by other user was also deemed useful.

2.1.2 Discussion

In this research we explored the issue of making practical knowledge acquisition and representation available and appealing to people who do not have a background in formal knowledge representation techniques.

**Research contributions.** The key research focus in this section was to identify the features a framework should provide that will allow people who are not knowledge engineers to develop and use practical knowledge representations (*practical ontologies*). More specifically, we endeavored to (1) find out what are the kinds of knowledge users are dealing with regularly and what are the types of problems they need to solve, and (2) extract accessibility and usability requirements for practical representations frameworks, e.g., what type of representational infrastructure might be easy to grasp by domain experts who may not be computer specialists.
We have also attempted to determine if the problems that users need to solve require advanced modeling constructs and what kind of reasoning tasks people care most about, i.e., what kind of inferencing support would be needed. Lastly, we tried to find out if the users consider the (general) principles of shareability and reuse important, i.e., if they require that practical representational frameworks provide mechanisms for importing other representations, support for managing alternative, contextual, definitions, and features for collaborative ontology development.

We obtained answers to these questions by administering a 20-question survey to 10 volunteers whose professional background is not related to knowledge engineering [Rosu2010].

The questions and answers in the preceding section are self-explanatory and therefore we only highlight several general findings.

*Types of knowledge and expressiveness requirements for representational languages.* After processing the types of information that our survey respondents indicated that they search for and work with, we extracted the following set of basic kinds of knowledge, equally applicable across domains: spatial knowledge (e.g., cellular localization of proteins), activities and causality (e.g., signaling pathways), complex associations (e.g., between drugs, genes and experimental conditions), transformations and mutations (e.g., progress of diseases), hypothetical (e.g., scientific hypotheses), uncertain and qualitative knowledge, and the representations of patterns (e.g., chemical compounds). We explore the representational requirements of these types of knowledge in the next section of this chapter.

Three quarters of our survey respondents indicated that they would need relatively simple representational mechanisms, mainly limited to the ability to construct inheritance hierarchies, and define concept structure, domain specific relationships between concepts, and simple cardinality constraints. The types of inferences listed by our survey respondents as useful were also largely confined to checking concept subsumption and equivalence. The aggregate responses suggest that the modeling requirements for designing practical ontologies may be well below the level of complexity offered by advanced ontology representation languages such as OWL.
Usability issues. It was reported by our survey takers that in some ontology-enabled frameworks it is not easily apparent how to use the available ontology to formulate queries. Other frameworks provide insufficient information, which causes users to misunderstand some of the ontological relations and leads to incorrect search results. Some ontology-enabled systems were reported as not allowing easy access to ontology information, such as the specificity of annotation terms (i.e., the depth of the term in the source taxonomy/ontology), and others were mentioned as not making explicit some of the assumption used by some of their tools, e.g., treating `isA` and `partOf` relationships between ontological terms equally when conducting gene enrichment analyses, which leads to inaccurate findings).

Sharing and collaboration. Based on the answers provided by the survey takers, we inferred that in order to foster collaboration, the best course of action is to implement a mechanism that (1) allows users to submit potential changes for review by the ontology administrators and (2) permits users with both domain knowledge and information engineering experience to directly modify ontologies, and (3) enable the import of concepts or modules from other practical ontologies.

Recommendations for a collaborative practical ontology framework.

From the information we collected through our survey and further discussions with the survey takers, it emerged that there is indeed a gap between the current ontology-enabled infrastructure and the practical needs of users who are not knowledge engineers and we make the following design recommendations:

(R1) Intuitive (human-readable) ontology representation language whose sentences can be translated automatically into a tractable fragment of first order logic.

This recommendation is supported by the fact that most of the representational and reasoning needs reported by the people surveyed did not go beyond having the ability to (1) represent hierarchies and simple constraints and (2) check concept subsumption and equivalence (answers to questions 1 to 5 in the survey). A representational language which is grounded in logic allows for the automated consistency checking of the ontologies being created within or uploaded to the framework as well as for the execution of diverse reasoning tasks. Moreover, a minimal entry barrier with regards to the skills required to develop and use ontologies is essential for the wide
adoption of such a framework, as is an intuitive, user friendly interface. Recent efforts such as the Distributed Ontology Language (DOL)\(^3\) (due to become an ISO standard in 2015) are aimed at fostering the interoperability of ontologies developed in a plurality of logic-based languages.

\((\text{R2})\) \textit{Easy to use mapping specification language.}

Support for integrating information from multiple sources featured often, either explicitly or implicitly, among the requirements reported by the survey respondents and the ability to define mappings to other ontologies is one of the key facilitating technologies. An easy to use interface and associated language for constructing mappings to external or internal ontologies is key for ensuring that the ontologies users create can serve their role as integration mediators.

\textbf{Limitations:} given the small number of respondents, it would be premature to draw any firm conclusions on the types of knowledge most commonly needed to be represented or on the most important reasoning tasks that a practical ontology framework would need to support.

2.2 Practical representations of knowledge

In this section we look into the challenges knowledge engineers must grapple with when developing practical knowledge representations.

\textbf{Research questions.} We investigate here are what are the representational requirements of types of knowledge of practical importance and how are the existing representational languages and frameworks addressing them.

There is no systematic overview in the literature of the types of knowledge of practical importance, their representational requirements and how they are met by the existing

\(^{38}\) Ontology Integration and Interoperability (ISO 17347) http://ontolog.cim3.net/cgi-bin/wiki.pl?OntoIOp
representational languages and we attempted to help fill this gap with the analysis presented in the remainder of this chapter.

The identification and selection of the types of knowledge deemed here to be of practical importance was guided by the information we gathered through the user survey presented in the first part of this chapter and by surveying the literature, e.g., [Antezana2009, Bhat 2011, Elahi2007, Hoehndorf2007, Horrocks2005, Kutz2011, Kutz2012, Mate2012, Leite2005, Mendonca2011, Richesson2012, Sojic2011].

We are focusing on the following categories: spatial and dynamic knowledge, activities and causality, complex associations, transformations and mutations, hypothetical, uncertain and qualitative knowledge, and the representations of patterns.

As there is a wide array of representational languages and frameworks to choose from, we endeavor to offer an analysis of representational requirements that can help knowledge representation practitioners with choosing between them when constructing practical ontologies. Although most examples we provide are from the biomedical domain, the issues covered in this chapter are generally applicable.

Knowledge representations are valuable both in themselves, as a way for people to organize and exchange their own information and beliefs, and also as a means for facilitating complex interactions among humans and semantically-enabled software agents, but we will focus here mainly on exploring requirements for representations that are machine understandable. We will, thus, let our analysis by guided by the often cited Artificial Intelligence desiderata [Poole2010] that a knowledge representation be: (1) of adequate richness for the purposes for which it is constructed, (2) compact, concise and natural, and (3) amenable to efficient computation.

In this work, we are not trying to establish which language is the best for practical representations of knowledge in general, as the most appropriate choice for a particular situation is likely to depend on a wide range of factors, from objective grounds, e.g., the expressive power of the languages under consideration, to more subjective reasons, e.g., personal preference for a representational paradigm over others. Instead, we explore the requirements that guide and constrain the selection of a suitable representational language, as well as the construction of knowledge models.
We are also deliberately not identifying our work as aimed at representing common-sense knowledge, as what qualifies as common-sense is still ambiguous at best, as already pointed out in the literature [Suh2006].

2.2.1 Spatial knowledge

Space information is at the core of our knowledge about the world and practitioners in many fields, from philosophers and linguists, to engineers and computer scientists, are searching for ways to represent and reason with it.

We start our discussion of spatial representation and reasoning with an example: the medical assessment of injuries, a difficult and knowledge intensive task which needs to be performed every time a patient walks into an emergency room. The medical practitioners combine the information provided by the physical examination, e.g., X-Rays, CT-scans, etc., with their biomedical knowledge in order to produce an assessment which is in turn used for determining the appropriate treatment for the patient. These triage-related tasks are time critical and knowledge intensive so an automated reasoning framework, e.g., [Rubin2006], that can combine image data and biomedical knowledge, e.g., anatomical, biomechanical, etc., into patient assessments can be of tremendous help in assisting medics perform their assessments more quickly, leading to a decrease in patient wait time and perhaps also in better outcomes for the critically injured patients.

In order to diagnose and evaluate the consequences of an injury, e.g., identifying the critical body structures that have been affected and the extent of the physical damage, such a framework needs to be able to integrate (spatial) image data into a three dimensional geometric model of the patient and identify the regions in this model with the appropriate anatomical organs and body cavities. These tasks require access to complex spatial representation and reasoning theories, as well as to comprehensive anatomical ontologies describing organs, body cavities, adjacencies, overlaps and other information that can be useful in making anatomical inferences. Current efforts in knowledge representation and reasoning are striving to provide computationally efficient models for these, and many other fields.
We will focus here on formalizations based on some form of spatial logic, but other approaches exist, such as representations based on semi-algebraic sets\(^{39}\), which have been used in spatial databases [Shekhar2001].

Spatial logics are formal languages whose variables range over geometrical entities and whose non-logical primitives denote relations among and operations on these entities. In brief, they are, as any other logic, characterized by:

- a syntax, i.e. along the lines of propositional logic, first order logic, higher order logic, etc.;
- a vocabulary of non-logical primitives, e.g., Convex (x), overlap(x, y), adjacent(x, y) etc.;
- a class of interpretations.

Spatial logics can include, of course, non-logical primitives that are not geometrical/topological in nature, such as parthood relations.

Several variants of spatial logic exist, of which the topological and Euclidian logics are perhaps the most well known. Topological logics are, in essence, spatial logics whose (principal) non-logical primitives have a topological character, with (topological) frames\(^{40}\) being the natural structures over which these logics are interpreted. One of the best known examples is RCC-8, for which we provide details later in this section and in Appendix C.

Euclidian logics are spatial logics whose variables range over subsets of an Euclidian space of fixed dimension, i.e., \(\mathbb{R}^n\), and whose non-logical primitives are representing geometrical properties, relations and operations over these sets, e.g., connected(x), convex(x), contact(x, y), closer-than(x, y, z), etc. Central to these logics is the notion of region\(^{41}\), as opposed to sets of points. Some logics also distinguish between the spatial regions and the physical objects that

\(^{39}\) Subsets of the n-dimensional Euclidean space \(\mathbb{R}^n\) defined by Boolean systems of polynomial inequalities

\(^{40}\) If X is a topological space, a frame on X is a pair (X, S) where S is a non-empty collection of subsets of X called regions. For example, (X, RC(X)), where RC(X) is a Boolean algebra over the closed sets of X, is the frame of regular closed sets in X.

\(^{41}\) A popular choice for set of regions is the set of regular closed sets over \(\mathbb{R}^n\), i.e., the sets which are equal to the topological closure of their interiors, although they include many pathological sets and are many times replaced by other sets, such as the regular closed semi-algebraic sets and the regular closed semi-linear sets.
occupy these regions. In terms of computational complexity, it has been shown that arbitrary first order theories of spatial logics, $\text{Th}_\sigma(K)^{42}$, are not, generally, decidable for arbitrary signatures $\sigma$ and classes $K$ of models. More complexity results in Appendix C.

**Geometry.** A complete axiomatization of the three-dimensional Euclidian geometry was given at the beginning of the 20th century by Hilbert who chose the following non-logical primitives for his axiomatization: $\text{point}()$, $\text{line}()$, $\text{incidence}(., .)$, i.e., point lying on a line, $\text{betweenness}(., ., .)$, i.e., point lying between two others, and $\text{congruence}(., .)$[Hilbert1988]. The models of Hilbert’s axiomatization are known as the Hilbert planes$^{43}$, and they include the (geometrical) plane with which we are all familiar.

Tarski proposed an alternative axiomatization, with only three non-logical primitives: $\text{point}()$, $\text{equidistant}(., .)$, and $\text{between}(., ., .)$ [Tarski1951, Tarski1999]. His first-order axiomatization of elementary$^{44}$ geometry is deductively complete and decidable$^{45}$ [Tarski1999] and his second-order axiomatization of full (solid) geometry is categorical$^{46}$, but undecidable [Tarski1956].

There are also several region-based, as opposed to point-based, axiomatizations of full geometry, the most well known of which, RGB, was put forward by Bennett and co-workers [Bennett2000]. This second-order theory is based on Tarski’s geometry of solids, and has as non-logical primitives the parthood relation and the concept of sphere. RGB is categorical, but it is undecidable$^{47}$. Weaker first-order variants of RGB are known to be incomplete.

---

42 $\text{Th}_\sigma(K)$, denote a collection of first order formulas describing $K$, a class of models, over $\sigma$, a vocabulary of non-logical primitives, e.g., $\sigma = (\text{convex}(.), \leq)$.

43 Elementary (plane) geometry is the study of Hilbert planes. It is also referred to absolute plane geometry, as the 13 axioms effectively eliminate the possibility that there are no parallels at all, thus precluding interpretations that would be models of the elliptic or spherical geometry [Greenberg2010].

44 Elementary in Tarski’s sense: he considered elementary the part of Euclidian geometry that can be formulated solely in first-order logic, without any set-theoretic constructs.

45 There is an algorithm to prove every statement in the theory, or its negation.

46 i.e., all models of the theory are isomorphic, in this case to the classical interpretation over the Cartesian spaces over $\mathbb{R}$

47 RGB can encode an undecidable set of axioms of first order arithmetic over $\mathbb{N}$ by representing equality, addition and multiplication as relations among regions whose number of components is associated with a natural number
Logical axiomatizations have also been given for the affine geometry, a generalization of Euclidean geometry characterized by scale distortions which deals with geometric properties that remained unchanged under affine transformation such as translation, contraction, expansion, dilation, reflection, rotation, shear, and similarity transformations.

In terms of representational power, affine geometry may be enough for stating qualitative, i.e., non-metric properties, but it should be noted that some notions of practical interest, such as that of an angle, are undefined in affine geometry. Other potentially useful properties, e.g., lengths of line segments, cannot be compared, as Euclid's third and fourth postulates are ignored in affine geometry. Tarski has proved that affine geometry does not admit a finite first-order axiomatization.

Affine geometry is also referred to as the theory of the betweenness relation operating on the open and convex subsets of the Euclidean plane, a view more akin to the meretopological approach to representing and reasoning about space, to which we now turn our attention.

**Mereotopology.** Among the formal theories of space, in contrast to the various geometries on metric spaces, which are concerned with quantitative spatial inferences, the meretopological theories aim to enable qualitative spatial reasoning, for the many practical scenarios where detailed and complete metric characterizations are not needed, e.g., it is sufficient to be able to infer that two objects are connected or disconnected, but not exactly how far apart they are.

Some approaches focus only on modeling mereological (parthood) information [Casati1999, Simons1987], others deal mainly with topological information [Cohn2001], while others try to build coherent mereotopologies which encompass both parthood and connectivity relations [Donnelly2004].

Mereologies are often referred to as the theories of “parts and wholes”, but, as it has been pointed out many times in the literature, e.g., [Casati1999], the monadic property of wholeness cannot actually be accounted for in these theories. Topologies partially solve this problem, but have shortcomings of their own in their difficulty in dealing with holes, e.g., describing sponge-

---

48 Affine geometry is in alternative description, Euclidian geometry with congruence left out.
like objects, body cavities, the interior of blood vessels, etc. In fact, their combinations, meretopologies, cannot actually provide a full characterization of the Euclidian space [Tsai2005] and, as it turns out, even the boundaries of holes cannot be determined/expressed using purely mereotopological tools [Varzi2007]. Furthermore, topologies are not expressible in classical first order logic and require higher order constructs, e.g., quantification over sets [Awodey2002]. From a knowledge modeling perspective, this inexpressibility result translates into the fact that many topological properties of practical interest, such as connectedness, or acyclicity, are not directly expressible in classical FOL. In fact, the class of FOL-definable topological properties is not yet completely characterized. This state of affairs has led to approaches where many topological properties of interest are handled by the introduction of non-logical primitives that directly express them, e.g. overlap(x, y), coincide(x, y), connected(x, y), where x and y are spatial regions, objects, etc. Other representational limitations of region-based topologies are handled by different types of logic, e.g., cone logic, see Appendix C.

Two types of parthood relations are common in the literature: (1) between instances, e.g., John’s kidney is part of John, and (2) between types, or classes, e.g., nucleus is part of cell. A parthood relations between two types A and B, is equivalent to saying that all instances of A are (instance-level) parts of some instance of B.

Most commonly, the parthood relation, e.g. part_of, is given a first-order formalization with the help of three ground axioms:

\[(A^P1)\] Reflexivity: \((\forall x)\) part_of(x, x)

\[(A^P2)\] Antisymmetry: \((\forall x,y)\) ((part_of(x, y) \land part_of(y, x)) \rightarrow x = y)

\[(A^P3)\] Transitivity: \((\forall x,y,z)\) ((part_of(x, y) \land part_of(y, z)) \rightarrow part_of(x, z)

It should be noted that, generally, part_of and has_part are not interchangeable, e.g., all cars have wheels, but not all wheels are parts of cars. On the other hand, all leaves are parts of plants (at least at some time), but not all plants have leaves. In practical representations the distinction between them has to be made clear so that classifiers can cope with situations in which both relations are used.
As pointed out by [Gangemi2001, Simons1987], part of, by itself, is not enough for representing and reasoning about the internal structure of entities and so relations expressing the notion of connection have been introduced in the literature, e.g., [Varzi2007, Simons1987], to supplement parthood. The relations modeling connection are generally assumed to satisfy, at least, the flowing ground axioms:

\[(A^C_1)\text{ Reflexivity: } (\forall x) \text{ connected}(x, x)\]

\[(A^C_2)\text{ Symmetry: } (\forall x,y) (\text{connected}(x, y) \rightarrow \text{connected}(y, x))\]

The relationship between parthood and connectedness can be stated as follows:

\[(\forall x,y) (\text{part\_of}(x, y) \rightarrow ((\forall z) (\text{connected}(x, y) \rightarrow \text{connected}(z, y))))\]

A widely used parthood and connection based approach for qualitatively describing relationships between objects is Region Calculus (RCC8) [Randell1992], whose eight base topological relations, disconnected (DC), externally connected (EC), equal (EQ), partially overlapping (PO), tangential proper part (TPP), tangential proper part inverse (TPPi), non-tangential proper part (NTPP), non-tangential proper part inverse (NTPPi) are illustrated in Figure 2.1. It is important to note that the basic relations are disjoint, in the sense that between any two objects in a topological space only one of the eight basic relations holds.

![Figure 2.1 The 8 basic relations of Region Calculus](http://www.informatik.uni-bremen.de/~till/)

---

49 For topological logics on arbitrary topological spaces, diagrams like the ones in Figure 2.1 are not reliable.

50 Image credit: Till Mossakowski, http://www.informatik.uni-bremen.de/~till/
As an example, TPP(tumor, left_breast) can be used to describe that a malignant breast cancer tumor is located inside of the left breast of a patient, right underneath the skin. If we wish to express the fact that the tumour is fully located inside the breast but does reach the surface of the breast, we would use the *not tangential proper part* relation, e.g. NTPP(tumor, left_breast). Many other useful facts can also be expressed as RCC-8 formulas, e.g. knowing that a region $x$ is contained in both regions $y$ and $z$ we can infer that $y$ and $z$ overlap: $(TPP(x, y) \land NTPP(x, z)) \rightarrow PO(y, z)$.

Although we can represent many useful spatial relationships, the expressive power of RCC-8 is rather limited and it cannot distinguish between connected and disconnected regions nor between regions with and without holes [Kontchakov2007], e.g., it is impossible to express the fact that there is a point where the occipital, parietal and temporal bone of a normal human skull meet, as RCC-8 only has binary relations. Depending on what type of connectedness we wish to describe, for instance point connections, e.g., that three geographical regions meet at a point, or surface connections, e.g., a cell colony is connected to the medium it is growing on, we need to introduce specializations of the basic connectedness relationship and appropriate ground axioms.

Various methods for extending RCC-8 expressiveness have been proposed in the literature. For instance, one can extend the RCC-8 with other relations, such as $convex(x)$, $containableInside(x)$, $congruent(x, y)$, $collinear(x, y, z)$, etc., e.g., [Cohn2001], or embed the RCC-8 concepts into a more expressive language. For example, Schultz and Hahn describe a description logic based approach which reconstructs six of the eight relations defined by RCC-8 and also includes several anatomy specific concepts and relations, i.e., $boundary$, anatomical proper part, $a$-$pp$, anatomically externally connected, $a$-$ec$, anatomically disconnected, $a$-$dc$, and partial overlap, $a$-$po$ [Schulz2000]. We note that integrating RCC-8 into description logic has to be treated carefully, however, as it is known that logics capable of supporting full RCC-8, e.g., $ALCRP(D)$, an extension of $ALC(D)$ with defined roles, are undecidable [Haarslev1999-a].

We can express facts that need the concept of a $boundary$, such as “the thoracic cavity touches the boundary of the abdominal cavity”, or a certain region is the proper part of the closure of another region, using other extensions of RCC-8, e.g., as in [Renz1999], or topological constraint
languages, such as \textit{TCC} \cite{Pratt-Hartmann2001}, but predictably, increased expressive power comes at a cost. For example, satisfiability\textsuperscript{51} is NP-complete for Renz and Nebel’s extension of RCC-8 \cite{Renz1999} and NEXPTIME-complete for \textit{TCC}. Complexity results for other topological languages can be found in e.g., \cite{Davies2012,Nenov2011}.

**Representing composition/aggregation.** We often think about objects in terms of their component parts or the materials they are composed of. More often than not, we probably do not pay too much attention to the mechanism(s) through different objects are assembled into a \textit{whole}, but such details become important for building an appropriate representation, e.g., we would like to be able to infer automatically that a fracture of the ulna is a fracture of the arm. It can be argued that whole-part associations are as much a product of our own mind as they are reflecting reality, but, without going too deep into a debate about the true nature of meronymic\textsuperscript{52} relations, we have to acknowledge that natural language abounds with examples of such relationships, e.g. “the torso is part of the body”, “growing up is part of life”, “the parts of a cell include the membrane, cytoplasm, nucleus”, etc.

Cognitive science research acknowledges that there are many different kinds of compositional relationship, depending, for example, on whether or not the parts have a functional or structural relationship to the object they constitute, and perhaps to one another as well, and whether they are the same kind of thing as the whole, or can be separated from the whole. Focusing on the various ways in which the English term “part of” and its cognates are used in natural language, Winston and co-workers \cite{Winston1987} have put forward a taxonomy of meronymic relationships that distinguished between six major types, according to whether the relation is functional\textsuperscript{53}, homeomerous\textsuperscript{54} and whether the part and the whole are separable:

\begin{itemize}
\item \textbf{Functional}: Wheels are a functional part of a car.
\item \textbf{Homeomerous}: The part and the whole are of the same kind, e.g slice-bread.
\end{itemize}

\textsuperscript{51} Caveat: as also pointed out in the literature, e.g., \cite{Kontchakov2011}, the satisfiability of a formula in an arbitrary topological space does not mean that the formula is actually \textit{realizable} by meaningful two or three dimensional objects.

\textsuperscript{52} Meronymic: relating parts and wholes, from Greek meros part + onuma name

\textsuperscript{53} E.g., wheels are a functional part of a car.

\textsuperscript{54} The part and the whole are of the same kind, e.g slice-bread.
(1) [functional, non-homeomerous, separable]: component-integral object, e.g. cell-membrane;
(2) [functional, non-homeomerous, non-separable]: feature-activity, e.g., respiration-living;
(3) [non-functional, homeomerous, separable]: portion-mass, e.g., slice-bread, grain-sand;
(4) [non-functional, homeomerous, non-separable]: place-area., e.g., oasis-desert;
(5) [non-functional, non-homeomerous, separable]: member-collection, e.g., cell-colony;
(6) [non-functional, non-homeomerous, non-separable]: stuff-object, e.g., coffee-cappuccino.

We note that this cognitive science view of the meronmy relations is consistent with the view of meronomy in knowledge representation, see e.g., [Varzi2007], i.e. partonomic relations are irreflexive, anti-symmetric and transitive, thus inducing a form of partial order between parts and wholes.

Understanding the distinctions between the various kinds of meronymic relationships, e.g., [Donnely2006], is important, as composing assertions involving different types of part-hood relations may lead to false inferences. For example, given that (p1) organelle is part of cytoplasm, (p2) cytoplasm is part of cell, (p3) cell is part of tissue, it is wrong to conclude that (c) organelle is part of tissue. This incorrect conclusion is due to the faulty application of the transitivity of part of across meronymic relations of different kinds, i.e., component-object type in the premises p1 and p2, and member-collection type in premise p3.

There are a number of relationships that are sometimes confused with part-whole relations, e.g., topological containment, e.g., coffee contained in a cup, and connection, e.g., that a refrigerator is connected to the electrical outlet does not make it part of the electrical system of a house. Other distinctions are more controversial and are beyond the scope of this thesis.

Representing structured objects, such as a heart, which has two ventricles separated by the intraverticular septum, goes beyond the capabilities of some currently widely used description logic based representational languages, e.g., OWL. Extensions to OWL that will allow for the modeling of structured objects have been proposed in the literature, e.g. description graphs and rules [Motik2008]. Representations built using extensions will allow inferences such as “if the septum of the left ventricle is perforated, then the septum of the right ventricle is also perforated”, since it would be possible to express the fact that the two ventricles share the same septum.
Representing locational information We also need a representational approach that allows us to relate objects to other objects that contain them, or to the regions of space in which they are located and permits us to describe information such as the coin mistakenly ingested by Tony remained within his stomach, a cork is blocking the patient’s trachea, the brain is inside the cranial cavity, or a ribosome is located inside the cytoplasm, and both are part of the cell, Toronto is located at 43.6481° N, 79.4042° W, York university is in Toronto, etc. in a way that is consistent to our intuition.

Mereotopology alone is inadequate for modeling this type of information, e.g., restricting ourselves to using only mereological concepts would results in representations such as (1) the coin is part of the stomach, (2) the cork is part of the trachea and (3) the ribosome is part of the cytoplasm. While the last description can be accepted as correct, if we agree that a ribosome is a part of the cytoplasm, albeit not a functional part of it, the first two descriptions clearly contradict our understanding of the respective scenarios.

A solution is augmenting the catalogue of knowledge representation relations with locative primitives that associate objects, such as the coin and the cork in the previous examples, with the objects they are related to or to the spatial regions which they occupy.

The question of whether locative primitives should be independent of the primitives describing parthood and connection, and what should be the interaction between them, is the type of investigation that forms the object of the many theories of location, some of which only allow physical objects to reside within spatial regions, and some which allow physical objects to also be located inside other physical objects, as is the case in some biomedical ontologies.

The location of an object can be given directly as spatial coordinates, e.g., “Toronto is located at 43.6481° N, 79.4042° W “, or in terms of spatial regions, e.g., “York university is located in Toronto”, or, at times, using rather more ambiguous natural language descriptions.

Following Varzi, [Varzi2007], we can distinguish between three types of location: generic, ubiquitous and entire locations, which we define below with the help of a binary predicate location\((x,y)\), whose intended meaning is “\(y\) is the precise/exact location of \(x\)”. It is important to note that, defined this way, location cannot be used in the following way: location(“YorkU”, “Toronto”) \(\wedge\) location(“YorkU”, “Ontario”), as either Toronto or Ontario is the exact location of
York University, but not both. To make this interpretation precise, it is assumed in this context that the following axiom is true:

\[ \forall x, y \text{location}(x, y) \land \text{location}(x, z) \rightarrow y = z \]

Precise definitions of the three types of locative predicates mentioned above can be given as follows:

- \[ \forall x, y \text{generic} - \text{location}(x, y) \leftrightarrow (\exists z (\text{overlap}(z, y) \land \text{location}(x, z))) \]
- \[ \forall x, y \text{entire} - \text{location}(x, y) \leftrightarrow (\exists z (\text{part} - \text{of}(z, y) \land \text{location}(x, z))) \]
- \[ \forall x, y \text{ubiquitous} - \text{location}(x, y) \leftrightarrow \exists z (\text{part} - \text{of}(y, z) \land \text{location}(x, z)) \]

and taking into account the axioms of classical mereotopology, several consequences can be thus inferred:

- \[ \forall x, y \text{location}(x, y) \leftrightarrow \text{entire} - \text{location}(x, y) \land \text{ubiquitous} - \text{location}(x, y) \]
- \[ \forall x, y \text{ubiquitous} - \text{location}(x, y) \leftrightarrow \text{generic} - \text{location}(y, z) \land \forall z \text{overlap}(z, y) \rightarrow \text{generic} - \text{location}(x, z) \]
- \[ \forall x, y \text{entire} - \text{location}(x, y) \leftrightarrow \text{generic} - \text{location}(x, y) \land \forall z \text{generic} - \text{location}(x, z) \rightarrow \text{overlap}(z, y). \]

This shows that, in this context, the choice of (exact) location as a foundational primitive is not mandatory, and one could just as well define locative relations in terms of the weakest primitive, generic location, instead. Of course, the validity of this conclusion rests on the background meretopological theory and the axioms used to define the intended meaning of the locative predicates. We just want to point out that it is difficult, if not impossible to argue in favour of choosing one foundational relation over another.

The primitives included here do not exhaust all the possible locative cases and many other, independent, choices of location relations exist, e.g. [Bittner2008, Casati1999, Parsons2006, Schulz2001, Varzi2007]. For example, Bittner and co-workers introduced locative relations to describe situations where \( x \) is located in \( y \) if \( x \)’s spatial region is part of \( y \)’s spatial region, \textbf{LocIn} (\( x, y \)), \( x \) and \( y \) partially coincide if \( x \)’s spatial region and \( y \)’s spatial region overlap, \textbf{PCoin}(\( x, y \)) [Bittner2008], while Schulz and Huhn defined \textbf{whole} - \textbf{location}(x, y), for modeling scenarios in
which *x is entirely, but not necessarily “exactly”, located within the boundaries of y* [Schulz2001]. We note that Parsons introduced a relation with a similar name but slightly different meaning [Parsons2006], which we would define in terms of the primitives introduced above as: \( \forall x,y \text{whole_location}(x,y) \leftrightarrow \forall z \,(\text{part_of}(z,x) \rightarrow \text{generic_location}(z,y)) \). Intuitively, this second definition is saying that is *x is wholly located in y iff all parts of x are precisely located in y*, i.e., no part of x is located somewhere else. It is also worth noting that in Parsons’s interpretation, *whole location* relates an object and the spatial region it occupies, while in Schulz and Huhn’s approach it relates two individuals, i.e. the coin and Tony’s stomach, and also leads to interesting consequences such as: if an object *x* is part of an object *y*, then *x* is also wholly located within *y*.

In the real world, of course, an object’s location can change over time so *location* might have to be defined as a ternary, rather than binary predicate in order to account for the temporal aspect as well, and we touch upon this elsewhere in this chapter.

Philosophical aspects such as the controversy between relationist, i.e., Leibnizian, and the substantivalist, i.e., Newtonian, conceptions of space are beyond the scope of this work and we refer the reader to, e.g., [Varzi2007], for a more in depth discussion.

**Limitations of existing approaches** To illustrate some of the mereotopological issues that current representations of knowledge are not able to grapple with, we give a few concrete examples from the Gene Ontology.

*Mitochondrial DNA replication* is defined in the Gene Ontology as “DNA replication that occurs in a mitochondrion”\(^{55}\), and *lysosomal membrane*, is defined as “the membrane that surrounds a lysosome”\(^{56}\). We give here a possible first-order formalization in which the locational context is specified with the help of the binary predicate *occurs_in*, which takes an occurrent, i.e., a process, as its first argument and a continuant, i.e., a region, or object, as its second argument.

\[ \forall x, \text{mitochondrial\_DNA\_replication}(x) \leftrightarrow (\text{DNA\_replication}(x) \land (\exists y \text{\_occurs\_in}(x,y) \land \ldots) \]

\(^{55}\) From http://geneontology.org/GO.ontology.structure.shtml#xp

\(^{56}\) From http://geneontology.org/GO.ontology.structure.shtml#xp
The current attempts to formalize biological knowledge use OBOL, a language of very limited expressive power that employs a genus-differentiae definitional pattern.

Let us consider the term “lytic vacuole within protein storage vacuole”, which is defined as a “membrane-bounded compartment containing crystals of phytic acid and proteins characteristic of a lytic vacuole, found within a storage vacuole”. Its corresponding OBO-format definition is:

```
name: lytic vacuole within protein storage vacuole
intersection_of: lytic vacuole
intersection_of: part_of protein storage vacuole
```

This definition does not properly convey the fact that the lytic vacuole is **fully located inside** a protein storage vacuole and that the relationship is **positional** rather than **functional**.

Some of the relations under consideration for inclusion in the Relations Ontology, such as `bounded_by`, would bring more clarity, as in:

```
name: cytosolic proteasome regulatory particle
intersection_of: proteasome regulatory particle
intersection_of: bounded_by cytosol
```

Nevertheless, we argue that the expressive power of the genus-differentiae definitional pattern is too limited for adequately representing complex biological knowledge. Furthermore, strictly adhering to this definitional pattern can lead to inaccuracies and inconsistencies, as in the following example:

```
name: nuclear telomere cap complex
intersection_of: telomere cap complex
intersection_of: part_of nucleus
```

The **nuclear telomere cap complex** (Figure 2.2) is a complex of DNA and protein located at the end of a linear chromosome in the nucleus and has the role of protecting and stabilizing that chromosome.
However, whether it should be considered a proper part of the nucleus is unclear, as this term refers to a protein complex and not to a chromosome, not even to region of the chromosome, as pointed out in the usage caveat attached to its Gene Ontology definition.

2.2.2 Dynamic knowledge

Dynamic knowledge is fundamental to our interaction with the world, e.g., describing and understanding the motion of objects or recognizing human behaviours. However, modeling knowledge about the entities that undergo transformations through time, e.g., losing or gaining parts, is a very difficult task, since classical mereological principles, e.g., the transitivity of parthood, are no longer valid.

Translating natural language descriptions of progressions into crisp formal descriptions encounters serious challenges since we need to be able to precisely define the apparition, disappearance and transformation of substructures in the course of a process so that we can use them to describe the beginning, end, as well as the key developmental stages of that process. For instance, an organism’s anatomy changes substantially between various developmental stages: in a mouse embryo the outflow tract is part of the primitive heart tube, but at a later stage the

---

57 Image credit: http://pharmacology.case.edu/department/faculty/Primary/Labs/Taylor/MacromolecularAssemblies.aspx

58 The conceptualizations of time differs greatly from culture to culture, e.g linear, in European cultures, cyclical, in some South-Asian cultures, a dynamic process, without past, present or future for the native American Hopi tribe [Hayden1987]. We use here the Western conceptualization of time.
outflow tract is part of the heart, as there is no primitive heart tube anymore [Hunter2009]. The medical description of a hepatic comma offers another example of developmental stages that are very difficult to precisely separate and define: “Since the brain function seen in liver failure (hepatic encephalopathy) progressively deteriorates as toxins accumulate, the syndrome of encephalopathy is categorized into four stages or grades. Grade I and II represent mild to moderate depression of brain function. In grade III, severe depression of consciousness occurs with some abnormal body posturing that signifies brain damage has occurred. The final stage of hepatic encephalopathy is grade IV. At this stage, the entire brain is affected and the individual becomes unconscious and unresponsive to painful stimuli (hepatic coma).”

Matters are also complicated by the fact that the order of events/stages can be given, or thought of, in absolute terms, i.e., dates or timestamps, or in relative terms, i.e., in reference to processes, e.g., prophase happens before metaphase in mitosis. Correct and complete formal descriptions of this type of information are, however, essential and thus, some form of spatio-temporal representation and reasoning is needed.

There exist several different philosophical approaches to tackle the integration of spatial and temporal knowledge, e.g., four-dimensionalist modeling, where a time dimension is added to the three spatial dimensions, phase-based modeling, where objects are “sliced” into instantaneous temporal sections, and presentism, which subscribes to the view that existence, in general, is equivalent to present existence. The first and second approach do not support the distinction between things (continuants) and events (occurrents), while the third approach does not support reasoning about objects that do not exist in the present, although one of its extensions aims to, at least partially, solve this issue by allowing for finitely many time-indexed presents, or histories [Smith2002].

We focus here on two main approaches to spatiotemporal representation and reasoning, the one that considers time and space to be dichotomous, and deals with “successions of instantaneous

---

59 http://www.mdguidelines.com/hepatic-coma

60 e.g., for our ability to formulate and answer questions such as What step follows anaphase in mitosis?
snapshots of the world” and the one which views time and space as a unified spatiotemporal theory, and models changes and processes [Grenon2004].

Although the two views appear to be incompatible, over time various attempts have been made to reconcile them and a spectrum of modeling options has been put forward, from (3 +1)-dimensional approaches with various types of temporal involvement, e.g., ordered sequence of temporally indexing of snapshots, complex snapshots and histories, to fully four-dimensional models dealing with one kind of hyperobject, a “chunk of space-time”, of sorts [Galton2004]. We review in what follows several of these attempts and the principles on which they are founded.

Grenon and Smith proposed a framework, now part of the Basic Formal Ontology (BFO)\(^{61}\), which rests on two first-order ontological theories, SNAP, which models continuants, and SPAN, which models occurrents\(^ {62}\), together with trans-ontological relations between the them [Grenon2004]. The top entities in the SNAP taxonomy are:

- substantial entities, i.e. substances, boundaries, fiat parts, aggregates of substances and sites;

- dependent entities, i.e., qualities, roles, functions, etc.;

- spatial regions, i.e., points, lines, surfaces, volumes;

while the top entities in SPAN are:

- processual entities, i.e., processes, fiat parts, aggregates, settings, instantaneous temporal boundaries;

- temporal regions, i.e., connected regions, scattered regions;

- spatiotemporal regions, i.e., connected regions, scattered regions.

---

\(^{61}\) http://www.ifomis.org/bfo

\(^{62}\) entities which unfold through time
It is worth noting that SPAN has *time* itself as a constituent, i.e., *Time* is its maximal temporal region.

The SNAP-SPAN framework includes a number of relationships distinguished into (1) *intra ontological*, e.g., parthood relationships between entities belonging to the same ontology, (2) *trans-ontological*, e.g., participation between an object (SNAP) and a process (SPAN), and (3) *meta-ontological*, e.g., temporal order between ontologies, or constituency, obtaining between an ontology and an ontological entity.

The integration of spatial and temporal knowledge can also be effected via a combination of spatial and temporal logics, the coherent merging of these logics has been realized according to various principles, three of which have been described in the literature [Kontchakov2007] as follows:

- The combined logical language should be able to express changes in time of the truth values of propositions describing purely spatial properties.
- The combined logical language should be able to express the evolution of spatial objects over finite periods of time.
- The combined logical language should be able to express the evolution of spatial objects over time⁶³.

The fine tuning of the interaction between the spatial and temporal operators in the chosen logics is essential to obtaining tractable, or at least decidable, spatio-temporal logics, as the straightforward combination of spatial and temporal logics generally leads to undecidable hybrids. Examples of decidable spatio-temporal formalisms include: RCC-8 + PTL⁶⁴ which is PSPACE-hard, RCC-8 + Allen’s interval calculus, which is tractable, TeRCC-8⁶⁵, whose satisfiability is NP-complete.

---

⁶³ The restriction to fixed, finite periods of time is dropped in this principle.

⁶⁴ Propositional Temporal Logic

⁶⁵ TeRCC-8 is designed for representing topological relationships between regions which hold over time. The time intervals are related using Allen’s time interval relations.
We note that, somewhat independently from the question of modeling spatial information, several formalisms have also been adapted for representing processes: e.g. a model-checking technique based on a combination between a rule-based language and a temporal logic language [Calzone2006], a form of graphical notation [Funahashi2008], petri nets [Matsuno2000] elementary actions [Peres2010, Maziere2004], π-calculus [Regev2001], situation calculus [Reiter2001], PSL [PSL]. As is to be expected, each of these frameworks has strengths, but also weaknesses due to various limitations, e.g., lack of separation between the representation of the processes and the objects they involve, oversimplifications, etc.

Somewhat independently of the logic-based efforts to combine spatial and temporal formalisms into coherent hybrids, researchers working on representing biomedical knowledge, e.g. the developers of the Relations Ontology (RO) and the members of the OBO Foundry, are proposing a hierarchy of temporal relations, such as starts, starts with, ends with, and ends, that (1) hold between occurrents, i.e., processes, and (2) implement some of the relations in Allen’s interval algebra for temporal reasoning, e.g., [Smith2005]. These relations are still under discussion, to our knowledge, and have not yet been included in BFO.

Relations for modeling biological knowledge have been mostly restricted to is_a, part_of, has_part, regulates, positively_regulates, and negatively_regulates, and although we welcome the initiative to significantly expand their range, we believe that, for the moment it falls short of providing a coherent solution to the representational challenges raised by biomedical knowledge since many of these foundational/primitive relations are only vaguely, if at all, defined, despite trying to convey complex relationships. The proposed relationships are also not general enough to be easily reusable. For instance, many of the relations proposed by RO’s developers contain in their naming (1) explicit locational information, e.g., dendrite synapsed in, downstream in neural circuit with, has soma location and has presynaptic terminal in, in neural circuit with, (2) explicit information about the result of an action, e.g., releases neurotransmitter, (3) explicit information about the result of a process, where the result seems to be a process itself, e.g., results in developmental progression of, results in formation of, results in maturation of, results in morphogenesis of and (4) results in disappearance of.

**Limitations of current approaches.** To the best of our knowledge, despite the above mentioned efforts, the temporally dynamic nature of biological relationships is not yet properly captured in
the existing biomedical ontologies and we subscribe to the opinion that this knowledge should be made explicit via formal logic definitions.

The expressive power of the current definitional approach, adhered to by the ontologies represented in the Open Biomedical Ontology (OBO)\textsuperscript{66} format, is insufficient for capturing the meaning of complex biological processes such as the exact location where a process unfolds and the temporal ordering between sub-processes, for example:

(1) “vesicle fusion with nuclear membrane involved in mitotic nuclear envelope reassembly”, defined in the Gene Ontology as “a cell cycle process that results in the joining of the lipid bilayer membrane around a vesicle with the lipid bi-layer membrane around the nucleus, and contributes to mitotic nuclear envelope reassembly”, and in obo format as:

name: vesicle fusion with nuclear membrane involved in mitotic nuclear envelope reassembly
is_a: vesicle fusion
is_a: cell cycle process
relationship: part_of mitotic nuclear envelope reassembly

(2) “protein insertion into mitochondrial membrane involved in apoptotic signaling pathway” defined as the “process in which a protein is incorporated into a mitochondrial membrane as the initial phase of the mitochondrial membrane permeabilization that takes place in the apoptotic signaling pathway” and in obo format as:

name: protein insertion into mitochondrial membrane involved in apoptotic signaling pathway
is_a: apoptotic mitochondrial changes
is_a: protein insertion into mitochondrial membrane
intersection_of: protein insertion into mitochondrial membrane
intersection_of: part_of apoptotic signaling pathway
relationship: part_of apoptotic signaling pathway

\textsuperscript{66} The OBO flat file format is an ontology representation language able to represent a subset of the concepts in the OWL-DL language family. In addition, OBO supports meta-data modelling and the modelling of concepts that cannot be expressed in DL languages. http://www.geneontology.org/GO.format.obo1_2.shtml
According to the Gene Ontology supplementary information for this term, it is intended to cover the insertion of pro-apoptotic proteins, e.g., Bax, into mitochondrial membranes at *an early step* in the apoptotic program, but this important temporal ordering information is in no way captured by the obo-format definition and remains confined to the textual definition.

**Comparison frameworks.** No general framework for comparing different spatio-temporal formalisms exists to our knowledge, but we list here a set of informal high-level requirements that we consider that a framework for representing transformations and progressions should meet:

1. *Support for representing transformations*, e.g., spontaneous and cumulative changes and conversions, as in catalyzed metabolic transformations, enzymatic reactions controlled by various proteins, cumulative changes in hepatic coma, etc.

2. *Ability to describe states and transitions*: e.g., the state of a particular gene is defined in terms of its expression rate.

3. *Ability to specify preconditions and postconditions*.

4. *Support for specifying stochastic/nondeterministic behaviours*: e.g., the effect of administering adrenaline to a patient can be either vasodilator, or vasoconstrictor depending on the expressions of the $\alpha_1$ or $\beta_2$ receptors in the patient’s cells, the $\lambda$-phages that infect Escherichia coli choose between a lysic or lysogenic life cycles\(^{67}\), etc.

5. *Support for defining inputs/outputs, or the creation/destruction of objects*: e.g., the lysosome organelles, the “stomach” of the cell, break down waste products, the pituitary gland produces melatonin, etc.

\(^{67}\) For a $\lambda$-phage, the choice between the lysic and lysogenic life cycle depends on the stability of the protein cII: a stable cII leads to the lysogenic pathway, while an easy to degrade cII leads to the lytic pathway.
2.2.3 Activities

The potential to accomplish an activity (function) versus the occurrence of an activity (function). The distinction between the potential to accomplish an activity, its realization/occurrence and the effector of the activity is very often implicit in natural language communications and many current representations do not, or are not able to, differentiate between these aspects.

For example the term histidine ammonia-lyase activity from the Gene Ontology refers to the potential of the enzyme histidine ammonia-lyase, a member of the lyase family, to act as a catalyst of the first step in the degradation of histidine according to the reaction: L-histidine = urocanate + NH3. Although histidine ammonia-lyase activity is defined as the “catalysis of the reaction: L-histidine = urocanate + NH3”, the implication is that the catalysis of this reaction can only be accomplished by the enzyme histidine ammonia-lyase.

A formal description of histidine ammonia-lyase activity should specify, at the very least, that it is (1) an occurrence of the (atomic) activity catalyzation, (2) as part of the complex activity degradation (3) of the object histidine, an essential amino acid, (3) effected by the agent histidine ammonia-lyase, an enzyme.

The lack of distinction between the potential to exercise a function and the occurrence/instantiation of that function can be problematic [Bada2011]: e.g., when this type of knowledge is used to annotate gene products that can perform different functions in the context of different biological processes, or different organisms [Wroe2003].

We also note that in recognition of the importance of distinguishing between the potential to participate in an activity and the actual participation, the developers of the Relations Ontology, the main ontology describing relations between entities in the biomedical domain, are currently considering the introduction of binary relations such as actively participates in, capable of,
capable of part of. However, these relations are still pending approval and it is not clear how, and if, they will be formalized.

**Activity versus participation in an activity.** Another distinction often blurred in natural language is that between activities and their participants. We contend that a proper knowledge representation should describe in which way an agent, or participant, takes part or influences an action, e.g., whether a biochemical compound (1) is an integral participant in the chemical reactions making up a biological process, or (2) it is a result of that process.

Improper or absent representational distinction between activities and their participants, reflected in conflation of references to functions/roles and the classes of entities performing them, can severely limit the potential for logical inferences.

In addition to limiting the types of inferences that can be made using such a representation, not distinguishing between activities and their effectors leads to confusing, or altogether wrong, definitions of certain terms, even at textual level, e.g., lyase activity, a molecular function, is defined in the Gene Ontology as the “catalysis of the cleavage of C-C, C-O, C-N and other bonds by other means than by hydrolysis or oxidation, or conversely adding a group to a double bond. They differ from other enzymes in that two substrates are involved in one reaction direction, but only one in the other direction. When acting on the single substrate, a molecule is eliminated and this generates either a new double bond or a new ring”. The definition starts with a reference to the action of cleaving chemical bonds but switches to talking about the effectors, the family of enzymes which perform these cleavages, i.e., the lyase enzyme family, rendering it neither a proper definition of the lyase activity nor a proper description of the proteins responsible for accomplishing this activity. We also note that the obo-formalization of this term in the Gene Ontology does not include any details other than the fact that lyase activity is a type of catalytic activity:

```
name: lyase activity
namespace: molecular_function
def: "Catalysis of the cleavage of C-C, C-O, C-N and other bonds by other means than by hydrolysis or oxidation, or conversely adding a group to a double bond. They differ from other enzymes in that two substrates are involved in one reaction direction, but only one in the other direction. When acting on the single substrate, a molecule is eliminated and this generates either a new double bond or a new ring."
is_a: catalytic activity
```
Functions versus roles. The distinction between functions and roles is at times not obvious. One view is that functions are intrinsic and roles are extrinsic and depend on context, e.g., an enzyme’s function is to catalyze a reaction and every time a protein acts as a catalyst in a reaction it would have realized an enzyme’s role [Smith2006]. According to other perspectives, roles are realizable dependent entities which are borne by continuants and realized by occurents, e.g., [Dumontier2008]. We also note that, at times, the role players that realize roles can be, at times, subparts of a continuant, e.g., the functional group gamma phosphate of the Mg\(^{2+}\)ATP molecule plays the role of transfer group during glucose phosphorylation. We review here several relations\(^ {69}\) aimed at making these distinctions explicit.

- **inheres_in**: connects dependent continuants such as qualities, functions, and dispositions to their bearers, e.g., catalytic activity function inheres_in catalytic molecule. In first-order formalization:

\[
\forall d, e \text{ inheres_in}(d, e) \leftrightarrow \\
(\forall d_i, t \text{ instance_of}(d_i, d, t) \rightarrow (\exists e_i, \text{ instance_of}(e_i, e, t) \land \text{ inheres_in}_i(d_i, e_i, t)),
\]

where \(d\) is a dependent continuant, \(e\) is an entity, \(t\) a time point and the instance level \(\text{inheres_in}_i\), in our notation, is taken as a primitive. bearer_of is the reciprocal relation.

We note that in the current proposal, inherence is a functional relation and thus “multi inherence” is not allowed, e.g., situations such as when a specific gene sequence “inheres in” multiple chromosomes are not covered.

- **function_of**: between a function and an entity (independent continuant), e.g., pumping blood is a function of the heart. has_function is its inverse.

- **has_role**: a relation between an entity (continuant) and a role. Its reciprocal relation is role_of.

---

\(^{69}\) These relations are proposals currently under review in the Relations Ontology (RO), http://obofoundry.org/ro/
• **realizes**: between a *process* and a *function*, where the process requires the execution of the function. At class level: a process $P$ realizes a function $F$ if and only if for any $p$ that instantiates $P$, there exists some $f$, and a time point $t$ such that $f$ instantiates $F$ at $t$ and $p$ realizes $f$, where $\text{realizes}^{i}$ is the primitive instance level relation of $\text{realizes}$, e.g., the process *histidine catabolism* realizes the function *histidine ammonia lyase activity*\textsuperscript{70}.

• **has_disposition**: between an independent and a dependent continuant.

A similar approach inheritance relation is also discussed in [Cocos2007], here presented in an adapted first-order logic form:

$$
\forall a_i, b_i, t, t' \in \text{inheres_in}(a_i, b_i, t) \land \exists \text{at}(a_i, t) \rightarrow \text{inheres_in}(a_i, b_i, t'),
$$

where $a_i$ and $b_i$ are variables ranging over instances of continuants and $t$ and $t'$ are time points.

At the class level, $\text{inheres_in}$ is defined as:

- $\forall a, b \text{ in}\text{heres}\text{in}(a, b) \iff \left( \forall a_i, t \text{ instance_of}(a_i, a, t) \rightarrow (\exists b_i, t \text{ instance_of}(b_i, b, t) \land \text{inheres_in}(a_i, b_i, t)) \right)$

- $\forall a, b \text{ bearer_of}(a, b) \iff \left( \forall b_i, t \text{ instance_of}(b_i, b, t) \rightarrow (\exists a_i, t \text{ instance_of}(a_i, a, t) \land \text{inheres_in}(a_i, b_i, t)) \right)$,

where $a$ and $b$ are variables ranging over classes of continuants and $t$ are time points.

SNAP [Grenon2004] provides another example of modeling of *inherence*, a form of existential dependence, which is an intra-ontological relation between a SNAP dependent entity and its bearer:

$$
\text{InheresIn}(x, y, t) \land \text{SpatialLocaion}(x, x', t) \land \text{SpatialLocation}(y, y', t) \rightarrow \text{Part}(x', y', t)
$$

The above axiom reveals that inherence is also a form of spatial subsumption in the sense that the spatial location of the inhering entity is part of the spatial location of its bearer.

\textsuperscript{70} Here, *activity* denotes a function and not a process.
2.2.4 Causality

In common parlance, a causal relation is the relationship between a set of factors, known as causes, and an effect (consequence), e.g. gene A regulates gene B, CXCL5 causes insulin resistance and Rck2 inhibits meiosis in yeast, an inherited AF508 mutation of the CFTR gene leads to cystic fibrosis. Often, but not always, causal knowledge also has an explicit temporal component while at other times the temporal causal nature of relationships is implicit, as in rain causes wet grass, and antibody to human leukocyte antigen triggers endothelial exocytosis.

Graphical models have a long tradition in representing causal associations. They range from simple directed graphs to (dynamic) Bayesian networks and Petri nets to systems of structural equations and logic. Although not inherently causal models, Bayesian networks are routinely used for modeling associations between discrete/categorical variables. Their interpretation is mainly qualitative, but the joint probability distributions of variable values can also be viewed as encoding quantitative causation information. For continuous variables, structural equation models, which capture the quantitative aspects of causal relationships as systems of linear equations, are a widely used choice, especially in engineering applications.

However, when causal inferences are sought for dynamic combinations of both discrete and continuous variables, logic-based approaches are better suited alternatives. It should be noted, however, that in general, conditional statements in logic are not statements of causality. An important distinction is that statements of causality require the antecedent to precede or coincide with the consequent in time, whereas conditional statements do not require this temporal order. Another sort of conditional, the counterfactual conditional, has a stronger connection with causality, yet even counterfactual statements are not all examples of causality.

Nonetheless, certain logics are naturally well suited for representing causal and dynamic (temporal) information. Among them, propositional non-monotonic causal logic (NCL) [McCain1997a, McCain1997b], various action languages build on generalizations of NCL, the linear time and computation tree logics (LTL, CTL, CTL*) [Huth2004], and their extensions, such as probabilistic CTL [Hansson1994, Kleinberg2011], originally intended for describing and reasoning about the behaviour of parallel programs and reactive systems. Many of these logics, however, have incomparable expressive power: each can express properties that cannot be expressed in the other. For instance LTL cannot express CTL formulas containing existentially
quantified paths, while CTL cannot express LTL formulas that select sets of paths without a specific property, i.e., the so called *strong-fairness* property. It should also be noted that, although decidable, the computational complexity of reasoning in these logics is not elementary. For example, checking satisfiability is PSPACE-complete for LTL, ExpTime-complete for CTL and 2ExpTime-complete for CTL*.

In biomedical ontologies, several “temporally-enabled” causal relations have been put forward71: *derived by descent from, derives from, developmentally contributes to, developmentally induces, developmentally preceded by, results in developmental progression of, results in disappearance of, results in maturation of, results in morphogenesis of, and results in regression of*. We note however, that these are all foundational relations whose intended meaning is not further formalized in a form that is amenable to automated computation.

We advocate here in favour of building formal, computational causal models which can then be used for determining and predicting the effects of interventions on the various constituents of a system, as well as for figuring out which components are informationally independent of each other, thus also assisting in the construction of more efficient, evidence-based, belief revision strategies.

2.2.5 Complex associations between entities

A representational framework needs to be able to deal with relations that associate more than two entities since examples of n-ary relations between entities abound in almost every domain, from the relation between buyers, products and suppliers in e-commerce, to biomedical annotations which relate a gene product, the annotator, the source of the evidence, and the time of the annotation, e.g., [McDonald2005].

The biomedical domain, in particular, teems with relationships that cannot easily, or naturally be transformed into binary relationships, such as (a) the transportation via the cytoskeleton of gene products from a cellular component to another cellular component by transporter proteins, under certain conditions, (b) transcription regulation, in which many different molecules take part with

71 submitted for final approval to the OBO Foundry
various roles, e.g., activated *rap1* functions as a negative regulator of *tcr* and *cd28*-mediated *il-2* transcription [Friedman2002], and (c) phosphorylation, by protein kinases\(^{72}\), as depicted in Figure 2.3.

![Figure 2.3 General scheme of kinase function\(^{73}\)](http://publications.nigms.nih.gov/order/index.htm)

Some representational frameworks, such as full first-order logic, support the representation of n-ary relations natively, via n-ary non-logical primitives. In other frameworks, e.g., description logic-based languages, if we wish to describe relations that involve more than two participants, in order to get around the restriction that all relations have arity two, we have to resort to workarounds that may not adequately convey the nature of the relationship, such as the potentially cumbersome and, at times, unintuitive statement reification mechanism. More specifically, in OWL, n-ary relations are simulated by turning the relation into a class as in the following example:

```turtle
:Diagnosis_Relation a owl:Class ;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:someValuesFrom :Disease ;
      owl:onProperty :diagnosis_value
    ] ;
  rdfs:subClassOf
    [ a owl:Restriction ;
      owl:allValuesFrom :Probability_values ;
    ]
```

\(^{72}\) A protein kinase is a kinase enzyme that modifies other proteins by chemically adding phosphate groups to them (phosphorylation).

\(^{73}\) Image source: Medicines by Design, National Institute of General Medical Sciences http://publications.nigms.nih.gov/medbydesign/images/ch4_kinases.jpg

Free of copyright, as per http://publications.nigms.nih.gov/order/index.htm
Things can get even more complicated if we need to capture some form of “internal” order, or structure within a relationship. For instance, relations modeling multi-substrate, enzyme-catalyzed reactions which follow a ternary-complex mechanism should specify that both substrates participating in the reaction bind to the enzyme in order to produce a ternary complex, e.g. the human DHFR enzyme reduces dihydrofolate acid to tetrahydrofolate acid, using NADPH as electron donor in a ternary-complex mechanism Figure 2.4.

Depending on whether the reaction follows a random or an ordered binding mechanism, the relation must also convey the order in which the substrates bind to the enzyme.

Figure 2.4 Human Dihydrofolate reductase, DHFR, with dihydrofolate (left) and NADPH (right) bound.

---

74 http://www.w3.org/TR/swbp-n-aryRelations/

75 An enzyme is a biological molecules that catalyzes chemical reactions.

76 A substrate is a molecule upon which an enzyme acts.

77 Image credit:tonycu88, licensed under the Creative Commons Attribution-Share alike

A mechanism similar to OWL’s reification scheme is implemented by BOWiki\textsuperscript{78}, a collaborative annotation and ontology curation tool. In this approach, similar to the one also put forward by the W3C Semantic Web Working Group\textsuperscript{79}, users can specify n-ary relations by representing their arguments as individual roles, i.e., named argument slots. For example, the fact that the \textit{glycoprotein hemopexin is a heme transporter in the blood plasma} can be represented in the framework provided by BOWiki as:

\[
\begin{array}{l}
\text{realizes::function = heme transporter activity;}
\text{realization = heme transport;}
\text{context = blood plasma }
\end{array}
\]

An alternative, first order logic representation, could use, for example, a ternary predicate $\text{transport}(\text{hemopexin, heme, blood plasma})$ whose intended interpretation is that the first argument is the transporter, the second is the entity being transported while the third is the location, or context in which the transport is taking place. This approach eliminates the need to reify relations, which can lead, at times, to cumbersome representations.

Each approach also has advantages and disadvantages: reification can provide a more economical representation in cases where there is a list of arguments for one, or several of the roles in a relation, while n-ary predicates offer a more natural and concise representation, in the sense that by defining just one predicate one can accomplish what reification can achieve only by introducing several.

### 2.2.6 Transformations, mutations, negative findings and pathological cases.

An important challenge in any attempt to represent knowledge is the description of entities that are transformations or mutations of other entities.

\textsuperscript{78} http://semanticweb.org/wiki/BOWiki

\textsuperscript{79} http://www.w3.org/TR/2004/WD-swbp-n-aryRelations-20040721/
Representing the results of transformations requires great care, as faulty descriptions can lead to wrong conclusions and unintended consequences, as in the following example drawn from the Gene Ontology. The term \textit{integral to membrane of membrane fraction} refers to fractions of cells that have been prepared from \textit{normal cells} by disruptive biochemical methods. Its inclusion in the Cellular Component hierarchy of Gene Ontology, leads to the top concept in CC, \textit{cellular component}, becoming its parent class and results in a conceptual inconsistency in the ontology. \textit{Cellular component} refers to locations, at the levels of subcellular structures, in viable cells, while \textit{integral to membrane of membrane fraction} refers to disrupted cells and may not correspond to structures that actually exist in intact cells.

Departure from normality takes many different forms: e.g. various genetic mutations cause in abnormal phenotypes mice, and other animals, such as deformities of the tail, syndactyly, albinism, and so on. For example, a statement such as \textit{Joe lacks cone cells}, is a proxy for saying that Joe suffers from monochromatism, a rare form of color blindness, and \textit{Mickey lacks tail} identifies Mickey as a tailless mouse, affected, perhaps, by the semi-dominant Brachyury mutation\textsuperscript{80}.

We can model the information we have about Mickey the mouse in an alternative way by defining a class denoting the strains of tailless mice, of which Mickey would be instance/member. We could also directly identify Joe as suffering from monochromatism by either expressing the fact the he is as a member-instance of the class \textit{color\_blind\_person (monochromatism)}, with the obvious definition, or by relating him to the class \textit{monochromatic\_vision}. However, this approach would not convey exactly the same information as the original statement, because monochromatism also afflicts some people whose eyes are not completely devoid of cone cells. We could, alternatively, define the class of “\textit{coneless}” people and declare that Joe is an instance of that class, but introducing a new class for every type of departure from normality may clutter the formal representation and negatively impact its usability.

\textsuperscript{80} a deletion on the T locus on mouse chromosome 17
Depending on the circumstances, one can deal with abnormalities at an instance level or at class level, but regardless of the approach, in order to describe atypical or anomalous situations, one has to first define normality, an endeavor fraught with difficult and often divisive philosophical challenges. Furthermore, in the examples discussed above, we assumed that the conditions afflicting Joe and Mickey are permanent, resulting perhaps from congenital mutations, but this type of information can also have a temporal aspect, as is the case with many clinical notes logged into patients records: e.g., the patient does not have cyanosis, and is in no apparent distress (at the time the ER physician made the assessment).

Information about the absence of a phenotype or a condition, the so-called negative observations, or negative findings, can be modeled in a similar manner and we distinguish here four categories, the first three of which have also been mentioned in [Ceusters2006]:

- \(<\text{individual, individual, } t>\) An individual is not related to another individual in a specific way at a specific time, e.g., the referral John received from his family doctor is missing from his hospital records.

- \(<\text{individual, class, } t>\) An individual is not an instance of a class at a given time, e.g. Mary’s edema is not an allergic angioedema.

- \(<\text{individual, class}>\) An individual is not related in a specific way to any instance of the class at some given time, e.g. the mouse housed in cage 10 did not exhibit abdominal contraction 5 minutes after being injected with acetic acid.

- \(<\text{class1, class2}>\) No instance of class 1 is related to any instance of class 2, e.g. tailless mouse lacks tail.

Several researchers have advocated in favour of including in the repertoire of foundational modeling primitives of lacks-type of relations, which would account for several types of negative findings as well as for departures from normality, e.g., [Ceusters2006, Cocos2007], for instance, the family of relations defined using the pattern \(i \text{ lacks } u \text{ at } t \text{ with respect to } r\), where \(r\) is the identity or a relation such as \(\text{quality_of}, \text{located_in}, \text{has_participant}\) and so on:

\[
i \text{ lacks } u \text{ at } t \text{ with respect to identity} \overset{\text{def}}{=} \text{there is no } x \text{ such that } x \text{ is identical_to } i \text{ at } t \text{ and } x \text{ instance_of } u
\]
The \textit{lacks_part} relation between an individual and a class/universal has also been discussed in [Cocos2007]. Its formal definition references, via what the author dubs as the “canonicity clause”, the normal/expected state of affairs modeled by a family of relations such as \textit{canonically_has_part}, \textit{canonically_has_initial_part_one}, \textit{canonically_has_at_least_one}. We reproduce here, in an adapted, formal form, several definitions from [Cocos2007].

- \textit{canonically_has_part}(a, p), with the informal meaning that all anatomical structures of type \(a\) have an anatomical structure of type \(p\) as a part:

\[
\forall a, p \ \text{canonically_has_part}(a, p) \iff \\
\forall a, t \ (( \text{instance_of}(a, a, t) \land \text{canonical_anatomical_structure}(a, t)) \rightarrow \\
\exists p_i \ \text{instance_of}(p_i, p, t) \land \text{part_of}(p_i, a, t))
\]

We note that in the absence of the constraint that \(p_i\) be a canonical anatomical structure, spurious assertions and inferences could creep unnoticed into a knowledge base that uses this terminology. Also, \textit{canonically_has_part}(a, p) only specifies that \(a\) has at least one part of type \(p\), so maximum and minimum cardinality constraints have to be specified separately.

- \textit{lacks_part}(a, p, t), i.e., an anatomical structure \(a_i\) of type \(a\) lacks an anatomical part of type \(p\), at time \(t\):

\[
\forall a, p, t \ \text{lacks_part}(a_i, p, t) \iff ( \exists a \ \text{instance_of}(a, a, t) \land \text{canonically_has_part}(a, p) \land \\
\neg \exists p_i \ (\text{instance_of}(p_i, p, t) \land \text{part_of}(p_i, a_i, t)))
\]
We note here that $lacks_{part}(a_i, p, t)$ specifies that $a_i$ is missing, at time $t$, all parts of type $p$, i.e., $lacks_{part}$(Mickey’s left front paw, toe, January 2012) means that Mickey’s left front paw is missing all 5 toes.

Although the formal and informal definitions in the above examples make some use of anatomical knowledge, they are however general in nature.

However, these approaches cannot deal with modal knowledge such as epistemic statements of the form as far as he knows, John is not allergic to anything or qualitative statements such as Mickey, the mouse, is almost tailless, i.e. he has a very short tail stump, which is the case for most “tailless mice”\(^81\).

Representing exceptions and other types of irregularities faces even more complex challenges. For example, “regular” eukaryotic cells are assumed to have a nucleus, as the existence of a nucleus or a nuclear envelope is considered a defining feature of eukaryotic cells [Youngson2006]. Mammalian red blood cells, however, do not have a nucleus, although they are eukaryotic cells, so one could be tempted to use the relations discussed above to define red blood cells as eukaryotic cells that lack nucleus. This approach would not be correct, however, as avian red blood cells happen to have nuclei.

To further complicate things, some eukaryotic cells, e.g. muscle cells, are actually polynucleate, and modeling this kind of knowledge requires the ability to specify some form of cardinality constraint, e.g., on the number of nuclei that a eukaryotic cell can, or should, possess.

Important information about exclusions/exceptions often remains confined to textual descriptions due to the inability of the adopted representational frameworks to account for them. For example the obo-format chosen by the Gene Ontology consortium is unable to account for exceptions such as the one contained in this description: “catalysis of the transfer of an alkyl or aryl (but not methyl) group from one compound (donor) to another (acceptor)”. As a consequence, this

\(^{81}\) Only a mutation of the Brachyury allele Tc has been documented to induce a true lack of tail phenotype in heterozygous mice.
description is represented as the term “transferase activity, transferring alkyl or aryl (other than methyl) groups”, in a format that is clearly not very suitable for automated inferences.

2.2.7 Hypothetical and uncertain knowledge

Uncertainty arises both from imperfect understanding of the domain we wish to model as well as from the incomplete information we have at our disposal and the way we acknowledge, model and manage uncertainty has important consequences to the perspectives we form from reasoning with that knowledge.

The need for these alternative models of representations arises, for example, when we:

- want to explicitly state possible exceptions and qualified statements, e.g., *pneumonia is typically caused by an infection but may also be caused by the invasion of the lung by eosinophils and the inhalation of dust and toxicants*, if a pregnant woman is infected with the Rubella virus within the first 20 weeks of pregnancy, the child *may be born with congenital rubella syndrome (CRS)*. Spontaneous abortion occurs *in up to 20% of cases* [Siegel1971];

- require a mechanism to express coexisting conflicting views, e.g., *Mary’s CT scan shows a hypodensity consistent with an infarct*, but she has other symptoms that seem to point to a different diagnosis;

- need to describe a change in beliefs due to advancement in the current state of knowledge, e.g., modeling molecular interactions requires a framework that is “elaboration tolerant” in the sense that it permits the continuous refinement of knowledge about biological actions, which may lead to changes in the order of events in the models of signaling pathways.

The nondeterministic, at times, nature of biological relations also makes adhering to the formal knowledge representation principle of univocity [Smith2004] very difficult. For example, during DNA replication, a structure called replication fork, made up of a template, a leading strand, a

82 http://en.wikipedia.org/wiki/Pneumonia
lagging strand, a primer, and Okazaki fragments, forms in the nucleus of the cell. We should be able to represent the fact that an Okazaki fragment is always part of a replication fork and that, in turn a replication fork is sometimes part of the nucleus of a cell.

We recall here that classical logics abide by the principle of monotonicity and do not support several types of reasoning that humans frequently perform, e.g., reasoning by default, i.e., conclusions may be derived in the absence of evidence to the contrary, abductive reasoning, i.e., conclusions are derived as the most likely explanations, and belief revision, i.e., new knowledge is allowed into the system, which may contradict existing beliefs.

However, a variety of strategies can be employed for representing different kinds of uncertain knowledge and which also permit us to operate outside the confines of monotonicity. Some of these formalisms provide solid theoretical underpinnings for non-monotonic reasoning, even though instead of the familiar rules of inference, one will use rules of conjecture, i.e., our conclusions, although appropriate given the current evidence, may be disconfirmed at a later date by the introduction of supplementary information. Modal logics, for example, provide mechanisms for representing necessity, possibility, eventuality and epistemic states. Hybrid frameworks also exist that combine classical logics with formalisms for representing uncertain knowledge, for example, first-order logic augmented with \{Yes, No\} values, first-order logic + Dempster-Shafer theory, e.g., [Saffiotti1990], Description Logic (OWL) augmented with constructs for expressing possibility, e.g., [Rector2008]. $ALCRP(S_2)$\textsuperscript{83} [Harslev1999-b], extensions of action languages with features that represent triggers and inhibitions [Tran2007], etc. Most of these approaches, however, do not include, or do not allow for, the explicit modeling of time.

Given the continuous flux of scientific discoveries, abductive reasoning has also been proposed in the literature [Bada2008, King2004, Rector2008] as a means to deal with the absence, or with the limited reliability, of scientific knowledge. When the information contained in a knowledgebase is insufficient to entail some property of interest, abductive reasoning, a form of guessing, allows us to add potential premises, e.g. missing steps in a biological process or

\textsuperscript{83} an instantiation of $ALCRP(D)$ with $S_2$, informally, the set of all regions, i.e., non-empty, regular, closed, planar subsets of $\mathbb{R}^2$. 

missing subsumption relationships, that will result in the entailment of the property under consideration. For example, from the known assertions (1) coenzyme binding is_a molecular function that results_in_binding_of (some) coenzyme and (2) FAD binding is_a molecular function that results_in_binding_of (some) FAD, abductive reasoning can help infer that FAD binding is_a coenzyme binding, which cannot be inferred through deductive reasoning [Bada2008]. Alas, inferences made through abductive methods cannot be guaranteed to hold, but, nonetheless, they can inform knowledge engineers about potentially missing relationships that should be added to the knowledgebase.

Finally, we note that, although the fact that uncertainty significantly influences our decisions has long been acknowledged, e.g., [Harrower2003], it is not yet clear what is the best/optimal way to represent, and present uncertain knowledge.

2.2.8 Qualitative information

Qualitative knowledge encompasses a large variety of types of information, usually, but not always, considered to be devoid of a quantitative, numerical dimension, and sometimes related to some form of uncertainty or deliberate vagueness, e.g., two chemical compounds are known to typically participate in a reaction, under some specified conditions, the maximum normal magnification of an optical microscope is about 1,000 times the size of the specimen being observed, but at greater magnification the image may become blurry, etc..

When some facts cannot be ascertained with absolute certitude, we need to be able to express qualitative information about relationships, e.g., degree of confidence, and although attaching such information to a relation in the form of annotations, or metadata, might be, at times sufficient, at other times, we might benefit from explicitly incorporating this type of knowledge in the definition of the relation of interest.

Qualitative observations such as roughly the same size as and negligible in size with respect to have been modeled in the literature, e.g., [Bittner2007], based on mereotopological primitive relations, e.g., part_of and sphere, and ad-hoc relations such as exactly-the-same-size-as, but these definitions are context-dependent and vague and their canonical models need to use precise numerical parameters for fixing the interpretations.
We also note here that topological information, e.g., overlap, connectedness, etc, is inherently qualitative and that research in spatial qualitative representation and reasoning aims, in essence, to provide calculi that allow for the automatic inferencing of knowledge about spatial entities, e.g., answering questions related to orientation, shape, relative size, and relative distance, without resorting to any quantitative techniques.

We also note that qualitative reasoning, although akin to the way humans often think, cannot replace quantitative reasoning in many, if not most, practical applications. For example, applications such as computer assisted surgery need to have access to numerical (quantitative) distance estimations as the effects of moving from one (qualitative) scale to another are not negligible, what is defined as very close in one context under one scale can become relatively distant under another scale and vice-versa.

2.2.9 Patterns

Many times we need to express knowledge in the form of patterns, or motifs, such as the DNA and miRNA motifs, patterns of protein and gene occurrences in scientific articles, chemical compounds or paths in graphs describing complex biochemical reactions.

The expressive power of first order logic over finite word structures is sufficient for representing many simple restrictions, e.g., the amino acid cystein appears before amino acid leucins in the expression of a protein, or any other starfree regular expression [McNaughton1971], but describing regular patterns of arbitrary complexity requires the expressive power of Monadic Second Order Logic (MSOL)\(^\text{84}\), as it is known that regular languages correspond to languages definable by sentences in the MSO\(^{\text{85}}\) [Büchi1960]. Among many useful features, in MSOL, one can directly refer to positions and subsets of positions in a string, a very helpful property when describing complex chemical structures.

The advantage of MSOL over regular expressions or their equivalent representations, the finite state automata, consist in the remarkably succinct representations afforded by the availability of

\(^{\text{84}}\) First Order Logic with added variables for subsets of the domain.

\(^{\text{85}}\) The Buchi-Elgot-Trakhtenbrot theorem
logical constructs. See [Jensen1994] for more elaborate discussion of the translation of regular languages to MSO formulas. Moreover, the same theorem proover, inference engine, can be used for checking patterns as for reasoning with the other parts of a knowledge representation, instead of using dedicated, external implementations of finite state automata, for example.

We note, however, that some patterns expressing relatively “simple” conditions can require the full expressive power of (monadic) second-order logic, higher-order logics, and the introduction of additional predicates, such arithmetic predicates for addition and multiplication. For example, the pattern (CC)*, which describes the fact that essential fatty acids (for humans) have an even number of carbon atoms, defines a regular language that, although “simple”, in a superficial, intuitive sense, it is not first-order definable, because it requires the ability to check parity. \(A^n T^n \cup C^n G^n\), which expresses the condition that in a normal DNA strand adenine (A) bonds only to thiamine (T), and cytosine (C) bonds only to guanine (G), defines a language that is not even regular, although it is definable in first-order logic augmented with arithmetic predicates for addition and equality.

Working with general context free languages would require us to go beyond monadic second order logic, but many patterns of practical interest are definable in decidable and tractable fragments of first order logic.

The connection between formal languages and logics is well studied and representational results for many languages of practical interest can be found in the literature, e.g., [Lautemann1995, Thomas2006].

### 2.2.10 Discussion

Explicit descriptions of domain knowledge can facilitate both the human and automatic interpretation and integration of information by providing means for meaning clarification and negotiation. They also provide the building blocks for frameworks that support the resolution a vast array of tasks: e.g., fact-finding, deep query answering 86, data exchange, information

integration, processing natural language texts, generating scientific hypotheses, composing web services and solving complex problems, e.g., [Staab2003, Stevens2007, Rubin2007].

As we argued in the beginning of this chapter, the demands of the domain to be modeled and the task(s) to be solved determine together the characteristics of a knowledge representation, which in turn determines the computational properties of the resulting knowledge-based system.

In this work we distinguish between knowledge that needs to be described, e.g., structure and substance, things, shape, size, number of things, functions and behaviours of things, parts of things, the order of things, processes and transformations, and inferred, e.g. classifications, and degree of similarity between things. In this chapter we have endeavored to identify and analyze, from a representational perspective, the main categories of “describable” (i.e., representable) knowledge. We deal with inferred knowledge, such as similarity and relatedness between ontological terms and entities annotated with ontological concepts, in Chapters 4 and 5 and 6.

**Research contributions.** In this section we have discussed representational requirements for the following categories of knowledge: spatial and dynamic knowledge, activities and causality, complex associations, transformations and mutations, hypothetical, uncertain and qualitative knowledge, as well as the representations of patterns. This led us to propose the following list of (informal) guidelines for choosing appropriate representational formalisms.

**Limitations:** We acknowledge that this list is not exhaustive and that its content may vary slightly, in the topics considered and granularity, depending upon whether we are restricting our attention to a particular domain. For an example of a slightly different categorization, see [LoBue2011], which identifies twenty common-sense knowledge categories identified as important for the task of semantic and logical inferences for recognizing textual entailment. There is significant overlap between their categories and ours, however, many of their categories are specifically pertinent to the task of entity recognition in natural language descriptions.

87 The formal specifications contained in a representation should allow for the automatic classification of concepts and roles into (poly)-hierarchies.
Guidelines for choosing the appropriate representational formalism

The main question that a knowledge engineer needs to answer before embarking on constructing a practical ontology is, of course, what purpose will the formal representation serve. In answering this question, one must determine what is more important: maximal expressiveness, i.e., representing all the details as comprehensively as possible, or having reliable computational support, i.e., tractable reasoning and query answering, since the two desiderata are conflicting.

The reason for this dichotomy is the direct relationship between the expressiveness of a formal ontology, and its underlying representational language, and the computational resources, i.e., the time and memory, required to perform inferencing and query answering using that ontology. We recall here that there is always a tradeoff when employing a reasoner between how precise the semantics is and how much information the reasoner has to consider, e.g., adding cardinality restrictions to properties significantly increases the amount of information that a reasoner needs to handle. For instance, reasoning with ontologies represented in DL-Lite requires polynomial time [Calvanese2006], while reasoning with ontologies represented in DLR, OWL1 and OWL Full is EXPTIME [Calvanese1998], NEXPTIME [CuencaGrau2006], and undecidable [McGuiness2004], respectively. In terms of expressivity, for example, the Gene Ontology and the Protein-Protein Interaction Ontology are in DL-Lite, BioPax is in DL-Lite, while the Foundational Model of Anatomy and the Cell Cycle ontology are in OWL1 [Keet2012]. Bittner and Donelly argue in favour of considering computational (practical) ontologies as being composed of two complementary parts, a DL-based component that enables tractable reasoning and a first order component which serves as ontology meta-data and specifies the properties of relations in the first component that are inexpressible in DL [Bittner2005].

There is a plethora of relationships to select from when modeling a knowledge domain and choosing between them can be challenging, especially if one needs to come up with relations to supplement/complement the ones already in use, all the while keeping in mind that too conservative, or parsimonious of an approach can lead to underspecified representations and impaired reasoning capabilities.

Apart from strictly computation-related questions, there are also many other practical and philosophical questions that knowledge engineers need to ask themselves, e.g., “Is there a need for defined classes or would just universals be enough?”,” “Should universals be allowed to be
subclasses of defined classes?”, etc. In short, deciding on the most suitable formalism is as much an art as it is a science, and without attempting to give an automated decision procedure for selecting the most appropriate representational formalism, we put forward a few informal guidelines.

If the knowledge representation is only meant for human processing, i.e., no automated computation is required, then the choice of representational language and primitives is unrestricted, i.e., one can use the language that best meets the expressive requirements of the information to be modeled.

If the feasibility of automated reasoning and querying is important, we advocate in favour of choosing a decidable representational language, which will exclude, for example full first order logic and OWL Full. One can further subdivide this category according to various restrictions on the available computational memory and time. For example, if both time and memory are at a premium, one can choose between variants of DL-Lite, Horn-SHIQ, etc., which have PTIME complexity. If neither time nor memory is likely to be in short supply, one can consider high-complexity, but decidable, variants of description logic, e.g., DLR, OWL1.1, and decidable fragments of first and second order logics. There is a vast grey area in between these two computationally-dependent cases, where the subtle interplay between expressive power and computational costs comes to the forefront. One can choose between an abundance of computationally enabled representational formalisms and there is simply no mechanistic way to dictate which formalism should be chosen. For example, if we need to model temporal information choosing between, e.g., temporal logic, temporal description logic and hybrid temporal approaches does not solely depend on their computational properties, but also on the type of statements they are allowing us to represent.

Because we did not restrict our discussion of representational requirements to only description logics and knowledge representation languages that are based on them, i.e., OWL in its many flavours, and instead discussed these demands in the larger context of general logic-based representational languages, we hope that the discussion we provided in this chapter of the various approaches to representing knowledge can serve as a guide for knowledge engineers in choosing the most suitable formalism when confronted with a scenario that lies somewhere in this intermediate area.
Challenges. Since each knowledge representation strategy offers a different set of modeling primitives, knowledge representation practitioners should keep in mind that a commitment to a modeling strategy translates into confinement to the limitations induced by that approach.

Debating the superiority of one approach over the other might look, at a first glance, as a strictly philosophical debate, but the outcome, has, in fact, serious implications for practical applications, as in the following example.

We briefly recall here the characteristics of several formal representations of geometrical knowledge. Coordinate geometry considers points to be the only primitive spatial entity, with lines, surfaces and geometrical bodies defined as collections of points, their properties and relationships defined using the coordinates of the points they are made up of. An alternative, qualitative perspective (especially attractive if no metric information is considered) takes regions as primitive entities, with points being constructed, if at all, as derived entities, based on regions. It has been conjectured that point-based axiomatizations of space induce richer ontologies than needed for practical purposes, as very useful applications that manipulate spatial data, such as geographic information systems and autonomous robots, use only planar and tridimensional polygonal regions\(^8\). Subscribers to this opinion, i.e., that qualitative knowledge of the space can be (more) naturally modeled in logics whose variables range over regions\(^8\) have developed an extensive collection of such logics\(^9\), but it has been shown that using regions rather than points as primitive ontological entities does not actually lead to more parsimonious spatial ontologies [Pratt1997]. Moreover, region-based approaches, i.e., mereotopologies, are not able to represent the full geometry [Tsai2005], an important thing to consider if the practical application calls for quantitative not just qualitative reasoning capabilities.

---

\(^8\) The boundary of a region of space is represented as a finite set of line segments and/or straight line (some regions are allowed to be unbounded). Usually, the higher the required accuracy, the higher the number of sides (segments) of the polygons.

\(^8\) subsets of \(\mathbb{R}^n\)

\(^9\) In Euclidian logics, variables range over subsets of \(\mathbb{R}^n\) and the non-logical primitives are classical geometrical properties and relations. Euclidian logics can have non-logical primitives that represent non-topological notions
Caveats. However tempting it might be to think that a language with more non-logical primitives is more expressive then another, this is not the case\(^\text{91}\).

It should also be noted that a formal representation is not necessarily correct, i.e., although specified in a precise language, it might not make sense as a depiction of reality. In fact, logic-based representation, in particular the representations based on description logic, have been criticized [Smith2008] as being artificial and divorced from the reality they were supposed to represent, in particular because of their perceived lack of attention to the treatment of instances belonging to different ontological categories, such as attribute types, e.g., blood pressure, heart beat, and event types, e.g., the beating of the heart, the heart pumping blood.

Very few tools exist to help knowledge engineers choose between formalisms, and those that do exist are very specialized, e.g. OntoParts, a tool to select mereotopological relations in OWL ontologies [Keet2012], so the analysis of representational requirements and the assessment of alternatives is still, largely an un-automated effort. Moreover, what one chooses to state as a fact in a representation, someone else can envision as best left to be inferred as the result of a reasoning method in an alternative representation. Choosing between the two approaches is, many times, far from straightforward and we acknowledge that selecting a representational framework from among the many competing formalisms is influenced not only by the demands of the intended application, but, potentially, by personal preferences as well.

We also note that there are several factors that hamper current efforts to build interoperable knowledge representations, regardless of domain.

Lack of community consensus. Unfortunately, naming conventions, object identifiers, and definitions of various entities may differ significantly between knowledge producers and publishers, e.g., the concept of *gene* is defined by the Human Genome Database as “a DNA fragment that can be transcribed and translated into a protein”, but is a “DNA region of biological interest with a name and that carries a genetic trait or phenotype” for Genebank [Schulze-Kremer1998].

---

\(^{91}\) E.g., the mereotopological languages L\((c, \leq)\) and L\((C)\) do not have the same expressive power over the domain of regular polygons in the open plane, R\(2\). L\((C)\) is, in fact, strictly more expressive [Pratt2000].
Another prominent example of lack of consensus in defining important domain concepts is provided by processes and sequences of events. As of yet, there are no clear guidelines for identifying what are the delimiters, the beginning and the end, of processes. This situation severely hampers any effort to coherently describe the sequence of events that constitute a process. In biology for example, some research groups conflate biological processes with their regulation, while others do not, complicating the task of curators, who may not be able to consistently choose whether to annotate a gene product to both a process and its regulation, just to the process or to only the regulation of the process. For example, ligands binding to receptors have been annotated to both signaling pathways and to the regulation of those signaling pathways by different groups with apparent different views on whether the binding action triggers a signaling pathway, it is its first step or it is rather a rate limiting, regulatory, step of the pathway. To complicate matters even more, a signaling pathway is interpreted, in some cases, as a sequence of regulatory events. Because of this current state of affairs, although pervasive throughout the biomedical domain, there is no widely adopted framework for describing biological processes.

In every domain there are also plenty of examples of imprecise, or ambiguous, use of terminology, the meaning of which becomes clear only when the context of the assertion is taken into account, e.g., in biomedical literature, Interleukin-6 is used to refer to a protein, the corresponding gene or a messenger RNA. Formalizing knowledge when the meaning of the terminology to be used is not entirely clear is a difficult, if not impossible, endeavor.

*Knowledge is in flux.* Ultimately, no matter what representational formalism is chosen, it would have to deal with the fact that knowledge is in constant flux. There are no easy or clear solutions for how to deal with this issue, but examples abound of bad modeling decisions taken with the (misguided) aim of building all-encompassing representations, e.g., many current medical vocabularies contain some curious catch-all constructs, such as *neoplasm of uncertain behaviour of heart*, which happens to be classified as a type of *structural disorder of heart* (SNOMED-CT).
3 From Information Models to Knowledge Models – Insights into Practical Ontology Design: towards a Core Business Data Interchange Ontology

Each of the architectural components of an enterprise has been investigated by different research groups operating in relative isolation, but it is becoming increasingly clear that enterprises would achieve maximum operating efficiency through the cooperative interaction between the models that describe the functionality of the enterprise from different vantage points and at different levels of abstraction.

The workflow logic of an enterprise is often captured using the Business Process Execution Language (BPEL), an XML-based orchestration language that allows for the specification of executable processes via message exchanges with other systems, according to sequences specified by the orchestration designers. For example, an orchestration could specify that the internal processing of a purchase order involves checking the stock availability, getting a credit card authorization and scheduling the delivery of the goods. Orchestrations, however, cannot fully represent scenarios in which agents, e.g., businesses, software programs, interact with outside agents. Such scenarios form the subject of choreography definitions.

In essence, a choreography defines a protocol for peer-to-peer interactions, where a protocol is represented by a legal sequence of messages that agents exchange in order to guarantee interoperability. For example, if an agent sends to another agent a purchase order message, the supplying agent needs to know what is the expected/appropriate response to this message in order to choose from a set of alternative actions, such as (1) acknowledge the order and send information about the availability, price and projected delivery of the goods or (2) immediately ship the goods accompanied by an invoice. A choreography specification is not usually directly executable, as it allows for multiple different realizations.

In a nutshell, choreographies specify interaction scenarios that can then be realized by composing orchestrations for each agent involved.
**How can ontologies help?** It has been documented in the literature that the semantic interoperability between various business information systems is frequently hindered by the absence of a common reference to the meaning of terms being exchanged, a *canonical business language* of sorts.

Existing information models syntactic standards, such as those describing patient records (HL7) or business processes and exchanges such as OAGIS, BPEL, WS-CDL, provide templates that promote interoperability, but they lack an underlying mathematical formalism that could be used to identify inconsistencies and contradictions and to test, or formally prove properties of the process and workflow specifications.

Formal ontologies can help by providing a reliable knowledge representation mechanism for modeling business processes beyond the capabilities of purely syntactic standards and can also contribute to their successful integration by fostering a shared understanding of the business domain. From a practical point of view, the availability of axiomatized application ontologies, and, implicitly, of logical reasoning, enables us to define machine-understandable semantic-preserving mappings between the terminologies used by different applications. This would normally require us to specify all the pair-wise mappings between the application ontologies, however, if a formal ontology exists that can be used as Interlingua, this requirement is replaced by that of writing translators between each application ontology and the shared ontology [Ciocoiu2001]. Furthermore, the model theoretic structures of the formal ontologies can be taken advantage of by specifying the definitions in the translational modules using model theoretic invariants [Grüninger2005]. In this work we take the Process Specification Language (PSL) [PSL], an industry standard\(^2\), to serve as an Interlingua between the various enterprise application ontologies.

**Goal of this research.** Our goal is to create an ontology that supports the integration and interoperability\(^3\) of business processes, while also analyzing the representational requirements

---

\(^2\) ISO 18629 (2004)

\(^3\) We abstract applications as first-order inference engines that exchange first-order sentences and assume that an ontology supports the interoperability between them, a.k.a. the Interoperability Hypothesis [Grüninger2004].
for developing such an ontology. More specifically, we endeavor to provide a formalism for expressing the semantics of the business messages at the core of business communications in a way that captures their meaning and purpose beyond the mechanics of the business interactions (which form the subject of the orchestration and choreography languages).

The formal ontology we are proposing could help not only human analysts trying to detangle the intricacies of business data exchanges, but we also anticipate that our work can be used to support several important tasks, such as

- the creation of enactable descriptions of business processes, i.e. descriptions that can be mapped automatically to executable specifications, e.g. via procedural attachments such as effectors and events.
- automatic composition of business processes
- monitoring, problem resolution, exception handling and verification of integration scenarios such as the purchase order scenario depicted in Figure 3.1.

\[ Figure \ 3.1 \ Production \ order \ scenario \]

In the remainder of this chapter we review previous work in the area, give a brief description of the ontology engineering strategy we employed as well as short overviews of the Process Specification Language and the OAGIS standard, followed by the presentation of the Core Business Data Interchange Ontology (CBDIO). We then examine CBDIO’s relationship with

---

\[94 \] Copyright (c) Open Applications Group. All Rights Reserved. Permission to reproduce granted under the licence agreement available at [http://www.oagi.org/oagis/9.0/](http://www.oagi.org/oagis/9.0/)
other business ontologies, and discuss representational requirements, design challenges and lessons learned.

3.1 Related work

Several research efforts have been directed at defining semantic frameworks for acquiring, organizing, sharing and using the knowledge intrinsic in business tasks and data exchanges. Several ontologies are aimed at modeling enterprises, in various degrees of detail [Grüninger2000, Ohgren05] others focus on business processes [Atefi1997, Haller2006, Pedrinaci2008, Fiorentini2009].

Tran and Tsuji proposed an ontology based task template language, OWL-T, for capturing business demands at a higher level of abstraction with the aim of enabling the automatic composition of task templates into executable processes [Tran2007]. Tasks are described in OWL-T terms of five main properties, relatesToTaskOnto, refersToDomainOnto, realizedBy, hasNonFunctionalProperty and hasFunctionalProperty, and are divided into the following task types: Atomic, Composite, Simple and Complex. hasFunctionalProperty, in particular, links a task with a description of its input, output, precondition, postcondition and effect.

OWL-P [Desai2006] is another recent agent-based business process modeling approach which aims at capturing, in OWL format, the interactions between agents, e.g., ordering, delivering payment and organizing shipping. These interactions are modeled as protocols and policies, where protocols only capture the essence of the interactions, the local details being left to be provided by each agent’s policy.95

Our ontology engineering efforts have, as a starting point, the Open Applications Group Integration Specification, OAGIS, a business communication standard (widely supported by major industry players such as IBM, Oracle and Microsoft), which provides a common horizontal message architecture to assist businesses in achieving interoperability. We give an

95 In essence, each policy agent’s provides the business logic details needed for generating and processing the messages exchanged during the interaction modeled by the protocol.
overview of the OAGIS standard in section 3.4, but we recall here that the standard allows messages to be sequenced together to form scenarios that detail the succession of information exchanges needed to perform specific business tasks. There are internal messages, such as Engineering Work Document, communicating the engineering design for a given item between the CAD and ERP applications, and external messages such as those between corporations and banks, which deliver credit and payment information. The content of the messages has a complex structure and contains details such as the type of data being delivered, the sender, the receiver and the action to be performed on the data. This information content forms the basis for the concepts included in our ontology.

A previous attempt at developing an ontology for the business domain based on the OAGIS standard was documented by Anicic and co-workers [Anicic2006] as part of their investigations into the applicability of Semantic Web technologies to industry-wide interoperability projects. The approach mentioned by Anicic et. al. is an automatic translation into OWL of the core XSD schemas from the OAGIS standard using the freely available tool xsd2OWL. The authors indicate that the translation was not successful for some core components but no reason is given for the failure, nor is any part of the ontology generated in their approach described formally. Fiorentini and co-workers [Fiorentini2009] also mention the intent to create a reference ontology to be used as a basis for their framework for ontology-based harmonization of information standards, including OAGIS, but to our knowledge, the results of their research efforts are not yet available.

We hope that given (1) our focus on data exchanges, (2) choice of OAGIS, a standard with a large industry support, as a source for developing the ontology terminology and (3) ontology development as an extension of an existing ISO standard, PSL, our research effort will help bridge the gaps between the previous formalization efforts.

3.2 Ontology engineering strategy

There are several ontology engineering approaches documented in the literature, a good survey of which can be found in [Ohgren2004], and we have chosen to follow the principles expressed
in the frameworks proposed by Uschold and King, [Uschold1995], Grüninger and Fox, [Grüninger1995] and further adapted by Ohgren and Sandkuhl [Ohgren2005].

The ontology engineering approach we adopt here is organized according to the following phases:

- **Requirements Definition.** Use cases or motivational scenarios are used to produce a specification of the intended models of the ontology. In this phase a set of informal ontology competency questions, which can be thought of as representing use scenarios formulated as questions, is produced to guide and inform the design of the ontology. As well as serving as justifications for the entities included in the ontology, the competency questions also provide criteria to evaluate the reasoning capabilities of the model and its correctness.

- **Ontology Design.** The outcome of this phase is the terminology to represent the concepts and relations in the domain of interest and axioms, i.e., first order sentences, to characterize the class of models that capture the requirements identified in the previous phase. The terminology alone is not sufficient for providing a formal model. Axioms are needed in order to provide formal definitions for the terms in the ontology and to constrain the possible interpretation of these terms. The informal competency questions are also translated in this phase into first order sentences.

- **Refinement.** A feedback loop between the requirements definition and the design phases in which the informal competency questions acquire a formal expression and the ontology’s terminology and axioms are refined.

Once the ontology design is stable a verification phase checks if the intended models specified by the requirements are equivalent to the actual models of the ontology. In essence, we need to check if the formalized competency questions are entailed by and consistent with the ontology and this step can be accomplished manually or with the help of automatic theorem provers.

The entailment of the competency questions can be formalized as:

\[ T_{ontology} \models CQ \]
where CQ designates the competency questions, $T_{\text{ontology}}$ represents the axioms of the ontology, in our case the theory formalizing the business message exchanges given by CBDIO together with the PSL theories.

In order to determine that the competency questions are consistent with the ontology, we need to verify that:

$$T_{\text{ontology}} 
\not\models CQ$$

**Requirements Definition.** In order to obtain a list of requirements needed to represent business data exchanges several avenues were pursued. The first was to explore the documentation developed for the existing (syntactic) business standards such as OAGIS, RosettaNet\textsuperscript{96} and ebXML\textsuperscript{97}. Another avenue for gathering data interchange requirements was to survey the literature on business process management and enterprise integration.

The following are the high level representational requirements that we have identified during our analysis:

- links between messages and their senders
- descriptions of data (business objects) being transferred as the payload of a message. This includes product specifications, accounting ledgers, customer information, employee timesheets, shipping information, etc.
- actions whose execution is requested, canceled or acknowledged, in a message
- actions preconditions and effects

---

\textsuperscript{96} http://www.rosettanet.org/

\textsuperscript{97} http://www.ebxml.org/
constraints, e.g., partial (temporal) ordering of operations/messages in business scenarios, certain operations must be followed by an acknowledgement, a message has one, and only one sender, etc.,

- roles played by the agents in different business scenarios

- agents’ goals

- grouping (classification) of agents, activities, resources, etc.

The types of knowledge we need to be able to represent overlap with the following among the categories of knowledge studied in Chapter 2: dynamic knowledge, complex associations, and activities. We decided not to consider, at this time, representing spatial knowledge, which would be needed e.g., for analyzing and optimizing shipping routes, hypothetical knowledge, e.g., hypotheses that a decision making agent might want to generate and test, or patterns and qualitative knowledge.

Many of these requirements are well addressed by PSL, which is why we decide to build our ontology as an extension of PSL, a first-order language. Since we have identified the need to represent complex (temporal) relationships, e.g., business objects are processed in an activity at a particular time, Description Logic based representational languages such OWL are not suitable.

**Competency Questions.** The competency of an ontology essentially defines the tasks that the given representation can be used to accomplish and the competency questions can be thought of as formalized requirements. As pointed out in the literature, [Cutting-Decelle2003, Fox1995], the competency questions do not generate ontological commitments, rather, they are used to evaluate the expressiveness of the ontology being designed and the ontological commitments that have been made during the engineering process. In our case, they can also be used to verify the compliance of the series of exchanged messages to some prescribed behaviours.

The following are some of the questions we have identified as determining the competency of the Core Business Data Interchange Ontology:

- Has the Receiver of a request, e.g. PurchaseOrder, PurchaseOrderChange, Invoice, acknowledged receiving the request?
- Has the receiver of a Request, e.g. Move Inventory, performed the necessary action, e.g. Process Inventory Move?
- Is the refusal to perform an action, e.g. PurchaseOrderRejection, justified and fully explained, by including in the rejection message the necessary explanation?
- What role(s) does an actor or a group of actors play in the organization?
- Are there other actors with a similar role/authorization level?
- What sub-activities does a specific protocol (complex activity) entail?
- What protocols is a certain activity part of?
- What is a valid execution order of the activities in a particular process?
- Which requests have been completed successfully and which ones have not?
- Which resources were involved in which activities?
- What actor was responsible for a certain activity?

Once the competency questions of interest are identified we can proceed to defining the terms and relations of the new ontology using the formal language of choice, first order logic in this case.

3.3 A short overview of the Process Specification Language

We have chosen to base our ontology design on the Process Specification Language (PSL) [Grüninger03], a formal language, based on situation calculus [Reiter1991], and developed at the National Institute for Standards and Technology as a foundation for process modeling.

PSL’s main purpose is to enable the semantic interoperability of manufacturing processes, from the engineering side to the business side, e.g. scheduling, workflow and project management, by providing a standard language for process specification. PSL’s applications, however, also include enterprise modeling as well as the design, analysis and implementation of business processes.

Our choice of extending PSL is motivated by the fact that PSL is an extensible International Standard, ISO 18629, which was designed to support interoperability and can consistently include ontologies for time and duration. In addition, basing our ontology on a First Order Logic (FOL) language provides important benefits such as a rich expressivity and a well understood model theoretic semantics. The reasoning tasks can thus be characterized in terms of deduction
and consistency, which enables us to take advantage of off-the-shelf reasoning systems: FOL theorem provers, model-checkers, DL reasoners, etc. While it is true that that FOL is only semi-decidable, many tasks of practical importance often prove to be in fact tractable.

Our approach to developing an ontology as an extension to PSL is not singular, as several other extensions exist for various domains, such as construction [Cutting-Ducelle2000], manufacturing [Cutting-Ducelle2003, Deshayes2005] and web services [Deshayes2005, Grüninger2006, Grüninger2004, Heller2006].

We briefly review here the relevant concepts. At the heart of PSL there is a central theory, PSL-Core, which provides the basic ontological commitment to a domain consisting of activities, activity occurrences, timepoints and objects. All the other modular theories, collectively known as the PSL Ontology, are consistent extensions of PSL-Core and formalize the intuition behind the composition of activities as well as between the occurrences of complex activities and their subactivities. The non-logical part of the lexicon used in PSL consists of constants, function symbols and predicates chosen to represent basic process-related concepts.

The PSL Ontology models both complex activities and their occurrences as elements of the domain of discourse. This representational choice enables us to quantify over both complex activities and their occurrences and to specify formally a broad variety of properties and constraints over them, while remaining in the realm of First Order Logic. The relationship between an activity and its different occurrences is captured by the occurrence_of relation

**PSL-Core** contains (1) four pairwise disjoint primitive classes: *object, activity, activity_occurrence, and timepoint*; (2) two primitive functions: *begin_of, and end_of*; (3) three primitive relations: *participates_in, before and occurrence_of*. The basic ontological commitments modeled by PSL-Core are as follows:

- Reasoning about processes requires four types of entities: *activities, activity occurrences, timepoints, and objects*.
- Some *activities* can occur multiple times, but there may exist *activities* that do not occur at all.
- *Timepoints* are linearly ordered, from the past to the future.
• *Activity occurrences* and *objects* have unique beginning and end *timepoints*

The outer core foundational theories axiomatize new primitive concepts, not found in PSL-Core, and thus provide definitions for the auxiliary notions needed to describe particular processes with various degrees of detail.

The basic intuition behind the composition of activities and occurrences is captured using the *subactivity* relation introduced in the Theory of Subactivities (PSL) and the corresponding composition relation over activity occurrences, *subactivity occurrence*, which is defined in the Theory of Activity Occurrences (PSL). *Complex activities* are composed of *atomic activities*, which are either *primitive*, with no proper subactivities, or concurrent aggregations of primitive (sub)-activities (Theory of Atomic Activities). For a given occurrence of a complex activity *a*, subactivity occurrences are occurrences of subactivities of *a* (Theory of Activity Occurrences).

More specifically, complex activity occurrences correspond to branches of *activity trees*, subtrees of the tree that consists of all possible sequences of activity occurrences, a.k.a. the *legal occurrence tree*. A subactivity occurrence corresponds to a sub-branch of the branch corresponding to the complex activity occurrence. Each complex activity occurrence has a unique root occurrence and relation *root*(sao, *a*) denotes that the subactivity occurrence sao is then root for an activity tree for *a*.

Of course not all sequences of activity occurrences are physically possible. The sequences that are possible are referred to as *legal*. *legal*(ao) specifies that the activity occurrence *ao* is a member of a legal sequence. *precedes*(ao₁, ao₂) specifies that *ao₁* happens earlier than *ao₂* in a legal sequence of occurrences.

Ordering constraints over subactivity occurrences of complex activities are captured by the *min_precedes* and *next_subocc* relations. *min_precedes*(sao₁, sao₂, *a*) has the following meaning: subactivity occurrence sao₁ precedes the subactivity occurrence sao₂ in all occurrences of complex activity *a*, with the understanding that there may be other subactivity occurrences between sao₁ and sao₂. *next_subocc*(sao₁, sao₂, *a*) is used to denote that sao₁ is the next subactivity occurrence after sao₂ in occurrences of the complex activity *a*. 
The extensions to PSL are modules that can be added to the Core when more expressive power is needed. Each extension contains new constants and/or predicates along with axioms that constrain the interpretation of the newly introduced terms.

3.4 A short overview of the Open Applications Group Integration Specification (OAGIS)\textsuperscript{98}

The main goal of the Open Applications Group Integration Specification (OAGIS 9.0) is to help the integration of the enterprise execution applications with the enterprise management applications, and further with the value chain applications.

OAGIS covers all major aspects of businesses, from descriptions of shopping carts and invoice requirements to complex customer resource management, financial services and engineering, all reviewed by committees of experts to ensure compliance with existing industry standards such as EU and North American invoice requirements and the Sarbanes-Oxley Act\textsuperscript{99}, many IT vendors are aligning their software with this standard, e.g., IBM WebSphere uses all the main business actions described by the standard. The current scope of the standard is summarized in Table 3.1 Scope of OAGIS 9.0.

<table>
<thead>
<tr>
<th>eCommerce</th>
<th>e-Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Price List</td>
</tr>
<tr>
<td></td>
<td>RFQ and Quote</td>
</tr>
<tr>
<td></td>
<td>Order Management</td>
</tr>
<tr>
<td></td>
<td>Purchasing</td>
</tr>
<tr>
<td></td>
<td>Invoice</td>
</tr>
<tr>
<td></td>
<td>Payments</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>MES</td>
</tr>
<tr>
<td></td>
<td>Shop Floor</td>
</tr>
<tr>
<td></td>
<td>Plant Data Collection</td>
</tr>
<tr>
<td></td>
<td>Engineering</td>
</tr>
</tbody>
</table>

\textsuperscript{98} Version 9.0

\textsuperscript{99} http://www.soxlaw.com/
The common architecture of the messages exchanged between business, supply chains or software applications is modeled by Business Object Documents, a common horizontal message standard that is independent of communication mechanism and business domain.

Table 3.1 Scope of OAGIS 9.0

<table>
<thead>
<tr>
<th>Logistics</th>
<th>Warehouse Management</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enterprise Asset Management</td>
</tr>
<tr>
<td></td>
<td>Orders</td>
</tr>
<tr>
<td></td>
<td>Shipments</td>
</tr>
<tr>
<td></td>
<td>Routings</td>
</tr>
<tr>
<td>CRM</td>
<td>Opportunities</td>
</tr>
<tr>
<td></td>
<td>Sales Leads</td>
</tr>
<tr>
<td></td>
<td>Customer</td>
</tr>
<tr>
<td></td>
<td>Sales Force Automation</td>
</tr>
<tr>
<td>ERP</td>
<td>Financials</td>
</tr>
<tr>
<td></td>
<td>Human Resources</td>
</tr>
<tr>
<td></td>
<td>Manufacturing</td>
</tr>
<tr>
<td></td>
<td>Credit Management</td>
</tr>
<tr>
<td></td>
<td>Sarbanes/Oxley &amp; Control</td>
</tr>
</tbody>
</table>

Figure 3.1 The architecture of a BOD

All BODs also contain an application area with information such as BOD id, sender id, creation time and digital signatures that can be used by the processing infrastructure for routing, authorization and receipt verification. The Application Area contains information that the

---

100 Copyright (c) Open Applications Group. All Rights Reserved. Permission to reproduce granted under the licence agreement available at http://www.oagi.org/oagis/9.0/
transport infrastructure can use in order to transmit the message, such as the sender of the message, authenticating digital signatures and global identifiers useful for transaction logging, reporting, resending, exception handling and security. OAGIS is meant to be agnostic to the specific type of transportation infrastructure that is used to carry the message so we have decided not to include this type of information in our ontology design at this point.

The Data Area contains the payload, the business data that is being exchanged, such as purchase orders, quotes, shipment information, and the actions to be performed on the data. This information is divided into two distinct areas:

- **Noun** – describes a common business object and is composed of Components, which are basic building blocks shared across all Nouns. There are 77 Nouns in OAGIS 9.0

- **Verb** – describes the action to be applied to the data (Noun). There are 19 Verbs in the standard, Acknowledge, Add\(^D\), Allocate\(^D\), Cancel, Change, Confirm, Create\(^D\), Get, GetList\(^D\), Issue\(^D\), List\(^D\), Load, Post, Process, Receive\(^D\), Respond, Show, Sync, Update, 7 of which, those marked with D, are deprecated.

For instance, a message that carries a purchase order to a business partner will contain information about the data to be acted on, the PurchaseOrder, i.e. the Noun, and the action to be performed upon receipt of the order, its Processing, i.e the Verb. It should be noted that the DataArea can include several PurchaseOrders, but only one occurrence of the action to be performed on the data. Within an enterprise, for example, an accountant performing an audit of the company can request, i.e., Verb = Get, from the Sales department details about some of their purchases, i.e., Noun = PurchaseOrder, with a request GetPurchaseOrder.
A Noun may be combined with various Verbs to describe portions of a business process, i.e. steps or tasks associated with the fulfillment of business functions. For example, if a business application sends a partner application a Get PurchaseOrder, the partner will respond with a ShowPurchaseOrder and later, the initiating application may issue a ChangePurchaseOrder.

A Verb can be associated with additional information, e.g., a Process verb associated to a specific task also indicates that that the execution of that task is acknowledgeable and confirmable.

In addition to the ApplicationArea and DataArea elements, each BOD also contains two mandatory attributes, (1) the releaseID, identifying the release of OAGIS to which the BOD belongs, and (2) the languageCode, indicating the default language of the data being carried, and two optional attributes (3) versionID, identifying the version of the BOD within a specific OAGIS release, and (4) systemEnvironmentCode, which flags the BOD as the result of a test or production level integration.

OAGIS defines a standard list of Verbs, e.g. Process, Cancel, Change, and Nouns, e.g. PurchaseOrder, RequestForQuote, needed in most business applications from supply chains to manufacturing integration scenarios. However, it should be noted that since OAGIS is a cross-industry standard, it focuses on general business functions and rather than attempt to contain all

---

101 Copyright (c) Open Applications Group. All Rights Reserved. Permission to reproduce granted under the licence agreement available at http://www.oagi.org/oagis/9.0/
elements needed by everyone, it provides mechanisms for extending the BODs with content external to OAGIS in order to meet the needs of user applications.

Figure 3.3 Verb and Noun components of the GetPurchaseOrder Business Object

Some of the actions describe in OAGIS are deemed to be requests and require a reply as described in Table 3.2.

<table>
<thead>
<tr>
<th>Request</th>
<th>Reply</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allocate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cancel</td>
<td>Confirm</td>
<td>Request: CancelReservation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response: ConfirmReservation</td>
</tr>
<tr>
<td>Change</td>
<td>Respond</td>
<td>Request: ChangePurchaseOrder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response: ConfirmPurchaseOrder</td>
</tr>
<tr>
<td>Create</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get</td>
<td>Show</td>
<td>Request: GetOrderStatus</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response: ShowOrderStatus</td>
</tr>
<tr>
<td>GetList</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Issue</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>Confirm</td>
<td></td>
</tr>
<tr>
<td>Post</td>
<td>Acknowledge</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Acknowledge</td>
<td>Request: ProcessPurchaseOrder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response: AcknowledgePurchaseOrder</td>
</tr>
<tr>
<td>Receive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sync</td>
<td>Confirm</td>
<td>Request: SyncCustomerAddress</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Response: ConfirmCustomerAddress</td>
</tr>
<tr>
<td>Update</td>
<td>Respond</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>Confirm</td>
<td>A confirmation can follow any request</td>
</tr>
</tbody>
</table>

Table 3.2 OAGIS Verb request-reply pairings

---

102 Copyright (c) Open Applications Group. All Rights Reserved. Permission to reproduce granted under the licence agreement available at http://www.oagi.org/oagis/9.0/
We also note that a confirmation message, i.e. a ConfirmBOD, can occur in response to any other BOD.

We have implemented an XSD parser in Python for automated extraction of concepts and relationships from XSD schemas, but this approach did not yield satisfactory results when applied to OAGIS, as the standard seeks to describe in detail the structure of the messages passed between business applications not to reflect the conceptual structure of the business domain. A thorough manual analysis of the standard and its accompanying documentation was required. In addition we have performed an extensive analysis of other existing business standards such as ebXML and RosettaNet.

### 3.5 Overview of the Core Business Data Interchange Ontology

In this section we introduce the ground terms and axioms of the Core Business Data Interchange Ontology. For details of the specification please refer to Appendix G.

CBDIO’s main purpose is to enable automated reasoning about the semantics underlying business exchanges. CBDIO does not attempt to be an exhaustive representation of all possible business exchanges, as this is an ever evolving domain, but rather an abstract model that is truthful to the semantic aspects underpinning these exchanges.

Our desiderata for CBDIO are: (1) an explicit representation of messages, (2) explicit modeling of the preconditions, outcomes and effects of business actions and (3) compatibility with existing and emerging standards, as well as with other ontologies that model activities. CBDIO’s role as an ontology with a well-defined model theoretic semantics is to support the construction of formal specifications not necessarily to provide an executable specification.

These desiderata are in line with the requirements listed in the Reference Architecture Foundation for Service Oriented Architectures, a cross-industry standardization effort who focuses on facilitating the integration between business and the information technology needed to support it, section 4.3.5 in [OASIS2009].
CBDIO’s terminology and axioms, are defined in first-order logic. Our choice of a logic-based approach to ontology design is motivated by two of its main advantages: (1) it allows the rigorous characterization of the completeness and consistency of the ontology and (2) it supports both the implementation of inference engines for models that use the ontology and the use of off-the-shelf first-order logic reasoners.

The primary categories of concepts we identified are as follows: (1) business actors, the entities that are capable of initiating or performing a business exchange, (2) business objects, the entities involved in business exchanges, and (3) business actions, the activities initiated and performed by business actors on business objects.

- **Actor** is a concept often found in the literature. In our context, an actor can be a human being, a group of humans, a business unit or even a computer system who performs activities in order to achieve a goal. Actors are defined by properties that do not change over time, i.e., they have the rigidity property, they remain actors even if the business context changes. We have identified the following actors: Person, GroupOfPeople, GroupOfEmployees, BusinessUnit, Enterprise, SoftwareApplication.

- **Roles.** Roles are behaviours, duties and prerogatives associated to actors. A role, thus, entails some form of *doing* from the part of the actor, e.g., performing or requesting an action in a specific context. In our model, a role can be, but is not limited to, the function/capacity of a person, group of people, business unit or software system, who is responsible for, or is generating various pieces of data, or requesting/performing various actions: e.g., clients ordering products, employee groups responsible for processing orders and invoices, engineers ordering, reviewing and approving product manufacturing changes and couriers responsible for delivering products and documents.

A particular actor can assume various roles, some concurrently, some at different points in time, e.g., a person is a *client* for his or her bank, and *project manager* for the company he or she works for, and each actor may have more than one role in an organization.

We have endeavored to provide a neutral and general set of concepts that would allow analysts to answer competency questions such as: How many roles does a certain actor perform in an organization? What roles are can performed concurrently be a business actor? The business roles
can be both internal and external, such as unqualified leads or contacts (SalesLeads), and qualified leads (Opportunity).

Although in many circumstances the goals of the agents are also modeled as part of their role descriptions, we have decided that modeling the goals of individual agents would add unnecessary complexity to our model, at this stage, and opted instead to allow the users of our ontology to import goal related concepts from other existing ontologies, such as the TOronto Virtual Enterprise (TOVE) [Fox1995], and the Business Intelligence Model (BIM), [Borgida2012].

Roles are usually organized hierarchically in an organization. For example, the administrative hierarchy at a small manufacturing company could look like this:

- CEO
  - Vice President
  - Director
  - Manager
  - Supervisor
  - Employee

Similar to other ontologies that include roles, e.g. TOVE [Fox1995], we introduce a hierarchical relation, \( \text{subordinate}(\text{role}_1, \text{role}_2) \), between roles, with the meaning that \( \text{role}_1 \) is a subordinate or \( \text{role}_2 \). We want the \text{subordinate} relation to induce a partial ordering on the set of roles so we impose that it is antireflexive, antisymmetric and transitive:

- **Antisymmetry:** \( \forall \text{role}_1, \text{role}_2 \ (\text{role}(\text{role}_1) \land \text{role}(\text{role}_2) \land \text{subordinate}(\text{role}_1, \text{role}_2) ) \rightarrow \neg \text{subordinate}(\text{role}_1, \text{role}_1) \)

- **Antisymmetry:** \( \forall \text{role}_1, \text{role}_2 \ (\text{role}(\text{role}_1) \land \text{role}(\text{role}_2) \land \text{subordinate}(\text{role}_1, \text{role}_2) ) \rightarrow \neg \text{subordinate}(\text{role}_1, \text{role}_1) \)

- **Transitivity:** \( \forall \text{role}_1, \text{role}_2, \text{role}_3 \ (\text{role}(\text{role}_1) \land \text{role}(\text{role}_2) \land \text{role}(\text{role}_3) \land \text{subordinate}(\text{role}_1, \text{role}_1) \land \text{subordinate}(\text{role}_1, \text{role}_3) ) \rightarrow \text{subordinate}(\text{role}_1, \text{role}_1) \)

Besides being related via the \text{subordinate} relationship, roles are also related via specialization/generalization, which induces a taxonomy of roles, or rather several, possibly intertwining, role taxonomies. For example, in a large IT company, a \textit{team leader} is a \textit{senior software developer}, which is turn a \textit{senior staff member}, and, further, a kind of \textit{employee}. In the
same time, however, a team leader is also a kind of supervisor, which is in turn a type of manager. We use $\text{isA}(\text{role}_1, \text{role}_2)$ to denote that $\text{role}_1$ is a specialization of $\text{role}_2$. As for the subordinate relation, we impose that $\text{isA}$ is antireflexive, antisymmetric and transitive:

- Antisymmetry: $\forall \text{role}_1 \, \text{role}(\text{role}_1) \implies \neg \text{isA}(\text{role}_1, \text{role}_1)$

- Antisymmetry: $\forall \text{role}_1, \text{role}_2 \, (\text{role}(\text{role}_1) \land \text{role}(\text{role}_2) \land \text{isA}(\text{role}_1, \text{role}_1)) \implies \neg \text{isA}(\text{role}_1, \text{role}_1)$

- Transitivity: $\forall \text{role}_1, \text{role}_2, \text{role}_3 \, (\text{role}(\text{role}_1) \land \text{role}(\text{role}_2) \land \text{role}(\text{role}_3) \land \text{isA}(\text{role}_1, \text{role}_1)$

$\land \text{isA}(\text{role}_1, \text{role}_3)) \implies \text{isA}(\text{role}_1, \text{role}_1)$

Actors and roles are related via $\text{has\_role}(\text{actor}, \text{role})$. This relationship will be refined in future extensions of the ontology to account for the fact that actors can have different roles in different contexts, at various points in time.

- **Actions.** The operations specified in a message conforming to the OAGIS standard can be one of the following types: Action, Response and Request. Each of these actions are further refined into more concrete types of actions, e.g., the possible actions a sender can specify are: Allocate, Add, Create, Cancel, Change, Get, Issue, Load, Post, Process, Sync, and Update. All these actions inherit criteria from their parent class Action, which they also extend with additional attributes: e.g. a Change also has a response code. The taxonomy of operations is captured by constraints of the form: $\forall a \, \text{SubTypeAction}(a) \rightarrow \text{SuperTypeAction}(a)$, e.g., $\forall a \, \text{Load}(a) \rightarrow \text{Action}(a)$.

Some actions have special properties, e.g., an acknowledgement action always occurs as the last action in a transaction. For instance, Acknowledge Purchase Order ends transactions initiated by a Process, Cancel, or Change Purchase Order action. Constraints capturing these types of properties need to be put in place when translating business scenarios into logical formulas.

Business actions can be composed to describe complex business scenarios (processes).

- **Business Objects.** The operations specified in a message will be performed on the object exchanged in that message or as a consequence of the data contained in that object. The types of objects that can be exchanged between two business applications cover a very extensive range,
from ledger and catalog entries to online orders, schedules, personnel information and carrier routes.

The objects in the ontology are linked together by relationships such as: is_a, e.g., a journal is a ledger, which is in turn financial document, part_of/has_part, e.g., journal entries are part of journals, and has_attribute, e.g. an e-mail has an attribute format specifying the type of encoding used. We use the PSL relation participates_in(object, activity occurrence, time point) to specify that an object participates in an activity occurrence at a particular time point.

**Formal specifications**

We formally specify the information flow in data interchanges by extending PSL, which provides predicates and axioms for the representation of core process modeling concepts, such as activities, activity-occurrences, and fluents. Our ontology is a non-definitional extension of PSL, in the sense that there are notions in CBDIO that cannot be defined solely using concepts in PSL-Core.

We attempt to achieve a seamless blending of business actions and their requisite exchanges in one framework by modeling processes and the sending and receiving of messages as activities, and messages themselves as objects and message contents as fluents. This approach allows us to formally describe data flows and enables the tracking of the history of business exchanges, providing means of generating explanations in case of failures and tracking the performance. Formal descriptions of the dynamic behaviour of business systems also facilitate forecasting related to achieving business goals.

**Activities.** In the spirit of PSL, we associate business actions with (PSL) activities, and make both actions and their associated activities elements of the domain of the ontology. Activities and activity occurrences are both first class citizens in the domain of the ontology, in the sense that activities are not considered to be classes and activity occurrences are not considered to be instances of activities, in the first order logic sense. We can, however, specify classes of activities: for example, create(order) and updateCustomerAddress(customer) are the classes of activities that generate an order, and update the profile of a customer, respectively.

\[ \forall ba \text{ business_action}(ba) \rightarrow \exists a \text{ business_activity}(ba, a) \]
\( \forall ba, a \ business\_activity(ba, a) \rightarrow activity(a) \)

An occurrence of a business action is thus an occurrence of the activity that corresponds to that action:

\[ \forall ba, a \ business\_action\_occurrence(ba, bao) \wedge occurrence\_of(bao, a) \rightarrow \]

\[ business\_activity(ba, a) \]

We model the atomic actions involved in a business process as occurrences of atomic activities. Examples of atomic activities include activities that modify (domain) facts, such as modifications of the inventory information stored in a database or engineering specifications, the creation of a purchase order or the updating of a customer profile.

These activities are modeled using input and output information and have an impact on domain specific fluents, e.g., purchase orders and customer profiles. The occurrence of these activities depends on information that was obtained by previously processing a message.

As in any formalism that aims to describe dynamic exchanges and processes, we also need to be able to compose simple activities into more complex ones. The converse, having access to a decomposition of complex processes into simple ones, is also necessary for a variety of tasks, such as the analysis of bottlenecks, deadlocks, or of the failure to achieve higher level business goals. We note that complex processes correspond to scenarios in the OAGIS parlance. They are accomplished by atomic processes whose coordination are modeled with the help of messages.

The information flow between the components of a complex process can then be described with the help of ordering, occurrence and state constraints. The ordering constraints specify the sequencing properties of the activities comprising a process while the occurrence constraints allow the specification of nondeterministic activities within business processes. The state constraints permit us to specify which business processes (actions) are initiated by changes in the state of the system.

In this framework, there are two types of composition: (1) between activities, and (2) between activity occurrences. The \( subactivity(activity_1, activity_2) \) and its corresponding axioms allow us to describe a discrete partial ordering among activities, but they do not specify the relationship
between the occurrence of an activity and occurrences of its subactivities, i.e., the same (sub)activities can be composed in different ways to form different (super)activities. The relationship between occurrence of an activity and occurrences of its subactivities is captured using the relation \textit{subactivity\_occurrence}, which can serve as a basis for the classification of complex activities, as well. Also, subactivity occurrences are considered to be a temporal part of the complex activity occurrence, and the axioms in the core theory of PSL guarantee that a subactivity occurs \textit{during} an occurrence of the complex activity.

For example, if \textit{PurchaseOrderTransaction(client, supplier, PurchaseOrder)} is a complex activity representing the business scenario in Figure 3.4\textsuperscript{103}, we can specify that \textit{ProcessPurchaseOrder(client, supplier, PurchaseOrder)} is part of this complex activity as:

\[ \forall \, \text{client, supplier, PurchaseOrder} \, \text{subactivity( ProcessPurchaseOrder(client, supplier, PurchaseOrder), PurchaseOrderTransaction(client, supplier, PurchaseOrder))}. \]

We can specify that a successful purchase order scenario requires the execution (occurrence) of all the operations prescribed by the information flow depicted in Figure 3.4\textsuperscript{104}, and that a confirmation of the success of the transaction must be sent by the supplier to the client, as the last operation of the transaction:

\[ \forall \, c, s, po \, \text{occurrence\_of (t\_occ, PurchaseOrderTransaction(c, s, po))} \rightarrow \]
\[ (\exists \, \text{occ1, occ2, occ3, occ4, occ5 subactivity\_occurrence(occ1, occ)} \land \]
\[ \text{subactivity\_occurrence(occ2, occ)} \land \text{subactivity\_occurrence(occ3, occ)} \land \]
\[ \text{subactivity\_occurrence(occ4, occ)} \land \text{subactivity\_occurrence(occ5, occ)} \land \]
\[ \text{occurrence\_of(occ1, ProcessPurchaseOrder(c, s, po))} \land \text{root\_occ(occ1, occ)} \land \]
\[ \text{occurrence\_of(occ2, AcknowledgePurchaseOrder(c, s, po))} \land \text{min\_precedes (occ1, occ2, occ)} \land \]
\[ \text{occurrence\_of(occ3, ShowDeliveryReceipt(c, s, po))} \land \text{min\_precedes (occ2, occ3, occ)} \land \]
\[ \text{occurrence\_of(occ4, ProcessInvoice(c, s, po))} \land \text{min\_precedes (occ3, occ4, occ)} \land \]

\textsuperscript{103} Copyright (c) Open Applications Group. All Rights Reserved. Permission to reproduce granted under the licence agreement available at http://www.oagi.org/oagis/9.0/

\textsuperscript{104} The OAGIS standard does not prescribe, as of now, the operations required for each business scenario or the order in which they must be executed. It does specify, however, constraints such as that an \textit{Acknowledgement} action must be preceded by a \textit{Request} action.
occurrence_of(occ5, ConfirmPurchase(c, s, po)) \land min_precedes(occ4, occ5, occ) \land leaf_occ(occ5, occ))

This scenario description can be further refined by adding more detailed conditions, e.g., that all operations listed in the scenario must succeed in order for the next operation to be initiated.

**Time.** Time points are objects distinct from activities and activity occurrences. They form an infinite linear ordering induced by the relation $before(t_1, t_2)$, with the obvious meaning that time point $t_1$ occurs before time point $t_2$. The metric for the timeline is supplied by the core PSL theory of $T_{duration}$, which maps every pair of time points to a time duration. The individuals of type $time\ duration$ form a logical sort that satisfies the familiar axioms of algebraic fields.

Each activity occurrence is associated to two unique time points that mark its beginning and its end, modeled with the help of two functions, $beginof(activity\ occurrence)$ and $endof(activity\ occurrence)$. The duration of an activity occurrence is, thus, equal to the time duration corresponding to the pair of time points denoting the beginning and the end of the activity occurrence, i.e., $duration(beginof(activity\ occurrence), endof(activity\ occurrence))$. 

**Figure 3.4 Purchase order scenario**

![Diagram of the purchase order scenario]

CustomerParty

SupplierParty

ProcessPurchaseOrder

AcknowledgePurchaseOrder

ShowDeliveryReceipt

ProcessInvoice

ConfirmBOD
Fluents. The abstraction of the business data interchanges presented here is based on first order predicates and terms. The predicates whose value do not change as a result of activity occurrences are referred to as relations, while those predicates and terms whose values are not immutable are referred to as fluents. The payload of messages consists of fluents which are accessed and manipulated by business actions.

We note that the relations and fluents are reified in our framework, i.e., they can appear as arguments of other predicates. A fluent can only be changed by the occurrence of activities, so a fluent holding after an activity occurrence will also hold prior to any immediately succeeding activity occurrence.

Both, relations and fluents, express the properties of the states of the system, and intuitively, the changes in state are captured by the set of fluents who are invalidated or made true by executing an action, i.e., an activity occurrence. PSL offers two relations that we can use to model state changes: prior(fluent, activity occurrence), and holds(fluent, activity occurrence) which specify that the fluent is true prior to and, respectively, after, the activity occurrence:

$$\forall a{\text{ocurrence}}, a{\text{ occurrence of}}(a{\text{ocurrence}}, a{\text{ity}}) \rightarrow$$

$$\neg \text{prior(fluent, activity occurrence)} \land \text{holds(fluent, activity occurrence)}$$

In our framework, Resource() is the most general type of fluent and all other fluents are a type of resource., e.g., ActualLedger(), representing the actual amounts, by account, within the ledger kept by a company, BudgetLedger(), denoting the budget amounts, by account, within the ledger belonging to a business area, etc.

Preconditions and postconditions. Preconditions describe the conditions that need to hold in order for an activity occurrence to happen. Several general types of preconditions are used in the literature, depending on whether the conditions are related only to state properties, the occurrence of other activities or temporal constraints. The general form of preconditions is:

$$\forall a_0, a_1 ((\text{occurrence of}(a_0, a) \land \text{legal}(a_0)) \rightarrow \varphi(a_0)),$$
where $\phi(ao)$ specifies the conditions that need to hold in order for the activity occurrence $ao$ to be possible.

The most common type of precondition is state-based, e.g. we must have the profile of a client stored in the database before updating it,

$$\forall ao, a \ ( \text{occurrence_of}(ao, a) \land \text{legal}(ao) ) \rightarrow \neg \text{prior} (\text{fluent}, ao)$$

where the $\text{activity_occurrence}$ is an occurrence of activity $\text{updateClientProfile}(pc)$, and the $\text{fluent}$ holds if the profile of the client, $pc$, is stored in the company’s database.

Other types of preconditions describe situations where the occurrence of other activities influences the possibility of another activity occurring, e.g., a company can only ship the items ordered by a client after payment has been received.

$$\forall ao_1, a_1 \ ( \text{occurrence_of}(ao_1, a_1) \land \text{legal}(ao_1) ) \rightarrow$$

$$\ ( \exists ao_2, a_2 \ \text{occurrence_of}(ao_2, a_2) \land \text{earlier}(ao_2, ao_1) \land \text{legal}(ao_2) )$$

where $ao_1, ao_2$ are occurrences of shipping and payment, respectively.

Time-based preconditions state conditions that are independent on state, for example that shipping goods will only happen during business hours on Tuesday and Thursday, or that a certain activity needs to be performed regularly, e.g., the orders queue has to be checked by an employee at the end of every business day.

These basic types of preconditions can be combined into complex formulas that incorporate conditions about the states of the systems as well as time and occurrence constraints.

The postconditions, or effects, describe the changes that activity occurrences bring about. As for preconditions, they can be divided, roughly, into state-based, time-based, occurrence-based, and complex preconditions which are combinations of the first three primary types. The general form of postconditions is;

$$\forall ao, a \ ( \text{occurrence_of}(ao, a) \land \phi(ao) ) \rightarrow \text{holds}(f, o),$$
where $\phi(ao)$ specifies conditions on the effects of the activity occurrence $ao$. For example, if the items on order are shipped after more than 7 days since receiving the payment, the shipping is free of charge, i.e. the client is reimbursed for the shipping costs.

$$\forall ao_1, a_1(\text{occurrence\_of}(ao_1, a_1) \land \text{occurrence\_of}(ao_2, a_2) \land \text{duration}(\text{end\_of}(ao_2), \text{beginof}(ao_1)) \rightarrow \text{holds}(\text{fluent}, o),$$

where $ao_1$ corresponds to shipping the goods that were paid for earlier, i.e., in $ao_2$, and $\text{fluent}$ corresponds to crediting the account of the client with the equivalent of the shipping costs.

**Messages**

Messages are at the core of the OAGIS standard and are an essential component of any process model, as processes interact by exchanging messages which, in essence, control their invocation, composition, interoperation and monitoring. Messages can be send by a human actor to another human actor, or can be automatically generated by an actor’s associated software application and sent to another application.

In the spirit of existing frameworks where the flow of information occurs via message passing, such as standards for semantic services\textsuperscript{105}, we model messages as conceptual objects with a non zero duration and focus on their semantic content rather than on how that content is formatted using XML, or other syntactic exchange format.

$$\forall m \ \text{message}(m) \rightarrow \text{object}(m)$$

Messages are classified according to the type of information they are transmitting: e.g. requests, confirmations, acknowledgements.

$$\forall m, \text{message}(m) \rightarrow \exists \text{msg\_type} \ \text{has\_message\_type}(m, \text{msg\_type}) \land \text{message\_type}(\text{msg\_type})$$

$$\forall m, \text{msg\_type} \ \text{has\_message\_type}(m, \text{msg\_type}) \rightarrow \text{message}(m) \land \text{message\_type}(\text{msg\_type})$$

\textsuperscript{105} E.g., the Semantic Web Services Framework (SWSF): http://www.w3.org/Submission/SWSF/
Each message type has an associated business action. This relationships is modeled by \textit{message\_type\_action}(msg\_type, ba).

Message types help organize the various kinds of messages into a taxonomy, via an \textit{isA} relationship between message types, and can also be used for defining message processing constraints.

The payload of a message is represented in our framework with the help of several relations:

\textit{message\_releaseID}(m, releaseID), where \textit{releaseID} is a mandatory attribute identifying the version of the OAGIS standard to which the message conforms;

\[\forall m \text{ message}(m) \rightarrow \exists r \text{ message\_releaseID}(m, r)\]

\textit{message\_language\_code}(m, languageCode), where \textit{languageCode} is a mandatory attributes identifying the language of the data payload;

\[\forall m \text{ message}(m) \rightarrow \exists l \text{ message\_language\_code}(m, l)\]

\[\forall m, l_1, l_2 \text{ message}(m) \land \text{message\_language\_code}(m, l_1) \land \text{message\_language\_code}(m, l_2) \rightarrow l_1 = l_2\]

\textit{message\_versionID}(m, versionID), where \textit{versionedID} is an optional attribute identifying the version of the message type;

\[\forall m, v_1, v_2 \text{ message}(m) \land \text{message\_versionID}(m, v_1) \land \text{message\_version}(m, v_2) \rightarrow v_1 = v_2\]

\textit{message\_systemEnvironmentCode}(m, systemEnvironmentCode), where \textit{systemEnvironmentCode} is an optional attribute which specifies if the message is the result of a test or production level process;

\textit{message\_ID}(m, msg\_ID). The id of a message is unique.

\[\forall m \text{ message}(m) \rightarrow \exists i \text{ message\_ID}(m, i)\]

\[\forall m, id_1, id_2 \text{ message}(m) \land \text{message\_ID}(m, id_1) \land \text{message\_ID}(m, id_2) \rightarrow id_1 = id_2\]
message_sender(m, sender_ID), where sender_ID is a mandatory attribute identifying the creator of the message.

∀m  message(m) → ∃ s  message_sender (m, s)

∀m, sd_1, sd_2  message(m) ∧ message_sender (m, sd_1) ∧ message_sender (m, sd_2) → sd_1 = sd_2

message_timestamp(m, t), where t is a mandatory attribute specifying the creation timestamp of the message. A message’s timestamp is unique.

∀m  message(m) → ∃ t  message_timestamp(m, t)

∀m, t_1, t_2  message(m) ∧ message_timestamp(m, t_1) ∧ message_timestamp (m, t_2) → t_1 = t_2

message_digital_signature(m, s), where s is the digital signature associated to the sender, actor or process who created the message. The digital signature of a message is unique

∀m  message(m) → ∃ s  message_digital_signature (m, s)

∀m, s_1, s_2  message(m) ∧ message_digital_signature (m, s_1) ∧ message_digital_signature (m, s_2) → s_1 = s_2

message_action(m, ba), where ba is a business action, e.g., Add, Create, Change, Get, Load, etc. required to be performed on, or using the data payload of the message. The action specified in a message m has to agree with the action(s) compatible with the type of m. There is only one action per message.

∀m  message(m) → (∃ a  business_action(a) ∧ message_action(m, a) )

∀m, ba_1, ba_2  message(m) ∧ message_action(m, ba_1) ∧ message_action(m, ba_2) → ba_1 = ba_2

message_bo_payload(m, bo_fluent) relates a message to its business object (data) payload. A message can be associated to several fluents describing its business object payload. A list of the current fluents in CBDIO is given in Appendix G.

We note that the OAGIS® standard prescribes that some types of messages should be followed by a response in the form of an acknowledgement message indicating, for example, that the
request to process a purchase order has been received and it has been processed. The sender and receiver in the acknowledgement message are the receiver and sender, respectively, in the original processing request message.

We do not include in the core ontology the notion of a communication channel for specifying how messages are transmitted between actors, but we envision an extension of the ontology with communication channels analogous to channels described in ontologies for the semantic web services.

3.6 Discussion and future work

Research contributions. In this chapter we presented a practical ontology, the Core Business Data Interchange Ontology, which aims to assist in improving the efficiency and interoperability of business organizations and in the development of intelligent decision support systems by providing: (1) a shared terminology, and (2) defining a precise and unambiguous semantics for business data exchanges.

Our research fits within a larger body of work which is anchored by the emerging understanding of the benefit of semantic models and tries to address, with the tools available in the field of knowledge representation, the increasing demand to connect internal and external business processes and move B2B\textsuperscript{106}, B2C\textsuperscript{107} and B2G\textsuperscript{108} interactions beyond data integration helping improve business objectives such as integrated supply chain management, increased supply chain visibility, and collaborative engineering and manufacturing. Domain and application ontologies provide not only a better understanding of the domain of interest but they also support important tasks such as (1) resource discovery, e.g., documents, web services, internal implementations of business activities, (2) harmonization of communication and the (3) semantic mediation and integration between processes.

The current Core Business Data Interchange Ontology describes:

\[ \text{106 Business to business} \]
\[ \text{107 Business to consumer, also business to community} \]
\[ \text{108 Business to government} \]
• **business actors**, the active components of a domain that are able to perform actions in support of fulfilling business goals, e.g., Person, BusinessUnit, softwareApplication;

• **business roles**, representations of the various actions and activities assigned to, required or expected of a business actor in a particular business setting, e.g. Employee, Client, Buyer, ShippingAgency;

• **business actions**, the various activities that business actors can perform, e.g allocate, acknowledge, process,

• **business objects**, entities that business processes operate on, some of which can also be exchanged via messages, e.g., ledgers, client profiles;

• **business messages**, the information exchanged during interactions among actors.

**Limitations**: it is likely that the ontology will continue to grow and evolve as it is updated to include more roles and actions as they become part of the newly emerging business standards. In particular, future extensions should capture other business related concepts such as goals and authorization rights.

In the remainder of this section we discuss our experience designing the ontology along with potential benefits and avenues for future work.

**Benefits of using CBDIO.**

*Desambiguation* CBDIO will help address the non-determinism and limited semantics inherent in the XML-based representations of business messages. Mostly due to the limitations of the XML typing mechanism, message fields had been named according to the intended type and not according to their meaning. For example, in order to enable XML parsers to do some minimal type checking of the content of a message, the date and time when the delivery of an order was expected by a buyer would be represented in an OAGIS-compliant message as:

```xml
...<PurchaseOrderLine>
...<DateTime qualifier="NeedDelivery">…</DateTime>”
...
<PurchaseOrderLine>
```
instead of something more explicit in terms of the intended meaning of the DateTime field

Worse still, if more than one DateTime fields would be associated to the purchase order, as in

an XML parser would raise a warning as it could not distinguish between the two DateTime elements. Furthermore, it would also be impossible, in this case, to specify in the XML schema that one of the DateTime fields is optional and the other is required.

Reusability of exchange specification by different participants under different contexts,

Semantics. The semantic specifications of exchanges are both human-readable and machine-understandable.

Composability/Modularity: complex exchanges can be created from components contained in several different specifications.

Cooperative development/ Information alignment: by sharing the same ontology, the participants in a sequence of business exchanges can collaborate in specifying how they should cooperate in fulfilling the exchange. The participants taking part in business exchanges can communicate and synchronize their state changes as well as the actual values of the exchanged information.

Exception handling/Semantic explanations. Semantic specifications of exchanges can define how exceptional or unusual conditions that occur during the exchange are handled in a manner that
allows for the generation of semantic explanations, e.g., semantic traces of execution, not just standard error messages.

*Compatibility with other Specifications.* This specification will work alongside and complement other semantic and syntactic specifications, such as existing ontologies for the business domain, e.g., TOVE [Fox1995] and BIM [Borgida2012], and syntactic standards such as the Business Process Execution Language for WS (BPEL4WS).

**Challenges.**

*Ontology acquisition.* Analyzing the OAGIS standard has proved quite challenging as the automated parsing of the XSD-based BOD specification has yielded only modest results in terms of getting a clear picture of the domain to be modeled, as the standard’s objective is to provide a detailed and precise structure for the messages to be exchanged not to make explicit a conceptualization of the business domain. The initial findings provided by the automated parsing needed to be supplemented with a careful analysis of the OAGIS related documentation and relevant literature in the area of enterprise modeling.

*Minimality, expressiveness and cognitive ergonomics.* An important open problem in ontology research, and concept formation in general, is the characterization of the set of concepts that are necessary in a representation. It is usually the case that the introduction of a new concept definition can be argued to be useful in some sense, but there are no clear criteria for demonstrating that all concepts are useful, in the sense than none can be removed without a loss of some kind. Approaches are documented in the literature [Wrobel994], which try to evaluate the necessity and/or the utility of new concepts with respect to the intended applications of the ontology, such as increasing the accuracy of a document retrieval system or improving the understanding by human users of the domain theory being developed. We would like to undertake such an analysis once the Core Ontology for Business Data Interchange reaches a stable core.

The ontology development methodology that we have followed here provides a natural evaluation strategy in the form of competency questions. We inferred the competency questions for the Core Business Data Interchange Ontology from the integration scenarios published in the OAGIS documentation and other business standards such as RosettaNet and ebXML, with the
aim of providing a well balanced set of requirements for both, general business knowledge, and knowledge particular to business data exchanges.

**Future work**

*Integration with existing ontologies.* Several ontologies exist that model to various degrees of detail enterprises [Grüninger2000, Ohgren2005] and business processes [Atefi1997, Haller2006, Pedrinaci2008], and we hope that given (1) our focus on business data exchanges, (2) choice of OAGIS, a standard with a large industry support, as a source for developing the ontology terminology and (3) ontology development as an extension of an existing ISO standard, PSL, our research effort will help bridge the gaps between the previous approaches.

CBDIO axiomatization can be extended to capture other business related concepts such as goals and authorization rights, and we are working on identifying bridges to existing ontologies in the field.

*Grounding.* The CBDIO concepts and the instantiations of these concepts are abstract specifications, in the sense that they do not describe the details of specific message formats and transport protocols.

A grounding has the role of making explicit these details by defining mappings between concepts in CBDIO to constructs in execution languages that describe the concrete realizations of these concepts, thus bridging the gap between the higher, conceptual, level, descriptions and the lower level details related to the interoperability between processes.

The two complementary formalisms that are partnered in a grounding cover overlapping but different conceptual spaces. Languages for concrete specifications are concerned with definitions whose syntax, generally expressed in XSD, is valid and checkable. CBDIO, on the other hand, allows the definition of types (Input/Output fluents) via logical assertions and is not concerned with syntactical constraints. A grounding, thus, uses the semantics provided by the ontology and the syntactical constraints defined by the concrete, execution, language.

Relying on groundings to provide the concrete execution details of abstract CBDIO descriptions allows us to benefit from the best of both worlds: on the abstract side from the expressiveness of
the ontology, relative to what syntax-oriented formalism like XSD could provide, and on the concrete side from the various existing transport mechanisms and communication protocols.

We envision that a semantic environment for enacting complex business processes whose data exchanges are specified using CBDIO would rely on groundings to some of the emerging standards, such as BPEL and WSDL, in order to provide the run-time details needed to handle the messages that conform to these syntactic standards and execute business transactions.
4 Measuring the Similarity between Ontological Concepts

The notion of similarity, although extensively used in various contexts, is difficult to define. In psychology it is regarded as a “factotum to cognition” [Larkey2005], and has been studied extensively to understand why we divide entities into categories and what makes categories comparable or not. When reviewing the literature, one encounters at least three different terms, used at times interchangeably, *(semantic) distance, similarity* and *relatedness*. Sometimes similarity and relatedness are byproducts of a semantic distance, and other times they are defined as measures in their own right. In the interest of simplicity some of the original notations and definitions have been modified in order to offer a unified presentation.

Measuring similarity in general, and word/concept similarity in particular, has been actively investigated by psychologists and researchers from various computer science communities, such as information retrieval, artificial intelligence and databases, see [Lee1993, Jurisica1998] for an overview. A good discussion of the cognitive aspects of similarity in provided by Tversky [Tversky1977] and by Ashby and Perrin [Ashby1988].

**Similarity versus Distance** In many instances a similarity measure is thought of as the inverse of a dissimilarity or distance function, which is, in general, a number assigned to a pair of points in a conceptual space to indicate how far those points are from one another. A distance function can be transformed into a similarity measure and vice versa by taking the reciprocal, but more elaborate transformations exist.

In physical sciences many distance functions are metrics, meaning that they are always nonnegative, symmetric and observe the triangle inequality; in contrast, distance functions defined on ontologies are not metrics.

There is no consensus on the properties a similarity function ought to have, but *symmetry* is many times among the requirements. Also, at times, *reflexivity*, $similarity(x, x) = 1$, is required instead of the 0-distance from self property, $similarity(x, x) = 0$. So far no real progress has been made in deciding which distance axioms are appropriate for comparing words and concepts, in
general. There are also long-standing psychological objections to the axioms used to define a metric. For example, a metric function is nondirectional, but in practice, people appear to be more likely to say that a subconcept is similar to a superconcept than they are to say that a superconcept is similar to a subconcept [Tversky1977, Ashby1988]. It was pointed out [Jurisica1994], however, that this perceived asymmetry might be attributed to a change in the implicit context people have in mind when assessing the similarity between two concepts.

Another unresolved issue transitivity: a relationship is transitive if from \( A \) is the same as \( B \) and \( B \) is the same as \( C \) it follows that \( A \) is the same as \( C \). However, if \( A \) is similar to \( B \) and \( B \) is similar to \( C \), any further inferences are questionable. For example, when the similarity between two objects is established based on spatial relationships such as is-adjacent-to, it cannot be inferred in all scenarios that \( A \) is similar to \( C \).

**Strategies for evaluating similarity measures.** The evaluation of a similarity measure is a complicated and not necessarily a straightforward matter, as the requirements or expectations and also the human perceptions of similarity vary greatly.

The evaluation methodologies adopted in the literature can roughly be divided into direct and task-based approaches. In the direct evaluation methods a list of pairs of concepts are ranked using the similarity values returned by the semantic measure being evaluated. This ranking is then either qualitatively assessed by human subjects or statistically compared to an aggregate ranking of the same set of pairs provided by domain experts. Two of the most popular statistical rank correlation assessment methods are the Spearman correlation coefficient and Kendall's \( \tau \) coefficient. All correlation coefficients range from \([-1, 1]\) and take the values of 1, 0, or -1 when the two rankings are identical, unrelated or inverse, respectively. However, a fact often overlooked in the field is that the results provided by the Spearman's coefficient and Kendall's \( \tau \) are not directly comparable, since the two correlation coefficients are sensitive to different biases in the data.

One of the most important benchmarks for direct evaluation of semantic similarity measures is a study performed by Miller and Charles [Miller1991], the de facto standard in the information retrieval community. In this study 38 undergraduate students assigned relatedness scores on a scale from 0 to 4, where 0 means that the two nouns are not related, to 30 word pairs, nouns of
general use. The pairs were ranked according to the mean of the scores assigned by all subjects such that the first pair was the most related and the last pair was the least related. To our knowledge, a benchmark for evaluating bio-ontology similarity measures has yet to emerge.

The indirect, or task-based evaluation consists of using the semantic similarity values in various scenarios such as word sense disambiguation (selecting the most appropriate sense according to the current context [Banarjee2003, Budanitsky2001], document retrieval [Budanitsky2006], predicting gene function [Hu2007, Stoica2006, Khatri2005], treating missing values in microarray experiments [Tuikkala2006], validation of predicted protein-protein interactions [Lin2004, Guo2005], and prediction of co-regulated gene groups [Tang2007]. Most of the semantic similarity evaluations are performed using, WordNet\textsuperscript{109} or, less frequently, Roget's Thesaurus\textsuperscript{110} for measures designed for use in information retrieval tasks and the Gene Ontology for measures intended to be used on bio-ontologies.

In what follows, we will discuss several existing semantic similarity measures, with an emphasis on information content-based measures and applications, which is the main focus of our research and introduce our contributions to the field. In our recent work [Rosu2008] we presented a comprehensive survey of over twenty semantic similarity measures and applications from several different fields and we will only review here in some detail those that are directly related to our work.

### 4.1 Feature-based similarity measures

The feature-based approach to computing similarity assumes that word/concept representations include features that are either specified explicitly (e.g., yeast is a eukaryote and a model organism and it multiplies by budding), or defined in terms of a distributional context (e.g., the term protein is observed within +/- 2 words of interacts). In this framework, the words/concepts are represented as vectors in the space of features, and the semantic similarity between them is defined as a metric over this vector space.

\textsuperscript{109} Available at: http://wordnet.princeton.edu

\textsuperscript{110} Available at: http://poets.notredame.ac.jp/Roget/contents.html
The cosine similarity measure, along with Jaccard’s and Dice’s coefficients, are the most widely used. Given two objects $A$ and $B$ with associated feature $A_i$ and $B_i$, $0 < i < n + 1$:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

$$similarity(A, B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}.$$

A fair criticism of this framework is that in practical applications the number of features per concept, as well as their distribution, is highly variable, which leads to several non-trivial computational issues such as choosing the relevant features to be included in the vector representation and reducing the dimensionality of the vector space. An entire line of research is dedicated to studying the computational aspects of vector representations, but this is beyond the scope of this thesis.

### 4.2 Edge-based similarity measures

Early on in the development of the semantic similarity field it had been hypothesized by several researchers [Brachman1983, Rada1989, Kim1990] that the semantic similarity and the edge distance between concepts organized in a hierarchy are closely related and even equivalent. This hypothesis is based on several important, but often overlooked assumptions. The concepts are regarded as having underlying features, that are simply not made explicit, and an *is-a* relationship is thought of as implying the inheritance of features such that a child concept differs from a parent concept in at least one feature. The hierarchy is also assumed to contain only *is-a* edges and most of the existing approaches either explicitly or implicitly disregard all the other relationships in an ontology.

The information retrieval community was among the first to propose and implement edge-based measures along with benchmarks and frameworks for evaluating them. The measures evolved from simple and weighted path distances [Rada1989, Lee1993, Sussna1993, Wu1994, Wu2006] to fairly complicated mathematical expressions with intricate systems of edge weights, or edge

4.3 Mutual information-based similarity measures

The more commonality two terms have, the more similar they are, we can hope to estimate their similarity by quantifying the amount of information they share. Information theory offers such a possibility through the use of *mutual information*, which measures the information that one random variable gives about the value of another random variable. We offer here a brief recapitulation of several important concepts from information theory.

Let X and Y be two random variables with joint probability distribution \( f(x,y) \) and marginal probability distributions \( f_1(x) = \sum_y f(x,y) \) and \( f_2(y) = \sum_x f(x,y) \).

\[
H(X) = -\sum_x f_1(x) \log(f_1(x)) \quad \text{and} \quad H(Y) = -\sum_y f_2(y) \log(f_2(y))
\]

represent the entropies of X and Y, respectively, and \( H(X,Y) = -\sum_{x,y} f(x,y) \log(f(x,y)) \) represents the entropy of the joint variable (X,Y). The conditional entropies are defined as follows:

\[
H(X|Y) = \sum_y f_2(y) H(X|Y=y) = -\sum_y f_2(y) \sum_x f(x|y) \log(f(x|y))
\]

and similarly for \( H(Y|X) \).

The *information* in the outcome \( X=x \) about \( Y \) is defined as \( I(X=x;Y) = H(Y) – H(Y|X=x) \) and, analogously, the *information* in the outcome \( Y=y \) about \( X \) is defined as \( I(X;Y=y) = H(X) – H(X|Y=y) \). Since \( I(X=x;Y) \) is a function of outcomes of \( X \) and \( I(X;Y=y) \) is a function of outcomes of \( Y \), they are not directly comparable. However, taking the expectation as

\[
E[I(X=x;Y)] = \sum_x f_1(x) I(X=x;Y) \quad \text{and} \quad E[I(X;Y=y)] = \sum_y f_2(y) I(X;Y=y)
\]

we deduce that

\[
E[I(X = x; Y)] = E[I(X; Y = y)] = -\sum_x \sum_y f(x,y) \log\left( \frac{f(x,y)}{f_1(x) \ast f_2(y)} \right)
\]

This quantity is termed the *mutual information* of variables \( X \) and \( Y \), \( I(X;Y) \). The term

\[
\log\left( \frac{f(x,y)}{f_1(x) \ast f_2(y)} \right)
\]

is called *pointwise mutual information*. 
A related empirical measure, the association ratio, was introduced by Church and Hanks [Church1990]. Let $x$ and $y$ be two random variables with marginal probabilities $f_1(x)$ and $f_2(y)$ and joint probability $f(x,y)$. The association ratio of two words $x$ and $y$ is defined as:

$$
\text{association ratio}(x, y) = \log \left( \frac{\hat{f}_w(x,y)}{\frac{w-1}{N} \hat{f}_1(x) \hat{f}_2(y)} \right)
$$

where $\hat{f}_w(x,y)$ is an estimate of the joint probability $f(x,y)$ as the number of times word $x$ is followed by word $y$ in a windows of fixed size $w$ and $\hat{f}_1(x)$ is an estimation of the marginal probability $f(x)$ as the frequency of $x$ in the corpus divided by $N$ the size of the corpus. $\hat{f}_2(x)$ is defined analogously.

The association ratio is in essence an asymmetrical estimation of the pointwise mutual information of words $x$ and $y$. The non-symmetry follows from the fact that the estimation of the joint probabilities does not count properly the number of times the two words appear together, regardless of order, and encodes the linear precedence of $y$ by $x$ in the window used for estimation. As Church and Hanks observed, the association ratio can be transformed into a symmetric measure by averaging the matrix of measurements with its transpose, but they argued in favour of keeping the measure asymmetric, as the order of information might prove interesting.

### 4.4 A new mutual information-based measure for the Gene Ontology

We have defined an estimation of mutual information for calculating the semantic similarity for concepts in the Gene Ontology as follows. Let $\text{GENES} = \{g_1, g_2, \ldots, g_N\}$ be a collection of genes and let $\text{GO} = \{t_1, t_2, \ldots, t_M\}$ be the set of terms from the Gene Ontology. Let $\text{ANNOTATIONS}_{ij} > 0$ be a matrix such that $\text{ANNOTATIONS}_{ij} = 1$ if gene $g_i$ is annotated with term $t_j$ and 0 otherwise. Genes are often annotated with several different terms and the parent child relationship between terms implies that a gene annotated with a term $t_i$ is also annotated with all
its ancestors in the Gene Ontology. Let \((A_j)_{0 \leq j \leq M+1}\) be a set of random variables ranging over \{0, 1\} such that, for all \(0 < j < M+1\), the event \((A_j = x)\) is the outcome of the following experiment “Let g be a gene. Assign 1 to x if term \(t_j\) is annotated to gene g and 0 otherwise”. The mutual information between any two random variables \(A_j\) and \(A_k\) is defined as

\[
\sum_{a_j \in \{0, 1\}} \sum_{a_k \in \{0, 1\}} P(A_j = a_j, A_k = a_k) \times \log\left( \frac{P(A_j = a_j, A_k = a_k)}{P(A_j = a_j) \times P(A_k = a_k)} \right)
\]

The joint probability \(P(A_j = a_j, A_k = a_k)\) is estimated as the number of genes annotated with both GO term \(t_j\) and GO term \(t_k\), normalized by \(N\), the total number of genes. The marginal probabilities for \(A_j\) and \(A_k\) are estimated as the total number of genes annotated with \(t_j\) and \(t_k\), respectively, also normalized by \(N\).

We define the similarity between two GO terms \(t_j\) and \(t_k\) as the relatedness ratio between the corresponding variables \(A_j\) and \(A_k\):

\[
\text{relatedness ratio}(a_j, a_k) = \log_2 \left( \frac{\frac{\hat{f}_w(a_j, a_k)}{N}}{\frac{\hat{f}_1(a_j)}{N} \times \frac{\hat{f}_2(a_k)}{N}} \right) = \log_2 \left( \frac{\frac{\hat{f}(a_j, a_k) \times N}{N}}{\frac{\hat{f}_1(a_j) \times \hat{f}_2(a_k)}{N}} \right).
\]

Our measure is similar to the association ratio estimation for mutual information between words and contexts introduced by Church and Hanks [Church1990]. However, the association ratio provides an asymmetric estimation of mutual information while the relatedness ratio is a symmetric measure.

We have calculated the association ratio for all pairs of terms in the Gene Ontology using a local copy of the Gene Ontology Database and also applied it to assessing the similarity of over 10000000 pairs of yeast genes.

Limitations: it was pointed out that the association ratio observed that the measure may become unstable for small counts [Church1990] and we are currently investigating whether discarding pairs with small counts is also warranted when assessing the similarity between terms in the Gene Ontology.
4.5 Information content-based measures

Although the edge based approach is an intuitive and easy to implement way of computing the semantic similarity between concepts organized in *is-a* hierarchies, it suffers from a widely acknowledged problem: it assumes that the links in the taxonomy represent uniform distances.

The reliance on the uniform distance assumption can be avoided by considering complementary statistical methods such as that introduced by Resnik [Resnik1995].

4.5.1 Modeling and estimating the information content of a concept

For any information content similarity measure, the accurate/appropriate modeling and estimation of information content are essential. We will examine them in some detail in this section.

Information content is by no means an easy concept to define, as information itself is a rather intuitive concept with a relatively wide ‘semantic halo’. Most of the work in information content based semantic similarity measures borrows heavily from the classical information theory. However, it is not very clear if the notions and results in information theory translate naturally into this field, as the premises and goals of information theory are different from those of information content based similarity measures. More specifically, classical information theory is concerned with modeling and analyzing technical aspects of communication are considered while the semantic aspects are not taken into account. We will revisit this point later in this section.

So what is information content? In classical information theory, the information content of a sequence of symbols is defined as the minimal number of bits required to transmit that sequence using an optimal encoding. According to this definition, even though we can use a less efficient encoding, which would require more bits, our choice of coding would not increase the amount of information transmitted. We would like to point out that this strictly quantitative interpretation of information as simply a number of bits, might not be an appropriate view of information in human communication in general. The debate on whether or not languages are actually
optimized for speech, and if not, why not, has been raging on in the natural language processing field for over 75 years, ever since Zipf formulated his famous law [Zipf1935].

Despite this debate, the dominant view in information theory is that the information content of a sequence, word, symbol, phoneme, etc. is linked with is predictability or associated information gain, and there are numerous studies who are striving to bring empirical evidence in its support. We present a short overview of recent approaches on estimating information content.

Cohen Priva studied the information content of phonemes in his work on the possible link between phoneme informativity, a term he coined, and phoneme deletion in spoken languages [Cohen-Priva2008]. In short, the author investigated the role of local predictability, such as the expectation that one has that a certain phoneme will follow a certain context, e.g., we are fairly certain that phoneme [n] should follow after [explanation], in the deletion of that phoneme from speech. He defined the surprisal at hearing a phoneme $p$ in some context $CONTEXT$, in other words $p$’s local predictability, using the conditional probability $P(p|CONTEXT)$. A phoneme’s informativity is then estimated as the negative log value of the phoneme’s local predictability averaged across every instance of the phoneme in spoken natural language. He concluded that his, and previous measures, are able to predict some of the variance that phonological theories do not account for. However, they do not explain many other aspects, such as why completely predictable phonemes are very stable in spoken English.

Cohen Priva’s work has recently been extended to words and contexts by Piantadosi et al. [Piantadosi2011] who studied the link between words frequency of use and their length and information content, measured as the average amount of information, and concluded that “information content is a considerably more important predictor of word length than frequency”. Their approach works as follows: let C and W be two random variables representing contexts and words with a joint distribution $f(w,c)$ and marginal distributions $f_1(w)$ and $f_2(c)$, let the amount of information conveyed by a word $w$ in context $c$ be $-\log(f(w|c))$. Noting that probability of word $w$ occurring in context $c$ is given by $f(w,c)/(f(w))$, Piantadosi and colleagues defined the average amount of information for a word $w$ as $-\sum_{c \text{ context}} \frac{f(w,c)}{f(w)} \log(f(w|c))$. They modeled context as an N-gram, a widely used approach in natural language processing, but conceded that there are many variables that may count as part of the “context”.
We would like point out that the inherent structure encoded in ontologies allows for alternative ways to model the information content of a concept, which do not rely on probabilistic assumptions and statistical information.

**Quantifying Information Content.** We review here the important issues in modeling and estimating information content with a discussion of the functions used in quantifying information. The logarithm is by far the most widely used function for information quantification, at least in classical information theory. Its use originates with Hartley [Hartley1928]. In his analysis of the generation and transmission of telegraph messages he noted that for a choice of $s$ primary telegraphic symbols the total number of messages that could be generated after making $n$ choices of primary symbols is $s^n$ and defined $n \ast \log(s)$ as a measure of information. He motivated this choice by arguing that a “measure of practical engineering value should be proportional to the length of the message”, in other words of the form $k \ast n$, where $k$ is a constant. Hartley obtained his, now famous, formula for quantifying information by solving the equation generated by imposing the condition that, given two systems with initial symbols $s_1$ and $s_2$, whenever the sets of messages that could be generated after $n_1$ and $n_2$ choices, respectively, are equal, the information associated with them should also be equal. He conceded that in human speech some words may have greater significance than others, but he noted that his measure was designed to be “independent of psychological factors”.

Shannon, one of the pioneers of information theory, also noted in his seminal work on a mathematical theory for communication [Shannon1948] that messages have meaning, but, as far as he was concerned, the “semantic aspects of communication are irrelevant to the engineering problem.” Although rarely, if ever, mentioned in the literature that evolved after him, Shannon’s reasons for continuing to use Hartley’s measure, although he acknowledged that it needed to be “generalized considerably”, are also very interesting and worthy of consideration. He believed, as did Hartley, that the logarithm was a natural alternative for quantifying information and he listed the following arguments in support of this choice: (1) practicality (parameters of engineering importance tend to vary linearly with the logarithm of the number of possibilities), (2) cognitive ergonomics (according to Shannon, we intuitively measure entities, by linear comparison) and (3) mathematical convenience (many limiting operations are simpler when using the logarithm).
Thus, it is not clear how the measures which were designed to ignore the meaning of the information can be best adapted for modeling the information content of ontological concepts and we identify this as an area that warrants careful and thorough investigation.

**The probabilistic approach.** Philip Resnik pioneered the use of information theoretic models for semantic similarity. In his seminal work on semantic similarity for taxonomies [Resnik1995] he introduced the probabilistic notion of information content to the field of semantic similarity measures and most subsequent information content based models in this area follow a similar approach.

One of the main assumptions of this model is that the frequency of use is related to how informative a concept is: the more frequently a term is used the less informative it is. This assumption translates into Resnik’s model in the estimation of the information content of a concept \( c \) as \(- \log(p(c))\), where \( p(c) \) is the probability of observing an occurrence of that concept in a text corpus, annotation database, etc.

He also postulated that the similarity between two concepts is the information that they share and that that shared information can be estimated as the information content of their most specific common ancestor. Later, Lin employed the same definition of information content, but defined the similarity between two concepts as the ratio between the amount of information needed to describe their commonality and the information needed to fully describe them [Lin1998]. For certain interpretations of common and distinguishing information, on hierarchies without multiple inheritance, Lin's measure is in fact equivalent to Resnik’s measure normalized by the combined information content of the two concepts. Appendix F offers a more detailed presentation of Resnik’s and Lin’s models as well as of recent effort to extend them.

In his model the set of concepts \( C \) in the ontology under consideration is associated with a probability function \( p: C \rightarrow [0, 1] \) such that for any concept \( c \) in \( C \), \( p(c) \) is the probability of encountering an instance of concept \( c \) in a text corpus or database. An occurrence of a concept will also count as an occurrence of its superconcepts, which implies that \( p(.) \) is monotonically nondecreasing. The top concept's probability is set to 1. In this framework, the information content of a concept \( c \), \( IC(c) \), is quantified as \(- \log(p(c))\). The intuition behind this approach is that the informativeness of a concept decreases as its probability increases, or in other words, the more abstract the concept, the lower its information content.
Another key assumption made in Resnik’s model is that the *common/shared information* of two concepts is encoded in their *common ancestors*. Resnik defined the similarity between two concepts $c_1$ and $c_2$ to be:

$$sim_{Resnik}(c_1, c_2) = \max_{c \in \text{Ancestors}(c_1) \cap \text{Ancestors}(c_2)} \{-\log(p(c))\}$$

Taking the maximum over all common ancestors in effect identifies the closest common ancestor with the maximum *information content*, termed *most informative subsumer*. Resnik argued that taking the maximum with respect to *information content* is analogous to choosing the shortest path with respect to edge distance. However, this observation does not necessarily hold in the case of taxonomies with multiple inheritance, where two distinct ancestors that offer the same minimal upper bound measured as edge distance in the graph might have very different *information content* values. It is also still unclear how the common information is shared/divided among multiple ancestors, which is an important issues in designing an effective measure.

In the original evaluation of $sim_{Resnik}(c_1, c_2)$ the probability of occurrence for each concept$^{111}$ was estimated using a text corpus$^{112}$, as the relative frequency $p(c) = \text{frequency}(c)/N$, where $N$ is the total number of concepts observed and frequency($c$) is calculated as:

$$\text{frequency}(c) = \sum_{c \in \text{Subconcepts}(c) \rightarrow |c|} \text{count}(c')$$

An alternative information-theoretic framework was introduced by Lin [Lin1998], with the intention of defining a universal measure that would not be tied to a particular application, domain or resource, as most of the earlier measures were designed specifically for WordNet. The following intuitions are at the foundation of Lin's approach:

1. the similarity between two objects is related to their commonality
2. the more commonality they share the more similar they are;
3. the similarity between two objects is related to their differences
4. the more differences they have the less similar they are;

---

111 a concept is a noun in the WordNet taxonomy  
112 The Brown Corpus of American English  
Available at: http://www.sscnet.ucla.edu/issr/da/index/techinfo/M0911.HTML
5. the maximum similarity between two objects is reached when the two objects are identical.

Since there are many alternative ways to define similarity that would be consistent with these intuitions, Lin made several additional assumptions.

Given two concepts $c_1$ and $c_2$, let $\text{description}(c_1, c_2)$ be a statement describing $c_1$ and $c_2$ and $\text{common}(c_1, c_2)$ be a statement describing the commonalities between $c_1$ and $c_2$.

- **Assumption 1.** The commonality between $c_1$ and $c_2$ is measured by
  \[ \text{IC}(\text{common}(c_1, c_2)) = -\log(p(\text{common}(c_1, c_2))). \]

- **Assumption 2.** The differences between $c_1$ and $c_2$ is measured by
  \[ \text{IC}(\text{description}(c_1, c_2)) - \text{IC}(\text{common}(c_1, c_2)). \]

- **Assumption 3** The similarity between $c_1$ and $c_2$ is a function of their commonalities and differences.

- **Assumption 4** The similarity between a pair of identical objects is 1.

- **Assumption 5** When there is no commonality between $c_1$ and $c_2$, their similarity is 0 regardless of how different they are.

- **Assumption 6** The overall similarity between two objects is a weighted average of their similarities computed from different perspectives, corresponding to their common ancestors.

Lin defined the similarity between two objects $c_1$ and $c_2$ as the ratio between the amount of information needed to describe their commonality and the information needed to fully describe them:

\[ \text{sim}_{\text{Lin}}(c_1, c_2) = \frac{\log(p(\text{common}(c_1, c_2)))}{\log(p(\text{description}(c_1, c_2)))} \]

In a hierarchy, assumed to be a tree, the commonality between two concepts $c_1$ and $c_2$ is considered to be fully described by the sentences “$c_1$ is a subconcept of $c_0$” and “$c_2$ is a subconcept of $c_0$”, where $c_0$ is the least common ancestor of $c_1$ and $c_2$. The similarity between the two concepts thus becomes:
If the tree restriction is relaxed and the taxonomy is allowed to be a directed acyclic graph, Lin's measure can be rewritten as:

\[
sim_{Lin}(c_1, c_2) = \frac{2 \log(p(\bigcup_{c_j \in Ancestors(c_1) \cup Ancestors(c_2)} c_j)))}{-\log(p(lca(c_1))) - \log(p(lca(c_2))) + \log(p(c_2))}
\]

On hierarchies without multiple inheritance Lin's measure is in fact equivalent to \(\text{sim}_{Resnik}\) normalized by the combined information content of the two concepts.

More recently, as part of their study on functional evaluation of domain-domain and human protein interactions, Schlicker et. al. [Schlicker2007] extended the work of Resnik and Lin by proposing the following measure:

\[
sim_{Schlicker} = \max_{c \in Ancestors(c_1) \cup Ancestors(c_2)} \left\{ \frac{2 \times IC(c)}{IC(c_1) + IC(c_2)} (1 - p(c)) \right\}
\]

where \(IC(c) = -\log(p(c))\). When used for terms in a hierarchy, \(\text{sim}_{Schlicker}\) ranges from 0, when the closest ancestor of the two terms is the root concept, to 1.

Although fairly intuitive and not overly complex, from a computational perspective, the frequentist approach to information content modeling can be problematic for several reasons:

1. **Bias in word usage or biases in research literature.** For example, the word *discombobulated* is used less often than its synonym, *confused*, but it is intuitively, no less informative. Also, *neuropeptide receptor activity* and *proteinase activated receptor activity* are equally informative, but while the first term appears as an annotation for over 100 gene products, the second is used less than 10 times, possibly because less research has been devoted to that area.

2. **Sparse data**, i.e. not all ontological concepts can be found in text corpora or annotation databases. In addition, there is no standardized method for the frequency-based estimation of the
information content of a concept. Wagner [Wagner2004] provides a few insights on this subject, in particular about estimating the frequency counts of concepts in multiple-inheritance hierarchies based on a text corpus.

The Intrinsic Information Content Approach. The shortcomings of the frequentist approach can be addressed by considering a purely intrinsic estimation of the information content of concepts.

A fundamental assumption of the information content models is that the similarity between two concepts is proportional to the amount of information they share. We have identified two distinct approaches for interpreting the shared information between two ontology concepts: (1) as being encoded in the least common ancestors and (2) as the mutual information between two random variables modeling the occurrence of the two concepts in a text corpus or annotation database.

We discussed the interpretation of shared information as mutual information in section 4.3 and we focus in this section on the interpretation of shared information as the information content of the least common ancestors. This approach is central to most information content based semantic similarity measures, but it can be problematic in ontologies which allow multiple inheritance since it is not clear what would be a suitable treatment of multiple common ancestors. Most measures only consider the least common ancestor with the largest information content; although it can be argued that valuable insight is lost by ignoring the other common ancestors. On the other hand, averaging over ancestors that belong to two separate hierarchies might not make sense, e.g., if it is clear that they represent distinct senses of a word.

Contextual information could be used to guide the selection of the appropriate common ancestor of the two concepts being compared, but the development of such measures is hampered by the absence of adequate formalisms for modeling context.

We have decided to investigate the impact on the estimation of semantic similarity of concepts of different strategies of treating their least common ancestors by implementing several similarity measures in two ways: (1) taking into consideration only the most informative least common ancestor and (2) averaging over all least common ancestors.

Our statistical analysis of GO has reveals that a significant proportion of term pairs share more than one ancestor. By far the most pairs with multiple least common ancestors, when considering
all edge types, are found in the Biological Process ontology (BP), 12,851,250 pairs, followed by Cellular Component (CC) and Molecular Function (MF) with 686,515 and 139,213 pairs, respectively. It is interesting to note that, even though the Molecular Function ontology has more than four times more terms than the Cellular Component, it has three times less pairs with multiple ancestors. Percentage-wise, when considering all edge types, 30% of the total number of Cellular Component term pairs have multiple least common ancestors, and so do over 10% of the Biological Process term pairs.

For each subontology we have determined that, considering only the \textit{is a} edges, the maximum number of least common ancestors for a pair of terms is 5 for Biological Process, 4 for Molecular Function and 5 for Cellular Component. If we consider all edge types, the maximum number of ancestors per pair of terms is, 7 (BP), 4 (MF) and 7 (CC).

We found that the correlation between the rankings induced by calculating the similarity score of a pair of terms as maximum or average over all the least common ancestors is very high, over 0.85, across the three GO ontologies, with the exception of Resnik’s measure and our IC\textsubscript{leaves} proposal, whose correlation coefficients are approximately 0.7.

The distributions of the values assigned by the similarity measures we have implemented follow, roughly, logarithmic distributions and the majority of the pairs of terms have small and very small similarity values.

### 4.6 New (intrinsic) information content-based similarity measures

We propose the following formulation of information content:

\[
IC_{\text{Leaves}}(c) = 1 - \frac{\log(|leaf\_subconcepts(c, relations)| + 1)}{\log(all\_leaves(relations))},
\]

\footnote{At the time of writing}
where \( \text{leaf\_subconcepts}(c, \text{relations}) \) is the number of most specific subconcepts of \( c \) in the graph induced by the \text{relations} under consideration, such as \text{is\_a}, \text{part\_of}, groups of relations or all available relationships. \( \text{all\_leaves}(\text{relations}) \) denotes all the leaf concepts in the ontology graph induced by the relations under consideration.

\( \text{IC}^{\text{leaves}}(c) \) distinguishes between the most specific descendents of a concept \( c \), but does not take into account the depth in the ontology of term \( c \). We are positing that both the depth and the local density of a concept \( c \) directly influence its informativity and we introduce the following method for estimating its information content:

\[
\text{IC}(c) = f(f_1(\text{depth}(c)), f_2(\text{local\_density}(c))),
\]

where \( f(f_1, f_2) = (f_1)^\alpha (f_2)^\beta \), \( \alpha \) and \( \beta \) are positive parameters controlling the influence of a concept’s level of abstraction and local density and

\[
f_1(\text{depth}(c)) = \frac{\text{depth}(c)}{\max \text{ \{depth(v)\}}}, \quad f_2(\text{local\_density}(c)) = \frac{\text{all\_leaves}}{\text{leaves}(c)}
\]

We define the specificity of a term \( c \) to be \( \text{specificity}^{\text{leaf}}(c) = \frac{\text{depth}(c)}{\max \text{ \{depth(v)\}}} \cdot \frac{\text{all\_leaves}}{\text{leaves}(c)} \).

We also consider an alternative definition for the concept depth factor and the local density that take into account all the descendents of a concept, not just the leaf concepts:

\[
f_1(\text{depth}(c)) = \frac{\text{depth}(c)}{\max \text{ \{depth(v)\}}} \quad f_2(\text{local\_density}(c)) = \frac{\text{all\_concepts}}{\text{descendents}(c)}
\]

\[
\text{specificity}^{\text{subconcept}}(c) = \frac{\text{depth}(c)}{\max \text{ \{concepts(v)\}}} \cdot \frac{\text{all\_concepts}}{\text{descendents}(c)}
\]

It follows that, when \( \alpha = \beta = 1 \), \( c \)'s information content is \( \text{IC}(c) = \text{specificity}(c) \).
Limitations: our measure assumes that interpretation of shared information as the information content of the least common ancestors, which can be problematic in ontologies which allow multiple inheritance since it is not clear what is the best way to deal with multiple common ancestors. Also, our method assumes that all edges have the same importance and that the children of an ontological concept cover it in its entirety, which may not be true for some ontologies.

4.7 Hybrid similarity measures

None of the information content, edge-based and feature-based approaches are free of criticism. The information content and feature based methods do not really take advantage of all the information contained in the structure of ontology, such as the depths of the concepts being compared, while the edge based measures ignore the relative importance of terms as illustrated by their frequency of use and the feature based measures.

The evaluation studies carried out so far have been unable to conclusively indicate that a certain feature-based, structure-based or information content-based measure clearly outperforms the others across a range of contexts. Ensemble approaches where the optimal measure is chosen dynamically and methods which incorporate multiple types of information might prove superior. In some cases, a combined approach is the only way to assess the similarity between terms, such as those belonging to different ontologies.

A few years after Resnik's seminal work on shared information as a measure of similarity, Jiang and Conrath [Jiang1997] introduced a semantic distance which combines knowledge about the ontology structure and information content with the goal of restoring the influence of the hierarchical edges in establishing the similarity between two concepts, using the statistics derived from the text corpus as a corrective factor. The distance proposed by Jiang and Conrath to assess the similarity between two words is the usual weighted shortest path length:

$$dist_{Jiang\;and\;Conrath}(w_1, w_2) = \sum_{c:\text{path}(c_1, c_2), c \in \text{lca}(c_1, c_2)} weight(c, parent(c))$$

where the weights controlling the contribution of an edge are of the form:
weight\left( c_{c}, c_{p} \right) = \left\{ \begin{array}{ll}
\beta + (1 - \beta) \times \frac{\text{local density}}{\text{node depth}} \\
\times \text{edge_strength}\left( c_{c}, c_{p} \right) \times \text{edge_coefficient}\left( c_{c}, c_{p} \right)
\end{array} \right\}^\alpha
\]

where \( c_{c} \) is the child concept and \( c_{p} \) is the parent concept and \( \alpha > 0 \) and \( 0 \leq \beta \leq 1 \) are parameters adjusting the influence of the node depth and the local density.

Jiang and Conrath introduced the notion of edge strength, approximated by the conditional probability \( p(c_{c} | c_{p}) \) of encountering an instance of the child concept \( c_{c} \) given an instance of the parent concept \( c_{p} \):

\[
\text{edge_strength}\left( c_{c}, c_{p} \right) = -\log\left( p\left( c_{c} \mid c_{p} \right) \right) = -\log\left( \frac{p\left( c_{c} \cap c_{p} \right)}{p\left( c_{p} \right)} \right)
\]

Since an occurrence of a concept is also counted as an occurrence of its superconcepts, \( p(c_{c} \cap c_{p}) \) is in fact \( p(c_{c}) \) and thus \( \text{edge_strength}\left( c_{c}, c_{p} \right) = -\log\left( p\left( c_{c} \right) \right) + \log\left( p\left( c_{p} \right) \right) \) and, using the definition of information content as \( -\log(p(c)) \), \( \text{edge_strength}\left( c_{c}, c_{p} \right) = IC\left( c_{c}, c_{p} \right) \).

There are many possible conversions of Jiang and Conrath’s distance into a similarity function, each with a different behaviour and performance. We reviewed some of the most popular transformations in [Rosu2008].

Li et al.[Li2003] introduced a hybrid approach for assessing the semantic similarity of words by suggesting that the similarity between two words \( w_1 \) and \( w_2 \) should be a function of the shortest path between them as well as the depth in the hierarchy of their least common ancestor and the semantic density of the two words: \( \text{sim}(w_1, w_2) = f_1\left( \text{path}(w_1, w_2) \right), f_2\left( \text{depth}(\text{lca}(w_1, w_2)) \right), f_3(\text{semantic_density}) \)\), where the transfer functions \( f_1, f_2 \) and \( f_3 \) model the independent contributions of the three factors.

Li and co-workers argued that the transfer functions should be non-linear and chose the following definitions:
\[ f_1(x) = e^{-\alpha x}, \quad f_2(x) = \frac{e^{\beta x} - e^{-\beta x}}{e^{\beta x} + e^{-\beta x}}, \quad f_3(x) = \frac{e^{\delta x} - e^{-\delta x}}{e^{\delta x} + e^{-\delta x}} \]

The local semantic density is defined as \( \max \{ \max_{s_1 \in \text{senses}(w_1)} \{ \max_{s_2 \in \text{senses}(w_2)} \{ \text{IC}(c) \} \} \), where \( \text{IC}(c) \) is calculated as \( -\log(p(c)) \). The authors experimented with various definitions for \( f \), such as \( f(x,y,z) = f_1(x) \ast f_2(y) \), \( f(x,y,z) = f_1(x) \ast f_2(y) + \lambda f_3(z) \) and \( f(x,y,z) = f_1(x) \ast f_2(y) \ast f_3(z) \). They concluded that in their experimental setting, Charles and Miller set of noun pairs from the WordNet taxonomy, the best performance was obtained taking \( f \) to be \( f_1 \ast f_2 \) and setting \( \alpha \), the parameter controlling the influence of the shortest path, to 0.2 and \( \beta \), the parameter adjusting the effect of the depth of the common ancestor, to 0.6. These results suggest that the semantic density of the least common ancestor may not have a major contribution to the similarity of two concepts.

### 4.8 A new hybrid similarity measure

In this section we present our proposal for a family of hybrid semantic similarity measures which combine information content and feature information for assessing the similarity between two concepts in different hierarchies [Rosu2009]. These measures employ a feature based similarity measure for “crossing the gap” between the two separate hierarchies and are formally defined as:

\[
\text{relatedness}(c_1, c_2) = \max_{b_1 \in O_1, b_2 \in O_2} \{ \text{sim}^{\text{inter ontology}}(c_1, b_1) \times \text{sim}^{\text{feature based}}(b_1, b_2) \times \text{sim}^{\text{intra ontology}}(b_2, c_2) \}
\]

Intuitively, given two concepts \( c_1 \) and \( c_2 \) from ontologies \( O_1 \) and \( O_2 \) respectively, their similarity is calculated as the strength of the path between them which may include two concepts \( b_1 \) and \( b_2 \) which provide a “bridge” between \( O_1 \) and \( O_2 \). The strength of this “bridge” is assessed using a feature based similarity measure and the strength of the intra-ontology segments is estimated with an information content based approach.

The definition of the measure allows the cross-ontology bridge to be at an arbitrary distance from the two concepts being compared. The two bridge heads are selected that maximize the overall strength of the path between the two concepts being compared.
We investigated the applicability of the *relatedness* family of measures to discovering associative relations between the three components of the Gene Ontology and we present our findings in the next chapter.

We also employed the relatedness measure to assessing the similarity of pairs of yeast genes which do not have annotations from the same ontology, e.g. one gene is annotated with terms from the Cellular Component and the Biological Process ontologies while the other has only Molecular Function annotations.

4.9 Discussion

Structural taxonomic information and shared information content are both associated with aspects of the similarity between ontological concepts most probably complementing each other in offering a more complete picture of similarity.

**Research contributions.** We proposed two intrinsic approaches to estimating the shared information content which do not rely on probabilistic assumptions and the availability of statistical data. In particular, our approaches can be used to assess the similarity of ontology concepts when corpus occurrence or annotation data are not available, so other information-content based methods cannot be applied. When occurrence information is available, we have also proposed to estimate the similarity between two concepts using the *relatedness ratio*, an estimation of the mutual information between the random variables representing the two terms being compared. We have also introduced a new hybrid measure, relatedness, which combines information based and feature based approaches to measure the similarity between two ontological concepts and can be particularly useful when comparing concepts in different ontologies.

**Partial evaluation.** We have performed an initial evaluation our proposals for new information content-based measures by calculating the similarity between all the pairs of terms in the Gene Ontology using a local copy of the Gene Ontology Annotation Database. The correlation coefficient between the rankings induced by our proposed measure, $IC^{leaves}$, and those induced by the measures proposed by: Resnik, Seco, Leacock and Chodorow, for all the terms in the
Gene Ontology, and over 15,000,000 pairs of yeast genes. $IC^{leaves}$ correlates well with all of them: e.g., on Cellular Component term pairs it has a Spearman correlation coefficient of 0.88 with Resnik’s measure and of 0.97 with Seco’s.

We have also investigated how our proposals to use the specificity of a concept to estimate its information content relate to existing measures by comparing the rankings induced over all pairs of terms in the Gene Ontology and also on the pairing of 5000 yeast genes. We found that both proposals correlate well with the other measures. The lowest Spearman correlation, 0.71, was measured between Resnik’s proposal and our proposal, $IC^{specificity}$, calculated for all descendents.

Our measures are more sensitive to the topography of the ontology graph, both when all types of edges are considered and also when only the $is\_a$ relation is taken into account. Applying our measures results in the smallest percentage of terms for which distinguishing between edges does not make a difference in the value of the information content, 61% and 62%, respectively, compared to 82% for Seco’s measure [Seco2004] and 82% for Resnik’s [Resnik1999] measure.

Our proposals also have more discriminating power, Table 4.1, assigning more distinct values for the information content of terms, across ontologies, than any of the other measures we compared them to. For example, they assigns 717, and 930 distinct values, respectively, to terms from the Biological Process ontology compared to 630 distinct values assigned by Resnik’s measure, which provides the next most diverse range of values. A direct consequence of this fact is that our proposal might be able to discriminate between different pairs of terms, genes, proteins, etc, when other measures will not be able to.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Distinct values</th>
<th>Min value</th>
<th>Max value</th>
</tr>
</thead>
<tbody>
<tr>
<td>information_content_leaves_all</td>
<td>293</td>
<td>0.00E+00</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>information_content_leaves_is_a</td>
<td>252</td>
<td>0.00E+00</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>information_content_resnik_all</td>
<td>786</td>
<td>0.00E+00</td>
<td>1.11E+01</td>
</tr>
<tr>
<td>information_content_resnik_is_a</td>
<td>694</td>
<td>0.00E+00</td>
<td>1.11E+01</td>
</tr>
<tr>
<td>information_content_seco_all</td>
<td>380</td>
<td>6.81E-02</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>information_content_seco_is_a</td>
<td>317</td>
<td>6.81E-02</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>information_content_specificity_descendents_all</td>
<td>1104</td>
<td>5.67E-06</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>information_content_specificity_descendents_is_a</td>
<td>940</td>
<td>5.65E-06</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>information_content_specificity_leaves_all</td>
<td>871</td>
<td>1.22E-05</td>
<td>1.00E+00</td>
</tr>
<tr>
<td>information_content_specificity_leaves_is_a</td>
<td>767</td>
<td>1.04E-05</td>
<td>1.00E+00</td>
</tr>
</tbody>
</table>

Table 4.1 Discriminating power of different Information Content Measures
We have calculated the correlation coefficient between the rankings induced by $IC^{leaves}$ and those induced by the measures proposed by: Resnik, Seco, Leacock and Chodorow, for both, all the terms in the Gene Ontology, and over 15 000 000 pairs of yeast genes. $IC^{leaves}$ correlates well with all of them: e.g on Cellular Component term pairs it has a Spearman Correlation coefficient of 0.88 with Resnik’s measure and of 0.97 with Seco’s.

We have also investigated how our proposals to use the specificity of a concept to estimate its information content relate to existing measures by comparing the rankings induced over all pairs of terms in the Gene Ontology and also on the pairing of 5000 yeast genes. We found that both proposals correlate well with the other measures. The lowest Spearman correlation, 0.71, was measured between Resnik’s proposal and $IC^{specificity}$ calculated for all descendents.

We conclude that our proposed similarity measures correlate well with existing measures in the literature and we advance the opinion that the characteristics of the task for which a similarity measure is used, and the topology of the ontology to which the measure is applied, matter the most to the choice of the method for calculating the shared information.

We have also investigated how our proposals compare with other measures in the literature via a task-based evaluation method: predicting the functional similarity of genes. We proposed a general integrative method, called Gene functional Association Predictor (GAP) which in its current incarnation integrates pathway information from many different public online databases (e.g., Reactome, KEGG, NetPath and NCI-Nature PID), Gene Ontology annotations as well as drug and disease associations mined from scientific literature. The predictive service is available on-line. We present the details of the method and the comparison with other existing predictive methods in Chapter 6.

Our proposal for a hybrid *relatedness* measure has shown very promising results in identifying relationships between separate hierarchies, which would have not been detected by lexical analysis or simple statistical analysis of co-occurrence. These relationships can be used as recommendations for improving the topology of ontologies such as the Gene Ontology and also for assessing the functional similarity of genes which do not have annotations from the same ontologies. We report on this project in Chapter 5.
5 Detecting Relationships between Concepts from Multiple Ontologies

Ontologies are increasingly important to a multitude of tasks in bioinformatics research, from predictive functional genomics to information retrieval and mediating between data sources in data integration engines. Among the many ontologies developed for the biomedical domain, Systematized Nomenclature of Medicine - Clinical Terms (SNOMED-CT)\textsuperscript{114} and the Gene Ontology (GO)\textsuperscript{115}, are the most widely used. Several dozen others are maintained by the Open Biomedical Ontology\textsuperscript{116} and Bioportal\textsuperscript{117}, and many more are being developed independently by various research groups around the world. Many bio-ontologies are relatively small in size, less than 1000 concepts, but some, such as SNOMED-CT and GO, have over 25,000 concepts.

As amply illustrated by the literature, the importance of the biomedical ontologies goes beyond that of gene and protein annotation vocabularies. Mainly through the assessment of similarity between sets of terms which are annotated to genes, proteins or documents of interest, they are central to a multitude of tasks, from predictive functional genomics [Stoica2006, Khatri2005, Bodenreider2005] to information retrieval and mediating between data sources in data integration engines [Bodenreider2008].

5.1 Motivation

The Gene Ontology, despite being often criticized for inconsistencies and for not adhering to formal ontology design principles [Smith2004-a, Smith2004-b], is nevertheless acquiring the status of a standard ontology across various biological domains.

\textsuperscript{114} Available at: http://www.ihtsdo.org/snomed-ct
\textsuperscript{115} Available at: http://www.geneontology.org
\textsuperscript{116} Available at: http://www.obofoundry.org
\textsuperscript{117} Available at: http://bioportal.bioontology.org/
GO consist of over 28,000 concepts connected via relationships of the following types: *is-a, part-of, regulates, positively_regulates*, and *negatively_regulates* and it is structured into three domain ontologies *molecular function, MF*, *biological process, BP*, and *cellular component, CC*.

The three constituents of the Gene Ontology (GO) were created as independent hierarchies and although a *biological process* is thought of as a series of events or molecular functions, until recently, there have been no associative relationships between terms indicating, for example, whether a *molecular function* is involved in a *biological process*.

In 2010 the Gene Ontology consortium started introducing *regulates* relationships between terms in the MF and BP ontologies and this initiative is ongoing. This decision came as a recognition of the necessity to make explicit some relatedness relationships in addition to the more common similarity, *is-a*, and parthood, *part-of*, relationships and to bridge the division between the two independently developed hierarchies. However, although relationships such as the one between *regulation of kinase activity* (BP) and *kinase activity* (MF) are made explicit with the introduction of the new *regulates* links, others, such as between *transcription* (BP) and *aryl hydrocarbon receptor binding* (MF), which should be linked, are not. Furthermore, this initiative does not include other relatedness relationships, such as *localization*: for example between *nucleus* (CC) and *DNA binding* (MF), or between *chromosome* (CC) and *sister chromatid biocondensation* (BP).

The availability of a richer set of links between the three constituent ontologies, BP, CC and MF, will bring several benefits:

1. **Eliminate the need for manual co-annotations with terms from all three sub-ontologies, help analyze existing annotations, and infer new knowledge.**

   For example, a gene annotated to a molecular function term *mf* can also be annotated to the biological process term related to *mf*.

2. **Provide a means to compare annotations from different hierarchies.**

---

118 According to the Gene Ontology documentation available at: http://www.geneontology.org/
Many yeast genes, and their products, are only annotated to either terms from the MF taxonomy or the BP taxonomy, but not both, and assessing their pair-wise semantic similarity depends on the availability of a method for comparing the relatedness of their annotations.

(3) assist in the semantic integration of biomedical databases.

The explicit relationships between molecular function and biological process terms can help integrate the Gene Ontology Annotation Database with pathway databases, such as REACTOME and KEGG, who describe pathways at different representational granularities.

(4) aid in the interpretation of the regulatory network induced by the regulates relationships;

If a function/process modulates the activity of one of the functions that are part of a process, then it can be inferred that it also regulates the entire process. However, the regulatory relationships are difficult to define in biology and there is still debate on whether some regulatory steps, such as feedback inhibition, are part of the biological process they are regulating. For example, isoleucine, the end product of a pathway transforming threonine into isoleucine, will, in excess amounts, switch off its own production by attaching to the inhibition site of the first enzyme in the pathway, threonine deaminase, thus down-regulating the pathway. This mechanism is illustrated in Figure 5.1.

(5) assist in the formulation of scientific hypotheses construction.

For instance, if a gene product $p$ is annotated to a molecular function $mf$ whose subtypes are related to biological processes $bp1$, and $bp2$, we can automatically formulate the hypothesis that $p$ may be involved in $bp1$, and/or $bp2$, and check this hypothesis against various data sources such as phenotype and gene expression data.
help answer queries that require associative information, such as “Which kinases are expressed in chloroplast?”.

In this work we present an approach for relating terms in different Gene Ontology hierarchies which takes advantage of the available gene annotation data, via an inter-ontological associative measure, as well the information encoded in the ontology hierarchy.

5.2 Methods

**Relationship discovery strategy.** Our technique for automatically discovering relationships between terms in different hierarchies is built around the family of hybrid relatedness measures we defined in Section 4.11.

\[
\text{relatedness}(c^{O1}, c^{O2}) = \max_{b^{O1}, b^{O2}} \{ \text{sim}_{\text{intra ontology}}(c^{O1}, b^{O1}) \times \text{bridge strength}(b^{O1}, b^{O2}) \times \text{sim}_{\text{intra ontology}}(b^{O2}, c^{O2}) \}
\]

---

119 Image credit: Enzymes - Biology Encyclopedia - cells, body, function, human, process, system, different, DNA http://www.biologyreference.com/Dn-Ep/Enzymes.html#ixzz1oI4kFUq
where \( c^{O_1} \) and \( c^{O_2} \) are the two terms being compared and \( b^{O_1} \) and \( b^{O_2} \) are the terms acting as *bridge heads* for a link between ontologies \( O_1 \) and \( O_2 \). Intuitively, we calculate the relatedness of \( c^{O_1} \) and \( c^{O_2} \) as the *strength of the path* between them. The strength of the *bridge* between the two ontologies is estimated using a feature-based similarity measure, \( \text{sim}_{\text{feature based}}(b_1, b_2) \), and the strength of the intra-ontology paths between \( c^{O_i} \) and \( b^{O_i} \), \( i \in \{1, 2\} \), is estimated using an information content similarity measure.

The two bridge heads, \( b^{O_1} \) and \( b^{O_2} \), are selected so that they maximize the overall strength of the path between the two concepts being compared, which is a very costly computational procedure that requires checking all possible bridges between the two ontologies. In order to reduce the computational effort, in semi-automatic implementations the initial bridges can be provided by the users, while in fully automatic approaches, heuristics can be used to prune the search space. These heuristics can employ methods such as lexical analysis of term names, feature-based distances as well prior information about the ontology terms, such as statistical correlation data.

The mappings selected for manual curation have been generated using our proposal for intrinsic information content measures

\[
(5.2) \quad \text{specificity}_{\text{subconcept}}(c) = \frac{\text{depth}(c)}{\max_{\text{values}(c)} \{\text{concepts}(v)\}} \times \frac{\text{all_concepts}}{\text{descendants}(c)}
\]

Our current implementation of the relatedness measures uses a heuristic which combines three well-known feature-based measures, cosine, Jaccard and Dice, to identify potential bridges between the three component hierarchies of the Gene Ontology. The strength of the association between the two terms that constitute a bridge is calculated as follows:

\[
(5.3) \quad \text{bridge_strength}(b_1, b_2) = \frac{\cos ine(b_1, b_2) + \text{Dice}(b_1, b_2) + \text{Jaccard}(b_1, b_2)}{3}
\]

**Materials.** We have calculated the relatedness between all the pairs of terms in the three GO hierarchies using a locally stored version of the Gene Ontology Annotation Database. In our prototype system, the semantic bridges between the three GO ontologies, as well as the intra-ontology semantic similarity scores for all pairs of GO terms have been pre-computed using
custom SQL scripts and stored in tables in the local database. The similarity scores range between 0 and 1 and have been calculated with a precision of two digits after the decimal point.

### 5.3 Results and discussion

The biomedical domain lacks generally accepted gold standards, i.e., human rated, evaluation datasets for semantic similarity, like the well known Miller and Charles (30 pairs of terms) [Miller1995] and Rubenstein and Goodenough (120 pairs of terms) [Rubenstein1965] benchmarks used to evaluate semantic similarity measures in natural language processing and information retrieval. Thus, we have opted to ask two biologists to assist with the evaluation of our relatedness measures.

**Research contributions.** We explored the applicability of our relatedness measures to finding mappings between concepts in different ontologies. We applied our measures to all pairs of terms \((b, m)\) where \(b\) is a biological process and \(m\) is a molecular function and submitted the top scoring pairs to the two experts for review. Each expert was presented with a list of pairs of GO concept names and had access to any other helpful information from the literature and the ontology itself. The experts were instructed to classify all relationships as either true or false. We calculated *correctness* as the ratio of true alignments to the total number of alignments considered, in a similar manner as precision and sensitivity. The inter-rater agreement was 78%.

The bridges between the Molecular Function and Biological Process hierarchies have been calculated using the inter-ontology measure in equation 5.3, and as intra-ontology measures we have used our two proposals for information content-based measures, *specificity leaves* and *specificity descendents*. We are going to refer to them in what follows as \(relatedness_{sl}\), and \(relatedness_{sd}\), respectively.

We ranked the (MF, BP) term pairs using each of the two measures and selected the top-10 ranked pairs, including tied pairs, 75 and 84 pairs, respectively, for our experts to review. The two biologists reviewed the mappings individually and scored each of them as either valid or invalid. This is in line with other evaluation approaches in the literature, e.g., [Al-Mubaid2009]...
and [Pedersen2007], which used 66 and 30 human-rated pairs of terms, respectively, for testing the performance of semantic similarity measures.

As depicted in Figure 5.2 the correctness of the \textit{relatedness}^{sl} measure is 100\% for all the pairs with rank 5 or above, and degrades gracefully to 75\% when all top-10 ranked pairs, 75 pairs in total, are considered.

![Figure 5.2 Correctness of the alignments between MF and BP terms, relatedness^{sl}](image)

\textit{Relatedness}^{sd} shows a similar performance, Figure 5.3. Its correctness is 79\% for the top-5 ranked pairs and decreases to 74\% for the top-10 ranked pairs.

![Figure 5.3 Correctness of the alignments between MF and BP terms, relatedness^{dl}](image)
We get a comparable performance when ranking the pairs of terms according to the average of \( relatedness^{sl} \) and \( relatedness^{sd} \): 100% correctness for the pairs ranked 5 or above, and a decrease to 76% correctly aligned pairs from the top-10, 70 pairs in total, Figure 5.4.

![Figure 5.4 Correctness of the alignments between MF and BP terms, average relatedness](image)

We have also assessed the performance of our measures when the evaluation dataset is expanded to the top-scoring 250 pairs. The precision of the \( relatedness^{sl} \) measure degrades gracefully from 100% for the top pairs to 71% for the whole set of 250 pairs, while the precision of the \( relatedness^{sd} \) measure goes from 100% for the top 10 pairs to 77% for the top 100 pairs and down to 71% overall precision. Ranking the pairs of terms according to the average of \( relatedness^{sl} \) and \( relatedness^{sd} \) shows a very similar performance, 100% precision for the first 10 pairs, 77% for the top 100 pairs and 71% for the top 250.

Our analysis has shown that the \( relatedness \) measure we are proposing is able to find term relationships that are not covered by \textit{regulates} relationships recently introduced in the Gene Ontology, such as between glycolipid transport and glycolipid transporter activity, \textit{alpha-glucoside transport} and \textit{maltose transmembrane transporter activity}. In fact, approximately 60%...
of the true positives in our evaluation dataset denote relations that are not regulatory in nature. The direct implications are two fold: (1) the cross-ontology relationships detected should be submitted to the Gene Ontology consortium for consideration to be included in future versions of GO, and (2) there is an opportunity to augment the portfolio of relationships types currently used for representing genomic knowledge.

Moreover, our measure is also able to detect relationships that would not be detected by the lexical methods, which are the most widely used ontology mapping strategies. Lexical methods rely on comparing the names of the terms to be related and thus would be expected to identify, for example, that Golgi organization and biogenesis (MF) is related to the Golgi apparatus (CC) and that vacuolar protein processing (BP) happens in a vacuole (CC), but cannot be counted upon to detect relationships between terms whose names share no lexical similarity. Our measure, however, provides a complementary approach, which takes advantage of gene annotation data, as well as the similarity information encoded in an ontology in order to detect relationships that are beyond the grasp of lexical methods, e.g., between lipoic acid metabolic process and bile acid binding, and between betaine transport and quaternary ammonium group: hydrogen symporter activity.

Last but not least, an important product of the work reported here is a gold-standard data set of 250 relationships between molecular function and biological process terms that have been examined by two experts. This data set contains 175 true positive relationships and 75 true negative, i.e., unrelated, pairs of terms [Rosu2012].

**Challenges.**

We have also looked at potential sources for incorrect mappings and we identified that similarities between some of the participants in various molecular functions and biological processes leads to the production of unwarranted mappings, e.g., between phenylalanine transport and high-affinity tryptophan transmembrane transporter activity. Both phenylalanine and tryptophan are amino acids and have some common transporters, such as AroP, a general aromatic amino acid permease (which is responsible for the transport of all three aromatic amino acids: phenylalanine, tyrosine, and tryptophan, across the inner membrane [Cosgriff1997]), but
they do not have identical transport mechanism. In particular, the high-affinity tryptophan transmembrane transporter activity exhibits 100-fold higher affinity for tryptophan than for other amino acids. [Seymour2006].

Other false positive findings can be explained by the fact that our measure does not distinguish between negative and positive regulation subclasses of terms referring to the regulation of molecular functions and biological processes. For example, the term guanylate cyclase activator activity, which refers to increasing the activity of guanylate cyclase, an enzyme that synthesizes cGMP from GTP in response to calcium levels, is linked with equal confidence to both negative regulation of cGMP biosynthetic process and to positive regulation of cGMP biosynthetic process, although the correct link is only to the latter.

We list here several issues that may affect generating and evaluating links between molecular function and biological process terms.

(1) A molecular function must be part of a biological process, but it can be part of many processes, not just one. Also, in some instances, a molecular function can be part of a given biological process only sometimes. These constraints however are difficult to model according to the principles adopted by the GO Consortium, since once a part_of relationship is established between a term A and a term B, the implication is that an instance/occurrence of A will always be part of an occurrence/instance of B. Similarly, a has part relationship between a biological process and a related molecular function does not always hold. For example, arginosuccinate synthase occurs in both urea cycle and polyamine biosynthesis, but a gene product can be an agent for arginosuccinate synthase activity only if this function occurs as part of a polyamine biosynthesis.

(2) Many molecular function terms do not really fit the stated definition of “elemental activities describing the actions of a gene product at the molecular level”. For example, the term ent-kaurene oxidation to kaurenoic acid, initially part of the Molecular Function hierarchy, has been moved to Biological Process as it represents three successive reactions, a.k.a. “The three successive oxidations of the 4-methyl group of ent-kaurene to form ent-kaur-16-en-19-oate, kaurenoic acid. This process may be carried out entirely by
the enzyme ent-kaurene oxidase.” Each of these reactions is now properly described by the subsequently added MF terms: ent-kaurene oxidase activity, ent-kaur-16-en-19-ol oxidase activity, and ent-kaur-16-en-19-al oxidase activity. The links ascertaining that each of these molecular function terms is part of the ent-kaurene oxidation to kaurenoic acid process do not yet exist in the ontology.

(3) Some Molecular Function and Biological Process terms describe the same action, e.g histone H3-K4 methylation (BP), also known as histone lysine H3 K4 methylation, denotes “The modification of histone H3 by addition of a methyl group to lysine at position 4 of the histone.” While histone methyltransferase activity (H3-K4 specific) (MF), also known as histone methylase activity (H3-K4 specific) denotes the “Catalysis of the reaction: S-adenosyl-L-methionine + histone H3 L-lysine (position 4) = S-adenosyl-L-homocysteine + histone H3 N6-methyl-L-lysine (position 4). This reaction is the addition of a methyl group onto lysine at position 4 of the histone H3 protein.”

(4) Some functions are not clear steps in a biological process: e.g. receptor activity (MF), “Combining with an extracellular or intracellular messenger to initiate a change in cell activity”, describes both a binding activity as well as the regulation of a cellular process.

(5) There are structural inconsistencies/omissions in the current organization of GO, e.g. in the BP hierarchy, glucose-6-phosphate transmembrane transport is specified to be a type of hexose phosphate transporter as well as a type of glucose transmembrane transport, while in the MF hierarchy in the MF hierarchy, glucose-6-phosphate transmembrane transporter activity is described only as a type of hexose phosphate transmembrane transporter activity Figure 5.5
Figure 5.5 Example of structural inconsistency in GO.

(6) In addition, there are issues related to the accuracy of annotations. For example, if several gene products are flagged by an experiment as being implicated in the locomotion process, a curator may choose to annotate them all to “locomotion”, or decide to annotate the gene products that are known to be transcription factors to the “regulation of locomotion”, instead. The distinction between the regulation of a biological process and the process itself is still unclear, as the community has not yet reached a consensus.

Limitations. One limitation of automatic methods for discovering relationships between ontology terms is the fact that many, if not most, of the relationships detected are indirect. Automatic methods, such as ours, also do not differentiate between various relationship types. Future work should attempt to distinguish between the various types of relationships between ontological concepts and also label the direct and indirect relationships.

Our relatedness measures have shown promising results in identifying alignments between concepts in different hierarchies of the Gene Ontology, however, as they are heavily dependent on the availability and quality of annotation data, we envision their use as a supplement rather than a replacement for the efforts to manually construct mappings between ontological terms.
Many different elements e.g., DNA, protein, RNA (coding and non-coding) associate and cooperate to form diverse networks that make up the functional machinery of a normal living cell: the interactome. These functional associations can be defined as physical binding, such as protein-protein or protein-DNA interactions, or may refer to a group of genes and/or proteins involved in a signaling pathway, or encompass genetic or phenotypic associations between candidate genes in a disease process.

Our knowledge of these functional associations is, however, still very limited. Considering protein-protein interactions (PPIs), the experimental coverage of the human proteome is at least one order of magnitude lower than the true proteome. Other forms of interactome data, such as protein-DNA or microRNA-mRNA interactions, lag behind PPIs both in coverage and curation, although there have been recent efforts to compile such known interactions into databases (e.g., hmChIP [hmChIP]).

Improving the coverage and connectivity of the interactome requires new experimental data; however, empirical methods, particularly high-throughput technologies, have a number of practical limitations and biases that cause the generation of false positives and false negatives. These can arise due to the nature and sensitivity of current methods, whereby stable interactions are more likely to be detected than weak or transient ones, and interactions among abundant proteins are more often detected than those with low copy number. Additionally, interactions can be condition-specific, adding yet another layer of complexity to the issue of false discovery [Wodak2009, Chen2003].

In silico methods, despite their own limitations, can provide a complementary accompaniment to experimental methods: they are useful to conduct data quality control such as reliability assessment and validation [Bader2004, Lin2009], facilitate knowledge exploration for deeper understanding of cellular mechanisms, through an integrative analysis of biological data from different sources such as protein/DNA sequences, gene expression, and pathways [Rhodes2005,
Reactome], and effectively reduce noise when combining predicted interactions with experimental data [You2010, Tong2002].

Many computational tools have historically concentrated on the prediction of protein-protein interactions; however, the prediction of gene functional association has a far broader application for the interactome since it can incorporate all forms of interaction (e.g., protein-DNA, microRNA-protein, pathway co-membership) involved in a biological process, pathway or disease. Such data can be useful in the elucidation of cellular pathways, and can help direct and prioritize future experimental hypotheses that best fill the current gaps in our knowledge.

We consider here the problem of deciding if two genes are functionally associated, and we propose an integrative method, called Gene functional Association Predictor (GAP) whose purpose is to provide a quantitative measure of gene functional relatedness. GAP improves on existing integrative methods by including a novel semantic similarity component to better take advantage of the available gene annotations. The proposed semantic mechanism of inferring gene associations is capable of processing implicit association evidence in order to identify and prioritize novel predictions of genes' functional relationships.

The current version of GAP integrates, in a principled way, pathway information from many different public online databases (e.g., Reactome, KEGG, NetPath and NCI-Nature PID), Gene Ontology annotations as well as drug and disease associations mined from scientific literature. GAP's semantic similarity measure is, however, general. It can take advantage of any type of biological data source, and can be extended to different organisms (provided that the employed databases and services cover multi-organism information). GAP

---

120 The nature of novel functional associations can be, in general, interpreted by the user via analysis of the corresponding association evidence, and should be eventually validated experimentally.

121 Available at: http://www.reactome.org/ReactomeGWT/entrypoint.html

122 Available at: http://www.genome.jp/kegg/

123 Available at: http://www.netpath.org/

124 Available at: http://pid.nci.nih.gov/index.shtml
predicts interactions among 19,027 human protein-coding genes; the gene names and symbols are those provided by the HUGO Gene Nomenclature Committee (HGNC\textsuperscript{125}).

Several research groups have already studied the potential of computational methods to combine multiple information sources to build different datasets and/or web servers of predicted protein/gene associations (e.g., I2D \textsuperscript{12D}, PIP \textsuperscript{PIP}, PIPs \textsuperscript{PIPs}, PPIFinder \textsuperscript{PPIFinder}, GeneMANIA \textsuperscript{GeneMANIA}, STRING \textsuperscript{STRING}). These services predict likely direct and indirect associations based on conserved sequence motifs, gene fusion events, gene co-expression, orthology, and pathway co-annotation. Some of these databases also incorporate evidence obtained by text mining of the scientific literature. Unlike text mining modules commonly used in other interaction predictors which simply look for the co-occurrence of genes in scientific abstracts (PubMed), GAP incorporates information provided by external and well-established semantic search engines (e.g., GoPubMed), and therefore, benefits from well-developed, sophisticated natural language processing techniques to acquire more comprehensive and accurate information about the genes of interest. We have reviewed a wide array of existing protein interaction predictors, and chose those that follow the same objectives\textsuperscript{126} as GAP's for an in-depth performance comparison.

We compared the predictive power of the selected methods with that of GAP over a gold standard dataset of experimentally validated direct and indirect interactions. We showed that GAP has a superior positive predictive value (i.e. precision rate), and specificity vs. sensitivity (i.e., true positive vs. true negative rates) in identifying known interactions as compared to the existing interaction predictors. In support of the novel potential interactions predicted by GAP, we selected a set of highly-scored novel interactions (i.e., not currently found in any known databases), and then manually searched the scientific literature for corroborating information. We also provided further supporting evidence for GAP's predictions from an external expert curated gene-disease association database.

\textsuperscript{125} The HUGO Gene Nomenclature Committee, http://www.genenames.org/

\textsuperscript{126} They look for functional associations (i.e., not specifically designed for predicting physical interaction) among human genes, and offer web server or make their predicted interactome available for download
Before moving further, we introduce a few notations. We denote by \( G = \{g_1, \ldots, g_N\} \) the set of \( N \) genes under consideration. We associate with each gene \( g_i \), \( d \) feature-sets, \(<F_1^i \ldots F_d^i>\), where each feature-set \( F_k^i \) is a subset of \( F_k = \{t_{k1}, \ldots, t_{kn}\} \), the set of all possible values the \( k^{th} \) feature can take. Prior to defining a measure of similarity between two genes (and their corresponding proteins), we need to define similarity measures between the feature-sets corresponding to these genes. Furthermore, in order to assess the similitude between corresponding feature-sets, we need to define a strategy to evaluate the similarity between the individual terms, i.e., feature values, contained in the sets being compared.

Each type of information denotes an attribute/feature describing a gene product. More specifically, each member \( g_i \) of our set \( \{g_1, \ldots, g_N\} \) of human gene products is associated with \( d \) attribute sets \(<A_{i1}, \ldots, A_{id}>\), where each attribute set \( A_{ik} \) is a subset of all the possible values that attribute \( A_k \) can take.

We conjecture that the propensity of two gene products \( g_i \) and \( g_j \) to interact is proportional to their similarity, which estimate with the aid of a similarity measure defined between the attribute sets \(<A_{il}, \ldots, A_{id}>\) and \(<A_{jl}, \ldots, A_{jd}>\)-associated with the two gene products. Given the similarity \( \delta_k(g_i, g_j) \) between the sets \( A_{ik} \) and \( A_{jk} \), we can calculate the similarity \( \delta(g_i, g_j) \) between each pair \((g_i, g_j)\) of gene products (1) as a scalar value, by combining the per-attribute similarities using aggregation approaches such as the weighted sum, \( \delta(g_i, g_j) = \sum_{k=1}^{d} w_k \delta_k(g_i, g_j) \), weighted mean, geometric mean, etc. or (2) as a \( d \)-dimensional similarity vector of per-attribute similarities \( \tilde{\delta}(g_i, g_j) = \langle \delta_1(g_i, g_j), \ldots, \delta_d(g_i, g_j) \rangle \). The scalar aggregation approaches generally require that the per-attribute similarity scores are in the same numerical range/scale, so a suitable scaling technique needs to be applied prior to combining the per-attribute scores. It should also be noted that some scalar aggregation approaches, e.g., the harmonic mean and the geometric mean, are not applicable when some of the per-attribute scores are allowed to be 0. Vectorial approaches, on the other hand, do not require the use of a common scale for the per-attribute scores, but are more difficult to handle and interpret, especially if the dimensionality of the vector space precludes an intuitive graphical representation. Selecting appropriate thresholds for interpreting the score vectors as indicative of high or low similarity is also more complex than for scalar aggregation approaches, as
Feature-set Similarity

As mentioned earlier, each gene \( g_i \) is associated with \( d \) feature-sets \(<A_{i1}, \ldots, A_{id}>\). In our framework, estimating the similarity between a gene pair \((g_i, g_j)\), requires calculating first the similarity between the respective feature-sets of \( g_i \) and \( g_j \).

Let \( \delta_k(g_i, g_j) \) be the feature-set similarity function which assesses the similarity between \( F_{ik} \) and \( F_{jk} \), the \( k \)th feature-sets of \( g_i \) and \( g_j \), respectively. We define our feature-set similarity function as an aggregation of the similarity scores of the terms contained in feature-sets as follows:

\[
\delta_k(g_i, g_j) = \frac{1}{|A_{ik} \ast A_{jk}|} \sum_{a_k \in A_{ik}, a_j \in A_{jk}} \delta_k(a_{ik} , a_{jk})
\]

and

\[
\delta_k(g_i, g_j) = \max \{ \delta_k(a_{ik} , a_{jk}) \}
\]

Gene Similarity

The similarity scores for each gene feature set can be combined in several different ways in order to obtain a similarity measure for pairs of genes. The weighted sum is a popular aggregation operator which provides a natural way of combining the similarity assessed according to the chosen axes of comparison. We define \( \delta : G \times G \rightarrow R\), the similarity measure for pairs of genes in \( G \), as the weighted sum of the gene feature scores:

\[
\delta(g_i, g_j) = \sum_{k=1}^{d} \delta_k(g_i, g_j)
\]

where \( w_k \) denotes the weight assigned to feature \( F_k \), and \( \delta_k(g_i, g_j) \) is the similarity measure associated to feature \( F_k \). A weighted sum of (scalar) similarity scores has the advantage of being easy to comprehend and calculate, but requires that the scores be on the same numerical scale (range). Recruiting a suitable technique for transforming the location and scale parameters of the similarity scores distributions is therefore essential to the meaningful aggregation of the similarity values obtained for each comparison feature. Generally, we require the transformations of the scores to be monotonic.

---

127 The effect of the transformation can be instead incorporated in deriving the weights assigned to each feature, i.e., \( w_k \)
When the distribution of the similarity values is known, several parametric normalization approaches can be applied, e.g. the Z-score standardization for Gaussian distributions, in which the population mean is subtracted from the individual raw scores and then the difference is divided by the population standard deviation. When the distribution of the similarity scores is not known, non-parametric methods, such as quantile normalization, can be used. One of the simplest, yet most commonly used transformations is *uniform scaling* which maps the scores to be aggregated to values in the interval $[0, 1]$ via the linear transformation

$$\text{raw\_score} \rightarrow \frac{\text{raw\_score} - \text{minimum\_score}}{\text{maximum\_score} - \text{minimum\_score}}.$$

We adopted two different strategies for merging feature similarities into a scalar gene similarity value: (1) *decile-only* aggregation in which each feature similarity score is mapped to an integer $h$ in $[1,10]$ such that the score is at the $h^{th}$ decile of the sample, and (2) *decile-weighted* aggregation which is a weighted sum calculation where the weights are decile values as described in decile-only method, and the scores are the $(0, 1]$-scaled feature similarity measures.

### 6.1 Results

Similar to the other protein interaction prediction servers, GAP can be queried with a gene (protein) and gives back a ranked list of related genes (proteins), and their corresponding interaction/association scores and evidences. The returned list of genes can be dichotomized into interacting and non-interacting groups using any threshold-setting techniques. Accordingly, GAP can be thought of as a binary classifier, and can thus, be evaluated using typical classifier evaluation measures (e.g., F1-score, AUC, and precision vs. recall curve) which rely on a "gold standard" of experimentally validated direct and indirect interactions. Such performance measures are useful for assessing GAP's predictive power over the dataset of known interactions, but, they provide no information for evaluating the predicted novel gene/protein associations. Even though, novel interactions can be validated experimentally, providing some external evidence on the strength of the associations can further support the validity of the interactions.

We first focus on evaluating the GAP predictive power over an inclusive set of known direct/indirect interactions, and then, provide supporting evidence on the putative novel
interactions. However, as the selected features and their corresponding sources are identical throughout all the experiments, before proceeding to discussing the performance evaluation, we first provide an overview of the chosen feature sources.

**Selected Features and their Sources**

As previously mentioned, GAP proposes a general integrative framework for gene association prediction, and can employ any type of biological data sources for extracting gene/protein features. We have extracted several features from two major sources of scientific texts, and online biological databases. The selected features and their corresponding sources are summarized in Table 6.1, and described in detail in the coming sections.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Feature Source</th>
<th>Feature Name</th>
<th>Coverage (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted by text mining tools</td>
<td>GOPubMed</td>
<td>GOPubMed-related-gene</td>
<td>75.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GOPubMed-related-Go</td>
<td>74.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>GOPubMed-related-disease</td>
<td>10.26</td>
</tr>
<tr>
<td>FACTA</td>
<td>FACTA-related-gene</td>
<td>71.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FACTA-related-disease</td>
<td>67.41</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FACTA-related-drug</td>
<td>42.85</td>
<td></td>
</tr>
<tr>
<td>Extracted from online databases</td>
<td>Pathway databases: KEGG, Reactome, NetPath, NCI-Nature PID related-pathway</td>
<td>34.09</td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.1 Information on Selected features.** Last column shows the coverage of individual features, i.e. 100% coverage gives the fraction of the 19K protein-coding human genes for whom the corresponding feature set is empty.

**Text-based feature sources**

To extract text-based features, we made use of two online freely available scientific text search engines namely FACTA and GoPubMed as described below.

**FACTA** (Finding Associated Concepts with Text Analysis) is a real-time text-mining system which processes MEDLINE abstracts for finding and visualizing direct and indirect associations between biomedical concepts. Given a query gene, FACTA returns related biomedical concepts (e.g. genes, diseases, and drugs), and presents them in a tabular format ranked based on co-
occurrence statistics. The concept IDs and their names or synonyms are collected from several biomedical databases such as UniProt, BioThesaurus, UMLS, and DrugBank. FACTA has been developed by the National Center for Text Mining (NaCTeM), University of Manchester, UK. We queried FACTA with all human protein-coding genes, and collected the returned related genes, diseases, and drugs. We accordingly assigned to each human gene a set of related gene IDs

**GoPubMed** [GoPubMed] is a *semantic* search engine that allows for searching PubMed documents with the help of the Gene Ontology (GO) terms and Medical Subjects Headings (MeSH). Beyond keywords, GoPubMed uses novel algorithms and state-of-the art natural language processing to provide a more comprehensive and accurate search. GoPubMed also provides the platform technology for GoDiseases which is basically used in this study. GoDiseases supports functional annotation of gene products by systematically linking genes to processes, functions, diseases, etc. GoPubMed is developed by Transinsight group which collaborates closely with the bioinformatics group of TU Dresden, Germany.

We queried GoPubMed with all human protein-coding genes and stored for each gene, the gene IDs, MeSH terms, and GO terms which were found by GoPubMed to be related to the gene of interest.

**Online biological databases used for feature extraction**

The current version of GAP incorporates information from different curated biological pathway data-sources including Reactome, KEGG, NetPath, and NCI-Nature Pathway Interaction Database. Pathways are an essential feature for discovering functionally interacting proteins as proteins co-concurring in similar metabolic pathways are more likely to be expressed together and have similar phylogenetic profiles. Accordingly, using the aforementioned pathway data-

---

128 The retrieved gene IDs represent the genes which are co-occurring with the query gene in MEDLINE abstracts. This feature provides transitive evidence for gene relationship, UMLS and DrugBank terms.
sources, we assign to each human protein-coding gene the set of all pathways in which the queried protein is a member.

It should be noted that GAP can be extended via minor modifications to its core similarity measures, to incorporate other types of information about genes and proteins, and it is in our future plans to integrate more biological knowledge (e.g., gene expression, sequence similarity, etc.) from online data sources into the GAP predictive feature set.

6.1.1 Assessment of GAP’ predictive capabilities - overlap with known gene associations

Positive Examples GAP is not specifically designed to capture direct protein interactions. We thus need to build up a positive gold standard consisting of both direct (i.e. physical) and indirect interactions (e.g. co-complex relationship and pathway co-membership). We sourced physical interactions from I2D (Interologous Interaction Database), an on-line database of known (and predicted) mammalian protein-protein interactions which collects experimentally known human protein interactions from many of the commonly used human PPI databases such as BioGrid129, BIND130, HPRD131, IntAct132, MINT133, etc. Co-complex protein pairs are acquired from CORUM (the comprehensive resource of mammalian protein complexes). The CORUM134 database is hand curated based on evidence derived from a variety of experimental techniques and does not include information from high-throughput datasets. Pathway relationships are extracted from the following online databases: Reactome, KEGG, NetPath, and NCI-Nature Pathway Interaction Database.

129 Available at: http://thebiogrid.org/
130 Available at: http://baderlab.org/BINDTranslation
131 Available at: http://www.hprd.org/
132 Available at: http://www.ebi.ac.uk/intact/
133 Available at: http://mint.bio.uniroma2.it/mint/Welcome.do
134 Available at: http://mips.helmholtz-muenchen.de/genre/proj/corum
Table 6.2 lists the number of protein pairs and the total number of proteins in each of the three types of protein interactions (i.e., physical interaction, co-pathway, and co-complex).

<table>
<thead>
<tr>
<th>Interaction Type</th>
<th>No. of proteins</th>
<th>No. of protein pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Interaction</td>
<td>13320</td>
<td>94431</td>
</tr>
<tr>
<td>Co-pathway</td>
<td>7170</td>
<td>1332037</td>
</tr>
<tr>
<td>Co-complex</td>
<td>2627</td>
<td>35062</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>14794</strong></td>
<td><strong>1432937</strong></td>
</tr>
</tbody>
</table>

Table 6.2 Number of protein and total number of related protein pairs contained in each data source

Figure 6.1 depicts the overlap of proteins and protein pairs between these three sets: 98% of co-pathway pairs, 56% of co-complex pairs, and 84% of physical interactions are distinct (i.e., do not overlap with other sets). This low overlap exemplifies the importance of including different types of interactions in the gold standard when evaluating predictors of gene or protein associations rather than physical interactions.\(^\text{135}\)

\(^{135}\) Unfortunately, most of the gene/protein association predictors in the literature are mistakenly treated as direct (physical) PPI predictors, and therefore, use incomplete or inappropriate gold standard databases.
Negative Examples. Unlike positive interactions, confirmed reports of non-interacting protein pairs are rarely available, especially for indirect PPI predictions. There exist collections known as the Negatome, consisting of pairs of proteins which are unlikely to be engaged in direct physical interactions. While Negatome [Negatome] is complementary to random datasets for training direct protein interaction prediction algorithms, it is not suitable candidates for the prediction of co-complex relationships or pathway co-memberships. Many of the protein pairs contained in the Negatome are derived either by retrieving not directly interacting pairs out of protein complexes with known 3D structures, or by including some pathway databases which are filtered against known directly interacting proteins. Since many of the Negatome protein pairs are indirectly interacting, this dataset is unsuitable to serve as a negative sample for the purpose of this study.

Most common approach to construct negative “gold-standards” is to randomly pair proteins having different cellular localization, or being involved in different biological processes. Such an approach, however, is prone to an over-optimistic estimation of the predictor performance due to the biased selection of negative examples. Furthermore, restricting negative data only to pairs of
proteins localized in different cellular compartments is an unsuitable gold standard choice for the
evaluation of predicted indirect interactions as indirectly interacting proteins such as those who
belong to the same pathway can be found in different cellular compartments (e.g., signal
transduction pathways). Even in the case of direct interactions, the fact that interacting protein
pairs have to be in the same place and time does not imply that all proteins in the same
compartment will be interacting with each other. In a nutshell, such negative datasets are an
appropriate choice for the task of protein co-localization prediction, rather than the prediction of
protein interactions. Eventually, in order to minimize the biased distribution of negative
examples, we opted for an unconstraint random sample of protein pairs. This selection is
justifiable according to the assumption often made in the literature that the fraction of interacting
proteins is small comparing to the total number of potential protein pairs. Therefore, the
possibility of including truly (unknown) interacting pairs among a random sample is low enough
for our approach to yield a fairly accurate test dataset. Using the set of ~ 19,000 human protein-
coding genes curated and approved by HGNC, we constructed the negative set by randomly
sampling 1,500,000 protein pairs out of the ~ 200,000,000 potential protein pairs.

**Performance evaluation measures**

To evaluate GAP performance, we employed three commonly used measures, namely F1-score,
precision versus recall curves, and area under the Receiver Operator Characteristic (ROC) curve
scores.

F1-score is a performance measure which combines precision and recall values into a single
score. In our context, for each query gene, precision refers to the fraction of retrieved interacting
partners which are truly interacting with the given gene. Recall, on the other hand, measures the
fraction of the known interacting partners of the queried gene which has been retrieved by the
PPI prediction tool. F1-score is the harmonic mean of precision and recall and ranges between 0
and 1. We used F1-score to compare GAP performance at different configuration settings
because the precision versus recall graph would be unreadable due to the large number of
comparing curves. Once the best performing configuration has been identified using F1-scores,
the precision versus recall curve is used instead for the subsequent performance comparisons.

**Precision vs. recall curve (PR curve)**
Although easy to calculate, F1-scores have certain limitations. They are not sensitive to the ranked order of the retrieved interactions as they are computed using the unordered set of retrieved interactions. As such, the rank (score) of the truly interacting protein pairs does not affect F1-scores, which is not appropriate as in general, users are more interested in the high-scored predicted interactions, and expect “true positives” to appear at the top of the ranked list of predicted interactions. To remedy this, we used precision versus recall curves which plot the precision values at every recall point. Given a query gene and a ranked list of predicted interacting partners, a PR curve is constructed by traversing down the list and plotting the precision value whenever arriving at a new recall point. In general, predictor A is assumed to be better than predictor B if, at every recall point, A's precision value is higher than B's.

The precision versus recall graph is defined for a single query. However, to arrive at a meaningful conclusion, performance comparisons should be done based on several queries. Therefore, some techniques for the interpolation of precision values is needed in order to evaluate the overall retrieval results for a given set of queries. In this paper, we employed a ceiling interpolation.

**Area Under the ROC Curve (AUC)**

Receiver Operator Characteristic (ROC) curves plot the true positive rate (i.e., recall) against the false-positive rate for different cutoff values of the predicted scores\(^ {136}\). ROC curves, therefore, measure the tradeoff between sensitivity and specificity.

The ROC curve can be aggregated into a scalar metric by computing the area under the curve (AUC). The AUC can be interpreted as “the expectation that a uniformly drawn random positive is ranked before a uniformly drawn random negative” [Flach2007], which is equivalent to the Wilcoxon-Mann-Whitney U statistic test of ranks [Hanley1982]). The AUC takes values between 0 and 1. However, because random guessing produces the diagonal line which has an area of 0.5, no realistic classifier should have an AUC less than 0.5 [Fawcett2003].

---

\(^{136}\) We used the ceiling interpolation of the ROC curves corresponding to a query gene.
GAP predictive power over the known interactions

To assess GAP's predictive power over the known interactions, we randomly selected a set of 115 human genes/proteins for which, based on our positive gold standard, some interactions have been experimentally reported\textsuperscript{137}. GAP was then queried with these genes, one at a time, and the performance measures (e.g., F1-scores, PR curves, or AUC scores) were calculated using the ranked list of the predicted interacting partners. The average of the scores or the interpolation of the curves were then depicted as the measures of the GAP's (and other benchmark methods') performance.

GAP performance at different configuration setting

We have used several methods to (1) calculate term-based similarities (using various data-driven and ontology-based approaches), (2) aggregate term-based similarities to feature-set similarities (using either maximum or average), and (3) combine feature similarities to gene similarity scores (using either weighted sum of feature similarity scores or $q$-quantile sum). To study the effect of each setting combination on GAP's performance, we ran GAP with different configuration settings and compared the resulting performance measures. For ease of reference, we assigned a number (i.e., config. number) to each combination of configuration settings as displayed in Table 6.3.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|c|c|c|}
\hline
Configuration Number & Term Similarity & Term Aggregation & Per-att. Aggregation \\
& & (per-attribute similarity) & (gene similarity) \\
& Leave & Resnik & Seco & Spec-desc & Average & Maximum & Score x decile & Decile only \\
\hline
1 & & & & & & & & \\
2 & & & & & & & & \\
3 & & & & & & & & \\
4 & & & & & & & & \\
5 & & & & & & & & \\
6 & & & & & & & & \\
7 & & & & & & & & \\
8 & & & & & & & & \\
9 & & & & & & & & \\
\hline
\end{tabular}
\end{table}

\textsuperscript{137} We impose this constraint to avoid selecting genes with no known interactions. All the performance measures would be 0.0 using these genes (i.e., no “true positives”)}
Table 6.3 Configuration Table: for ease of reference, at any threshold setting, different combinations of similarity measures at different levels of term, per-feature, and gene similarity are given specific configuration numbers

Figure 6.2 F1-scores for different configuration settings. Error bars show the standard error of the mean depicts the F1-scores at different configuration settings when the threshold is set to select the 25% and 10% of the highest-scored interactions, i.e., upper quartile and upper decile, respectively. The overall trend of GAP performance at different configuration settings is similar using either upper quartile, Figure 6.2(a), or upper decile, Figure 6.2(b), threshold settings. F1-scores, however, are generally higher using the upper quartile threshold, although intuitively one may expect a decrease in performance when using a looser threshold setting, as retrieving more interaction typically introduce more noise into the prediction performance. However, GAP displays a graceful degradation of its predictive power, i.e. the chance of retrieving noise (false positive) stays low while predicting more and more interacting pairs.

(a) Upper quartile threshold setting  (b) Upper decile threshold setting
Moreover, varying gene similarity aggregation methods (decile-weighted vs. decile-only) insignificantly affect the overall performance and therefore, no definite conclusion on the advantage of one over the other can be made according to our experiments.

Regarding the feature similarity aggregation methods (average vs. maximum), except for when the specificity descendants (SD) similarity measure is used, maximum aggregation performs slightly better than the average aggregation. The average method, in one hand, incorporates more information from term similarities, but on the other hand, it is more prone to including noisy information, if the adopted term similarity method is not accurate or meaningful enough.

Last but not least, as expected, the employed term similarity method seems to be the most influential factor, as its quality would be propagated through the aggregation chain. As Figure 6.2 shows, GAP best performs when setting the term similarity method to specificity-descendant.

Configuration number 14 (SD-Ave-Decile) is the best performing configuration, and therefore, for all the coming sections, GAP (by itself) refers to the configuration 14 of our proposed tool.

In order to more precisely evaluate and visualize the effect of the term similarity setting, we have also reported precision versus recall curves, Figure 6.3 and the AUC scores, Figure 6.4, at different term similarity settings (i.e., configuration numbers 2, 6, 10, and 14). As these Figures reconfirm, SD outperforms the rest of term similarity configuration settings.
In order to more precisely evaluate and visualize the effect of the term similarity setting, we have also reported precision versus recall curves, Figure 6.3 and the AUC scores, Figure 6.4, at different term similarity settings (i.e., configuration numbers 2, 6, 10, and 14). As these Figures reconfirm, SD outperforms the rest of term similarity configuration settings.

Figure 6.4 AUC score comparison at different term similarity measures. The average of AUC values are shown as bars, and for each bar, the standard error of the mean is shown as a thin error bar.
**Predictive power of individual features**

To assess the predictive power of each individual feature, we ran GAP with only one feature at a time, and reported the interpolated PR curves and averaged AUC scores as shown in Figure 6.5 and Figure 6.6 respectively. In order to investigate the advantage of using ontology annotations (for ``Go'' feature), we tried out both ontology-based (i.e., SD) and data-driven (i.e., frequency-based) term similarity methods. As expected, the ontology-based similarity performs better, specifically at higher recall points.

![Graph showing PR curves for different features]

**Figure 6.5 Precision vs. recall curves using every single feature as the only predictive feature of GAP**

As Figure 6.5 and Figure 6.6 show, drug is the least and GO is the most influential feature. Such an observation, however, does not diminish the value of drug and disease features in deriving gene functional associations.
Figure 6.6 AUC score comparison when individual features are set to be the only predictive feature of GAP. The average of AUC values are shown as bars; the standard error of the mean is shown as a thin error bar at the top of each bar.

These features, in fact, play a role in extracting more implicit associations which are not typically included in our gold standard dataset. For instance, two genes which are affected by the same drug, are in some sense related to each other, but may not necessarily interact or be members of identical pathways or complexes. The advantage of these features is more evident in other GAP applications such as molecular pharmacology studies rather than the PPI prediction. We conclude that, combining all features together brings about the best and the steadiest performance.

**GAP performance as compared to other PPI prediction servers**

Even though GAP maintains its own exclusive features and methodology, one may speculate its advantage as compared to the existing well-developed and widely-used PPI predictors or gene association mining tools. To ascertain GAP's predictive power in extracting true gene associations or protein-protein (direct/indirect) interactions, as compared to the state of the art PPI predictors, we have selected several existing tools as benchmark predictors and compared their performance using our compiled gold-standard.

We were interested in selecting benchmark methods which are not specifically designed for direct protein interactions (for instance, tools using molecular docking algorithms were excluded). Furthermore, to avoid implementing any of the method chosen for comparison, we
looked for tools which offer web servers through which gene/proteins can be easily queried. Last but not least, we favored well-developed and widely-cited tools which have been intensively used for various bioinformatics tasks.

We applied our selection criteria to over 20 different PPI predictors or gene association mining web servers, and eventually chose the following five tools:

**STRING: Search Tool for the Retrieval of Interacting Genes/Proteins**

The database and web-tool STRING is a meta-resource of known and predicted protein-protein associations derived from four sources of genomic context, high-throughput experiments, coexpression, and scientific literature. STRING has been developed by a consortium of academic institutions. It is regularly updated and the last version covers about 5.2 millions proteins from 1133 species.

STRING imports protein association knowledge from databases of physical interactions and databases of curated biological pathways (e.g., MINT, HPRD, BIND, DIP, BioGRID, KEGG, and Reactome). Besides the experimentally derived protein-protein associations, STRING also stores computationally predicted interactions from the text mining of scientific texts as well as interactions computed from genomic features.

More precisely, for each query gene, STRING retrieves all genes which repeatedly occurred within the same cluster as the query gene. STRING defined the gene cluster as introduced by [Overbeek2000]. Different genomic features (e.g., gene neighborhood, gene fusion events, and coexpression) are incorporated to form the gene clusters. Text based predicted interactions are simply derived by processing the content of scientific articles (e.g., PubMed) to search for gene name co-occurrences.

**I2D-Pred: Interologous Interaction Database-Predicted**

I2D (Interologous Interaction Database) is an on-line database of known and predicted mammalian protein-protein interactions. We used I2D's known human PPIs in the construction of our gold standard, and compared GAP's performance against I2D's set of 59,373 interolog-based predicted interactions.
I2D-Pred is constructed by mapping model organism (i.e., S. cerevisiae, C. elegans, D. melanogaster and M. musculus) protein interactions to human protein orthologs using BLASTP and the reciprocal best-hit approach. Using the constructed database of model organism-to-human orthologs, each model organism protein was translated to its human ortholog, and a predicted human interaction was added if both proteins in the model organism interaction were conserved in humans. The predicted interactions are evaluated using protein domains, gene co-expression and Gene Ontology terms.

**PPI Finder: A Mining Tool for Human Protein-Protein Interactions**

PPIFinder is a web-based tool for mining human protein-protein interactions from PubMed abstracts based on their co-occurrences and interaction words. PPIFinder proposes a hybrid frame-based approach to take advantage of both statistical and computational methods. It follows a typical frequency-based statistical method of retrieving genes related to the query gene based on their co-occurrences in the PubMed literatures. However, PPIFinder also adopts computational linguistic methods, to extract semantic descriptions of the predicted interactions from the co-occurring literature. In addition, PPIFinder searches for Gene Ontology annotations and used the shared GO terms to infer potential PPIs. According to the reported statistics, only 28% of the co-occurring protein pairs in PubMed abstracts appeared in any of the frequently used human PPI databases (HPRD, BioGRID and BIND). On the other hand, of the known PPIs in HPRD, 69% showed co-occurrences in the literature, and 65% shared GO annotations.

**PIP: Potential Interactions of Proteins**

PIP is a web server for evoking human, rat, and fission yeast potential protein-protein interactions which are derived from homology with experimentally known protein-protein interactions from various species. To identify homologous interacting pairs of experimentally supported PPIs, BLAST searches are run for the entire genomes of the species of the interest (e.g., rat) against all proteins in DIP [DIP] and MIPS [MIPS] PPI databases. The putative protein interactions were given confidence scores based on their homology to experimentally observed
interacting proteins. The confidence scores are then weighted according to the amount of the experimental evidence (i.e., higher weight is given to more frequently observed interactions).

Once the networks of interacting proteins are constructed, the number of individual interactions is reduced by using a clustering method aimed at identifying key interconnected network nodes. Some networks constructed around key proteins were then identified as being highly up or down-regulated in rat cell lines of high metastatic potential.

**PIPs: Human protein-protein interactions prediction database**

The PIPs database is a web resource for predicting human protein-protein interactions using a naïve Bayesian method that combines information from gene expression, orthology, domain co-occurrence, post-translational modifications, co-localization, and the analysis of the local topology of the predicted PPI network.

Each evidence type is considered as a separate module providing a score of interaction. The individual module scores are combined into a prediction score corresponding to the overall likelihood of the potential interaction given the available data. PIPs contains 37606 high probability interactions (i.e., with a score < 1 indicating that the interaction is more likely to occur than not) out of which ~3400 are not reported in the interaction databases HPRD, BIND, or DIP.

To set up the experiments, we queried each of the benchmark tools with 115 randomly selected human genes/protein, and accumulated the related genes or proteins ranked according to their similarity or confidence scores\(^\text{138}\).

---

\(^{138}\) We used the default settings of each tool except for STRING in which the cutoff number of interactors for each gene is increased to 100 where the default value was 5 (the confidence value of the interactions is set to *medium and above* which is STRING’s default setting). Moreover, we excluded STRING interactions which are directly derived from the databases of known interactions.
Figure 6.7 Precision vs. recall curves of GAP as compared to the selected benchmark PPI predictors.

Figure 6.7 and Figure 6.8 respectively show the precision versus recall curves, and the AUC scores of GAP as compared to the benchmark PPI prediction tools.

Figure 6.8 AUC score comparison between GAP and the five benchmark PPI prediction tools. The average of AUC values are shown as bars; error bars correspond to the standard error of the mean.
As shown in Figure 6.7 and Figure 6.8, GAP is the absolute winner in extracting the experimentally known protein interactions. PPIFinder and STRING place on the second position while the difference between their performance is insignificant. The other three predictors behave almost identical and rather poorly on the given set of queried genes.

6.1.2 Assessment of GAP’s predictive capabilities - supporting evidence from the scientific literature

The value of the PPI prediction and gene association mining tools lies in their ability to predict reliable novel interaction and association candidates. Even though the actual interactions can only be validated experimentally, providing supporting evidences from external data sources, can further encourage biologists to try out confirming the predicted interactions in the wet lab.

Hence, we manually curated the existing literature hunting for strong supporting evidences or confirming some of the novel predicted interactions. We also used manually curated databases of genetic association to a high profile disease (e.g., autism), to extract corroborating information endorsing our predicted functional interactions, operating under the assumption that genes known to be implicated in a disease susceptibility, should be also predicted to be functionally associated.

Support for the predicted associations in the scientific literature

We selected 40 high-scored novel139 predicted protein interactions, and asked an expert curator to search scientific literature for any information corroborating these predictions. The result were very promising, and we could either confirm or strongly support several direct/indirect predicted interactions as listed in Table 6.4 List of the confirmed PPI interactions, types of the interactions and the PubMed IDs of the articles confirming the predicted interactions experimentally..

---

139 We identify these predicted potential interactions as novel because they do not appear in any commonly used direct or indirect interaction databases.
<table>
<thead>
<tr>
<th>Interacting Pairs</th>
<th>Interaction Type</th>
<th>PubMedID of confirming article</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD82 and DARC</td>
<td>physical</td>
<td>16862154</td>
</tr>
<tr>
<td>CD82 and AMFR</td>
<td>physical</td>
<td>18037895</td>
</tr>
<tr>
<td>ILK and ACTN4</td>
<td>physical</td>
<td>16837631</td>
</tr>
<tr>
<td>ILK and NPHS1</td>
<td>physical</td>
<td>16837631</td>
</tr>
<tr>
<td>NPHS1 and ACTN4</td>
<td>physical</td>
<td>16837631</td>
</tr>
<tr>
<td>BRMS1 and SAP130</td>
<td>physical</td>
<td>16914451</td>
</tr>
<tr>
<td>DACTN4 and SYNPO</td>
<td>co-complex</td>
<td>?</td>
</tr>
<tr>
<td>KISS1 and MED23</td>
<td>regulation</td>
<td>16964286</td>
</tr>
<tr>
<td>FGF2 and VEGFA</td>
<td>co-pathway</td>
<td>?</td>
</tr>
</tbody>
</table>

Table 6.4 List of the confirmed PPI interactions, types of the interactions and the PubMed IDs of the articles confirming the predicted interactions experimentally.

**Integrin-linked kinase physically interacts with α-astinin-4 and nephrin; α-astinin-4 physically interacts with nephrin**

Dai and co-workers in their study on the function of Integrin-linked kinase (ILK) in glomerular podocyte biology, showed that the ILK deficiency caused an aberrant distribution of nephrin and α-actinin-4 in podocytes. Using co-immunoprecipitation, Dai and co-authors also demonstrated that ILK physically interacts with nephrin and α-astinin-4 in vivo, and that in the resulting ternary complex, nephrin and α-astinin-4 are also physically interacting.
Support for predicted associations from an expert curated gene-disease association database

We have also found supporting evidence for our predictions by examining the Human Gene module of SFARI Gene\textsuperscript{140}, a web-based, publicly available and highly trusted, expert curated repository of genetic associations to autism spectrum disorder (ASD).

The content of the Human Gene module originates from published, peer-reviewed scientific literature. The information is extracted through a systematic search of the literature, followed by a multi-step annotation and curation strategy to derive ASD associated genes.

Intuitively, genes associated to a specific disease function, should be functionally related to each other. We are, therefore, interested in seeing how intra-connected the autism related genes would turn out to be based on the predictions delivered by GAP, under the assumption that the more the autism genes are predicted to be related to each other, the better the predictor is.

The current version of SFARI human gene module includes 306 autism-related genes, and we needed to reduce this number in to be able to analyze and visualize the prediction outcome. In order to select genes which are strongly and positively associated to autism, we subtracted the total number of negative from the total number of positive associations that were directly reported in the SFARI database, and selected those for which this difference is more than 10. Following this procedure, we ended up with 25 genes out of 306.

The functional intra-connectivity of autism genes predicted by GAP (using upper decile threshold setting) is depicted in Figure 6.9. The transparency of the edges is inversely proportional to the rank of the predicted associations. In other words, the lower the score of

\textsuperscript{140} Available at: http://sfari.org/
similarity is, the more transparent an edge would be. In addition, the size of a node and a node's highlight size are, respectively, proportional to the node degree and the SFARI score\(^{141}\) of the node.

Figure 6.9 Functional intra-connectivity of autism genes predicted by GAP (a), predicted by STRING (b), and drawn from databases of known direct and indirect interactions (c).

\(^{141}\) number of negative subtracted from the number of positive associations reported by SFARI Gene dataset.
The transparency of the edges is inversely proportional to the rank score/score of the predicted associations. The size of each node and the node highlight size are, respectively, proportional to the node degree and the SFARI score. The node highlight colors change from red (the lowest SFARI score) to blue (highest SFARI score).

As Figure 6.9 clearly shows, autism genes are densely associated to each other, according to the GAP prediction. Furthermore, nodes with a lower degree of connectivity (smaller node size), in most cases, also have a lower SFARI scores (smaller highlight size). Such an observation to some extend justifies the low degree of some autism nodes in the GAP predicted graph.

We have followed the same process using STRING as the gene association predictor. Of all the benchmark methods we considered, STRING is the most comparable technique to GAP (i.e., seeking for gene associations in a general way), and also is among the best-performing benchmark predictors. As Figure 6.9(b) shows, STRING's predicted intra-connection graph is by far more sparse than GAP's.

Furthermore, for readers interested in comparing the advantage of GAP prediction over the datasets of known direct/indirect interactions, we have also included the interactions among autism genes derived from databases of known physical interactions and pathways (we used our previously defined gold standard to extract the interactions). Figure 6.9(c) shows known associations among the autism genes, and clearly reveals the limitation of the existing databases of known associations.

We also used GAP to predict novel autism associated genes. To be able to visualize the predictions, we only selected 10 autism genes with the highest SFARI scores, ran GAP to extract the predicted associated genes, and excluded out the known autism genes. For the sake of readability of the corresponding graph, we only report the associated novel genes whose ranks are less than 5. Figure depicts the known autism genes (red nods at the center of the graph) and the novel autism genes (black nodes placed around the red nodes); black edges corresponds to the associations between known and novel autism genes.
Figure 6.10 Novel autism genes predicted by GAP, and the evidences from the related diseases, drugs, GO terms, and pathways.

To support the predicted associations, Figure 6.10 also shows evidence from pathways, GO annotations, drugs and diseases. Red nodes at the center of the graph correspond to the 10 known autism genes with the highest SFARI score. The novel autism genes are shown by black nodes placed around the red nodes; black edges corresponds to the associations between known and novel autism genes. Supporting evidence is shown in different shapes and colors as described by the legend. When an interaction is supported by an evidence, the corresponding genes are connected to the evidence node by a transparent edge if the gene node is an autism gene (red node), and by a dark edge if the gene node is a novel gene (black node).

A promising observation is that each evidence is strongly related to almost all of the autism genes (the transparent lines between evidence nodes and red nodes), and therefore, their association to the predicted novel genes (the dark lines between evidence nodes black nodes) should be also meaningful.
6.2 Discussion

**Research contributions.** In this chapter we presented GAP, a general-purpose *Gene functional Association Prediction* tool that semantically integrates information from various sources in order to infer if genes are functionally related. Using common performance evaluation measures and a gold-standard database of experimentally validated direct and indirect interactions, we have shown that GAP significantly outperforms the existing interaction prediction tools in correctly identifying *known* interactions. We have also confirmed a subset of novel predictions, with respect to our gold standard database, by manually searching the scientific literature\(^{142}\).

We also corroborated a subset of GAP’s novel predictions using an expert curated database of genes associated to the autism spectrum disorders, under the assumption that genes participating in the same phenotype are expected to be functionally related. Our primary reason for selecting autism spectrum disorders for supporting GAP's predictions was data availability. However, autism is one of the most common neurological disorders with a very complex genetic architecture whose underlying genetic determinants are still largely unknown, i.e., based on GWAS studies only a few genes with common polymorphisms have reached genome-wide significance [Wang]. It is, therefore, particularly useful to propose high-confidence candidate genes associated with autism susceptibility.

Since GAP successfully predicted functional associations among *known* autism genes, we were encouraged to use it for predicting *novel* genes associated with autism. We queried GAP with 306 known autism genes from the SFARI human gene module, and retrieved all functionally interacting partners predicted by it whose interaction p-values are less than 0.05. 11,215 *novel* genes, with respect to SFARI database, were predicted to be functionally associated to the SFARI-known autism genes. However, as each of the SFARI genes may contribute to different processes, and participate in diverse genetic phenotypes, not all the functionally related genes should be necessarily implicated in autism spectrum disorders. Yet, if a new gene is predicted to

\(^{142}\) Though these confirmed predictions were experimentally validated in the published literature, they are new to the publicly available datasets of known direct and indirect interactions.
be functionally associated to several known autism genes, it is more probable to be involved in autism, a phenotype which is shared by all the SFARI genes.

We define the association-degree of a predicted gene to be the number of SFARI autism genes which are predicted by GAP to be functionally associated to it. The histogram of the association-degrees of all the 11,215 predicted novel genes is provided in Figure 6.11.

![Histogram of association-degrees](image)

**Figure 6.11 Histogram of the association-degree of 11,215 genes predicted to be functionally associated to SFARI-listed autism genes.**

We selected predicted genes with the highest 1\% of association-degrees (i.e., association-degree greater than 15). 114 genes were accordingly selected and reported as *novel* candidate autism genes not yet listed in the SFARI database. Figure 6.12 displays the network of functional associations of these 114 predicted genes to SFARI autism genes.
Figure 6.12 Network of functional associations among SFARI-listed autism-related genes (yellow) and GAP-predicted autism-related genes (black)

For each of the 114 genes, the gene name, its interacting partners from SFARI database, and supporting evidence, i.e., pathways, GO annotations, disease and drug information used by GAP to produce the predictions can be found at: http://genomebiology.com/imedia/2366542188000498/supp2.pdf.

GAP’s predicted functional interactome\textsuperscript{143} contains ~1M predicted functional associations whose associated p-values are less than 0.01, out of which about 0.9\% are novel. GAP’s novel predictions connect previously disconnected components and singletons to the main body of the known interactome.

\textbf{Limitations:} Gap’s performance depends on the quality of the data available for integration as well as on the limitations of the individual methods used for assessing the similarity between

\textsuperscript{143} By functional interactome we mean a network in which nodes correspond to genes and edges correspond to the functional associations between the adjacent nodes.
genes, e.g., the assumptions made by the semantic similarity measures that depend on gene annotation data.

In summary, GAP can be applied to a wide range of tasks such as phenotype prediction, gene clustering, and pharmacology analysis. The underlying approach is general, can take advantage of any type of biological data sources, and can be extended to different organisms, provided that the employed databases and services cover multi-organism information [Vafaee2012].
The Semantic Knowledge Management field has become increasingly active in recent years with high profile efforts such as Google’s Knowledge Graph leading the way towards making the collective knowledge accumulated by humanity machine-processable in ways that were not too long ago thought out of the realm of possibility. It has taken the resources of a search engine giant to bring forth the era of searching for and expecting answers to our queries instead of just a list of (more or less) relevant documents, but the technology employed by the Knowledge Graph can be traced, in part, to several open source collaborative efforts on knowledge acquisition and representation such as Freebase and DBPedia. Although the contribution of ontologies is not specifically mentioned, the language describing the new technology reveals its grounding in knowledge representation techniques.

Improved semantic technologies for acquiring, representing and integrating knowledge will get us closer to the establishment of an interoperable network of personal and global knowledge representations and knowledge discovery tools and realizing the promises of the Semantic Web.

From requirements to the design of practical knowledge and their application to the semantic integration of knowledge and knowledge discovery.

In this thesis we presented several studies that advance the field of Semantic Knowledge Management by making contributions to its core components, from requirements and

144 http://www.google.ca/insidesearch/features/search/knowledge.html
145 A large, open-access, on-line collaborative knowledge base developed by Metaweb which was acquired by Google in 2010. http://www.freebase.com/
146 A crowed-sourced knowledge base of facts extracted from Wikipedia. http://dbpedia.org/About
147 “[The Knowledge Graph] currently contains more than 500 million objects, as well as more than 3.5 billion facts about and relationships between these different objects.” From the official Google blog http://googleblog.blogspot.ca/2012/05/introducing-knowledge-graph-things-not.html
technologies for practical knowledge representations (Chapters 2 and 3), to the semantic integration of knowledge and knowledge discovery, by providing new ontology mapping techniques and their applications to advancing biomedical knowledge (Chapters 4, 5 and 6).

Traditionally, knowledge representations have been created by knowledge engineers who were also domain experts, but there is an increased perception that with the advent of the Semantic Web\textsuperscript{148}, of which the efforts mentioned above are an integral part, more and more end users will be able to influence the development of domain representations through collaborative development efforts, or build their own representations as foundations for Personal Knowledge Management systems\textsuperscript{149}.

In Chapter 2 we present the results of the survey we conducted in order to identify the real-life requirements of the users who currently work with practical ontologies, or plan to make use of them in the future and also performed an in-depth investigation of the representational requirements for modeling domain knowledge as practical ontologies. We extracted representational and functional requirements that frameworks for practical ontologies should meet in order to support the knowledge intensive tasks users need to engage in and provided recommendations for implementing such frameworks.

We also identified the following basic categories of knowledge as important for the practical purposes reported by our survey takers: spatial and dynamic knowledge; activities and causality; patterns and complex associations; transformations and mutations; hypothetical; and uncertain and qualitative knowledge. We provided an in-depth analysis of these and other basic types of knowledge we identified from surveying the literature by discussing existing approaches to codifying knowledge and supplying formal representation solutions to some of the issues we identified. Although most of the examples we provided are from the biomedical domain, the issues and solutions covered in chapter 2 are generally applicable.

\textsuperscript{148} a collaborative movement to promote the development and use of common data formats for the World Wide Web led by the international standards body, the World Wide Web Consortium (W3C).

\textsuperscript{149} http://www.anderson.ucla.edu/faculty/jason.frand/researcher/speeches/PKM.htm
Formal knowledge representations are essential for realizing the true potential of semantic technologies and a better understanding of how the expressiveness of the languages used for representing and querying knowledge influences computability will help us not only with our representational choices but also with our decisions to pursue exact or approximate answers for our questions. We also provided informal guidelines for choosing among the various formal language the one that is the most suitable for the purpose for which the knowledge representation is to be built, when taking into account the available computational resources.

Knowledge acquisition and representation is just the first building block in Semantic Knowledge Management frameworks. Semantic interoperability and integration lie at their heart and occupy an increasingly prominent role as it is unlikely that a unified conceptual framework can be found that we can all agree upon and will be adequate to represent all types of knowledge and it is to be expected that we will continue with the current practice of developing separate ontologies for describing diverse domains according to different points of view. Building ontologies that bridge the gap between the various models of knowledge is one of the ways in which semantic interoperability can be achieved.

In Chapter 3 we make contributions to the interoperability efforts by introducing the Core Business Data Interchange Ontology, based on the informational models contained in several of the standards developed by the industry, as well as insights into the design process and representational requirements.

The success of semantic interoperability and integration depends not only on the availability of quality domain and application ontologies but also on our ability to define or discover mappings between the various entities and relationships in the plurality of ontologies already in use or forthcoming and many efforts are currently underway to align the various representations on hand. For instance, the W3C Semantic Web Interest Group coordinates the efforts to publish mappings between established standards such as Dublin Core\textsuperscript{150} and Simple Knowledge Organization System, SKOS\textsuperscript{151}, and newcomers such as Schema.org and GoodRelations. In the

\textsuperscript{150} http://dublincore.org/

\textsuperscript{151} http://www.w3.org/2004/02/skos/
natural sciences field, the Open Biological and Biomedical Ontologies Foundry, a collaborative effort with the stated aim of developing a suite of orthogonal and interoperable reference ontologies for the biomedical domain, oversees the creation of mappings between the various vocabularies, thesauri and ontologies it is hosting.

At the core of mapping discovery techniques there is a measure, or an ensemble of measures of similarity, or, more generally, of relatedness.

In Chapter 4 we have explored this topic in depth and proposed several such measures, some for assessing the similarity between concepts in the same ontology, some for estimating the similarity between concepts in different ontologies. The intra-ontology measures we proposed makes sole use of the information encoded in the ontology they are applied to and are thus complementary to measures that depend on the availability and quality of ontology instance data. The inter-ontology family of measures we put forward takes advantage of both types of information: instance and structural.

In Chapter 5 we describe an application of our method to detect relationships between concepts in different ontologies. More specifically we used our similarity measures to discover links between the three components of the Gene Ontology and had the top mappings evaluated by expert biologists. Our efforts proved to be complementary to the existing lexical techniques for mapping detection and to GO Consortium’s own initiative to define mappings between its ontologies. Our method discovers mappings that cannot be detected via lexical techniques and are outside the (restricted, for now) scope of the mappings introduced by the GO Consortium.

In Chapter 6 we present our contribution to one of the most important aspects of Semantic Knowledge Management: knowledge discovery. We incorporated our intra-ontology measure into an integrative method that predicts the functional association of human genes and can be used to advance knowledge in the biomedical domain by helping with a variety of tasks such as phenotype prediction, gene clustering, and pharmacology analysis. The underlying approach is sufficiently general that it can take advantage of any type of biological data sources, and can also be extended to different organisms. We reported on the evaluation method and the results in chapter 6.
Future work

There are many extensions of the work presented in this thesis that can be undertaken in future research. We discuss several of them in what follows.

The survey that helped identify requirements for practical ontology frameworks, discussed in Chapter 2, can be expanded to include a larger variety of ontology users. Its findings can be used to guide the design of software tools that would allow non-computer specialists to develop and use personal ontologies and link them up to the growing number of online resources and tools.

The analysis of the representational requirements of different types of knowledge, also undertaken in Chapter 2, can be extended to include more types of knowledge, and can also be further refined in order to divide some categories into subtypes that are of particular interest in certain domains and merit to be studied separately, in their own right. It would be also interesting to investigate if ontology users from different domains have different points of view regarding the representational requirements for the knowledge in the overlapping area of their respective domains and to what extent the conventions and practices existing in their fields.

The design of specialized representational languages can benefit tremendously from being informed by these requirements, as many, if not most of the existing languages have a generic purpose and have not been designed with specific representational requirements in mind.

Formal ontologies are knowledge representations aimed at unambiguously capturing the meaning of and the relationships between concepts in a variety of domains, and there are many opportunities to use them alongside existing information standards.

The development of the Core Business Data Interchange Ontology (CBDIO), the contribution we presented in Chapter 3, is part of the research efforts aimed at formalizing the concepts within existing information standards as ontologies. Our work can be advanced by extending it to include more of the existing and emerging industry standards and integrating it into the Ontology-based Standards Initiative\textsuperscript{152}, is a leading effort in the area of incorporating ontology approaches in the development and application of standards.

\textsuperscript{152} http://ontolog.cim3.net/cgi-bin/wiki.pl?OntologyBasedStandards
Ontologies can assist in the standardization efforts across domains in various ways, e.g., by helping improve the quality of standards by increasing their precision and thus reducing ambiguity and misinterpretations, and by making it easier to check conformance, via automated methods supported by the availability of formal axioms. From a knowledge management perspective, ontologies can also make the management of information models, standards, and applications, easier, leading to a better reuse of the existing knowledge. These are all very promising avenues for future research.

In particular, CDBIO can be used in the future to perform an ontological analysis of the standard it is based on (OAGIS), and potentially those it would be extended to cover, in order to identify potential inconsistencies and ambiguities, and subsequently help repair these issues and redesign the standard(s).

Formal ontologies can also be used to support integration and interoperability by acting as mediators or via mappings defined between the ontologies associated to the knowledge to be integrated. Semantic similarity measures are at the core of ontology mapping technologies and is an area of active research.

The work on semantic similarity measures we presented in Chapter 4 can be extended by considering all the different types of relationships between ontological concepts. Another interesting issue to explore in future research is the incorporation of formal ontology axioms in the design of similarity measures as well as the design strategies that can recommend the most appropriate measures to be used depending on the features on the ontology, e.g., simple taxonomy, or complex ontology with several different types of relations between concepts and formal axioms. In particular, designing similarity measures for ontologies represented in Description Logic-based languages would be a particularly useful endeavor, as OWL ontologies are becoming more and more common.

The ultimate goal, the establishment of an interoperable network of personal and global knowledge representations and knowledge discovery tools, would require many more research projects but we believe that the work presented in this thesis is a step in the direction of realizing it due to its contributions to improving the understanding of the requirements for representing knowledge, adding new insights into the design of practical representations; aiding ontology interoperability by supplying an ontology for data exchange and providing new tools for
semantic integration (similarity measures to help find mappings between ontologies) and knowledge discovery (a web-based service for predicting functional gene associations).
References


## Appendix A ISO Terminology– related standards

<table>
<thead>
<tr>
<th>Technical Subcommittee</th>
<th>Domain</th>
<th>Standard</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC 37/SC 1: Principles and methods</td>
<td>Terminology policies Development and implementation</td>
<td>ISO29383:2010</td>
</tr>
<tr>
<td>TC 37/SC 2: Terminographical and lexicographical working methods</td>
<td>Bibliographic references and source identifiers for terminology work</td>
<td>ISO 12615:2004</td>
</tr>
<tr>
<td>TC 37/SC 2: Terminograpical and lexicographical working methods</td>
<td>Assessment and benchmarking of terminological resources – General concepts, principles and requirements</td>
<td>ISO 23185:2009</td>
</tr>
<tr>
<td>TC 37/SC3: Systems to manage terminology, knowledge and content</td>
<td>Computer applications in terminology – Machine-readable terminology interchange format (MARTIF) – Negotiated interchange</td>
<td>ISO 12200:1999</td>
</tr>
<tr>
<td>TC 37/SC3: Systems to manage terminology, knowledge and content</td>
<td>Terminology and other language and content resources – Specification of data categories and management of a Data Category Registry for language resources</td>
<td>ISO 12620:2009</td>
</tr>
<tr>
<td>TC 37/SC3: Systems to manage terminology, knowledge and content</td>
<td>Ontology Integration and Interoperability (OntoIOP)</td>
<td>ISO/NP 17347</td>
</tr>
<tr>
<td>TC 37/SC3: Systems to manage terminology, knowledge and content</td>
<td>Systems to manage terminology, knowledge and content – Concept-related aspects for developing and internationalizing classification systems</td>
<td>ISO/FDIS 22274</td>
</tr>
<tr>
<td>TC 37/SC3: Systems to manage terminology, knowledge and content</td>
<td>Systems to manage terminology, knowledge and content – Design, implementation and maintenance of terminology management systems</td>
<td>ISO 26162</td>
</tr>
<tr>
<td>TC 37/SC3: Systems to manage terminology, knowledge and content</td>
<td>Systems to manage terminology, knowledge and content - TermBase eXchange (TBX)</td>
<td>ISO 30042:2008</td>
</tr>
</tbody>
</table>

**Figure A.1 ISO Terminology-related Standards**
Appendix B OBOL

OBOL [Mungall2004] is one of the several languages designed for representing biomedical knowledge.

OBOL is used to formalize the knowledge encoded in the vocabularies which are part of the OBO Foundry and contains several atomic vocabularies, organized around domains, such as molecular functions, cell, protein, and generic anatomy, lexical categories, e.g., nouns, adjective and relational adjectives, and type designators, i.e. Roman and Indo-Arabic numerals and other special symbols. This language does not include verbs as a distinct category, because its primary goal is to formalize knowledge that has already been amassed in the controlled vocabularies managed by the OBO Foundry, as illustrated by the example in Table B.1.

<table>
<thead>
<tr>
<th>OBOL phrase</th>
<th>Lexical category</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive regulation of Interleukin - 6 secretion</td>
<td>adjective</td>
<td>general biological process</td>
</tr>
<tr>
<td></td>
<td>noun</td>
<td>general protein family</td>
</tr>
<tr>
<td></td>
<td>preposition</td>
<td>general biological process</td>
</tr>
<tr>
<td></td>
<td>special token</td>
<td></td>
</tr>
<tr>
<td></td>
<td>type designator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>noun</td>
<td></td>
</tr>
</tbody>
</table>

Table B.1 OBOL representation of a GO term

Legal OBOL phrases follow a simple formal grammar, given below, A finite set of terminal symbols $\Sigma_{OBO}$, containing all elements in the atomic vocabularies.

- A finite set of non-terminal symbols $N_{OBO}$ containing the set of lexical categories (nouns, adjectives, etc.) and the set of phrase types (noun phrases, prepositional phrases, etc.)

- A start symbol $S_{OBO}$

- A set of production rules $P$, a sample of which follows below:

  - $S_{OBO} \rightarrow$ NounPhrase
  - $\rightarrow$ NounPhrase $\rightarrow$ Noun | AdjectiveNounPhrase | NounPhrase’-‘Token |
  - $\rightarrow$ NounPhrase NounPhrase | NounPhrase PrepPhrase
  - $\rightarrow$ Prep NounPhrase
The semantics of an OBOL phrase is intended to be specified as an *OBOL definition*, of the form *genus* and *differentiae* (a set of necessary and sufficient conditions that distinguish between terms belonging to the same genus). Each word in the atomic vocabularies is considered to be a distinct genus. Below is the OBOL definition corresponding to the example in Table B.1

(regulation qualifier = (negative) affects = (secretion secrets = ( interleukin type_token(6) ) ) )

One of the main goals of OBOL is to assist with the creation of semi-computable definitions, of the *genus*-differentia form, that Gene Ontology Consortium now makes available for download. All records in the obo format files have the form:

\[
\text{defined_term} := \text{genus_term} \quad \text{that relation term}
\]

The genus differentia definition in the Gene Ontology of the term in Table B.1, in the obo format is as follows:

```
name: positive regulation of interleukin-6 secretion
namespace: biological_process
def: "Any process that activates or increases the frequency, rate or extent of interleukin-6 secretion."
intersection_of: biological regulation
intersection_of: positively_regulates interleukin-6 secretion
```

where the genus is *biological regulation* and the differentiae is *interleukin-6 secretion*. Assuming that each term and relation has a corresponding unary and binary predicate, respectively, a possible translation of these definitions into first order logic is immediate:

\[
\forall x \quad \text{positive\_regulation\_of\_interleukin\_6\_secretion}(x) \leftrightarrow \\
(biological\_regulation(x) \land (\exists y \ \text{positively\_regulates}(x, y) \land \text{interleukin\_6}(y)))
\]
Limitations. On the one hand OBOL cannot express all terms in the Gene Ontology, and on the other hand, the full set of legal OBOL phrases is infinite, and only a finite subset of them correspond to meaningful biomedical knowledge.

The OBO-Edit reasoner is very limited at the moment. Things that it currently cannot do include:

- Take advantage of necessary/inverse_necessary info
- Identify cycles of non-cyclic relations
- Reason about inverse_of links
- Reason about union_of links
- Reason about symmetrical relations
- Do cardinality checks
- Use domain and range information
- Make full use of sub-relations
Appendix C Representations of Spatial Knowledge

On spatial logics. Spatial logics are formal languages whose variables range over geometrical entities and whose non-logical primitives denote relations among and operations on these entities. In brief, they are, as any other logic, characterized by:

- a syntax, i.e. along the lines of propositional logic, first order logic, higher order logic, etc.;
- a vocabulary of non-logical primitives, e.g., Convex (x), overlap(x, y), adjacent(x, y) etc.;
- a class of interpretations.

Some complexity results: the first order theory of regular, closed, polyhedra in $\mathbb{R}^2$, over a signature that contains only the contact relation, is undecidable [Dornheim1998] and so are many other theories with non-logical primitives that are able to express the notion of closer than, over various domains of quantification$^{153}$. First-order theories over the 1-dimensional Euclidian space which have a restricted expressive power, i.e., they cannot express the closer than relationship, are decidable [Nenov2011].

On axiomatizations of Geometry. Euclid presented in his book, Elements, the first known axiomatic description of geometry, i.e., the first application of a formal method to the task of representing common (mathematical) knowledge, but it was only at the beginning of the 20th century that a complete axiomatization of the three-dimensional Euclidian geometry was given [Hilbert1988], which repaired the many mistakes and omissions in Euclid’s work, e.g., the missing axiom for betweenness, and errors in angle calculations. He chose the following non-logical primitives for his axiomatization: point(.), line(.), incidence(., .), i.e., point lying on a line, betweenness(., ., .), i.e., point lying between two others, and congruence(.,.) [Hilbert1988]. From these few basic primitives all the other familiar geometric entities and properties, such as segments, triangles, angles, circles, etc., can be defined, with the help of thirteen axioms in all. Tarski, in fact, gave several axiomatizations of geometry and continued to revise his work for

$^{153}$ E.g., the regular closed semi-algebraic sets over $\mathbb{R}^n$, the regular closed semi-linear sets over $\mathbb{R}^n$, the set of algebraic polytypes over $\mathbb{R}^n$, etc.
almost half a century. The first, a first-order axiomatization of elementary geometry, Tarski’s sense, is decidable but contains a countably infinite set of axioms, known as the continuity axiom schemata\textsuperscript{154}. It is also worth noting that Tarski’s first-order axiomatization of geometry is non-categorical, i.e., admits non-standard models.

**Affine Geometry.** Tarski has also provided an axiomatization for affine geometry and proved that affine geometry does not admit a finite first-order axiomatization. He has, nevertheless, provided a logical characterization for it, which is both incomplete, in the logical sense, and undecidable [Tarski1979].

**Cone logic** [Kuijpers1999], allows us to express many useful things, such as the fact that two regions meet at a point or share a “demarcation line”. Regions here are a special class of spatial information, expressible as sets in the plane $R^2$, consisting, intuitively, only of “filled” surfaces.

**Region Calculus.** A typical inference task in a system modeled using the RCC-8 relations is, for example, finding out, for three given objects (regions) $x, y, z$, what relations might hold between $x$ and $z$, given that some base relations $r_1$ and $r_2$ hold between $x$ and $y$, and $y$ and $z$, respectively. The solution to this task relies on calculating the composition between $r_1$ and $r_2$, usually based on some predefined relational compositions.

RCC-8 and BRCC-8, and extension of RCC-8 with Boolean operators, allow variables that range over *regular closed sets*\textsuperscript{155} in topological spaces and are not sensitive to the domains they are interpreted over, i.e., in order to show that a formula is satisfiable in, e.g., the 3-dimensional Euclidian space, it is enough to prove that it is satisfiable in an arbitrary (*connected*, for BRCC8) topological space [Renz1998]. This relative inexpressiveness translates into the decidability, albeit not necessarily tractability, of the satisfiability problem, which is NP-complete for RCC8 and PSPACE for BRCC8.

---

\textsuperscript{154} The schemata is needed as a replacement for the second-order Dedekind-style “cut” axioms needed to axiomatize the (real) Euclidian geometry [Tarski1999].

\textsuperscript{155} A set is a regular closed set if it is equal to its closure. The regular closed sets are of particular practical importance because they exclude from the interpretation a logical theory pathological cases.
Appendix D Ontology Web Language (OWL)

The web Ontology Language (OWL) is a language for the Semantic Web with formally defined semantics. OWL is a W3C standard \(^{156}\).

There are various syntaxes available for OWL: e.g., (1) functional-Style syntax, designed to be easier for specification purposes, (2) RDF/XML,

---

\(^{156}\) http://www.w3.org/TR/owl2-overview/

\(^{157}\) Image source: http://www.w3.org/TR/owl2-overview/OWL2-structure2-800.png
the only syntax required to be supported by all OWL tools, (3) Manchester syntax designed to be
easier for non-logicians to use, (4) OWL XML, a syntax defined by an XML, and tools are
available that can translate between them.

OWL 2 is not a programming language. Instead, it can be used to describe in a logical way
knowledge about a domain of interest using:

- **Entities:** elements used to refer to real-world objects: objects (*individuals*), e.g. Mary,
  Kitty, categories (*classes*), e.g., person, cat, and relations (*properties*), e.g., owns,
  ownedBy.

  ClassAssertion ( :Person :Mary )
  ClassAssertion ( :Cat :Kitty )

- **(Property and Class) Expressions:** complex descriptions of entities formed by
  combining existing entities using specialized *constructors*.

  DisjointClasses ( a:Man a:Woman )

- **Axioms:** logical statements

  SubClassOf ( Annotation( rdfs:comment "Cats are animals." ) a:Animal a:Cat )

Reasoners can then be used to infer further information (consequences) based on the descriptions
made using OWL. The complete, normative BNF grammar describing OWL can be found at:


OWL has several variants, listed below in decreasing order of expressiveness:

- **OWL Full:** allows the use of all OWL language elements, Table D.1, and constructs and
  also permits the unconstrained use of RDF constructs. In OWL Full the resource
  *owl:Class* is equivalent to *rdfs:Class* and all data values are considered also to be
  part of the individual domain, which consists of all resources (*owl:Thing* is equivalent
  to *rdfs:Resource*). Moreover, object properties and datatype properties are not
  disjoint: i.e., *owl:ObjectProperty* is equivalent to *rdf:Property.*
• **OWL DL**: is based on Description Logic and has the maximal number of constructs from OWL Full for which current theoretical results can guarantee the existence of decidable reasoning procedures. OWL DL requires a pair-wise separation between classes, datatypes, datatype properties, object properties, annotation properties, ontology properties, individuals, data values and the built-in vocabulary. Since the set of object properties and datatype properties are disjoint, the following property characteristics: *inverse of*, *inverse functional*, *symmetric*, and *transitive*, cannot be specified for datatype properties. OWL DL also requires that no cardinality constraints be placed on transitive properties, their inverses and any of their superproperties.

• **OWL Lite**: was designed with the tractability of reasoning tasks in mind and allows, besides the definition of subclass relationships, only very simple constraints on relations, e.g. cardinality constraints of 0 and 1. OWL Lite forbids the use of the following constructs: oneOf, unionOf, complementOf, hasValue, disjointWith, DataRange.

**Note**: In OWL DL and OWL Lite, `owl:Class` is a proper subclass of `rdfs:Class` i.e., not all RDF classes are OWL classes in OWL DL and OWL Lite.

<table>
<thead>
<tr>
<th>OWL Language Element</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>owl:AllDifferent</code></td>
</tr>
<tr>
<td><code>owl:allValuesFrom</code></td>
</tr>
<tr>
<td><code>owl:AnnotationProperty</code></td>
</tr>
<tr>
<td><code>owl:backwardCompatibleWith</code></td>
</tr>
<tr>
<td><code>owl:cardinality</code></td>
</tr>
<tr>
<td><code>owl:Class</code></td>
</tr>
<tr>
<td><code>owl:complementOf</code></td>
</tr>
<tr>
<td><code>owl:DataRange</code></td>
</tr>
<tr>
<td><code>owl:DatatypeProperty</code></td>
</tr>
<tr>
<td><code>owl:DeprecatedClass</code></td>
</tr>
<tr>
<td><code>owl:DeprecatedProperty</code></td>
</tr>
<tr>
<td><code>owl:differentFrom</code></td>
</tr>
<tr>
<td><code>owl:disjointWith</code></td>
</tr>
<tr>
<td><code>owl:distinctMembers</code></td>
</tr>
<tr>
<td><code>owl:equivalentClass</code></td>
</tr>
<tr>
<td><code>owl:equivalentProperty</code></td>
</tr>
<tr>
<td><code>owl:FunctionalProperty</code></td>
</tr>
<tr>
<td><code>owl:hasValue</code></td>
</tr>
<tr>
<td>owl:imports</td>
</tr>
<tr>
<td>----------------------</td>
</tr>
<tr>
<td>owl:incompatibleWith</td>
</tr>
<tr>
<td>owl:intersectionOf</td>
</tr>
<tr>
<td>owl:InverseFunctionalProperty</td>
</tr>
<tr>
<td>owl:inverseOf</td>
</tr>
<tr>
<td>owl:maxCardinality</td>
</tr>
<tr>
<td>owl:minCardinality</td>
</tr>
<tr>
<td>owl:Nothing</td>
</tr>
<tr>
<td>owl:ObjectProperty</td>
</tr>
<tr>
<td>owl:oneOf</td>
</tr>
<tr>
<td>owl:oneOf</td>
</tr>
<tr>
<td>owl:onProperty</td>
</tr>
<tr>
<td>owl:Ontology</td>
</tr>
</tbody>
</table>

**Table D.1 Web Ontology Language elements**
Appendix E Logic definability

We give here a brief overview of logic definability.

(Informal) Definition: a **structure** consists of a set along with a collection of finitary operations and relations which are defined on it.

**Definition:** A **structure** is a triple \( \mathcal{A} = (A, \sigma, I) \) where \( A \) is a **domain**, a non-empty set, a **signature** \( \sigma \), consisting of function and relation symbols, and an **interpretation function** \( I \) that indicates how the signature is to be interpreted on the domain.

**Definition:** Let \( \mathcal{L} \) be a first-order language, \( \mathcal{M} \) an \( \mathcal{L} \)-structure with domain \( M \), \( X \) a fixed subset of \( \mathcal{M} \), and \( m \) a natural number. Then:

A set \( A \subseteq M^m \) is **definable in** \( \mathcal{M} \) **with parameters from** \( X \) if and only if there exists a formula \( \varphi[x_1, \ldots, x_m, y_1, \ldots, y_n] \) and elements \( b_1, \ldots, b_n \in X \) such that for all \( a_1, \ldots, a_m \in M, (a_1, \ldots, a_m) \in A \) if and only if \( \mathcal{M} \models \varphi[a_1, \ldots, a_m, b_1, \ldots, b_n] \).

A set \( A \) is **definable in** \( \mathcal{M} \) **without parameters** if it is definable in \( \mathcal{M} \) with parameters from the empty set (that is, with no parameters in the defining formula).

A function is definable in \( \mathcal{M} \) (with parameters) if its graph is definable (with those parameters) in \( \mathcal{M} \).

An element \( a \) is definable in \( \mathcal{M} \) (with parameters) if the singleton set \( \{a\} \) is definable in \( \mathcal{M} \) (with those parameters).
Appendix F Computability and Complexity

On querying databases/knowledgebases. Initially, investigations into the complexity of query answering for knowledge bases have focused on queries over the terminological portion, where the main issues are concept subsumption and satisfiability. Increasingly, however, issues related to formulating and evaluating queries over the extensional part of knowledge bases, a.k.a., the assertional box (ABox)\textsuperscript{158}, are moving to the forefront.

The most basic ABox queries are simple instance retrievals, e.g., return all instances (from the Abox) of a particular TBox concept. These basic data look-ups are, however, of limited utility and many practical applications require much more complex queries. Much of the research on complex query evaluation has been focusing on the class of conjunctive queries, first order logic formulas\textsuperscript{159} of the form $\varphi(x_1, x_2, \ldots, x_k) : \exists x_{k+1}, x_{k+2}, \ldots, x_m A_1 \land A_2 \land \ldots \land A_r$, where $x_1, \ldots, x_k$ are free variables, also called distinguished variables, and $x_{k+1}, \ldots, x_m$ are bound variables, also known as undistinguished variables, and $A_1, \ldots, A_r$ are atomic first order formulae. These queries correspond to the select-project-join queries in relational algebra, i.e., queries that do not use the operations union or difference, and also to the select-from-where queries in SQL.

The combined complexity of answering general conjunctive queries is known to be NP-complete, a result derived in the ’70 by Chandra and Merlin [Chandra77], but the class of acyclic conjunctive queries, which covers many problems of practical interest, is known to have a polynomial combined complexity [Gotlob2001]. The data complexity of evaluating general conjunctive queries is in fact much lower, i.e., $\text{AC}^0$, and, thus, (at most) polynomial. Further complexity results in, e.g., [Calvanese2006, Lutz2008, Lutz2009, Kontchakov2009, DeGiacomo2011, Rosati2012, Immerman1999].

\textsuperscript{158} An ABox is a finite set of of membership assertions such as $C(a)$, i.e., the object denoted by the constant $a$ is an instance of $C$, and $R(a, b)$, i.e., the two objects denoted by the constants $a$ and $b$ are in the relationship denoted by $R$, in other words the pair $(a, b)$ is a instance of relationship $R$.

\textsuperscript{159} Essentially first order formulas without $\forall$, $\lor$, and $\neg$. 
### Appendix G OAGIS

<table>
<thead>
<tr>
<th>1. ActualLedger</th>
<th>27. InventoryBalance</th>
<th>53. PriceList</th>
</tr>
</thead>
<tbody>
<tr>
<td>3. AllocateResource</td>
<td>29. InventoryCount</td>
<td>55. ProductionOrder</td>
</tr>
<tr>
<td>4. BOD</td>
<td>30. Invoice</td>
<td>56. ProductRequirement</td>
</tr>
<tr>
<td>5. BOM</td>
<td>31. InvoiceLedgerEntry</td>
<td>57. ProjectAccounting</td>
</tr>
<tr>
<td>7. CarrierRoute</td>
<td>33. ItemMaster</td>
<td>59. PurchaseOrder</td>
</tr>
<tr>
<td>8. Catalog</td>
<td>34. JournalEntry</td>
<td>60. Quote</td>
</tr>
<tr>
<td>9. ChartOfAccounts</td>
<td>35. Location</td>
<td>61. Receivable</td>
</tr>
<tr>
<td>11. CostingActivity</td>
<td>37. MaintenanceOrder</td>
<td>63. ReceiveInventory</td>
</tr>
<tr>
<td>12. Credit</td>
<td>38. MatchDocument</td>
<td>64. RecoverWIP</td>
</tr>
<tr>
<td>14. CreditTransfer</td>
<td>40. MoveInventory</td>
<td>66. RFG</td>
</tr>
<tr>
<td>15. CreditTransferIST</td>
<td>41. MoveWIP</td>
<td>67. RiskControlLibrary</td>
</tr>
<tr>
<td>16. CurrencyExchangeRate</td>
<td>42. OnlineOrder</td>
<td>68. Routing</td>
</tr>
<tr>
<td>17. CustomerPartyMaster</td>
<td>43. OnlineSession</td>
<td>69. SalesLead</td>
</tr>
<tr>
<td>18. DebitTransfer</td>
<td>44. Operation</td>
<td>70. SalesOrder</td>
</tr>
<tr>
<td>19. DebitTransferIST</td>
<td>45. Opportunity</td>
<td>71. SequenceSchedule</td>
</tr>
<tr>
<td>21. EmployeeWorkSchedule</td>
<td>47. Payable</td>
<td>73. ShipmentSchedule</td>
</tr>
<tr>
<td>22. EmployeeWorkTime</td>
<td>48. PaymentStatus</td>
<td>74. ShipmentUnit</td>
</tr>
<tr>
<td>23. EngineeringChangeOrder</td>
<td>49. PaymentStatusIST</td>
<td>75. SplitWIP</td>
</tr>
<tr>
<td>25. Field</td>
<td>51. PickList</td>
<td>77. UOMGroup</td>
</tr>
<tr>
<td>28. InspectDelivery</td>
<td>52. PlanningSchedule</td>
<td>78. WIPStatus</td>
</tr>
</tbody>
</table>

**Figure G.1 All OAGIS 9.0 Nouns**
Appendix H  The Core Business Data Interchange Ontology - Terminology

**Actors:** entities of the domain capable of requesting and performing actions in which they are producing, modifying and consuming business objects. Actors work in a coordinated fashion towards accomplishing their goals.

**Roles:** functions that actors play in a context. They are context dependent, i.e. anti-rigid, e.g., a person can be a *client* for a Financial Services company and an *employee*, for a manufacturing plant.

We have identified the following roles: Accountant, Addressee, AssessmentProvider, AssessmentSubject, Broker, Beneficiary, BenefitsOutsourcer, BenefitsPlanAdministrator, Buyer, BusinessPartner, BusinessAssociate, Carrier, Client, Customer, Competitor, Consignee, ChiefEngineer, CAO, CEO, CFO, CTO, CIO, Engineer, Employee, Exporter, FinancialAgent, ForwardingAgent, Importer, Inspector, Initiator, Invoicee, Manager, DivisionManager, Dealership, DepartmentManager, Director, Manufacturer, Payee, PayrolOutsourcer, PayrollServiceEmployee, PlanParticipant, PlanSponsor, ProjectManager, Publisher, Recruiter, Receiver, Referer, Requester, ResearchStaff, RiskManager, SalesContact, SalesLead, SalesPerson, ShippingAgency, ScreeningServiceProvider, ScreeningServiceRequester, Sender, StockPlanAdministrator, StockPlanTrustee, Supplier, Supervisor, TechnicalStaff, TransferAgent, User, VP, President.

We give here several examples of correspondences between the roles in CBDIO and OAGIS XML types:

Payer :: BillToParty, Broker :: BrokerParty, Carrier :: CarrierParty, Customer :: CustomerParty, Employee :: EmployeeParty, Payer :: FreightBillToParty, BusinessAssociate :: JointVentureParty, Manufacturer :: ManufacturerParty, Remitter :: PayFromParty, Sender :: ShipFromParty, Addressee :: ShipToParty, Buyer :: SoldToParty/BuyerParty, Supplier :: SupplierParty, Dealership :: DealershipParty.
We envision that, as industrial and business standards continue to develop, this initial set will be continuously expanded to include other types of roles that will be introduced in the new versions of these standards.

**Scenarios** are generalized representations of business collaborations and a realized through appropriate compositions of actions. Scenarios describe:

- a business purpose
- a collaboration among actors
- events that trigger the collaboration
- post-conditions, i.e., error and success.

For example, a buyer sends a request for a quotation to a supplier and upon comparing the quotation received from the supplier with quotes received from other providers, places a purchase order. The supplier who receives the purchase order accepts it, depending on stock availability, sends the goods and issues an invoice to the buyer.

**Fluents** CBDIO contains the following (extendable) list of fluents:

*Account*(general ledger entity id, general ledger nominal account, effective time period, account type). General ledger (GL) entity id is the primary balancing segment of the GL account structure. Typically, this is the owning entity for the transaction. A G/L entity is the smallest organizational unit for which individual financial statements must be drawn up according to relevant commercial law. A GL nominal account is the primary account classification within a chart of accounts, e.g. fixed asset, land, buildings, etc. Account type is an account classification that specifies the use of this account, e.g., asset, equity, expense, liability, revenue

*CarrierRoute*(tracking ID, actual ship date and time, scheduled delivery date and time, required delivery date and time, promised delivery date, loading date and time, earliest ship date and time) describes a scheduled journey that a particular goods transportation service provider is requested to perform on the behalf of a shipper, customer or other transportation coordinator. A *CarrierRoute* is typically generated by a transportation planning system to coordinate and
optimize the movement of many goods shipment to reduce overall freight cost, improve overall delivery schedules, and ensure that a carrier’s equipment is utilized efficiently. Other business systems typically interested in a CarrierRoute document include supply chain execution, supply chain visibility, warehouse management, and transportation management systems.

Catalog(catalog id, classification scheme, content): a list of items or commodities. The items may be arranged according to a classification scheme. Within the Catalog, each item can be classified into one or more categories. A Catalog has at least one publisher and one or many suppliers for the items in the Catalog.

ChartOfAccounts(): represents the accounting structure of a business. Each account represents a financial aspect of a business, such as its accounts payable, or the value of its inventory, or its office supply expenses.

Credit(customer id, total invoiced, order amount, available amount, order limit amount, total credit limit amount, date time, effective time period, ledger id) represents customer credit information, and is used in the context of credit checking new sales orders. If the sender is the accounts receivable application, the total invoiced is the only required amount field. When the sender is the customer order management application, the order amount is the only required amount field. The order amount is the total of the sales order that is typically used to compare against a credit limit during credit checking. Available amount is the credit amount available for a trading partner to make an additional order against. This amount is typically a combination of orders approved but not yet invoiced and orders invoiced yet still considered open items in the receivables balance. Order limit amount is the credit limit of a single sales order used during credit checking. Total credit limit amount is the total credit limit for a trading partner used during credit checking. Ledger id identifies the financial ledger to be used when updating balances.

CreditStatus(customer id, sales order reference, purchase order reference) represents the credit approval status of a customer or a specific customer order.

CustomerRecords(CRID, content) identifies the customer records.
DispatchList(receiver ID, work center ID, operation reference, production order reference, forecasted time period, setup time period, tear down time period, content) prioritized detailed status of orders and operations scheduled or in-process at a specific work center.

EmployeeWorkSchedule(employee ID, content) represents data related to the planned work hours for an employee; it includes relatively static employee information, such as employee ID and name and also schedule-specific information such as dates and amount of time to be worked.

EmployeeTimeSheet(employee ID, content) refers to time sheet information for an employee. This information may be collected in an external source, and then transferred to a HRMS or Payroll application.

EngineeringChangeOrder(priority code, approval date and time, estimated engineering cost, estimated manufacturing cost, implementation date and time, approval required by date and time, requester, engineering change reviewer, engineering change revised BOM, engineering change revision) can be used to request a change to a manufactured item. Engineering change reviewer identifies the reviewers of the change, whose responsibility it is to approve or reject the engineering change document. Engineering change revised BOM identifies the components of a Bill of Material that will be effected by the given engineering change while engineering change revision describes the actual revision information.

EngineeringWork(work type) can be (1) an engineering work request, a solicitation for some work to be done (2) an engineering work order, the authorization for one or more activities to be performed, e.g. design deviation permit, design release, management resolution, manufacturing release, etc. (3) engineering bill of materials.

InspectionDeliveryReport(inspector ID, supplier ID, purchaser ID, order ID, content) reports the inspection of items identified in the source document, which is typically, a purchase order.

Inventory(content) includes all stocked items and primarily represents the quantities of each item by location.

Invoice(customer ID, order ID) invoice of a customer for goods/services the have been provided.
MaintenanceOrder(requester ID, supervisor ID, planner ID, project reference, failure date and time, reported date and time, scheduled time period, budgeted amount, machine, location ) is an order for a machine, building, tooling or fixed asset to be repaired or for preventative maintenance to be performed.

OnlineOrder(buyerID, date and time, order ID, content)

PersonnelInformation(EmployeeID) is the human resource information maintained for each employee and it includes data such as job code, employee status, department or place in the organization, and job-related skills.

ProductionOrder( execution time period, forecasted time period, need delivery date, due date time, required date time)

PurchaseOrder(order date and time, buyerID, supplierID, accept by date time, content) represents an order to purchase goods from a buyer to a supplier.

Quote(effective time period, promised ship date time, promised delivery date time, delivery time tolerance, content) describes the prices of goods or services provided by a vendor. The quote includes the terms of the purchase, proposals for delivery, and lists the goods or services ordered, as well as their quantities.

Requisition(requester, required delivery date and time, content) is a request for the purchase of goods or services. Typically, a requisition leads to the creation of a purchase order to a specific supplier.

Shipment(content, carrierID, receiverList) describes the goods to be transported by a carrier and delivered to one or more buyers.

WorkInProgressConfirmation(confirmation ID, WIP content): represents confirmation of the movement of WIP materials.
Messages: actors communicate by sending messages to each other requesting actions and receiving acknowledgements. Depending on the type of action in the payload of the message, we distinguish between OperationRequests, InformationRequests, and Responses.

OperationRequests: Cancel, e.g. a purchase order, Post, e.g. an update to an employee time sheet, Add, e.g., Issue, e.g., a product quote, etc.

InformationRequests: Get the content of a purchase order, payroll, general ledger, etc.

Responses: Acknowledgement, e.g., a request for performing an operation was received, it must always follow a request for a processing operation, e.g., to process a purchase order; Confirm, e.g., that a requested operation was performed successfully.

There are 531 message types in CBDIO, corresponding to the BODs included in OAGIS 9.5.

<table>
<thead>
<tr>
<th>AcknowledgeAllocateResource</th>
<th>NotifyPurchaseOrder</th>
</tr>
</thead>
<tbody>
<tr>
<td>AcknowledgeCommercialInvoice</td>
<td>NotifyQuote</td>
</tr>
<tr>
<td>AcknowledgeConfirmWIP</td>
<td>NotifyRemittanceAdvice</td>
</tr>
<tr>
<td>AcknowledgeCostingActivity</td>
<td>NotifyRequisition</td>
</tr>
<tr>
<td>AcknowledgeCredit</td>
<td>NotifyRFQ</td>
</tr>
<tr>
<td>AcknowledgeCreditStatus</td>
<td>NotifyRiskControlLibrary</td>
</tr>
<tr>
<td>AcknowledgeCreditTransfer</td>
<td>NotifyRouting</td>
</tr>
<tr>
<td>AcknowledgeCreditTransferIST</td>
<td>NotifySalesLead</td>
</tr>
<tr>
<td>AcknowledgeDebitTransfer</td>
<td>NotifySalesOrder</td>
</tr>
<tr>
<td>AcknowledgeDebitTransferIST</td>
<td>NotifySequenceSchedule</td>
</tr>
<tr>
<td>AcknowledgeDispatchList</td>
<td>NotifyShipment</td>
</tr>
<tr>
<td>AcknowledgeEmployeeWorkTime</td>
<td>NotifyShipmentSchedule</td>
</tr>
<tr>
<td>AcknowledgeEngineeringChangeOrder</td>
<td>NotifyShipmentUnit</td>
</tr>
<tr>
<td>AcknowledgeEngineeringWorkDocument</td>
<td>NotifyShippersExportDeclaration</td>
</tr>
<tr>
<td>AcknowledgeHazardousMaterialShipmentDocument</td>
<td>NotifyShippersLetterOfInstruction</td>
</tr>
<tr>
<td>AcknowledgeInspectDelivery</td>
<td>NotifySupplierPartyMaster</td>
</tr>
<tr>
<td>AcknowledgeInventoryBalance</td>
<td>NotifyUOMGroup</td>
</tr>
<tr>
<td>AcknowledgeInventoryConsumption</td>
<td>NotifyWarrantyClaim</td>
</tr>
<tr>
<td>AcknowledgeInventoryCount</td>
<td>PostCostingActivity</td>
</tr>
<tr>
<td>AcknowledgeInvoice</td>
<td>PostJournalEntry</td>
</tr>
<tr>
<td>AcknowledgeIssuedInventory</td>
<td>PostMatchDocument</td>
</tr>
<tr>
<td>AcknowledgeJournalEntry</td>
<td>ProcessAllocateResource</td>
</tr>
<tr>
<td>AcknowledgeMaintenanceOrder</td>
<td>ProcessCommercialInvoice</td>
</tr>
<tr>
<td>AcknowledgeMatchDocument</td>
<td>ProcessConfirmWIP</td>
</tr>
<tr>
<td>AcknowledgeMergeWIP</td>
<td>ProcessCredit</td>
</tr>
<tr>
<td>AcknowledgeMoveInventory</td>
<td>ProcessCreditStatus</td>
</tr>
<tr>
<td>AcknowledgeMoveWIP</td>
<td>ProcessCreditTransfer</td>
</tr>
<tr>
<td>AcknowledgeOnlineOrder</td>
<td>ProcessCreditTransferIST</td>
</tr>
<tr>
<td>AcknowledgeOperation</td>
<td>ProcessDebitTransfer</td>
</tr>
<tr>
<td>AcknowledgeOpportunity</td>
<td>ProcessDebitTransferIST</td>
</tr>
<tr>
<td>AcknowledgePaymentStatus</td>
<td>ProcessDispatchList</td>
</tr>
<tr>
<td>AcknowledgePaymentStatusIST</td>
<td>ProcessEmployeeWorkTime</td>
</tr>
<tr>
<td>AcknowledgePickList</td>
<td>ProcessEngineeringChangeOrder</td>
</tr>
<tr>
<td>AcknowledgePlanningSchedule</td>
<td>ProcessEngineeringWorkDocument</td>
</tr>
<tr>
<td>AcknowledgeProductAvailability</td>
<td>ProcessHazardousMaterialShipmentDocument</td>
</tr>
<tr>
<td>AcknowledgeProductionOrder</td>
<td>ProcessInspectDelivery</td>
</tr>
<tr>
<td>AcknowledgeProductionPerformance</td>
<td>ProcessInventoryBalance</td>
</tr>
<tr>
<td>AcknowledgeProductionSchedule</td>
<td>ProcessInventoryConsumption</td>
</tr>
<tr>
<td>AcknowledgePurchaseOrder</td>
<td>ProcessInventoryCount</td>
</tr>
<tr>
<td>AcknowledgeQuote</td>
<td>ProcessInvoice</td>
</tr>
<tr>
<td>AcknowledgeReceiveDelivery</td>
<td>ProcessIssueInventory</td>
</tr>
<tr>
<td>AcknowledgeReceiveItem</td>
<td>ProcessMaintenanceOrder</td>
</tr>
<tr>
<td>AcknowledgeRecoverWIP</td>
<td>ProcessMergeWIP</td>
</tr>
<tr>
<td>AcknowledgeRemittanceAdvice</td>
<td>ProcessMoveInventory</td>
</tr>
<tr>
<td>AcknowledgeRequireProduct</td>
<td>ProcessMoveWIP</td>
</tr>
<tr>
<td>AcknowledgeRequisition</td>
<td>ProcessOnlineOrder</td>
</tr>
<tr>
<td>AcknowledgeRFQ</td>
<td>ProcessOperation</td>
</tr>
<tr>
<td>AcknowledgeRiskControlLibrary</td>
<td>ProcessOpportunity</td>
</tr>
<tr>
<td>AcknowledgeSalesLead</td>
<td>ProcessPaymentStatus</td>
</tr>
<tr>
<td>AcknowledgeSalesOrder</td>
<td>ProcessPaymentStatusIST</td>
</tr>
<tr>
<td>AcknowledgeShipment</td>
<td>ProcessPickList</td>
</tr>
<tr>
<td>AcknowledgeShipmentUnit</td>
<td>ProcessPlanningSchedule</td>
</tr>
<tr>
<td>AcknowledgeShippersExportDeclaration</td>
<td>ProcessProductAvailability</td>
</tr>
<tr>
<td>AcknowledgeShippersLetterOfInstruction</td>
<td>ProcessProductionOrder</td>
</tr>
<tr>
<td>AcknowledgeSplitWIP</td>
<td>ProcessProductionPerformance</td>
</tr>
<tr>
<td>AcknowledgeWarrantyClaim</td>
<td>ProcessProductionSchedule</td>
</tr>
<tr>
<td>AcknowledgeWIPStatus</td>
<td>ProcessPurchaseOrder</td>
</tr>
<tr>
<td>AddPurchaseOrder</td>
<td>ProcessQuote</td>
</tr>
<tr>
<td>AddQuote</td>
<td>ProcessReceiveDelivery</td>
</tr>
<tr>
<td>AddRequisition</td>
<td>ProcessReceiveItem</td>
</tr>
<tr>
<td>AddRFQ</td>
<td>ProcessRecoverWIP</td>
</tr>
<tr>
<td>AddSalesOrder</td>
<td>ProcessRemittanceAdvice</td>
</tr>
<tr>
<td>AllocateCostingActivity</td>
<td>ProcessRequireProduct</td>
</tr>
<tr>
<td>CancelDispatchList</td>
<td>ProcessRFQ</td>
</tr>
<tr>
<td>CancelEmployeeWorkTime</td>
<td>ProcessRiskControlLibrary</td>
</tr>
<tr>
<td>CancelEngineeringChangeOrder</td>
<td>ProcessSalesLead</td>
</tr>
<tr>
<td>CancelEngineeringWorkDocument</td>
<td>ProcessSalesOrder</td>
</tr>
<tr>
<td>CancelInspectDelivery</td>
<td>ProcessShipment</td>
</tr>
<tr>
<td>CancelInventoryBalance</td>
<td>ProcessShipmentUnit</td>
</tr>
<tr>
<td>CancelInvoice</td>
<td>ProcessShippersExportDeclaration</td>
</tr>
<tr>
<td>CancelMaintenanceOrder</td>
<td>ProcessShippersLetterOfInstruction</td>
</tr>
<tr>
<td>CancelPickList</td>
<td>ProcessSplitWIP</td>
</tr>
<tr>
<td>CancelProductionOrder</td>
<td>ProcessWarrantyClaim</td>
</tr>
<tr>
<td>CancelPurchaseOrder</td>
<td>ProcessWIPStatus</td>
</tr>
<tr>
<td>CancelQuote</td>
<td>ReceiveMoveInventory</td>
</tr>
<tr>
<td>CancelReceiveDelivery</td>
<td>ReceiveProductionOrder</td>
</tr>
<tr>
<td>CancelRequireProduct</td>
<td>ReceivePurchaseOrder</td>
</tr>
<tr>
<td>CancelRequisition</td>
<td>RespondConfirmWIP</td>
</tr>
<tr>
<td>CancelRFQ</td>
<td>RespondCredit</td>
</tr>
<tr>
<td>Change Confirm WIP</td>
<td>Respond Credit Status</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>Change Credit</td>
<td>Respond Dispatch List</td>
</tr>
<tr>
<td>Change Credit Status</td>
<td>Respond Employee Work Time</td>
</tr>
<tr>
<td>Change Dispatch List</td>
<td>Respond Engineering Change Order</td>
</tr>
<tr>
<td>Change Employee Work Time</td>
<td>Respond Engineering Work Document</td>
</tr>
<tr>
<td>Change Engineering Change Order</td>
<td>Respond Inspect Delivery</td>
</tr>
<tr>
<td>Change Engineering Work Document</td>
<td>Respond Inventory Balance</td>
</tr>
<tr>
<td>Change Inspect Delivery</td>
<td>Respond Invoice</td>
</tr>
<tr>
<td>Change Inventory Balance</td>
<td>Respond Maintenance Order</td>
</tr>
<tr>
<td>Change Invoice</td>
<td>Respond Online Order</td>
</tr>
<tr>
<td>Change Maintenance Order</td>
<td>Respond Opportunity</td>
</tr>
<tr>
<td>Change Online Order</td>
<td>Respond Pick List</td>
</tr>
<tr>
<td>Change Opportunity</td>
<td>Respond Production Order</td>
</tr>
<tr>
<td>Change Pick List</td>
<td>Respond Purchase Order</td>
</tr>
<tr>
<td>Change Production Order</td>
<td>Respond Quote</td>
</tr>
<tr>
<td>Change Purchase Order</td>
<td>Respond Receive Delivery</td>
</tr>
<tr>
<td>Change Quote</td>
<td>Respond Require Product</td>
</tr>
<tr>
<td>Change Receive Delivery</td>
<td>Respond Requisition</td>
</tr>
<tr>
<td>Change Require Product</td>
<td>Respond RFQ</td>
</tr>
<tr>
<td>Change Requisition</td>
<td>Respond Sales Lead</td>
</tr>
<tr>
<td>Change RFQ</td>
<td>Respond Sales Order</td>
</tr>
<tr>
<td>Change Sales Lead</td>
<td>Show Actual Ledger</td>
</tr>
<tr>
<td>Change Sales Order</td>
<td>Show Allocate Resource</td>
</tr>
<tr>
<td>Create Maintenance Order</td>
<td>Show BOM</td>
</tr>
<tr>
<td>Create Production Order</td>
<td>Show Budget Ledger</td>
</tr>
<tr>
<td>Create Requisition</td>
<td>Show Carrier Route</td>
</tr>
<tr>
<td>Get Actual Ledger</td>
<td>Show Catalog</td>
</tr>
<tr>
<td>Get Allocate Resource</td>
<td>Show Chart Of Accounts</td>
</tr>
<tr>
<td>Get BOM</td>
<td>Show Commercial Invoice</td>
</tr>
<tr>
<td>Get Budget Ledger</td>
<td>Show Confirm WIP</td>
</tr>
<tr>
<td>Get Carrier Route</td>
<td>Show Costing Activity</td>
</tr>
<tr>
<td>Get Catalog</td>
<td>Show Credit</td>
</tr>
<tr>
<td>Get Chart Of Accounts</td>
<td>Show Credit Status</td>
</tr>
<tr>
<td>Get Commercial Invoice</td>
<td>Show Credit Transfer</td>
</tr>
<tr>
<td>Get Confirm WIP</td>
<td>Show Credit Transfer IST</td>
</tr>
<tr>
<td>Get Costing Activity</td>
<td>Show Currency Exchange Rate</td>
</tr>
<tr>
<td>Get Credit</td>
<td>Show Customer Party Master</td>
</tr>
<tr>
<td>Get Credit Status</td>
<td>Show Debit Transfer</td>
</tr>
<tr>
<td>Get Credit Transfer</td>
<td>Show Debit Transfer IST</td>
</tr>
<tr>
<td>Get Credit Transfer IST</td>
<td>Show Dispatch List</td>
</tr>
<tr>
<td>Get Currency Exchange Rate</td>
<td>Show Employee Work Schedule</td>
</tr>
<tr>
<td>Get Customer Party Master</td>
<td>Show Employee Work Time</td>
</tr>
<tr>
<td>Get Debit Transfer</td>
<td>Show Engineering Change Order</td>
</tr>
<tr>
<td>Get Debit Transfer IST</td>
<td>Show Engineering Work Document</td>
</tr>
<tr>
<td>Get Dispatch List</td>
<td>Show Field</td>
</tr>
<tr>
<td>Get Employee Work Schedule</td>
<td>Show Hazardous Material Shipment Document</td>
</tr>
<tr>
<td>Get Employee Work Time</td>
<td>Show Inspect Delivery</td>
</tr>
<tr>
<td>Get Engineering Change Order</td>
<td>Show Inventory Balance</td>
</tr>
<tr>
<td>Get Engineering Work Document</td>
<td>Show Inventory Consumption</td>
</tr>
<tr>
<td>Get Field</td>
<td>Show Inventory Count</td>
</tr>
<tr>
<td>Get Hazardous Material Shipment Document</td>
<td>Show Invoice</td>
</tr>
<tr>
<td>GetInspectDelivery</td>
<td>ShowInvoiceLedgerEntry</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>GetInventoryBalance</td>
<td>ShowIssueInventory</td>
</tr>
<tr>
<td>GetInventoryConsumption</td>
<td>ShowItemMaster</td>
</tr>
<tr>
<td>GetInventoryCount</td>
<td>ShowJournalEntry</td>
</tr>
<tr>
<td>GetInvoice</td>
<td>ShowLocation</td>
</tr>
<tr>
<td>GetInvoiceLedgerEntry</td>
<td>ShowLocationService</td>
</tr>
<tr>
<td>GetIssueInventory</td>
<td>ShowMaintenanceOrder</td>
</tr>
<tr>
<td>GetItemMaster</td>
<td>ShowMatchDocument</td>
</tr>
<tr>
<td>GetJournalEntry</td>
<td>ShowMergeWIP</td>
</tr>
<tr>
<td>GetListActualLedger</td>
<td>ShowMoveInventory</td>
</tr>
<tr>
<td>GetListBOM</td>
<td>ShowMoveWIP</td>
</tr>
<tr>
<td>GetListInventoryCount</td>
<td>ShowOnlineOrder</td>
</tr>
<tr>
<td>GetListItemMaster</td>
<td>ShowOnlineSession</td>
</tr>
<tr>
<td>GetListMaintenanceOrder</td>
<td>ShowOperation</td>
</tr>
<tr>
<td>GetListPickList</td>
<td>ShowOpportunity</td>
</tr>
<tr>
<td>GetListProductionOrder</td>
<td>ShowPartyMaster</td>
</tr>
<tr>
<td>GetListPurchaseOrder</td>
<td>ShowPayable</td>
</tr>
<tr>
<td>GetListQuote</td>
<td>ShowPaymentStatus</td>
</tr>
<tr>
<td>GetListReceiveDelivery</td>
<td>ShowPaymentStatusIST</td>
</tr>
<tr>
<td>GetListRequisition</td>
<td>ShowPersonnel</td>
</tr>
<tr>
<td>GetListRFQ</td>
<td>ShowPickList</td>
</tr>
<tr>
<td>GetListRouting</td>
<td>ShowPlanningSchedule</td>
</tr>
<tr>
<td>GetListSalesOrder</td>
<td>ShowPriceList</td>
</tr>
<tr>
<td>GetListUOMGroup</td>
<td>ShowProductAvailability</td>
</tr>
<tr>
<td>GetLocation</td>
<td>ShowProductionOrder</td>
</tr>
<tr>
<td>GetLocationService</td>
<td>ShowProductionPerformance</td>
</tr>
<tr>
<td>GetMaintenanceOrder</td>
<td>ShowProductionSchedule</td>
</tr>
<tr>
<td>GetMatchDocument</td>
<td>ShowProjectAccounting</td>
</tr>
<tr>
<td>GetMergeWIP</td>
<td>ShowProjectMaster</td>
</tr>
<tr>
<td>GetMoveInventory</td>
<td>ShowPurchaseOrder</td>
</tr>
<tr>
<td>GetMoveWIP</td>
<td>ShowQuote</td>
</tr>
<tr>
<td>GetOnlineOrder</td>
<td>ShowReceiveable</td>
</tr>
<tr>
<td>GetOnlineSession</td>
<td>ShowReceiveDelivery</td>
</tr>
<tr>
<td>GetOperation</td>
<td>ShowReceiveItem</td>
</tr>
<tr>
<td>GetOpportunity</td>
<td>ShowRecoverWIP</td>
</tr>
<tr>
<td>GetPartyMaster</td>
<td>ShowRemittanceAdvice</td>
</tr>
<tr>
<td>GetPayable</td>
<td>ShowRequireProduct</td>
</tr>
<tr>
<td>GetPaymentStatus</td>
<td>ShowRequisition</td>
</tr>
<tr>
<td>GetPaymentStatusIST</td>
<td>ShowRFQ</td>
</tr>
<tr>
<td>GetPersonnel</td>
<td>ShowRiskControlLibrary</td>
</tr>
<tr>
<td>GetPickList</td>
<td>ShowRouting</td>
</tr>
<tr>
<td>GetPlanningSchedule</td>
<td>ShowSalesLead</td>
</tr>
<tr>
<td>GetPriceList</td>
<td>ShowSalesOrder</td>
</tr>
<tr>
<td>GetProductAvailability</td>
<td>ShowSequenceSchedule</td>
</tr>
<tr>
<td>GetProductionOrder</td>
<td>ShowShipment</td>
</tr>
<tr>
<td>GetProductionPerformance</td>
<td>ShowShipmentSchedule</td>
</tr>
<tr>
<td>GetProductionSchedule</td>
<td>ShowShipmentUnit</td>
</tr>
<tr>
<td>GetProjectAccounting</td>
<td>ShowShippersExportDeclaration</td>
</tr>
<tr>
<td>GetProjectMaster</td>
<td>ShowShippersLetterOfInstruction</td>
</tr>
<tr>
<td>GetPurchaseOrder</td>
<td>ShowSplitWIP</td>
</tr>
<tr>
<td>GetQuote</td>
<td>ShowSupplierPartyMaster</td>
</tr>
<tr>
<td>GetReceivable</td>
<td>ShowUOMGroup</td>
</tr>
<tr>
<td>GetReceiveDelivery</td>
<td>ShowWarrantyClaim</td>
</tr>
<tr>
<td>GetReceiveItem</td>
<td>ShowWIPStatus</td>
</tr>
<tr>
<td>GetRecoverWIP</td>
<td>SyncAllocateResource</td>
</tr>
<tr>
<td>GetRemittanceAdvice</td>
<td>SyncBOM</td>
</tr>
<tr>
<td>GetRequireProduct</td>
<td>SyncCarrierRoute</td>
</tr>
<tr>
<td>GetRequisition</td>
<td>SyncCatalog</td>
</tr>
<tr>
<td>GetRFQ</td>
<td>SyncChartofAccounts</td>
</tr>
<tr>
<td>GetRiskControlLibrary</td>
<td>SyncCommercialInvoice</td>
</tr>
<tr>
<td>GetRouting</td>
<td>SyncCredit</td>
</tr>
<tr>
<td>GetSalesLead</td>
<td>SyncCreditStatus</td>
</tr>
<tr>
<td>GetSalesOrder</td>
<td>SyncCurrencyExchangeRate</td>
</tr>
<tr>
<td>GetSequenceSchedule</td>
<td>SyncCustomerPartyMaster</td>
</tr>
<tr>
<td>GetShipment</td>
<td>SyncDispatchList</td>
</tr>
<tr>
<td>GetShipmentSchedule</td>
<td>SyncEmployeeWorkSchedule</td>
</tr>
<tr>
<td>GetShipmentUnit</td>
<td>SyncEmployeeWorkTime</td>
</tr>
<tr>
<td>GetShippersExportDeclaration</td>
<td>SyncEngineeringChangeOrder</td>
</tr>
<tr>
<td>GetShippersLetterOfInstruction</td>
<td>SyncEngineeringWorkDocument</td>
</tr>
<tr>
<td>GetSplitWIP</td>
<td>SyncField</td>
</tr>
<tr>
<td>GetSupplierPartyMaster</td>
<td>SyncHazardousMaterialShipmentDocument</td>
</tr>
<tr>
<td>GetUOMGroup</td>
<td>SyncInspectDelivery</td>
</tr>
<tr>
<td>GetWarrantyClaim</td>
<td>SyncInventoryBalance</td>
</tr>
<tr>
<td>GetWIPStatus</td>
<td>SyncInventoryConsumption</td>
</tr>
<tr>
<td>IssueMoveInventory</td>
<td>SyncInventoryCount</td>
</tr>
<tr>
<td>ListActualLedger</td>
<td>SyncInvoice</td>
</tr>
<tr>
<td>ListBOM</td>
<td>SyncItemMaster</td>
</tr>
<tr>
<td>ListInventoryCount</td>
<td>SyncLocation</td>
</tr>
<tr>
<td>ListItemMaster</td>
<td>SyncMaintenanceOrder</td>
</tr>
<tr>
<td>ListMaintenanceOrder</td>
<td>SyncOnlineOrder</td>
</tr>
<tr>
<td>ListPickList</td>
<td>SyncOperation</td>
</tr>
<tr>
<td>ListProductionOrder</td>
<td>SyncOpportunity</td>
</tr>
<tr>
<td>ListPurchaseOrder</td>
<td>SyncPartyMaster</td>
</tr>
<tr>
<td>ListQuote</td>
<td>SyncPersonnel</td>
</tr>
<tr>
<td>ListReceiveDelivery</td>
<td>SyncPickList</td>
</tr>
<tr>
<td>ListRequisition</td>
<td>SyncPlanningSchedule</td>
</tr>
<tr>
<td>ListRFQ</td>
<td>SyncPriceList</td>
</tr>
<tr>
<td>ListRouting</td>
<td>SyncProductionOrder</td>
</tr>
<tr>
<td>ListSalesOrder</td>
<td>SyncProductionPerformance</td>
</tr>
<tr>
<td>ListUOMGroup</td>
<td>SyncProductionPerfromance</td>
</tr>
<tr>
<td>LoadActualLedger</td>
<td>SyncProductionSchedule</td>
</tr>
<tr>
<td>LoadBudgetLedger</td>
<td>SyncProjectMaster</td>
</tr>
<tr>
<td>LoadInvoiceLedgerEntry</td>
<td>SyncPurchaseOrder</td>
</tr>
<tr>
<td>LoadMatchDocument</td>
<td>SyncQuote</td>
</tr>
<tr>
<td>LoadPayable</td>
<td>SyncRemittanceAdvice</td>
</tr>
<tr>
<td>LoadProjectAccounting</td>
<td>SyncRequisition</td>
</tr>
<tr>
<td>LoadReceivable</td>
<td>SyncRFQ</td>
</tr>
<tr>
<td>NotifyAllocateResource</td>
<td>SyncRiskControlLibrary</td>
</tr>
<tr>
<td>NotifyBOM</td>
<td>SyncRouting</td>
</tr>
<tr>
<td>NotifyCarrierRoute</td>
<td>SyncSalesLead</td>
</tr>
<tr>
<td>NotifyCatalog</td>
<td>SyncSalesOrder</td>
</tr>
<tr>
<td>NotifyChartofAccounts</td>
<td>SyncSequenceSchedule</td>
</tr>
<tr>
<td>NotifyCommercialInvoice</td>
<td>SyncShipment</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NotifyCredit</td>
<td>SyncShipmentSchedule</td>
</tr>
<tr>
<td>NotifyCreditStatus</td>
<td>SyncShipmentUnit</td>
</tr>
<tr>
<td>NotifyCurrencyExchangeRate</td>
<td>SyncShippersExportDeclaration</td>
</tr>
<tr>
<td>NotifyCustomerPartyMaster</td>
<td>SyncShippersLetterOfInstruction</td>
</tr>
<tr>
<td>NotifyDispatchList</td>
<td>SyncSupplierPartyMaster</td>
</tr>
<tr>
<td>NotifyEmployeeWorkSchedule</td>
<td>SyncUOMGroup</td>
</tr>
<tr>
<td>NotifyEmployeeWorkTime</td>
<td>SyncWarrantyClaim</td>
</tr>
<tr>
<td>NotifyEngineeringChangeOrder</td>
<td>UpdateConfirmWIP</td>
</tr>
<tr>
<td>NotifyEngineeringWorkDocument</td>
<td>UpdateCredit</td>
</tr>
<tr>
<td>NotifyField</td>
<td>UpdateCreditStatus</td>
</tr>
<tr>
<td>NotifyHazardousMaterialShipmentDocument</td>
<td>UpdateDispatchList</td>
</tr>
<tr>
<td>NotifyInspectDelivery</td>
<td>UpdateEmployeeWorkTime</td>
</tr>
<tr>
<td>NotifyInventoryBalance</td>
<td>UpdateEngineeringChangeOrder</td>
</tr>
<tr>
<td>NotifyInventoryConsumption</td>
<td>UpdateEngineeringWorkDocument</td>
</tr>
<tr>
<td>NotifyInventoryCount</td>
<td>UpdateInspectDelivery</td>
</tr>
<tr>
<td>NotifyInvoice</td>
<td>UpdateInventoryBalance</td>
</tr>
<tr>
<td>NotifyItemMaster</td>
<td>UpdateInventoryCount</td>
</tr>
<tr>
<td>NotifyLocation</td>
<td>UpdateInvoice</td>
</tr>
<tr>
<td>NotifyMaintenanceOrder</td>
<td>UpdateMaintenanceOrder</td>
</tr>
<tr>
<td>NotifyOnlineOrder</td>
<td>UpdateMatchDocument</td>
</tr>
<tr>
<td>NotifyOperation</td>
<td>UpdateOnlineOrder</td>
</tr>
<tr>
<td>NotifyOpportunity</td>
<td>UpdateOpportunity</td>
</tr>
<tr>
<td>NotifyPartyMaster</td>
<td>UpdatePickList</td>
</tr>
<tr>
<td>NotifyPersonnel</td>
<td>UpdateProductionOrder</td>
</tr>
<tr>
<td>NotifyPickList</td>
<td>UpdatePurchaseOrder</td>
</tr>
<tr>
<td>NotifyPlanningSchedule</td>
<td>UpdateQuote</td>
</tr>
<tr>
<td>NotifyPriceList</td>
<td>UpdateReceiveDelivery</td>
</tr>
<tr>
<td>NotifyProductionOrder</td>
<td>UpdateRequireProduct</td>
</tr>
<tr>
<td>NotifyProductionPerformance</td>
<td>UpdateRequisition</td>
</tr>
<tr>
<td>NotifyProductionSchedule</td>
<td>UpdateRFQ</td>
</tr>
<tr>
<td>NotifyProjectMaster</td>
<td>UpdateSalesLead</td>
</tr>
<tr>
<td></td>
<td>UpdateSalesOrder</td>
</tr>
</tbody>
</table>

**Table 0.1** Messages types in CBDIO
Appendix I Benefits of Formal Representations of Knowledge

We least here several benefits of representing knowledge using formal representational languages.

*Extensibility.* Frameworks using fine-grained logic-based descriptions facilitate the construction of new concepts by providing users with a mechanism that allows them to compose existing term definitions and/or restrictions on relationships between them.

Formal term definitions can also help with the extension of manually maintained classifications. Multi-axial hierarchies, in particular, are very difficult to expand, especially if the extension requires the enumeration of all cross-products between the terms describing two sub-domains. For instance, the creators of the Gene Ontology had to manually introduce terms for all types of *organization and biogenesis* processes, by location, e.g., *Golgi organization and biogenesis, polar granule organization and biogenesis, extracellular structure organization and biogenesis, mitotic spindle organization and biogenesis in cell nucleus*, etc. Such a task could have greatly benefited from a computer-assisted way to generate these terms, and could also have helped avoid possible omission of term definitions.

*Recommendations.* A system built around formally specified ontologies can benefit from more reliable curation and error checking as well as easier alignment with related ontologies.

Logical rules integrated into annotation tools can be leveraged to:

- remind curators to make additional annotations, e.g., if the biomedical community decides that allosteric enzymes are considered to be involved in process and also in its regulation, a recommender can remind curators that if they annotate an allosteric enzyme to a process they should also annotate it to the regulation of that process;

- spot and correct annotation mistakes, e.g., curators could have avoided mistakenly annotating proteins which are involved in the transport of *iron sulphur precursors* to the
term iron ion transport, as these proteins are involved in the transport of iron-containing compounds rather than in the transport of iron ions per se.

*Automated classification.* The manual classification of terms is error prone and the inability to employ automated reasoners to verify the correctness of the classification has led to inconsistencies in the design of many existing knowledge representations.

The translation of some of these manually maintained classifications into logic-based formalisms can uncover many such inconsistencies, e.g., in the Gene Ontology, glutamate receptor activity was initially classified as a type of neurotransmitter activity, which implied that glutamate is always acting as a neurotransmitter, an assertion that is not actually true.

Furthermore, manually constructed hierarchies lack the flexibility that would allow them to aid in solving problems of different natures, in different contexts. There may be multiple ways to organize the concepts in the description of a domain, but a rigid classification may reflect only one way of organizing things, which may not be the most appropriate for the task at hand.

*(Formal) mappings.* In addition to making possible the classification of terms according to various features, fine-grained logic-based descriptions can also help integrate various subdomain representations, e.g., Gene Ontology can be integrated with various external ontologies such as Chebi\(^{160}\), an ontology for chemical compounds, and the Protein Ontology\(^{161}\).

The mappings introduced between various domain ontologies lead not only to (1) their synergistic integration, e.g., deoxyribose 5-phosphate phosphorylation, the “metabolic process which results in the addition of a phosphate group to a deoxyribose 5-phosphate” and given that phosphate group is a term defined in CHEBI, but not in the Gene Ontology, a mapping can be established between GO to CHEBI which will help explicate the meaning of deoxyribose 5-phosphate phosphorylation, but also to (2) the discovery of potential coverage incompleteness in some of the ontologies being integrated, e.g., CHEBI does not include deoxyribose 5-phosphate.

\(^{160}\) [http://www.ebi.ac.uk/chebi/](http://www.ebi.ac.uk/chebi/)

\(^{161}\) [http://pir.georgetown.edu/pro/pro.shtml](http://pir.georgetown.edu/pro/pro.shtml)
Establishing explicit formal mappings between representations of biomedical knowledge has further benefits. For example, the terms defined in the Gene Ontology are meant to take into account the Enzyme Commission classification system, and thus, a change in this classification system will require, for now, the manual re-grouping and re-classification of the corresponding terms in GO. Logic based definitions of sufficiently fine granularity could, however, significantly simplify this task by enabling the automatic re-classification of the terms in the Gene Ontology affected by the changes.

The availability of formal definitions can also expose as inconsistent some of the existing (informal) mappings. For example, as the work on introducing formal definitions for the Gene Ontology terms progresses, some of the current mappings between GO terms that are defined in terms of overlapping cellular components and terms in other conceptual models, e.g., in the database Reactome, which considers non-overlapping cellular components, may be exposed as invalid.

Currently, mappings to related ontologies are in many cases rather difficult to construct, as the granularity of the term descriptions is not adequate and there are substantial differences in coverage, e.g., GO contains terms referencing chemicals that are not defined in CHEBI. In some cases, constructing mappings became not just difficult but also very error prone, due to different term usage in the two ontologies: e.g. finding mappings between CHEBI terms and GO terms is complicated by the fact that GO uses (a) a singular chemical name to represent a class of chemicals that contain the same chemical skeleton or (b) uses the 'chemical and derivative' nomenclature\(^{162}\), e.g. *thiamine and derivative*.

We note, however, that although formal knowledge specifications offer numerous benefits they require a comparatively high level of specialized knowledge to construct and may turn out to be very expensive to maintain.

---