Semantic Query Answering with Time-Series Graphs

Leo Ferres[1] Michel Dumontier[2] Natalia Villanueva-Rosales[3]
[1]Human-Oriented Technology Laboratory
[2]Department of Biology
[3]School of Computer Science
Carleton University
Ottawa, Ontario, K1S 5B6, Canada
{leo_ferres, michel_dumontier}@carleton.ca
nvillanu@scs.carleton.ca

Abstract

Statistical graphs are ubiquitous mechanisms for data visualization such that most, if not all, enterprises communicate information through them. However, many graphs are stored as unstructured images or proprietary binary objects, making them difficult to work with beyond the reports in which they are embedded. While graphs can be mapped to more common XML representations, these lack expressive semantics to discover new knowledge about them or to answer queries at various levels of granularity. This paper describes an OWL ontology that facilitates the representation, exchange, reasoning and query answering of statistical graph data. We illustrate the advantages of using an ontological approach to discover and query about time-series statistical graphs.

1. Introduction

Graphs (bar and line graphs, pie charts, stacked bar graphs, etc.) are complex visual objects aimed at either communicating or exploring numerical data [8]. As such, most, if not all, enterprises (private companies, government organizations and scientific communities) find themselves in the situation of having to work with graphical representations of numerical data. However, few of these enterprises, if any at all, have techniques and standards in place to exchange, search and query these graphical representations. In fact, all graph workers can hope to do is store these representations in their native application's format (such as MS Excel, SPSS, OpenOffice, JGraph, etc.) and simply work with unstructured images (JPEG, GIF, etc.). Although unstructured images of graphs may be enough in certain situations (e.g. during a business meeting, or for typesetting), it seems it would be valuable to archive this graphs as structured or semi-structured representations for 1) the exchange of graph information, 2) building new graphs from existing graphs and 3) discovering new knowledge about the graph.

We have reason to believe (wrt (1) above) that once they are used, graphs usually become archived and are rarely referred to again. To our knowledge, this is so not because these graphs have lost their value, but because there is currently no way of querying, retrieving and manipulating those graphs. It is even problematic for users to find these graphs, unless they know, for example, in which Excel spreadsheet they created them, or where and under what name they saved them [21]. Likewise, even if these graphs could be retrieved, there is no structure in the graphical information such that it would allow for simple querying, such as asking whether the y-axis (or vertical axis) is in "thousands of dollars". In the same vein, (wrt (2) above) even though unstructured information, like images of graphs, can be exchanged easily over email or accessed through databases, most statistical organizations must exchange *structured* information with their counterparts all over the world. Exchanging raster, unstructured images is usually not enough for these organizations. With respect to (3) above, being able to query these graphical representations may provide a level of interaction with them not possible to-date.

It may appear that it is maybe more worthwhile to exchange the original data based on some business or other domain ontology, rather than exchange the *visualization* of the data. However, in creating graphs for mass consumption, authors usually have some communicative purpose in mind [6, 19]. While the data is a necessary condition of the representation, merely adding structure to the data does not seem to be enough as important information may be lost: for instance, the author of the graph may have put special effort in choosing one scale over another for the y-axis, may have put text in parts of the graph will not otherwise be included, may have highlighted part of the graph with a different color, may have assumed that by writing 2003, 2004, etc. in the x-axis was enough to implicitly label the x-axis as 'year', etc. The addition of this information requires a semi-structured approach in which other groups can easily add new knowledge to the graph.

Two additional related problems could be potentially solved by adopting a semi-structured approach to representing graphical information. The first problem has to do with the fact that images are sometimes not easily accessible in certain contexts, such as users of small devices like PDAs or cell phones. Given the small factor of the screen on these devices, it is sometimes hard to display the full graph image. A semi-structured graphical representation would partly solve these issues by allowing intelligent zooming of a rendered graph, or querying the important parts of the graph with some intelligent user interface, etc. Second, it is well known that North American governmental organizations have an "accessibility" mandate, by which they are to provide accessible information to a variety of people in a variety of contexts [1, 18]. Graphical representations of numerical data are quite inaccessible to blind and visually impaired individuals, for example. If graphs carried more structured information, for instance, it would be much easier to render these graphs in other modalities, such as textually. Inspiration for the present work was born from this accessibility problem [12].

The purpose of this paper is to provide a rich formal model (ontology) of graphical representations of numerical data such that it becomes possible to reason about and query the model. The model carries the spirit of our accessibility work (i.e. its requirements), but we provide some evidence that it may provide the basis for a solution to the other problems mentioned above; namely, i) the inability to retrieve and semantically query graphs, ii) the inability to work with a graph model that helps graph exchange, and, of course, iii) the possibility of encoding graphs in other modalities for certain demographics. We first start with a survey of related work (Section 2), continue by detailing the ontology development process (Section 3) and discussing some important aspects of the final ontology (Section 4). We finish by providing some guidelines as to how the ontology presented here may help to solve problems (i–iii) discussed above (Section 5) and with a discussion of future work (Section 6).

2. Related work

There is a large body of work on the different kinds of knowledge needed to create an ontology for graphs; namely, expressive yet decidable and tractable logics, spatial, temporal and diagrammatic reasoning, cognitive studies on graph comprehension, etc. However, despite being such ubiquitous objects, graphs (bar graphs, line graphs, etc.) have not been studied as thoroughly in the knowledge representation (KR) arena as other kinds of allegedly more complex diagrams. For instance, in the diagrammatic representation and reasoning literature [2, 3], graphs are usually mentioned and categorized as diagrams, but their definition using their composing objects and properties are rarely dealt with. In all fairness, this "overlooking" is not hard to defend, given that this literature is more concerned with generalizations about *all* diagrams, rather than the definition of a simple subset.

Application specific data models provide insight into graphs and their components. For example, Office Open XML (commonly referred to as OOXML or Open XML) [9] and Open Document Format [17], are XML-based file format specification for electronic documents such as memos, reports, books, spreadsheets, charts, presentations and word processing documents. These documents define, in the form of XML schemas, all the elements of graphs, such as titles, data points, axes, trend lines, etc. and how they must be encoded to be readable in other applications. Even though a good first step to a defining graphs and their components, these specifications fall short in that they do not really formally encode the *relations* that hold among graph objects, except in maybe indirect ways. As it is, for instance, there is no way of knowing, simple facts such that both the x-axis and the y-axis of a graph are both axes of that graph, or even deeper relations such that if x-axis categories are in years (2001, 2002, etc.) that they stand for some time interval.

Two projects that have tried to integrate many of the techniques described above (a comprehensive graph object model with the reasoning capabilities from the diagrammatic reasoning literature) are those described in [21] and [10]. In the first case, the author provides a solution to graph search by storing graphical information in an XML file (driven by a graph XML Schema) and later, when retrieved, rendering them graphically by the Scalable Vector Graphics (SVG) standard tools. The querying of the XML was carried out by pattern matching, and some inference was carried out on the values of the y-axis to pose queries such as "man && 3.1" to return the year at which that value is present for the keyword "man", say "2003". As the author points out, the inference mechanisms for this system is still quite limited, and the graph representation model is only an XML schema, with no (or only indirect) relationship between the composing objects and very limited querying as to the visual objects composing graphs (such as axes, titles, etc.).

In turn, in [10], the authors attempted a formal definition of graphs and ways to query them by means of a formal language, albeit without the web interface of the previous author. Although fairly comprehensive wrt the kind of information about graphs that it encoded, this work suffered from the peculiarities of the modeling language used (Prolog). In that occasion, we had to limit ourselves to relatively trivial relationships between objects and could say very little about properties that involved some object property, as more recent modeling languages allow. In recent work [11], we designed an XML data structure to encode graphs to support iGRAPH-Lite, an accessibility tool that represents and queries the different parts of a graph in order to render them accessible by a natural language interface. Although comprehensive in the objects it allows, this representation also fell prey to the generality of purpose of XML, compared to RDFS and OWL [16].

The closest to a real ontology for (cartesian) graphs is *GraphRep*, described in [7]. *GraphRep*, however, is not exactly an ontology in itself, but a framework for mapping the visual components of a graph (lines, points, distances, text labels, etc. to a more general ontology of abstract algebra, *EngMath* [14], which defines the conceptual scheme of a mathematical domain (engineering mathematics). This latter domain includes concepts such as measurement scales, length, time and vector quantities, to name but a few. *GraphRep* then allows an ontology of some syntactic graphical objects to be mapped to a conceptual modeling ontology, effectively giving "meaning" to the former kinds of objects. For example, in the case of Figure 1, the length unit for each segment in each axis (expressed in centimeters, pixels, etc.) maps to a unit of measurement (0, 10, ..., 70 billions of dollars) for the y-axis and I, II, ..., or quarters for the x-axis.

Quite a few of these mappings are reported in the paper and, when possible, we have incorporated them. However, at the time of [7] there was no alluding to any knowledge representation language other than KIF [13], a language for first-order logic in which knowledge encoded may be undecidable for reasoning and there is no principled account of where these objects and relations used in the ontology come from, or their coverage.

The purpose of this paper is to extend the ontology development effort of these early authors by publishing a reusable graph ontology that is suitable for solving the problems mentioned in the introduction. The next section describes the methodology used to design and formalize the graph ontology.

3. Methodology

3.1. Requirements

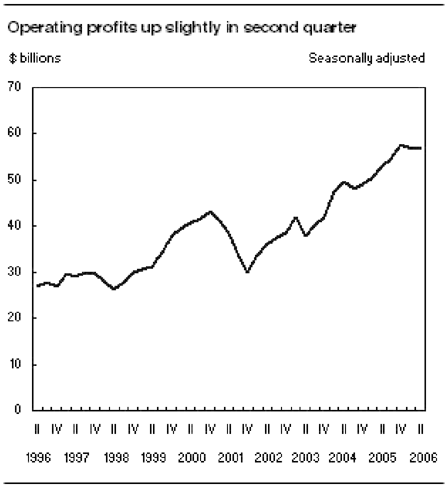

Figure 1. Sample graph from the corpus

The requirements for the ontology presented here were strongly influenced by our work on iGRAPH-Lite, which provides a natural language interface to an enriched XML representation built from application specific graphs. Thus, the ontology would minimally cover the classes and attributes that are required by iGRAPH-Lite. Moreover, we designed a set of competency questions [20] from which to evaluate the coverage and utility of the ontology. Some of these questions are:

1. What is the graph title?
2. What are the titles of the axes of the graph?
3. Do the x-axis categories of a graph correspond to years?
4. What is the unit of the y-axis? (e.g. currency, weight, etc.)
5. What time interval (month, year) corresponds to the hold?
6. Which graphs plot time series data?
7. Which graphs are line graphs?

In order to provide an empirical grounding to the modeling efforts, we compiled a large corpus of graphs, obtained from a Canadian statistical agency's online publication: Statistics Canada's "The Daily"[1]. This publication spans about ten years (mid 1996 to mid 2006) and contains exactly 5060 graphs, of which 3203 (63.3%) were line graphs Essential elements of the ontology were obtained from the 2883 line Graphs (90%) while the remaining 10% (320 graphs) was used to test coverage of the ontology. "Deconstructing" graphs into their components resulted in a "collection of concepts" that was subsequently pruned where overlapping or redefined where ambiguous. To ensure broad coverage, we considered objects defined in both the OOXML and the ODF and that objects in the test-set of graphs could be instantiated in the ontology. Following [20], we grouped the concepts in a simple taxonomy (e.g. both x- and y-axis are a kind of axis) and started defining the relationships in light of existing upper ontologies (see below) and the already available *GraphRep* ontology.

---

[1] http://www.statcan.ca/english/dai-quo/

3.2. The modeling tools

To model the graph domain carved from the graph corpus, we have chosen Description Logics (DL) as the modeling language and OWL-DL (the Web Ontology Language, in its DL form) as the implementation language.

OWL-DL is an XML/RDF(S)-based language designed to represent information about objects and how these are interrelated. In general, OWL extends RDF/RDFS in important respects, primarily on how properties that are local to a document behave. The design of OWL-DL has been strongly influenced by research on DL, a Knowledge Representation (KR) formalism (a fragment of predicate logic) characterized by i) the use of different kinds of *constructors* to define complex concepts out of simpler ones and ii) an emphasis on the soundness, completeness and computational tractability of reasoning problems [4]. OWL-DL is, in effect, a language to implement DL KBs, including assertions (ABoxes) and terminologies (TBoxes)[2].

The basic types of DL are *concepts* (*classes* in OWL) and *roles* (*properties*, in OWL). Concepts are unary predicates ranging over the set of individuals in the domain of discourse, while roles are binary relations associating individuals to some value. There are several families of languages in DL, defined primarily by the kinds of constructors that they will allow. In this case, we will restrict our constructors to those allowed by the OWL-DL 1.1 specification; namely, $\mathcal{SHOIQ}$(D) [16], although the actual ontology is less complex ($\mathcal{ALHI}$(D)). For clarity, we use the Manchester OWL syntax [15] instead of the concrete syntax of OWL-DL to refer to our ontology design decisions, following the convention of capitalizing the first letter in classes (SimpleClass) and writing relationships starting with a lowercase (hasProperty).

4. Graph Objects and Relations

4.1 Ontology Design Principles

In designing the graph ontology we determined several desirable features. First, to maximize reuse graph terminology should be available for any purpose where the terminology was required. Second, logical assertions on graph entities should be universally true, that is to say, true for every instance and under all circumstances. Third, application specific logic including document validation should be described using the same knowledge representation language and use logical assertions that simply place further restrictions on what is universally true.

To address these issues we applied a novel three layered approach in our ontology design. The first layer is composed of graph primitives[3] that are defined axiomatically from fundamental notions about graphs and their components. The second layer describes graph complex class expressions [4] that aim to enhance the description of primitives such that automated reasoning may be performed. The third layer, the iGraph layer[5], places specific restrictions on the types of relations between graph components in order to define the necessary conditions for creating valid iGraph documents.

4.2 Statistical Graph Ontology

The Statistical Graph Ontology (SGO) describes graphs and its components. It is composed of sixty-six classes, twelve object properties and six datatype properties. For the purposes of this paper, we will concentrate only on those issues that are of particular interest while defining a time-series data graph such as the one in Figure 1, which is used as the example throughout the paper.

A Graph hasPart one or more Plot and may (hasTitle) one or more Title (PrimaryTitle or SecondaryTitle) or hasSource Source of origin. The Plot hasPart one or more Series and hasPart one or more CoordinateAxis (XAxis, YAxis or ZAxis) which may hasPart one or more CategoryAxis (PrimaryCategoryAxis or SecondaryCategoryAxis) or one or more ValueAxis (LeftValueAxis or RightValueAxis).

A CategoryAxis hasPart CategoryData, and more specifically PrimaryCategoryAxis hasPart PrimaryCategoryData whereas SecondaryCategoryAxis hasPart SecondaryCategoryData. CategoryData may be ordered using isBefore or isAfter properties. In contrast, a ValueAxis may be used to plot continuously varying numerical data (ValueData) and may be scaled (ScaledValueAxis such as LinearValueAxis, LogarithmicValueAxis and SemiLogarithmicValueAxis).

A Series hasPart two or more DataPoint hasPart Data to be plotted from the set of CoordinateAxis. Data, or more specifically CategoryData or ValueData hasValue a concrete datatype value (e.g. int, float, double). Certain properties are defined as transitive (hasPart and its inverse isPartOf) such that if graph1 hasPart plot1 and plot1 hasPart series1, then a graph1 hasPart series1.

4.3 Augmenting the SGO

Since we can expect that applications that instantiate the SGO will use the most basic primitives (i.e. Title) instead of the most specific class (i.e. GraphTitle), we would like to provide the capability to automatically discover such cases.

---

[2] For an in-depth treatment of the issues we do not cover here for lack of space, the reader is invited to consult [16], on which we base most of our theoretical background.

[3] http://ontology.dumontierlab.com/statistical-graph-primitive

[4] http://ontology.dumontierlab.com/statistical-graph-complex

[5] http://ontology.dumontierlab.com/statistical-graph-igraph

In realizing instances as their most specific type, we may formulate queries using more complex terminology. Thus, we augment the SGO using the expressiveness afforded by OWL-DL such as disjunction, negation, union, intersection, class equivalence, existential and universal restrictions and (qualified) cardinality restrictions. Thus, we can define a GraphTitle as:

Title **that** isTitleOf **some** Graph

Thus, a DL reasoner would classify all graph titles during the reasoning process. Importantly, this terminology becomes part of the vocabulary that can be used to formulate a meaningful query.

4.4 Defining the iGraph requirements

While the previous section alludes to how graphs and their components are related, we can strictly enforce this logical description in another ontology. A benefit then is that documents may be validated on whether instantiated graph ontologies have the necessary class membership restrictions and relations between objects. Moreover, this can provide the mechanism by which the integrity of application data may be specified. For instance, the corpus of LineGraph from Statistical Canada is defined in the iGraph ontology[6] and described here using the Manchester OWL syntax:

```
Graph and hasTitle some PrimaryTitle
 and hasPart some (Plot
     and hasPart some (XAxis
         and hasPart some (PrimaryCategoryAxis
             and hasPart some CategoryData))
     and hasPart some (YAxis
         and hasPart some (ValueAxis
             and hasTitle some Title
             and hasStart some float
             and hasEnd some float
             and hasStep some float))
     and hasPart some Line
     and hasPart some (Series
         and hasPart min 2 DataPoint
         and hasPart some (DataPoint
             and hasPart some (CategoryData
                 and hasValue some string
                 and isBefore some CategoryData)
             and hasPart some (ValueData
                 and hasValue some float))))
```

Following this description, the Graph in 1 is classified as a LineGraph because it satisfies the restrictions. More specifically: it has the PrimaryTitle "Operating profits up slightly in second quarter" and contains the Plot *plot*. *plot* contains an XAxis *xaxis* that contains the PrimaryCategoryAxis *primary_category_axis* that contains PrimaryCategoryData *x1*, *x2*, etc. *plot* also contains the YAxis *yaxis* that contains the ValueAxis *value_axis*. *value_axis* holds the title of "$ billions" also has three datatype properties which specify the axis start value of "0", an end value of "70" with steps of "10" units.

plot also contains Series *series1* with more than two DataPoint such as *datapoint1*, *datapoint2*, etc, each of which contain CategoryData and ValueData. For instance *datapoint1* contains PrimaryCategoryData *x1* and ValueData *y1*. *x1* holds the string value of "III" and *y1* holds the float value of "27.9".

4.5 Adding Contextual Knowledge to Graphs

An important consideration in the interpretation of graphs is that they represent the comparison of certain *types* of data. For instance, 1 plots currency (Dollar) against time intervals (Quarter and Year). While these types fall outside the scope of the SGO, there exists mechanisms (ontology integration and mapping) to reuse ontologies pertaining to those domains and be able to explicitly categorize the data. Moreover, new inferences may be drawn from the assignment of types to our data.

We have imported a time interval ontology[7] and asserted the specific type (FirstQuarter, SecondQuarter, ThirdQuarter and FourthQuarter) to the PrimaryCategoryData. This approach enables new reasoning to occur with respect to time over our graph data. We demonstrate the utility of this approach in the next section.

4.6 Ontology Population

The ontology was populated using XML documents of statistical graphs generated by iGRAPH-Lite . iGRAPH-Lite is composed of at least three general processing parts. The first is application specific and involves communicating with different kinds of graph applications (MS Excel, OpenOffice, etc.). This module outputs a simple representation R of the necessary conditions to reconstruct a graph G. A second module takes R as input and runs several algorithms aimed at enriching R by, for example, recognizing the x-axis categories (years, quarters, months, etc.), recognizing the unit of the y-axis, recognizing titles (if they have not been set by the usual "chart properties" option, finding minimums and maximums, correlating both categories (say, year and quarters) in two-category x-axis, to name but a few. This module outputs a much richer representation of R, let us call it R'. Finally, the third module is the language generation module, which queries R' and generates natural language descriptions of the graph.

The iGRAPH-Lite ontology was populated by mapping the data contained in R' from the test set of iGRAPH-

---

[6] http://ontology.dumontierlab.com/igraph

[7] http://ontology.dumontierlab.com/time-interval-primitive

Lite processed graphs. Given that OWL can be represented with an RDF/XML syntax, the mapping was performed using XSLT transformations[8] based on the iGRAPH-Lite XML schema for data export. These XSLT transformations can be extended/adapted to support ontology evolution and further data exchange between the ontologies described in this paper and iGRAPH-Lite.

5. Practical Implications

This section discusses the problems we set out to solve and how having an ontology of graphs may be able to solve them. We start by giving the reader a general idea of the kinds of queries that can be posed to the system, and the kinds of retrieval to be expected. We then concentrate on the advantages of this approach for data exchange and finally we discuss our own work on accessibility and how the ontology facilitates multimodal presentation.

5.1 Semantic query answering over Statistical Graphs

To illustrate the utility of the SGO, we first populated the ontology [9] with the data from 1, and then used the Query Plugin of Protégé 4 alpha (build 53)[10] with the embedded Fact++ DL reasoner for consistency checking and query answering.

We use the Manchester OWL syntax to construct our query expression. A query to retrieve all the data points in the graph would be written as follows:

DataPoint **that** isPartOf **some** Graph.

Since parts of a graph are related with the hasPart transitive relation, this query is answered because each data is part of a series, which in turn is part of a plot, which in turn is part of a graph. Thus, the data returned, among others, is *datapoint0, datapoint1, ..* These data points are composed of the x-axis/categorical data and the y-axis/value data.

A second query demonstrates how we can leverage asserted contextual knowledge to find value axis data plotted against time interval data (i.e. the second quarter of any year):

ValueData **that** isPartOf **some**
(DataPoint **that** hasPart **some**
(SecondQuarter)).

The individuals that satisfy this class contain, among others, *y0* with value "26.7" and *y4* with value "28.9".

A third query demonstrates how the reasoner will make infer class membership based on a role-based description similar to that described in 4.3. Querying solely for AxisTitle, we discover a *primary_cat_axis_title* with value "Quarter", *secondary_cat_axis_title* with value "Year", and *val_axis_title* with value "$ billions". Notice that each was asserted as instances of Title, and were *inferred* to be instances of AxisTitle.

A fourth query shows how complex definitions using concepts defined elsewhere can be made. In this case, we want to retrieve all graphs that contain time intervals, or TimeSeries:

Series **and** hasPart **some** TimeInterval.

This query returns the individual *Series1*. This is because we have asserted the category data as specific type of time interval whose type relationship to the TimeInterval class is defined in an imported ontology (using owl:imports). Since *series1* is *asserted* as a Series that hasPart **some** DataPoint and each data point (e.g. *datapoint1*) contains CategoryData (e.g. *x1*) of a specific type of Quarter (e.g. ThirdQuarter), and Quarter is a TimeInterval, it follows that each data point of *series1* is a TimeInterval and *series1* can be classified as a TimeSeries

Finally, the semantic annotation of the graph data also provides a mechanism by which we can query a knowledge base to find all graphs containing a certain "type" of data. Querying about the example graph and a second graph [11] which has months as the primary category axis, we ask:

Graph **that** hasPart **some** TimeInterval.

This query returns the two graphs *example1:graph, example2:graph* defined in the instantiated ontologies. Additional annotation of the value axes will allow users to create more sophisticated queries in which they can find graphs having certain types of category and value axes, thus being able to differentiate between graph having the same type of category axis.

5.2 Data Exchange

The ontology facilitates data exchange by i) providing a standard (syntactic) representation of the data for machine consumption, ii) the meaning (semantics) of the data represented is unambiguous, formally represented and machine understandable, iii) the three layer approach offers a more flexible model to exchange the data encoded, i.e. a potential user does not have to "commit" to semantics defined over the hierarchical organization of the classes and properties or the requirements of iGRAPH-Lite. Instead, a potential user can exchange the assertions included in the primitive ontology and then extend this ontology to satisfy her

---

[8] http://www.w3.org/TR/xslt

[9] http://ontology.dumontierlab.com/igraph-example

[10] http://protege.stanford.edu/

[11] http://ontology.dumontierlab.com/igraph-example2

requirements, and finally iv) given the flexibility of XSLT transformation, new inferences, for example new defined class memberships obtained by the DL reasoner can be returned to the application systems.

5.3 Accessibility

A word should be said about accessibility to graphical representations by blind and visually impaired individuals, which inspired this work. Structured representations of graphical (or any other kind of) information are a *sine qua non* step for rendering information accessible to different communities. In our case, we have started to use the present ontology to construct an alternative representation of graphs in the form of a Natural Language (NL) descriptions of them. We are presently investigating how to associate linguistic ontologies (see, for example, [5]) to each of the concepts in the graph ontology discussed here. In this fashion, algorithms at the application level will be able to query the instances and associate concepts with linguistic expressions, or, in other words, to how those concepts should be communicated in natural language.

6. Conclusions and Future Work

A statistical graph ontology is useful for several reasons. First, in our own work on accessibility, having such structured and well-defined semantics for graphical representations means that ever more things can be "stated" when rendering graphs accessible multimodally [11]. Second, our supporting organizations (Statistics Canada and Cognos, Inc.) are currently unable to work with graphical representations of data. Although they have all the necessary elements of a rendered graph either at the application or the relational database level, they do not possess a principled and empirically tested way to infer new knowledge about graphs, to search their "parts" and to query information across different graphs. By using the SGO, these limitations are effectively overcome.

As a visual representation, it would be desirable to reason in terms of the spatial locations occupied by graph objects. Examples of these queries might include "does the title *overlap with* the plot area?", "what is adjacent to the y-axis title". Similarly, the assertion of time intervals on category axes opens the door to perform temporal reasoning, something we do now somewhat primitively. While there is no explicit support in OWL for spatial and temporal reasoning and inference, this can be done by i) asserting a set of spatial/temporal relationships that are decidable, ii) create extensions that operate over OWL ontologies, or iii) we can transform the OWL KB into a system that can perform specialized spatial and temporal reasoning.

We believe this ontology is a contribution to enterprises working with graphical representations of statistical data at a good level of granularity and it also provides a solution for the three problems identified above: i) the inability to retrieve and semantically query graphs, ii) the inability to work with a graph data model such that it helps graph exchange and graph merging, and iii) the possibility of encoding graphs in other modalities for certain demographics, by coherently handling the different parts of graphical objects into a unified conceptual model.

7. Acknowledgments

The authors would like to thank the three anonymous reviewers who suggested valuable changes to improve the quality of this paper. This paper was written with the support of Statistics Canada (Funds 301706 and 301710), ORNEC, NSERC (CRD 3463814-06) and Cognos, Inc. (Funds 360275) to L. Ferres, a CONACYT scholarship (#150581) for N. Villanueva-Rosales and NSERC DG for M Dumontier.

References

[1] U.S. Section 508 of the Rehabilitation Act: Electronic and Information Technology Accessibility Standards. Available at `http://www.section508.gov/index.cfm?FuseAction=Content&ID=12`, 1998.

[2] M. Anderson, editor. *Reasoning with Diagrammatic Representations II, Papers from the 1997 Fall Symposium*. Technical Report FS-97-03. AAAI Press, Menlo Park, CA, 1997.

[3] M. Anderson. *Diagrammatic Representation and Reasoning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[4] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[5] J. A. Bateman, R. Henschel, and F. Rinaldi. Generalized Upper Model 2.0: documentation. Technical report, GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany, 1995.

[6] J. Bertin. *Semiology of Graphics*. The University of Wisconsin Press, Madison, WI, 1983.

[7] P. C. H. Cheng, J. Cupit, and N. R. Shadbolt. Supporting diagrammatic knowledge acquisition: an ontological analysis of cartesian graphs. *Int. J. Hum.-Comput. Stud.*, 54(4):457–494, 2001.

[8] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229:828–833, 1985.

[9] Ecma International. Office Open XML File Formats. Available at `http://www.ecma-international.org/publications/standards/Ecma-376.htm`, 2006.

[10] L. Ferres, A. Parush, Z. Li, Y. Oppacher, and G. Lindgaard. Representing and querying line graphs in natural language: The iGraph system. In *SmartGraphics 2006*, number 4073 in Lecture Notes in Computer Science, pages 248–253, Berlin Heidelberg, 2006. Springer-Verlag.

[11] L. Ferres, A. Parush, S. Roberts, and G. Lindgaard. Helping People with Visual Impairments Gain Access to Graphical Information Through Natural Language: The iGraph System. In *Proceedings of the 10th International Conference on Computers for Handicapped Persons*, Lecture Notes in Computer Science. Springer–Verlag, 2006.

[12] L. Ferres, P. Verkhogliad, G. Lindgaard, L. Boucher, A. Chretien, and M. Lachance. Improving accessibility to statistical graphs: The inspectgraph system. In *Proceedings of the Ninth International ACM SIGACCESS Conference on Computers and Accessibility*, Tempe, Arizona, 2007.

[13] M. L. Ginsberg. Knowledge interchange format: the KIF of death. *AI Magazine*, 12(3):57–63, 1991.

[14] T. R. Gruber and G. R. Olsen. An ontology for engineering mathematics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94). Bonn, Germany, May 24-27, 1994.*, pages 258–269. Morgan Kaufmann, 1994.

[15] M. Horridge, N. Drummond, J. Goodwin, A. Rector, R. Stevens, and H. H. Wang. The manchester owl syntax. In *Proceedings of the OWL: Experiences and Directions Workshop Series*, 2006.

[16] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1(1):7–26, 2003.

[17] Organization for the Advancement of Structured Information Standards (OASIS). Open Document Format for Office Applications (OpenDocument) v1.0. Available at `http://www.oasis-open.org/committees/download.php/12572/`, 2005.

[18] Treasury Board of Canada. Common Look and Feel for the Internet - Accessibility. Available at `http://www.tbs-sct.gc.ca/clf-nsi/inter/inter-01-00_e.asp`, 2004.

[19] E. R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, CT, USA, 1986.

[20] M. Uschold and M. Grüninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[21] M. Wollowski. Search and inference with diagrams. In *Proceedings of the The Ninth IASTED International Conference on Internet & Multimedia Systems & Applications*, Honolulu, HI, August 2005.