An Object Zooming Model for the Semantic Web

Tuan Anh Ta, Jean-Marc Saglio
GET-ENST, CNRS-UMR5141, 46,rue Barrault, Paris, 75013, France
Email:{ta|saglio}@enst.fr

Abstract

Zooming techniques have been used in many applications to supply users with a cognitive way of exploring data/information. By zooming, a user changes focus to have an overall or a detail view. On the Web, there is always a sizable amount of resources which we can not visualize all in a window. At any moment users are only capable of concentrating on a visualization space that does not contain too dense information. Starting from the idea of semantic zoom where the fewer objects we see the more details we get, we propose a model that helps users to explore resources dynamically. Our model presents a way for metadata querying where users can specify not only a filter but also a restriction window defined by a number of objects appearing. We believe that this work can provide a new framework for browsing the Semantic Web.

1. Introduction

Due to the continuous growth of the amount of reachable web resources and their intricate nature, future semantic web applications will face the need of new efficient techniques for exploring resources. Actually, web applications are mostly based on a browsing interface where users navigate from a category of resources to another category, like web portals. Certainly, this technique is useful for many application fields, but it is inefficient when the number of resources per category become very large. Consequently, users will have difficulties finding the interesting information.

The aim of this paper is to contribute to the definition of an efficient technique for exploring the Semantic Web. Our proposal was inspired by the dynamic exploration techniques developed in the field of information visualization [Spe01]. All of these techniques rely on a presentation model which can help users to find objects in a dense information space.

Let's briefly introduce the main techniques that we considered, namely "dynamic query", "attribute explorer" and "neighbourhood explorer". In the first technique, dynamic queries are used as interactive filters on the object attributes. They focus user interest on the result by applying restrictions progressively on the objects as they appear. Second technique, "attribute explorer" technique [Smi01] differs from "dynamic query" by the way the restrictions are set. While in the first restrictions are specified directly on the values of each attribute, the second enables restrictions to be set on aggregate values from an histogram presentation.

In the third technique of "neighbourhood explorer", when a user concentrates on a centric object, he can change focus on one of the object's neighbours, following an attribute-dimension. In this last case, the neighbourhood characteristics are based on an ordering of attribute values.

This paper introduces a dynamic exploration model which relies, like the "attribute explorer" technique, on the statistical distribution of object attribute values.

Its main idea is to define a dynamic view as a window which automatically selects a given amount of objects whose minimum "worth" depends on the size of the window. Putting such views together in sequence will create an exploration framework for objects in a metadata base. A panning of view would help to see neighbourhood objects of the same "worth", while a zooming-out or zooming-in would help to see different sets of objects, all of the same given cardinality, with greater or lesser minimum "worth".

The rest of the paper is organized as follows: in section 2 we will point out the problem of Semantic Web browsing that motivates our research. Section 3 describes our zooming model on objects. Section 4 presents related works on the zoom concept. Finally, we have a provisional conclusion with some perspective on further works.

2. Semantic web browsing problem

Initiated by W3C, the Semantic Web is now an active research direction that aims at bringing semantics to the Web itself. As an infrastructure of the Semantic Web, a resource description framework, RDF in short, has been recommended for describing metadata on the Web.

By using description languages as RDF [Las99] and RDFS [Bri02], users can describe web resources under descriptor classes by which semantic frames are formalized for resource descriptions. As an example of resource descriptions on the Semantic Web, we reuse the knowledge schema developed in the C-Web project [Ama01]. In this schema, a description base is constructed with ontology classes, semantic objects, and resource descriptors.

Figure 1 shows a simple example of description base on the solar system. The top of the figure shows a class of celestial bodies with three properties: name, mass (i.e. weight) and aphelion (i.e. distance from the sun). For each body object we have to supply information on these properties. In our example, mass and aphelion are given as ratio values with respect to the Earth. Note that the number of celestial bodies can reach into the thousands (the TransNeptunian Object Varuna was registered at the end of year 2000 with number 20 000). At the bottom of the figure, resource descriptors are represented as triangles. A resource descriptor reports metadata for a resource on the Web. Such metadata could give documentary descriptions such as resource type, created date, etc.. Its presentation could be a post like in Weblogs systems where description is mainly a comment. It could also bring more semantics if the description refers to a semantic object such as Venus, Earth or any other celestial body in the sample. Thus we use resource descriptors to link resources to their celestial body subjects. In the figure an arrow denotes a description predicate on RDF model.

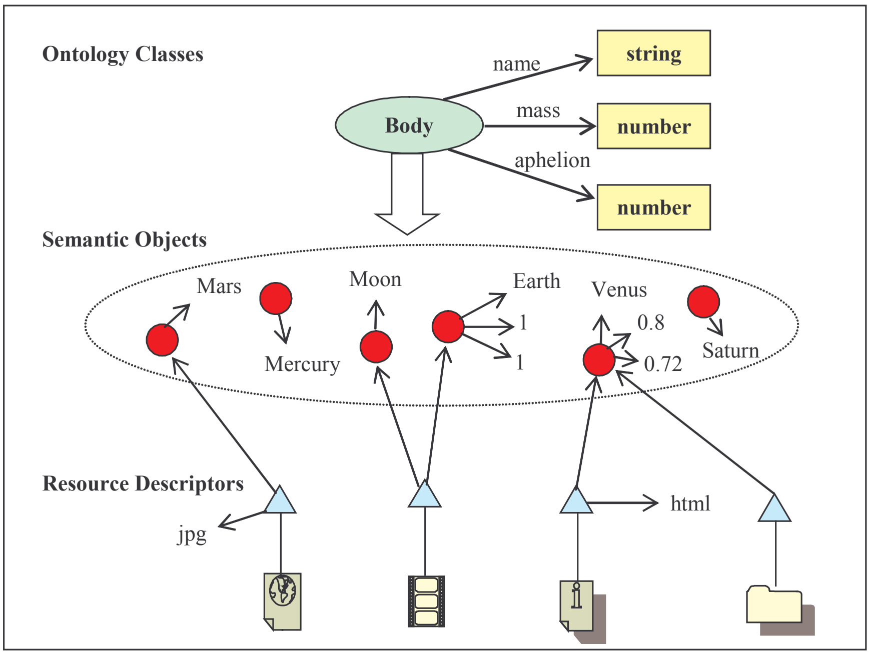

Figure 1: A sample for resource descriptions on the Semantic Web

To explore resources, users first ought to find, from this metadata base, interesting objects that define the subject of resource searching. In the example, when one chose the solar system, he will see the celestial bodies on which he could discover resources. Since the number of bodies, in scientific solar system databases, is, as we said, very large, he can not browse them one by one, neither view all of them at a time. He needs an efficient selective browsing technique. Zoom based techniques are quite good answers for this need. By enabling interactively changing the focus based on the appearance of objects in the result, these techniques can make the user tasks easier. Like a lens, users can zoom-out to see many objects with less details/attributes or zoom-in to see fewer objects with more details/attributes. They can also pan the lens to change view target on other objects, with fixed level of details. A mix of both could create a kind of adaptable pan & zoom.

This pan and zoom interactive exploration is classical in Geographical Information System, or in Zoomable Interfaces, as we will see in section 4. The model we shall present in next section is, as with "attribute explorer" technique, more data distribution sensitive and adaptable.

3. A zooming model for exploring object metadata

Since our exploration model only considers literal attributes of objects, without loss of generality we use a relational data model to simplify the presentation in this paper. Suppose that the body objects in the solar system are represented by a simple relation

Body = (**oid**, name, mass, aphelion).

A zoom is thus defined as a restriction view on the relation in order to capture only a limited number of objects for visualizing. Assuming a zoom on celestial bodies could be made on a restriction from two attributes mass and aphelion. We narrow the zoom on bodies having a mass greater than m and aphelion around target A in interval w (i.e., A-w ≤ aphelion ≤ A+w). We assign F to the maximum number of objects that could appear in the zoom. The data view for the zoom is defined by the query: "try to select F records from R where A-w ≤ aphelion ≤ A+w and mass ≥ m". In this query, w and m are adjustable parameters, while F and A are constant. By adjusting m together with w, we could select a maximum of visualized objects that do not exceed F. There is certainly a relationship between m and w to assure that at least F objects, and no more, will appear in the zoom.

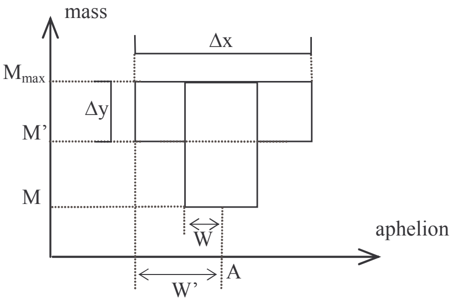

Figure 2: A zoom is a restriction window

Graphically, a zoom can be represented as a restriction window on an exploration space (figure 2). Suppose that after an adjustment for a maximum objects appearing in the zoom, w gets a value W and m a value M. The restriction window can be represented as a rectangle which bounds all objects appearing in the zoom. Note that M_{max} is the maximum mass of all celestial bodies in the database. If users now adjust w to W' where W' > W, automatically, m should also be adjusted to M' where M' > M in order to keep an amount of F objects within the restriction window. This adaptation of the zoom window is an interpretation on the need of users as follows. Suppose that the zoom is focused on 3 objects (F=3). It means that users want to select only 3 celestial bodies for discovering related web resources. To allow the users to see bodies starting from the Earth, the target A is set at the beginning to the Earth's aphelion (i.e., A=1). By seeing objects appearing in the zoom, they find the Earth and two smaller celestial bodies (e.g., Venus, Earth, and Mars). In order to find out planets heavier than the Earth, they increase the aphelion parameter w. Automatically, the m minimum mass is increased till two heavier bodies appear, Jupiter and Saturn.

For a generalization, we denote Δx filter range on the aphelion and Δy on the mass. Δx and Δy are called adjustable restriction zoom parameters. We call F focal threshold which specifies a maximum number of objects appearing in a zoom. If we change the focal threshold of a zoom, we obtain a zoom in/out operation. Figure 3 shows a zooming in where the focal threshold is reduced from F to F'. The zoom parameter Δx and Δy will be automatically adjusted for each focal threshold of the zoom.

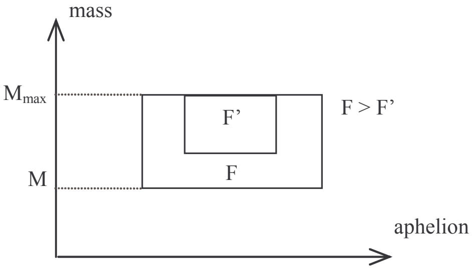

Figure 3: Changing the zoom to view fewer objects

A zoom panning in our model means a shift of restriction window on an attribute dimension. For example, changing the aphelion target A makes a panning (figure 4).

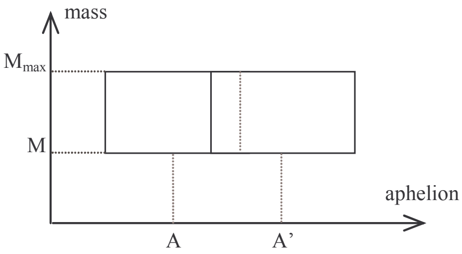

Figure 4: Panning the zoom

More generally, a zoom in/out on n-dimension restriction window could be also modelled. Such a zoom allows users to exploit an exploration on n attributes of objects. A zoom is well defined if all its restriction parameters (i.e., Δx, Δy, …) are computable. In the presented example,

$$\Delta x = [A-w, A+w] \text{ and } \Delta y = [m, M_{max}].$$

In our model, we choose one attribute which we call the "worth" or "granularity" attribute – m in the example above – to be dependent of all the others which we call "exploration" attributes – w in the example above. Technically, users have to trim values for n-1 restriction intervals. The restriction on the last parameter is automatically calculated from the others, according to object distribution in the exploration space. In the solar system database, changing w involves an automatic change of minimum m.

We can imagine the sample zoom as a magic lens with three buttons: the first tunes the number of celestial bodies needed (F), the second allows users to pan the lens (A), and the last serves to adjust the width of the lens (w), then simultaneously the granularity (m).

On object zoom computing

A zoom is in fact a view on data which can be defined as a query with parameters. Let us return to the preceding example where a zoom was created on three parameters F, A, and w for a mean of "zooming on the F heaviest celestial bodies whose aphelion stands between A-w and A+w". The query for this zoom can be written with a free syntax

```
SELECT oid FROM Body
WHERE |aphelion-A|>w
ORDER BY mass TRUNCATE TO F.
```

Changing parameter values of the zoom involves recalculating on the data view. Calculating this view always needs a truncation on F first ordered objects of the result. A simple cut on complete results is the simplest way of evaluating this kind of query, but we waste time selecting and then rejecting most.

[Ozs02] proposed an algebra extension for relational-object databases that help to resolve a similar problem: select only objects of highest importance. More recently [Lern03] proposed a new query language, called "AQuery", as an extension of SQL2 with a new clause called ASSUMING ORDER which allow to define as array-tables any of the tables identified in the FROM clause. This proposition is very promising for computing queries like our object zoom query, similar to the classical "top 10" - here above "top F" - query.

However, such truncating queries fail to indicate the boundary of the restriction window because we may lose objects in the boundary window in the case it contains more than F objects. Thus it is better either to get all the objects found on the boundary or to reduce a little the boundary. In both these ways, the number of objects in the zoom just approximates to F. This tactic avoids the case where two objects have the same value, one appearing in the zoom while the other does not.

We propose a new marker restriction operator in databases which can help to resolve this problem. Unlike a relational restriction based on attribute filters, a marker restriction is computed by propagatively scanning objects in a region that is positioned and limited by markers. A marker is used to instruct how to scan objects on an attribute (e.g., from which value to which value, or from the nearest to the farthest of a centric value,…). The scanning process will be stopped when the number of scanned objects reaches the threshold of the zoom. Normally, a n-dimension zoom need n markers. Note that a marker restriction could product different results if the given markers are not enough to specify a unique "way of scanning". The final goal of a maker restriction is to find an adapted restriction window for a zoom.

We believe that an extension with a scanning operator will improve the response time in comparison with the use of truncating queries. By creating a multidimensional index [GG98] on the scanning attributes, we could find rapidly neighbours of an object without scanning all the objects in the space. Such an indexing technique would be more efficient when the number of objects is very large. We hope that our further researches will prove this hypothesis.

Interacting on a zoom (zooming in/out, panning) is simply to pass from a view to another one among the aggregate views generated for all zoom parameter values. Since materialized views promise high performance improvements, zooming interaction could be more rapid if all zoom views are computed in advance. Unfortunately, this plan is not viable, as the number of possible aggregate views is exponential in the range of zoom parameter values. Furthermore, much like a cache, the views get dirty whenever the object data is modified. There is a risk of paying a high price each time new data is shipped to the database. For a balance between response time and costs of storage and view maintenance, a view selection problem has been adopted in [Chi01]. In fact, a view selection consists in selecting some better views among others that minimizes query response time the most. The technique of dynamic view management [Kot99] is more adequate when it minimizes query response time by computing answers from materialized view fragments in a view pool. It is dynamic because instead of selecting views, it computes and stores only view fragments that can be further exploited. We believe that a study on these techniques will also help us to find an optimization zoom computing based on materialized view fragments.

4. Related work

The concept of zooming in/out appears to be rather universal and easy to understand whatever could be the information set it is applied on. It is closed to concepts like unfolding/folding, developing/contracting, detailing/aggregating and many others. In GUI it is closely related to the expand/collapse button associated to nodes of a tree structure. As tree structures are used either for class hierarchies where top-down navigation has a specialization semantic (with "narrower terms generic" for subclass names) or for set hierarchies where top-down navigation has a partitive semantic (with "narrower terms partitive" for subset names), zooming semantic of GUI using such buttons is application dependent.

Multidimensional zooming

Hypercube is the core model [Agra95] in the multidimensional databases research field. This model distinguishes two kinds of attributes: attributes that define object dimensions and the ones that represent object measurements. Usually dimensions are associated with hierarchies that specify aggregation levels and hence the amount of the objects to be measured. A dimension structure is a partition tree.

In *Hypercube* models, first operations are *slicing* (given a value for one dimension, projecting measures on the n-1 other dimensions of the hypercube) and *dicing* (selecting some sub-hypercube).

The idea of zooming up along a dimension is implemented by another operation, the *roll-up*; the reverse of *roll-up* is *drill-down*. Drill-down + dicing are triggered by users when they want to increase the detail of measurements around a dimensional value of interest.

Note that if they have several values of interest a dimension should be ”shearable”. M. Scholl et al. [SVPRR96] formally defined dimensional shearing. This definition can be generalized to multidimensional shears.

Our research differs from this multidimensional zooming where facts are aggregated as measures which hide individual objects, and where there is no constraint between depth of drill-down and count of objects. However the way drill-down/roll-up selects intervals along dimensions is similar.

Interface zooming

Zoomable User Interface (ZUI) is a part of research on information visualisation. A semantic zoom is a geometric zoom which increases magnification as it decreases fraction of the space – with a kind of “semantic” magnification. It means that a zooming-in allows users to see more (description) details about objects. A ZUI is based on the concept that data are modelled in a two-dimensional virtual world. Users can travel in this world and focus on areas of interest [Fur95]. When users approach an object, the representation is modified and more details appear. Therefore a ZUI model involves a specification for presenting information in different scales.

In semantic zoom visualization, the fewer objects found in a zoom the more attributes can be displayed on the screen. One can say that the constraint F is on information quantity displayed and not on object count as in our object zooming model. Our research differs also from ZUI as the later does not automatically depend on the data distribution .

Cartographic zooming

Generalization is well known as a cartographic process that generates many scale-dependent maps from a single database [Rig94, Mül95]. It allows a zoom-out view to be created on a set of geographic objects belonging to a given area. A generalization process deals with generalization operators such as geometric simplification, object selection, object abstraction, etc. If we do not consider the aspect of geometric representation, cartographic generalization involves a zoom on semantic resolution details where the more abstract we are or the lower resolution we use, the simpler and more synthetic objects we get. For example, depending on the resolution of a map, a map area could be drawn for city, state or country.

Our zooming model is rather different from such kind of thematic zooming on data. We suppose, unlike a cartographic visualization, navigation on the (Semantic) Web involves browsing topics first and related resources after. At a moment users have chosen an interesting topic, they can see “more details”, i.e. see related resources, by a click on a selected topic. Thus we are only interested in a zoom in/out on a set of objects which belong to the single topic class. Object attributes form a multidimensional space that users can explore dynamically. As a matter of fact, our model does not implement the idea “*the fewer objects we see, the more (description) details we get*” of semantic or cartographic zoom [Spe01].

Text zooming

From the textual linguistic point of view a text is logically not only a written line to be read sequencially but a complex tree or graph structure thanks to many marks between sentences. Cast in a document with paragraphs, titles, or as many presentation marks as one can found in an HTML document, the tree structure may be underlined per se or via a table of contents.

A few classical or pedagogical texts can be found with systematic summaries, written by authors, attached to each node of the table of contents often at multiple depth levels.

One can think of a summary as being an extract or an abstract, with rather different implications :

- an abstract is a gloss that describes the contents of a document without necessarily featuring any of that content

- an extract is a summary that is constructed mostly by choosing the most relevant pieces (sentences or paragraphs) of the source text.

Text processors may be used to a posteriori create such summaries either as most representative sentences or paragraphs or as real new sentences or as real new sentences build with different approaches: see [Mar00] or [Moe00].

Whatever is the way to summarize, a reader can say that he or she is zooming in the text when he or she navigates down the table of contents tree displaying longer and longer summaries and eventually the full text.

A priori, text zooming is very far from object zooming we have only considered in this paper. But we have to keep in mind that attributes that make objects visible – i.e. by which objects are displayed through GUI - are very often text-valued, and that the idea of information density should apply too with this type of attributes.

5. Conclusion

In this paper, we have presented a positioning research on an object zooming model which can help to explore efficiently a large quantity of resources from the Semantic Web according to objects they are linked with, assuming these objects belong to a single root class, with common attributes. We have suggested that this model is somehow intermediate between multidimensional drill-down/roll-up technique and “attribute explorer” technique, in the one hand, and semantic, cartographic and text zooming techniques, in the other.

In the future, we will develop a declarative tool that enables exploring dynamically and efficiently a large amount of intricate resources described by semantic objects. We also pursue the zoom computing perspectives that have been mentioned in this paper.

6. References

[Agra95] R. Agrawal, A. Gupta, S. Sarawagi, *Modeling Multidimensional Databases*, Research Report, IBM Almaden Research Center, San Jose, California, 1995, and in Proc. ICDE '97

[Ama01] Bernd Amann, Alain Michard. *The C_Web Architecture Specification V1.0*. Technical report, INRIA, 2001. Available at http://cweb.inria.fr/Resources/resources.html

[Bri02] Dan Brickley and R.V. Guha. *RDF Vocabulary Description Language 1.0: RDF Schema*. W3C Working Draft 30 April 2002. Available at http://www.w3.org/TR/rdf-schema/

[Chi01] R. Chirkova, A. Y. Halevy, and D. Suciu. *A formal perspective on the view selection problem*. Proc. of VLDB, pages 59--68, 2001.

[Fur95] George W. Furnas, Benjamin B. Bederson. *Space-Scale Diagrams: Understanding Multiscale Interfaces*. *CHI 1995:* 234-241.

[GG98] V. Gäde, O. Günther, *Multidimensional Access Methods*, ACM Computing Surveys, 30(2), 1998.

[Kot99] Y. Kotidis and N. Roussopoulos. *Dynamat: A dynamic view management system for data warehouses*. In SIGMOD, pages 371--382, 1999.

[Las99] Ora Lassila and Ralph R. Swick. *Resource description framework (RDF) Model and syntax specification*. Recommendation, W3C, February 1999. Available at http://www.w3.org/TR/REC-rdf-syntax/

[Lern03] A.Lerner and D.Shasha, *Aquery: Query Language for Ordered Data, Optimization Techniques and Experiments*. Proceedings of the 29th VLDB Conference, Berlin, 2003.

[Mar00] Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. Cambridge, MA:MIT press.

[Moe00] Moens, M.-F. (2000). *Automatic Indexing and Abstracting of Document Texts*, Norwell, MA, Kluwer Academic.

[Mül95] Müller, J.C., Weibel, R., Lagrange, J.P., Salgé, F. *Generalization: state of the art and issues*. In Gis and Generalization: Methodology and Practice, J.CMüller, J.P.Lagrane and R.Weibel eds., Taylor and Francis 1995, pp.3-17.

[Ozs02] G. Ozsoyoglu et al. *Sideway value algebra for object-relational databases*. Proceedings of the 28th VLDB Conf, Hong Kong, 2002.

[Rig94] P. Rigaux & M. Scholl (1994) *Multiple Representation Modelling and Querying*. In: International Workshop on GIS (J. Nievergelt et al., eds.), LNCS No. 884, Springer-Verlag, Berlin, pp. 59-69.

[Smi01] A.Smith, *Attribute Explorer, A Dynamic Query Mechanism*, IBM, 2001. Available at http://www-106.ibm.com/developerworks/library/us-atex/

[Spe01] R.Spence, *Information Visualization*, ACM Press Books, 2001.

[SVPRR96] M. Scholl, A. Voisard. J.-P. Peloux, L. Raynal, P. Rigaux, *SGBD Géographiques: spécificités*, International Thomson Publishing France, Paris, 1996.