

# Content-Preserving Graphics

Timothy Lebo, Alvaro Graves, and Deborah L. McGuinness

Rensselaer Polytechnic Institute, Troy NY 12180, USA  
lebot@rpi.edu, gravea3@rpi.edu, dlm@cs.rpi.edu

**Abstract.** Visualization is a common method used to communicate information about underlying data. It requires prudent editorializing and appropriate abstractions to produce a meaningful and compelling result. And, to be most effective, visual designers and analysts should understand the context in which their visual message will be observed, understood, and used. Unfortunately, if the observation context that a visual designer *anticipates* varies from the *actual* context in which a visual message is used, many kinds of problems may arise. This situation is especially common in the Web environment, where the audience is distributed and detached from the original author. We propose a Linked Data technique to preserve content within graphics, so that analysts may, when needed, augment the original content in an isolated graphic to satisfy tasks unanticipated by the original visual designer. Our technique uses common web standards to publish, integrate, and access data among disparate but coordinating agencies and enables a new class of knowledge discovery that goes beyond the information in any one visualization to enable the discovery of patterns among the content presented within a corpus of visualizations. By adopting the perspective that graphics are merely derived data subsets, we offer a new consumption method that treats data and visuals uniformly such that sharing the graphic is sharing the data. We offer a prototype implementation of the technique and demonstrate the advantages of this new means for sharing and consuming Linked Data.

## 1 Introduction

Although data and visualizations are ubiquitous, the analytical process surrounding them is rarely straightforward and often costly. Analytics can be viewed as two major loops [15], where a *foraging loop* seeks, searches, and filters information and a *sense making loop* develops a mental model to suit some objective. To complicate matters, both loops can be performed either *bottom-up* or *top-down* at different stages. Additionally, technological disparities among tools, user skills, and data sources often fosters an *ad hoc* environment that reduces effectiveness. A 2012 survey of 35 data analysts highlights these challenges, reporting that users' most tedious and time-consuming tasks are discovering and wrangling data [10]. When finished, analytical results are typically static and shared via email, a shared file system, or during group meetings. Analysts also reported difficulties discovering when relevant data becomes available, and some analysts avoid visualization altogether because it adds a barrier to the underlying data.

The Web has emerged as a mature, but disjointed, visualization platform. The New York Times regularly produces mature web-based graphics tailored to specific stories; Many Eyes [19] enables non-experts to create, share, and annotate views of their own

data; d3.js [4] offers web developers a toolkit that embodies years of visualization research; sites such as *visualizing.org* host regular visualization challenges<sup>1</sup>; and social media sites are streams of pointers to (and commentary about) others' visual works. Viégas *et al.* [19] suggest that a visualization should scale not only by the size of the data, but also by the size of its audience. They point out that the Web provides the broadest audience possible and offers a democratic and deliberative style of data analysis.

Despite the current benefits of the Web, a significant opportunity cost can be found by considering the following three conditions:

- Many analytical graphics result from a costly and opaque analytical process.
- Each analytical graphic may inspire an entirely new analytical process.
- New analytical processes often require additional data seeking and filtering.

Consumers often face challenges when they are inspired to extend, elaborate, or repurpose the content of an existing analytical graphic because *sharing the graphic* is currently not equivalent to *sharing the data*. By omitting the underlying or supporting data for a result, producers impose non-trivial costs for any subsequent consumer wishing to perform unanticipated analyses.

To reduce the cost of analyses inspired by pre-existing analytical graphics, we offer four main contributions:

1. **Depiction** - An abstract relation *depicts* that holds between a graphical element and the entity that it portrays and represents.
2. **Content-preservation** - A concrete design to employ the depicts relation using existing Linked Data standards for data representation.
3. **Content-augmentation** - A method and prototype implementation to automate foraging for the underlying content of a graphic, along with supplemental content for alternative analyses.
4. ***ex situ* repurposability** - An aspect by which visualizations may be evaluated by their ability to enable unanticipated repurposing even when detached from their original creation environment.

These techniques empower analysts to do more with content-preserved graphics than just taking them at face value. Starting with only the graphic itself, they can inspect the original data that led to the graphic, and they may synthesize it with diverse data sources to enable novel analyses. Currently, the content-preservation technique can be applied to any XML-based graphical format that permits metadata on individual graphical elements (we have demonstrated it for SVG, GraphML, OmniGraffle) and the technique can be incorporated as a feature in a variety of existing graphics production systems (we have demonstrated it in d3.js and our own XSLT-based system).

In the rest of this paper, we review work related to repurposing analytical graphics (Section 2), introduce the technical components of our design using a simple example (Section 3), describe our prototype content-preserving graphics creation system (Section 4), and demonstrate the benefits of content-preservation and content-augmentation using a real-world example from the visualization literature (Section 5). We conclude by discussing some strengths, limitations, and future work of our approach (Section 6).

<sup>1</sup> <http://visualizing.org/open-challenges>

## 2 Related Work

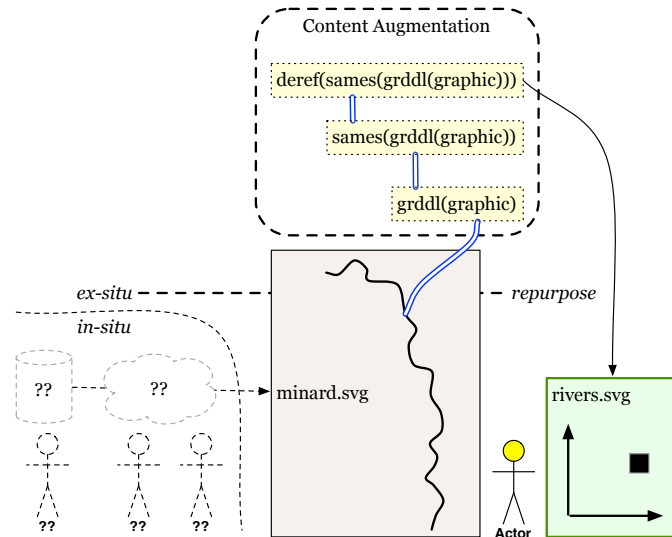
Shneiderman’s “Extract” task [17] is central to the distinction that we draw between *in situ* and *ex situ* repurposability. While the first six tasks (overview, zoom, filter, details-on-demand, relate, and history) tend to occur *within* a single environment, the extract task results in detaching content *from* its original environment. Revision [16] and Grover [12] are two recent systems that enable analysts to repurpose graphics extracted from their original *in situ* environments, but at the same time their limitations illustrate the shortcomings that one faces with a graphic that is not content-preserving. The two tools reverse-engineer quantities from a rasterized image of a pie chart or bar chart, render alternatives using visual design guidelines that increase perceptual effectiveness, and permit users to annotate individual data elements. Although these approaches can estimate graphically-encoded quantities, they cannot recover the content that was originally used to create the visualization and are thus limited in the kinds of tasks they can support (e.g., only annotations and highlighting). The technique presented in this paper, by preserving and providing access to the original content encoded in a graphic, re-enables the full spectrum of Shneiderman’s seven tasks even after the graphic is detached from its original creation environment.

Preserving content within graphics aligns with Dasgupta’s and Kosara’s [6] suggestion that data be traced through the visualization pipeline, so that the impact of any transformation may be measured and understood. They argue that the endpoint for future visualization models should go beyond the screen representation. Instead, they suggest a “measurement framework” that would provide visual designers a quantitative environment to create tools that adapt to the data, physical display parameters (e.g., screen size or pixel resolution), and even the user’s abilities to understand. The content-preservation technique presented in this paper may be seen to advance this objective, and broaden the scope of their measurement framework beyond a single system by using Web-based data representation and publishing standards.

## 3 Approach

In this section, we outline the conceptual and technical components of our approach to preserve content within graphics and subsequently augment the content to fulfill new and potentially unanticipated analytical tasks. Figure 1 illustrates an overview using an example that is used throughout this section. Centered near the bottom of the figure is a notional SVG file depicting Minard’s map of Napoleon’s march during *La Campagne de Russie, 1812-1813*. The black squiggle depicts the Neman River, which Napoleon crossed during his attempt to invade Moscow.

The important distinction between *in-situ* and *ex-situ* environments is shown in the lower left hand corner of Figure 1. By *in-situ* environments, we mean the original environment within which a graphic was created. This includes the data sources, transformation processes, and tooling – along with any human or mechanical agents involved. These *in-situ* environments may be repeatable or *ad hoc*, and they may be stand-alone or comprise many systems. Our content-preserving and augmentation approach is most beneficial to consumers who do not have insight or access to the graphics’ original creation environment. We refer to these detached environments as *ex-situ* environments.



**Fig. 1.** By preserving content within graphics, we enable novel tasks that extend beyond those anticipated by their original creators – even when analysts are consuming graphics detached from the environment in which the graphics were originally created.

To the right of the map in Figure 1 is the analyst who benefits from content preservation and triggers content augmentation to suit her new analytical purposes. In this example, the analyst observes the squiggle depicting the Neman River and is inspired by a new question that requires more information than what is available in the graphic. Because this graphic is detached from its original environment, the analyst does not have the affordance to modify the original data selection or view parameters – she has only the static graphic that she may have found on the web, on a shared server, or in an email from a colleague. After content augmentation, the analyst is able to create a scatter plot comparing rivers’ lengths to their discharge rates using values that were automatically synthesized from two disparate Linked Data providers. Although we use a scatter plot in our example, content augmentation does not limit the kinds of visual techniques used in subsequent analyses. This is because content-augmentation precedes and thus informs the visualization construction process. The next five subsections describe the details of content preservation and augmentation.

### 3.1 The Depicts Relation

We name a relation *depicts* that holds between a graphical element and the entity it portrays and represents:

$$depicts : graphic \rightarrow content \quad (1)$$

By *graphical element*, we mean any drawing, painting, or other visual form that may exist at any level of abstraction or granularity (e.g., from a single pixel or stroke to a

full display’s composition). For example, the *black vertical squiggle* that appears to the far left in Minard’s famous Russian march map *depicts* the *Neman River*.

$$\text{depicts}(\text{squiggle}, \text{neman\_river}) \quad (2)$$

### 3.2 Resource Description Framework (RDF)

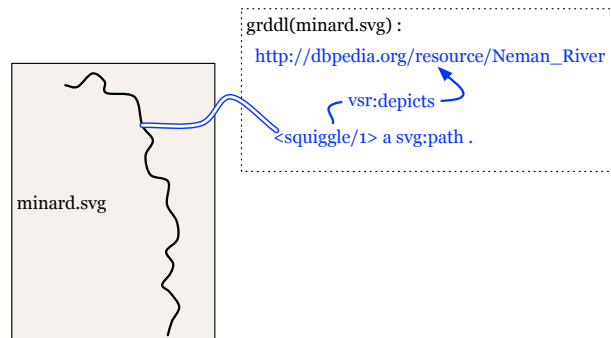
We express the *depicts* relation concretely using RDF [11]. RDF extends the linking structure of the Web by using URIs [7] to name the relationship between things. We can express that the black squiggle from the previous section depicts the Neman River by creating (or reusing) URIs for each and relating them with the following RDF triple. The prefix<sup>2</sup> *vsr* is an abbreviation for the namespace <http://purl.org/twc/vocab/vsr#> and stands for “Visual Strategies in RDF”, a vocabulary we introduce in Section 4.

```
<http://ieeevis.tw.rpi.edu/squiggle/1> a svg:path;
  vsr:depicts <http://dbpedia.org/resource/Neman_River> .
```

### 3.3 Gleaning Resource Descriptions from Dialects of Languages (GRDDL)

GRDDL [5] provides a common way to obtain RDF assertions from arbitrary XML documents. We use this W3C standard to obtain RDF depiction assertions from a wide variety of existing graphical formats such as SVG, GraphML, and OmniGraffle. A GRDDL-compliant XML document contains a *grddl:transformation* attribute whose value is a URL of some source code or Web service. When a GRDDL processor finds this link, it can input the XML into the code or service to receive an RDF representation of the XML document’s contents. We summarize GRDDL in the following function. For our purposes, it maps graphics to RDF. Figure 2 illustrates a snippet of RDF that is extracted from an SVG file of Minard’s map.

$$\text{grddl} : \text{graphic} \rightarrow \text{RDF} \quad (3)$$



**Fig. 2.** Depiction annotations can be uniformly extracted from graphical files using GRDDL. Obtaining an RDF description of the depicted resources is the first step of content augmentation.

<sup>2</sup> Prefixes used are defined at <http://prefix.cc/owl,dcterms,skos,prov,sio,dbo,stream,dt,yago>.

### 3.4 Dereferencing URIs and Linked Data

RDF is most useful when its node and edge URIs follow the Linked Data principles [1]:

- Use HTTP URIs to name things, so that people can look up those names.
- When someone looks up a URI, provide useful information in RDF.
- Include links to other URIs, so that they can discover more things.

An HTTP URI appearing in RDF can be *dereferenced* in the same way that web page URLs are requested with HTTP. For example, the URI that we used to name the Neman River in the previous examples can be dereferenced to get a web page about it. Additionally, because the URI that we chose for the Neman River is also a Linked Data URI, we can use HTTP's *content negotiation* feature to request a description of the river in a format other than HTML. By requesting the *text/turtle* MIME type when dereferencing the Neman River's URI, we receive more than 200 triples about the river, including the following three:

```
<http://dbpedia.org/resource/Neman_River> a dbo:River;
  stream:discharge "616.0"^^dt:cubicMetrePerSecond;
  stream:watershed "98.0"^^dt:squareKilometre .
```

We summarize Linked Data dereferencing as a function mapping an HTTP URI to an RDF graph:

$$\text{deref} : \text{URI} \rightarrow \text{RDF} \quad (4)$$

Because URIs are reusable, they may and should appear within a variety of data sources. For example, when we reuse DBPedia's URI [3] for the Neman River, it is in both their dataset and our example dataset. It also appears in a third dataset maintained by sameas.org [9], so we can find more about the river by requesting another URI for the same river<sup>3</sup>, which dereferences with the following triple, among others:

```
<http://dbpedia.org/resource/Neman_River>
  owl:sameAs <http://yago-knowledge.org/resource/Neman_River> .
```

The `owl:sameAs` property is used to link one data source's URI to another data source's URI. Knowing that two URIs denote the same thing allows a consumer to combine descriptions from both into a single unit. For example, when we dereference YAGO's [18] URI for the Neman River and find its length, we can infer that DBPedia's URI for the same river has the same property.

```
yago:Neman_River
  yago:hasLength "937000.0"^^yago:m .
<http://dbpedia.org/resource/Neman_River>
  stream:discharge "616.0"^^dt:cubicMetrePerSecond;
  stream:watershed "98.0"^^dt:squareKilometre;
  yago:hasLength "937000.0"^^yago:m . # # # Inferred
```

Analytical foraging can be achieved by consuming Linked Data in the following pattern:

1. Dereference the URI of some interesting thing (initially, one found in the original graphic).
2. Inspect the RDF that is returned about the thing.
3. Dereference more URIs of other interesting things.
4. Accumulate the gathered RDF into a working collection.

<sup>3</sup> <http://sameas.org/n3?uri=http%3A%2F%2Fdbpedia.org%2Fresource%2FNieman>

To guide the pattern above, a *property path* may be used to specify how Linked Data agents should automatically forage for a graphics' content and related information. A property path is a list of RDF relations. Starting with an initial RDF graph, all objects of the first RDF relation are dereferenced to create a new, larger RDF graph. Next, all objects of the second RDF relation are dereferenced to create a newer, larger RDF graph. The process is repeated until the property path is exhausted. We describe dereferencing of an RDF graph using a property path as a function:

$$follow : \langle RDF, PROPERTY+ \rangle \rightarrow RDF \quad (5)$$

### 3.5 Repurposing

Applying the *grddl*, *deref*, and *follow* functions to a content-preserving graphic file produces an aggregated RDF graph that contains not only the data elements depicted within the original graphic (i.e., the Neman River), but also those that were related, did not appear in the graphic, and were accumulated *post hoc* from any number of disparate sources. This enables an analyst to repurpose the content of the original graphic to satisfy novel analytical needs. In our example, the analyst is inspired by the Minard map to compare rivers according to their physical characteristics. Using content augmentation, the analyst is able to automatically forage from DBpedia, sameas.org, and YAGO to determine the river's discharge and length. By doing this for all rivers of interest, the analyst is able to accumulate RDF that constitutes a new derived dataset that is suitable for her own novel computations, depiction, and redistribution.

## 4 Visual Strategies For RDF (VSR): A Prototype Implementation

This section describes a prototype implementation of the technique introduced above.

### 4.1 Implementing Abstract Visual Strategies

Visual Strategies in RDF<sup>4</sup> (VSR) generates SVG, GraphML, and OmniGraffle files that depict arbitrary input RDF graphs. VSR accepts a user-defined *visual strategy* to specify how the input RDF graph should be rendered – each triple is processed to determine *if* it should be depicted and, if so, *how* it should be depicted. Basic visual strategies are created by categorizing RDF classes and properties into one of 26 groups that control the triple's coloring, labeling, or visual connectivity according to the URI, namespace, or type of the triple's subject, predicate, or object. More advanced visual strategies can be specified by adding XSLT-based templates to override the core system to call either of two API functions that generate graphical nodes or edges, respectively. Because visual strategies are decoupled from the target graphical format, the same visual strategy can be used to produce any available output format, and new output formats can be added (without regard for the visual strategy) by implementing the two API function calls.

### 4.2 Automated vsr:depicts Annotation

Because VSR creates graphical elements within the context of a *single* triple, it is straight forward to automatically include the vsr:depicts annotation for every resulting graphical element. Annotating vsr:depicts relations is implemented once for each graphical format and then simply used each time a user invokes VSR to visualize an RDF graph. The vsr:depicts annotations are

<sup>4</sup> Available at <https://github.com/timrdf/vsr/wiki>, see <http://purl.org/twc/paper/lebo/2013/cold>

never manually asserted by the user, allowing them instead to focus on *what* they want to see and *how* they want to see it. Each VSR graphical format implementation provides the appropriate *grddl:transformation* annotation within its XML output, so that GRDDL processors can obtain an RDF representation of their output. One GRDDL processor is required for each graphical format, since the *vsr:depicts* relations need to be encoded differently to suit each format’s requirements.

### 4.3 An Abstract Graphical Description

VSR maintains an ontology<sup>5</sup> to describe graphical properties regardless of graphical format. For example, *vsr:Graphic* is the superclass of *svg:rect*, *graphml:node*, and *omni:ShapedGraphic*. The ontology is inspired by the Semiology of Graphics [2] and also defines the *vsr:depicts* relation used throughout this paper. By abstracting the differences among concrete graphical formats, the VSR ontology offers a higher level description that may be used by visual designers and end users to describe visualization requirements, specify visual designs, describe graphical results, and perform rigorous evaluation. Thus, the ontology is applicable throughout the visualization lifecycle. VSR currently uses the ontology to describe visual strategies, to record the provenance of visual encoding decisions made to create graphics, and to describe the graphics in their GRDDL output.

### 4.4 Provenance of Visual Encoding

When VSR applies a visual strategy to create a graphical element, it records the conditions that led to the creation using the W3C PROvenance Ontology (PROV-O) [14]. These records can be used to explain why a graphical element has (or does not have) certain properties, such as why it is red, why it was omitted, or why it is visually connected to another graphical element. Using RDF to record the provenance of each graphical element allows this metadata to be accessible as Linked Data, just like the content data (e.g. the Neman River) that led to the graphic. In addition to PROV-O, the provenance records use VSR’s abstract graphical description introduced above and the SemanticScience Integrated Ontology (SIO) [8] to employ PROV-O’s directed qualification pattern for attributes. For example, if a visual strategy decides to draw a black squiggle to depict the Neman River, the following portion of PROV-O records that a “rule 43” was responsible for deciding that it should be black.

```
<http://ieeevis.tw.rpi.edu/squiggle/1>
  a svg:path, :Squiggle, vsr:Graphic;
  vsr:depicts      <http://dbpedia.org/resource/Neman_River>;
  vsr:fill         vsr:black;
  sio:has-attribute <black-fill> .

<black-fill>
  a sio:Attribute, vsr:fill;
  sio:refers-to    vsr:black;
  prov:wasAttributedTo strategy:rule_43 .
```

## 5 Demonstration of Technique

In this section we demonstrate how *content preservation* and *augmentation* overcome the challenge to repurpose analytical graphics that are produced using traditional visualization techniques. We repurpose the content from an isolated but content-preserved graphic to perform a

<sup>5</sup> Ontology is available at <http://purl.org/twc/vocab/vsr>



new analysis and generate a new graphic. The graphics described in this section are encoded in up to three different formats (SVG, GraphML, OmniGraffle) and were created by one or two different production methods (d3.js, VSR). The variety of graphical formats and production methods emphasizes that the content preservation and augmentation technique is not dependent on any one tool or format, i.e., it is broadly applicable to existing analytic environments and should reflect a relatively low adoption cost to include in existing systems. We chose to demonstrate SVG and GraphML due to their popularity and OmniGraffle because of its mature user interface.

*Repurposing a Visualization Publication Survey.* Lam *et al.* [13] present a survey of more than 800 publications from four visualization venues: InfoVis, EuroVis, VAST, and the Journal of Information Visualization. They categorized the 361 papers that discussed some form of evaluation into seven scenarios. Their categorization reflects the state of practice within the field and also offers visualization researchers a guide to select an appropriate evaluation method for their own work. Lam *et al.* provided two graphics in their survey. The first is a stacked bar chart that depicts the number of evaluations in each of two categories (data process and visualization) from the years 1995 to 2010. The second graphic is a line chart depicting the number of evaluations in each of seven scenario categories during the same timespan.

Figure 3 shows our *Replica 1*<sup>6</sup> of Lam *et al.*'s first figure. The graphic was content-preserved using the d3.js and VSR production systems to create SVG<sup>7</sup> and OmniGraffle files, respectively. Next, we applied the content-augmentation technique to the OmniGraffle file using the following dereference function.

$$\text{deref}(\text{grddl}(\text{graphic})), \quad (6)$$

$$\text{vsr:depicts}, \text{ sio:has-member}, \quad (7)$$

$$\text{dcterms:subject}, \text{ skos:broader} \quad (8)$$

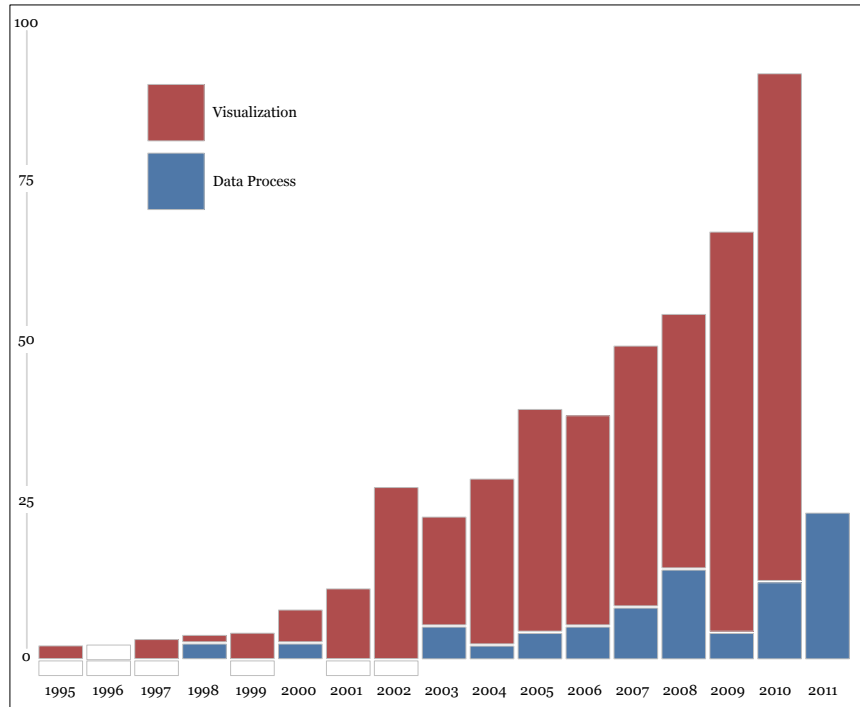
The dereference function above merged 413 Linked Data RDF graphs to produce a final aggregate of 9,854 triples. The resulting graph described the publications counted in the stacked bar chart replica shown in Figure 3. The RDF graph also described the categories that Lam *et al.* assigned, including the hierarchical organization of the subcategories that they defined.

Figure 4 shows key details about the dereference function that merged 413 Linked Data RDF graphs. First, GRDDL'ing the content-preserved OmniGraffle resulted in 891 RDF triples. Of these, 33 resources were depicted by graphics (these were the paper counts for the histogram). After dereferencing each of these 33 URIs, 1,527 total RDF triples were accumulated. Based on this graph, the 348 objects of the sio:has-member property were dereferenced to accumulate a total of 7,621 RDF triples (these additions included details about the publications that were counted in the original graphic, e.g. titles, authors, and pointers to IEEE Xplore). After dereferencing the 31 objects of the dcterms:subject property, a total of 9,854 triples were accumulated (these additions described all nine categories that Lam *et al.* used in their analyses).

Using the aggregated RDF graph as input, the calculations required to create the second figure were performed and encoded as an additional RDF graph, which was rendered using d3.js. The second figure is a line graph that groups paper evaluations by 7 subcategories instead of *Replica 1*'s 2 main categories. We note that we were unable to replicate the exact quantities depicted by Lam *et al.* because they used a scaling operation not reported in their paper. The original authors were unable to provide clarification when requested by email, so our paper counts are not scaled.

<sup>6</sup> The graphic is available in OmniGraffle and SVG formats, created by VSR and d3.js, respectively, at <http://ieevis.tw.rpi.edu/lam-2012-evaluations-2-categories>

<sup>7</sup> `svg-crowbar` permits users to save d3.js graphics as external SVG files. See <http://nytimes.github.com/svg-crowbar/>



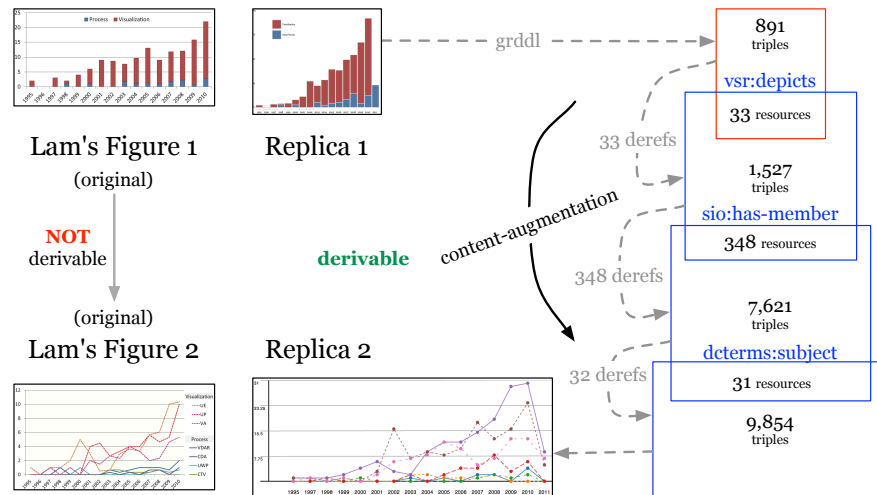
**Fig. 3.** *Replica 1*: Stacked bar chart replicated from Lam *et al.*, with added content-preservation. It depicts the number of publications in each of two evaluation categories from 1995 to 2011.

## 6 Discussion and Conclusions

Visual analytics is often a nonlinear process that can take many unanticipated turns. Often, one analyst's result is another analyst's inspiration for a new investigation. When one wishes to extend, elaborate, or repurpose the contents of an existing analytical graphic, problems may arise because *sharing the graphic* is currently not equivalent to *sharing the data*.

We demonstrate advantage to using the content preservation and augmentation technique, which empowers analysts to do more with content-preserved graphics than just taking them at face value. Instead, graphics serve as a useful handle to portions of Linked Data that not only serve as seeds for further *post hoc* exploration, but more importantly *inspire* users to engage with the results and *inform* their explorations when they do. We showed how a *more detailed* figure from the visualization literature could be created using a replica of a *less detailed* figure from the same publication, where the required additional details were obtained automatically by accumulating Linked Data seeded from URIs available within the Figure itself.

Although the content preservation and augmentation technique is relatively straightforward for graphics creators and consumers to adopt, it requires a larger commitment from data providers, since the technique is only effective if the data is available as RDF and Linked Data. However, because an up front cost to publish can minimize foraging costs during countless subsequent analyses, the effort may already be worthwhile in certain environments.



**Fig. 4.** The Linked Data RDF graph growth at each stage of content-augmenting the *Replica 1* stacked bar chart. The accumulated RDF graph that resulted provided the information required to repurpose Lam’s first figure to Lam’s second figure (*Replica 2*, at <http://ieevis.tw.rpi.edu/lam-2012-evaluations-7-scenarios>).

Using Linked Data to drastically reduce foraging costs emphasizes the need for effective filtering and searching of information that has been, or will be, accumulated. In the example and demonstration presented in this paper, the property paths that we used in the *deref* function were determined manually using a relatively comprehensive understanding of the data that would be collected. To fully leverage Linked Data foraging in less controlled environments, more powerful techniques are required to 1) provide *context-free* overview+detail of arbitrary RDF data that has been accumulated and 2) empower the analyst to steer and throttle the automated foraging. For example, one forage execution from our demonstration retrieved Linked Data from the *dx.doi.org* domain<sup>8</sup>. Although this provided valuable bibliographic information that could be used in related analyses, it was ancillary for the current task, invoked their servers unnecessarily, and cluttered the accumulated data. It will therefore be important in the future to control not only what kind of data should be accumulated, but also from where (or, correspondingly, from whom).

Because content preservation explicitly identifies what each graphic depicts, it gives rise to externalizing information that is traditionally ephemeral, i.e. never explicit outside the scope of a visualization system. For example, in our replicas of the stacked bar chart and line graph, we needed to create a URI to represent the count of publications within each category so that analysts could dereference them in the future. The “new” URIs for the tallies, in turn, led to the existing URIs of the publications that were included in the tally, and foraging the publication details enabled novel subsequent analyses. It is thus important to ensure that derived information is modeled with the same care, attention, and accessibility as the original data. This would grow the amount of *useful* Linked Data, and motivates the need for Linked Computation techniques.

Because visual designers cannot anticipate every context in which their analytical graphic will be used, it is important to provide their audiences with the ability to repurpose the content of any result, especially in situations where the graphic is detached from its original creation

<sup>8</sup> curl -H “Accept: text/turtle” -L <http://dx.doi.org/10.1109/INFVIS.2005.1532122>

environment. We refer to this ability as *ex situ* repurposability, and, based on the benefits we demonstrated here, recommend that it be an aspect used to evaluate visualizations in the future.

## References

1. T. Berners-Lee. Linked data. <http://www.w3.org/DesignIssues/LinkedData.html>, 2013.
2. J. Bertin. *Semiology of graphics*. University of Wisconsin Press, 1983.
3. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A crystallization point for the Web of Data. *Web Semant.*, 7(3):154–165, Sept. 2009.
4. M. Bostock, V. Ogievetsky, and J. Heer. D3; data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, Dec. 2011.
5. D. Connolly. Gleaning resource descriptions from dialects of languages (grddl), September 11 2011.
6. A. Dasgupta and R. Kosara. The importance of tracing data through the visualization pipeline. In *Beyond Time and Errors – Novel Evaluation Methods for Visualization (BE-LIV)*, 2012.
7. M. Duerst and M. Suignard. Internationalized Resource Identifiers (IRI). <http://www.ietf.org/rfc/rfc3986.txt>, January 2013.
8. M. Dumontier. The SemanticScience Integrated Ontology (SIO). <http://code.google.com/p/semanticscience/wiki/SIO>, 2013.
9. H. Glaser, A. Jaffri, and I. C. Millard. Managing co-reference on the semantic web. In *In WWW2009 Workshop: Linked Data on the Web (LDOW2009)*, 2009.
10. S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2917–2926, Dec. 2012.
11. G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>, 2004.
12. N. Kong and M. Agrawala. Graphical overlays: Using layered elements to aid chart reading. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2631–2638, Dec. 2012.
13. H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *Visualization and Computer Graphics, IEEE Transactions on*, 18(9):1520–1536, Sept. 2012.
14. T. Lebo, S. Sahoo, and D. McGuinness. PROV-O: The PROV Ontology. <http://www.w3.org/TR/prov-o/>, 2013.
15. P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, 2005.
16. M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. Revision: Automated classification, analysis and redesign of chart images. In *ACM User Interface Software & Technology (UIST)*, 2011.
17. B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *IEEE Visual Languages*, number UMCP-CSD CS-TR-3665, pages 336–343, College Park, Maryland 20742, U.S.A., 1996.
18. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, pages 697–706, New York, NY, USA, 2007. ACM.
19. F. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1121–1128, 2007.