Undefined 0 (2014) 1–0
IOS Press

A Five-Star Rating Scheme to Assess Application Seamlessness

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Timothy Lebo *,**, Nicholas Del Rio, Patrick Fisher, and Chad Salisbury
Air Force Research Laboratory, Information Directorate
Rome, NY, USA

Abstract. Analytics is a widespread phenomenon that often requires analysts to coordinate operations across a variety of incompatible tools. When incompatibilities occur, analysts are forced to configure tools and munge data, distracting them from their ultimate task objective. This additional burden is a barrier to our vision of *seamless analytics*, i.e. the use and transition of content across tools without incurring significant costs. Our premise is that standardized *semantic web* technologies (e.g., RDF and OWL) can enable analysts to more easily munge data to satisfy tools' input requirements and better inform subsequent analytical steps. However, although the semantic web has shown some promise for interconnecting disparate *data*, more needs to be done to interlink user- and task-centric, analytic *applications*. We present five contributions towards this goal. First, we introduce an extension of the W3C PROV Ontology to model analytic applications regardless of the type of data, tool, or objective involved. Next, we exercise the ontology to model a series of applications performed in a hypothetical but realistic and fully-implemented scenario. We then introduce a measure of seamlessness for any ecosystem described in our Application Ontology. Next, we extend the ontology to distinguish five types of applications based on the structure of data involved and the behavior of the tools used. By combining our 5-star application rating scheme and our seamlessness measure, we propose a simple Five-Star Theory of Seamless Analytics that embodies tenets of the semantic web in a form which emits falsifiable predictions and which can be revised to better reflect and thus reduce the costs embedded within analytical environments.

Keywords: analytics, interoperability, Linked Data, semantics, evaluation

1. Introduction

Linked Data is a large, decentralized, and loosely-coupled conglomerate covering a variety of topical domains and slowly converging to use well-known vocabularies [1,2]. To more fully reap the benefits of such diverse data, linked data analysts must employ an equally diverse array of analytical *tools*.

Meanwhile, the Visual Analytics community has been forging a science of analytical reasoning and interactive visual interfaces to facilitate analysis of "*overwhelming amounts of disparate, conflicting, and dynamic information* [3]." Although the community has produced a vast array of tools and techniques that could assist [4], it remains difficult to easily reuse those tools in evolving environments such as the world of linked data analytics – perhaps because they rely on more mundane representations that make it difficult to establish and maintain connections across analyses.

Our work considers *applications* as an analyst's contextualized use of a tool to suit a specific objective. We focus on the more difficult kind of application where analysts use third-party tools to process third-party data and thus have limited control on design and behavior. Regardless of which community's approaches are adopted, the need to continually form

*Corresponding author. E-mail: Timothy.Lebo@us.af.mil
**DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. Case Number: 88ABW-2014-5577

interconnections among the triad consisting of *data*, *analyst*, and *tool* remains a costly endeavor – and to benefit from both visual analytics and linked data research, the costs need to be more clearly portrayed, assessed, and overcome.

Our hypothesis is that analysts are better able to chain applications together to support threads of analyses when tools:

1. declare their input semantics and
2. provide their derived results as linked data.

With respect to *declaring input semantics*, the flexibility afforded by new APIs such as D3 [5] has resulted in a proliferation of "one-off" visualization tools that inhibit low-cost reusability. These new visualizations regularly assume specific input data formats about specific topics that are not explicitly expressed. Without explicit and formal input semantics, analysts are deprived of a meaningful munging target and must fall back to rummaging through example input datasets or inspecting source code to reconstruct the visualization's implicit data requirements, which can consume up to 80% of the analytical process [6].

With respect to *providing results as linked data*, even if analysts could easily reuse the near two-thousand cataloged D3 visualizations[1], each visualization is a sink from the standpoint of an analytical ecosystem. Results derived from previous applications, including interactions and selections, are often not codified in forms that can reduce integration costs incurred by subsequent analyses that can benefit from previous results.

Our contributions and sectioning of this paper are illustrated in Figure 1. As shown at the bottom of the stack, Section 2 introduces an extension of the W3C PROV Ontology to model analytic applications regardless of the type of data, tool, or objective involved. Section 3 exercises the ontology to model a series of applications performed in a hypothetical but realistic and fully-implemented scenario. Section 4 introduces a measure of seamlessness based on the cost of performing applications in ecosystems described using our application ontology. Section 5 extends the application ontology to distinguish five types of applications that comprise our five-star theory, which states that the structure of data and tools employed in applications dictates the level of analytical seamlessness.

[1] http://christopheviau.com/d3list/ maintains a list of public D3 visualizations. The current count as of December 10, 2014 was 1,897 visualizations.

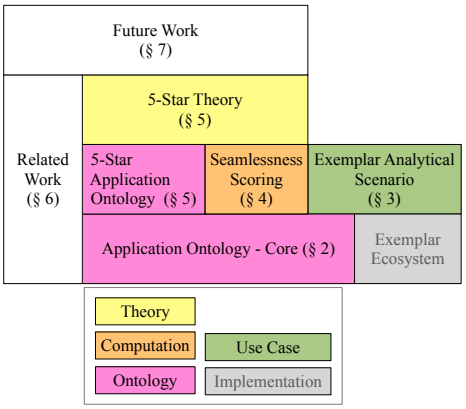

Fig. 1. A theory of seamless analytics comprises four elements that are illustrated with an exemplar and realistic, implemented scenario.

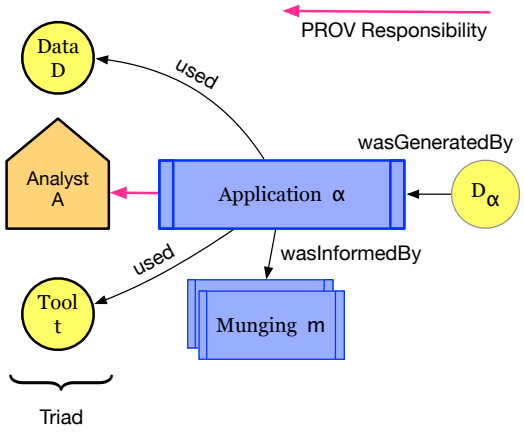

Fig. 2. Application Ontology Core is an extension of PROV. Applications use tools to generate new datasets which could include visualizations. Applications are informed by munging activities that transform data representations. Figure 10 illustrates an extension to further distinguish among five types of applications.

Section 6 describes past work in the area of analytical models and techniques for supporting interoperability in analytical environments. Finally, Section 7 discusses future work before concluding in Section 8.

2. An Ontology of Analytical Applications

Our core Application Ontology (AO) provides a minimal set of concepts to describe an analytical step, herein known as an **application**; a complete thread of analysis can be chained together when subsequent applications use materials from previous applications, as exemplified in Section 3. Application chains can then be assessed using the seamlessness measure in-

troduced in Section 4 and can be further distinguished into five sub-types using the constraints introduced in Section 5.

An application refers to an analyst's contextualized use of some dataset within a tool to achieve some implicit objective, which contrasts prior work of modeling applications as a piece of software [7]. Our applications are a kind of PROV Activity [8], defined as "*something that occurs over a period of time and acts upon or with entities; it may include consuming, processing, transforming, modifying, relocating, using, or generating entities*." An application associates three key entities, which we refer to as the application *triad*: 1) the dataset used, 2) the performing analyst who also most immediately benefited from the result, and 3) the tool that derived a result from the dataset. Figure 2 illustrates these relations using the PROV layout conventions[2] – D_α is the result derived from dataset D by analyst A with tool t during application α.

The distinguishing aspect of our AO is the focus on **munging** activities that may be required to suit a dataset to a tool's input requirements. This relation is also shown in Figure 2 using PROV, but we further relate munging activities as also being *part of* the application[3]. Munging, also known as wrangling [6], is the imperfect manipulation of data into a usable form. Munging has been recognized in the field for decades[4], yet continues to be a ubiquitous and costly problem [9]. We focus on munging because it persists and dominates as a cost factor for applications.

As shown in Figure 3, we establish seven subclasses of munging and group them into three intermediate super-classes. These intermediate classes (*mundane*, *semantic*, and *trivial* munging) are distinguished according to a dichotomy that can be found within Tim Berners-Lee's Linked Data rating scheme [1]. Broadly speaking, Berners-Lee's scale can be used to partition data into two groups: non-RDF and RDF. Let $D_{[1,3]}$ denote the union of all data earning one, two, or three stars according to the popular scheme, and $D_{[4,5]}$ the union of all four or five-star data. We call any dataset within $D_{[1,3]}$ "*mundane*" and any dataset within $D_{[4,5]}$ "*semantic*," reflecting the perspective of the Semantic Web and Linked Data communities that more highly rated data are easier to use or inherently provide more value. Let the function tbl return the star rating of a

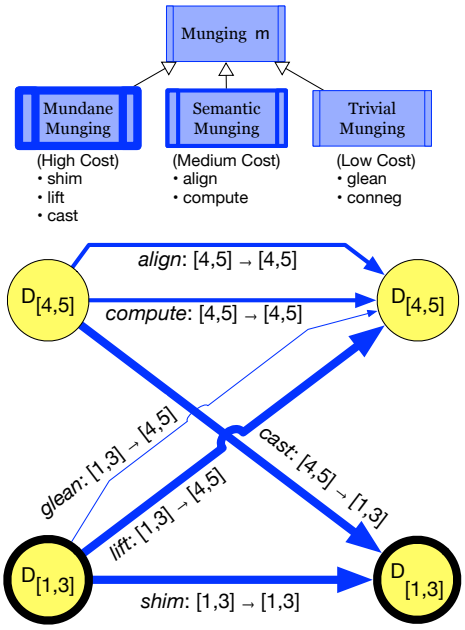

Fig. 3. Munging activities defined in terms of the Tim Berners-Lee's linked data scale. Not shown is content negotiation because it applies to all data types (an ideal situation).

dataset, i.e., $tbl(D_s) = s$). The seven sub-classes of munging (*shim*, *lift*, *cast*, *align*, *compute*, *glean*, and *conneg*) are defined in terms of using[5] data from either $D_{[1,3]}$ or $D_{[4,5]}$ and generating data from the same.

$$\text{munge} : \{D_{[1,3]}, D_{[4,5]}\} \mapsto \{D_{[1,3]}, D_{[4,5]}\}$$

Mundane munges incur the highest cost and are shown in Figure 3 with heaviest edges. Semantic munges are less expensive than mundane munges and are shown with medium weight lines. Finally, trivial munges are the least expensive of all and are shown with lightest lines. The abstract and coarse level cost is intended to reflect the ease at which data can be used within and across applications.

2.1. Mundane Munging

Three kinds of munging activities are common in that they all require the analyst to understand *both* the structure *and* semantics of mundane datasets ($D_{[1,3]}$).

Shimming ($shim$): generates $D_{[1,3]}$ from $D_{[1,3]}$; it is any data transformation that does not involve RDF and is the kind of activity that the Linked Data community is working to ameliorate.

[2] http://www.w3.org/2011/prov/wiki/Diagrams
[3] Using Dublin Core, http://purl.org/dc/terms/
[4] The New Hacker's Dictionary http://catb.org/jargon/html/M/munge.html
[5] We continue to follow PROV terminology to describe activities.

Lifting (*lift*): generates $D_{[4,5]}$ from $D_{[1,3]}$; it creates RDF from non-RDF and has occupied the Linked Data community's attention for most[6] of the past decade [10,11,12].

Casting (*cast*): generates $D_{[1,3]}$ from $D_{[4,5]}$; it creates mundane forms from RDF and, unfortunately, is regularly performed by many Linked Data applications today, typically by using SPARQL to create browser-friendly HTML or SVG.

2.2. Semantic Munging

Two kinds of munging activities are common in that they require the analyst to understand *only* the semantics of datasets ($D_{[4,5]}$).

Aligning (*align*): generates $D_{[4,5]}$ from $D_{[4,5]}$; it derives new relationships from RDF and can often be achieved using ontological mappings [13].

Computing (*comp*): generates $D_{[4,5]}$ from $D_{[4,5]}$; it derives new information from RDF that is itself also expressed in RDF. While aligning is a special kind of computing, there are many other kinds of computing that are not aligning. Computing is relatively less common in current practice but can be found in a few works such as Linking Open Vocabularies[7] and SPARQL-ES [14].

2.3. Trivial Munging

Two kinds of munging activities are common in that they *do not* require the analyst to understand any of the dataset's structure or semantics.

Gleaning (*glean*): generates $D_{[4,5]}$ from $D_{[1,3]}$; the GRDDL[8] and RDFa recommendations are both approaches that can be used to glean RDF from non-RDF representations without the need for contextual knowledge.

Content Negotiation (*conneg*): generates $D_{[1,5]}$ from $D_{[1,5]}$ and "*refers to the practice of making available multiple representations via the same URI.*"[9]

When a mundane dataset can be gleaned, its embedded semantic content will be denoted using an exponent. For example, an RDFa dataset embedding four-star RDF is expressed as 3^4; the HTML's three-star rating is denoted as the base while the four star rating of the RDF is denoted with the exponent. The schema for the gleanable notation is therefore $[1,3]^{[4,5]}$. The exponent representing the star rating of the semantic content is returned when these datasets are gleaned, for example: $tbl(glean(D_{3^4})) = 4$. These kinds of datasets are flexible and allow analysts to use tools that process either the mundane *carrier* data or the embedded semantic content.

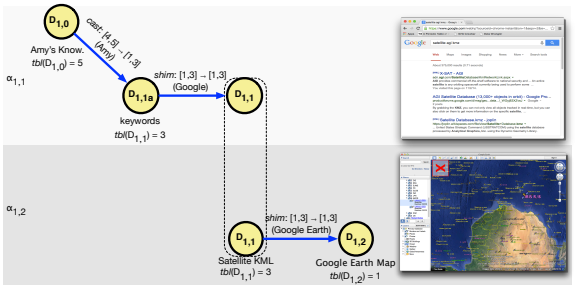

Fig. 4. Munges performed to the support the applications during Task 1.

3. An Analytical Scenario: Space Junk

This section presents a representative analysis modeled according to our application ontology presented in the previous section. The analysis is centered on the broad topic of Earth's artificial satellites, e.g., where they are, who launches them, and how they have changed over time. As our analyst performs applications and inspects generated results, she will incrementally and serendipitously gain insight, formulate new questions, and perform subsequent applications.

In addition to exercising our application model, we use this scenario to highlight certain munging "anti-patterns" that are representative of the state of practice in both analytics and linked data disciplines. These anti-patterns increase the cost associated with generating and reusing materials and in some cases directly influence how analysts design their applications to mitigate compatibility issues [9]. Particular, we focus on two anti-patterns that we name the "house top" and "hill slide," which can be identified visually in provenance munge traces that zigzag between the semantic and mundane levels.

3.1. Task 1: Where are Earth's Artificial Satellites?

Amy, a student enrolled in a physics course, is learning about trajectories as applied to satellite launches

[6] http://triplify.org/challenge
[7] http://lov.okfn.org/dataset/lov/
[8] http://www.w3.org/TR/grddl/
[9] http://www.w3.org/TR/webarch/

and becomes curious about the extent of material launched into space. Although her professor mentions there are more than 2,000 satellites launched by various countries, she remains curious about the location of these satellites and seeks some visualizations to gain perspective.

3.1.1. Task 1, Application 1 ($\alpha_{1,1}$): Google

Without prior materials to work from, Amy performs a Google search to find relevant information about satellite orbits. She stumbles upon a KML file provided by Analytical Graphics Inc. (AGI) [10] that describes satellites known to be in orbit, their launch dates, launch sites, and ownership by country. Her resulting query response page is iconified in the top lane of Figure 4.

In terms of munging, Amy relies on her existing knowledge of satellites to formulate a search query, also shown in the top lane of Figure 4. Dataset $D_{1,0}$ represents Amy's cognitive model [15] that is casted into a mundane sequence of keywords $D_{1,1a}$. We consider cognitive models as a kind of semantic data (i.e.,$[4, 5]$) since both embody interconnections among concepts, despite the fact that cognitive models can never be fully embodied by a dataset. Amy then uses Google to shim her keywords into a results page that contains the AGI KML dataset $D_{1,1}$ that satisfies her criteria.

When analysts cast semantic data to the mundane level, they "fork" the analysis into two parallel threads. Upon casting, explicit interconnections maintained at the semantic level are lost or become implicit and can only be maintained within analysts' cognitive models. As analysts continue to derive mundane results, their cognitive models must adapt in order to maintain the correspondence between the threads that parallel at semantic and mundane levels. If analysts eventually want to unify the two threads, by lifting back to the semantic level, they will have to rely on their burdened cognitive models to reconstitute the connections lost at the data level, thereby duplicating effort and wasting resources. These casts can be seen in the remaining applications.

3.1.2. Task 1, Application 2 ($\alpha_{1,2}$): Google Earth

Now that Amy has a relevant KML file, she uses Google Earth to view it as an interactive geospatial view that plots the location of satellites, as iconified at the bottom lane of Figure 4. The resulting interactive map is littered with many satellites, providing Amy with some perspective regarding the quantity of satellites in orbit.

When analysts cast semantic data to mundane datasets and continue to work at the mundane level, the hill slide anti-pattern can be observed. Visually, hill slides can be identified by a downward slope (i.e., a cast) from the semantic level to the mundane level, followed by a series of shims as shown in the bottom lane of Figure 4. Using results generated by hill slides incurs higher costs because shims and lifts would be required to return to the semantic level. In terms of munging, Amy continues to hill slide and uses Google Earth to shim the mundane KML satellite data ($D_{1,1}$) data to mundane pixels ($D_{1,2}$), shown in the bottom lane of Figure 4.

In the process of inspecting the Google Earth visual, Amy notices folders labeled "Rocket Bodies", "Inactive Satellites", and "Debris", an unanticipated artifact leading to the insight that many of the objects in space are junk rather than active, useful satellites.

3.1.3. Reflection on Task 1's Two Applications

In this first task, Amy set out to understand where Earth's artificial satellites are located, a task that required her to first find relevant satellite data. Once located, Amy visualized the data using Google Earth, which provided the perspective she sought: there is an abundance of satellites throughout orbit. In the course of answering her first question, she also realized that many of the satellites are rocket bodies, inactive satellites or debris, which she herein considers junk. This new insight motivates Amy to compare the relative proportions of active satellites to space junk.

3.2. Task 2: How Many of Earth's Satellites are Space Junk?

Unlike her previous effort, Amy can begin her second inquiry using materials generated by her first task: an AGI KML file and a Google Earth visualization. Amy could reuse the result from Google Earth *if* the connections between data and the representative graphics of interest had been codified and persisted during the hill slide. However, the same visualization from which Amy drew insight now serves as a blockade, preventing her from "reaching through" the graphical layer and easily accessing specific elements of interest in the data layer. Amy could employ *expensive* image processing to map specific groupings of pixels to data elements [16], but it would require her to use new experimental tools that keep data at the mun-

[10] http://adn.agi.com/SatelliteDatabase/KmlNetworkLink.aspx

dane level. She therefore falls back to using the satellite KML file that is less in sync with her current objective regarding satellites.

3.2.1. Task 2, Application 1 ($\alpha_{2,1}$): SPARQL

Amy first chooses to generate an HTML table consisting of satellite counts grouped by "satellite type" using SPARQL, as iconified in the top lane of Figure 5. From the values contained in the table, she can calculate the ratio of active satellites to junk. Before she can leverage SPARQL, or any RDF tool for that matter, she needs an RDF representation of the KML satellite data. Amy first tries to negotiate [17] for an RDF representation of the satellite data but fails because the AGI server does not recognize her RDF ACCEPT headers. She then runs the satellite KML through GRDDL and RDFa processors that return empty sets.

At this point Amy, like many other Linked Data researchers, must lift AGI's satellite KML obtained from her previous task ($D_{1,1}$) to an RDF representation ($D_{2,1b}$) using one of the converters provided by the linked data community [11]. Amy first performs a shim to obtain a CSV representation $D_{2,1a}$ required by her converter, shown at the top right in Figure 5. The converter, in turn, generates an RDF representation as depicted by the rise in the munge trace.

The munge trace from $D_{1,1}$ to $D_{2,1b}$ is a stop-gap approach to obtain linked data, which is tolerable since standards for converting to linked data are relatively new. [11] The problem is that Amy's derived results will likely fall back down to the mundane level when she applies some visualization technique to re-present the data. This lift-then-cast anti-pattern can be observed in the munge sequence as a "house top" as seen in Figure 5 between $D_{2,1a}$ and $D_{2,1c}$.

Once Amy obtains RDF, she poses a SPARQL query that results in an HTML table ($D_{2,1c}$) that is in turn rendered by a web browser. The rendered table provides exacts counts for the different kinds of satellites orbiting Earth: 1,118 active satellites; 1,120 rocket bodies; 8,686 pieces of debris; and 13,754 inactive satellites. From the table, Amy continues the hill slide by calculating the ratio (0.047, $D_{2,1c}$) between active satellites and junk, which she manually determines using her desktop calculator.

[11] R2RML `www.w3.org/TR/r2rml` is a more recent standard for mapping relational data to RDF.

3.2.2. Task 2, Application 2 ($\alpha_{2,2}$): Sgvizler

Although the quantities obtained from the previous application answer Amy's question, she prefers to view this information in a visual form, resulting in a second application as shown in the second lane of Figure 5. To get a histogram, Amy uses Sgvizler [18]. Sgvizler is a JavaScript tool that allows developers to annotate HTML with instructions on how to execute SPARQL queries. The results of the queries are then integrated with the Google API, which generates SVG corresponding to a variety of visualizations, including Amy's histogram iconified in the bottom right of Figure 5.

Rather than reusing the mundane HTML table generated by the previous application, Amy falls back to using the RDF version of the satellite data (i.e., $D_{2,1b}$) since it is less costly to munge; casting the RDF will be less expensive than shimming the HTML table. Her munge trace is presented at the bottom of Figure 5 and illustrates how Amy uses Sgvizler to cast the satellite RDF into the mundane SVG form ($D_{2,2a}$) before rendering it into mundane pixels ($D_{2,2}$) using a web browser.

3.2.3. Reflection on Task 2's Two Applications

In her second task, Amy set out to uncover the relative proportion between active satellites and space junk. Although the results of the SPARQL query allowed Amy to calculate the ratio of useful satellites to junk, the ratio itself was not as insightful as the visual provided by Sgvizler. Sgvizler's histogram provided easy, side-by-side comparison of relative bar lengths corresponding to satellite counts. From the visual, Amy could clearly show that the majority of orbiting objects are in fact useless junk.

Amy does not know, however, which countries are most responsible for the resulting environmental condition. Is it her home country launching the majority of junk, or some other developing country new to space exploration and less sensitive to environmental awareness? She undergoes the next series of applications to explore which countries are launching junk and at what frequency.

3.3. Task 3: Which Countries are Most Responsible for Earth's Space Junk?

After performing only two inquiries, Amy has accumulated a non-trivial set of materials that she can consider in this next task: various mundane representations of the AGI satellite data and one alternate rep-

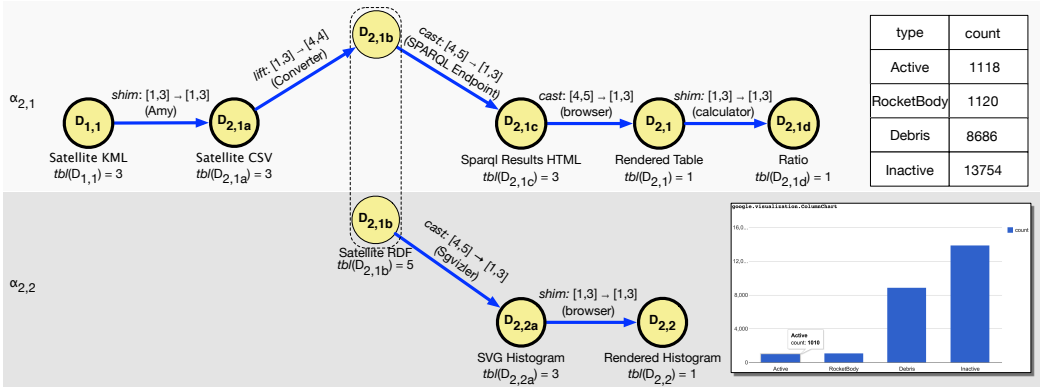

Fig. 5. Munges performed to support the applications during Task 2.

resentation in RDF. Amy could use the results from Sgvizler as a starting point if only the tool had preserved the interconnections between the satellite data and the histogram bars that were lost during the hill slide. Once gain, Amy is unable to get a low-cost handle on specific data elements needed for her analysis and must fall back to using the less contextualized satellite RDF file ($D_{2,1b}$).

3.3.1. *Task 3, Application 1 ($\alpha_{3,1}$): Semantic Histogram*

In the interest of gaining a hold on the information emitted from Sgvizler, Amy switches to use a semantic histogram tool to recreate the same view, but in a semantic form that maintains the connections between data elements and representative graphics. The resulting histogram is iconified at the top right in Figure 6 and represents the same statistical information generated by Amy's use of Sgvizler (satellite counts grouped by type).

Because the semantic histogram tool accepts RDF and describes its input data requirements in OWL, Amy is able to align the satellite RDF to fit the input requirements of the semantic histogram, as depicted by the link between $D_{2,1b}$ and $D_{3,1a}$ in the top lane of Figure 6. Although Amy eventually casts her linked data to a mundane histogram visualization ($D_{3,1}$), the visual maintains connections between data and graphics by embedding a link within the image's metadata to the RDF represented by the histogram (i.e., 1^5). Amy can therefore perform a low cost glean to access the RDF about junk satellites associated with the histogram bars ($D_{3,1c}$). Now that Amy has overcome the shortcomings of Sgvizler, she can begin to pursue an answer to her actual question.

We believe the cast-then-glean inversion pattern as seen in this application is a hallmark of using linked data to a fuller potential. Analysts should use tools to generate visualizations that preserve the semantic connections between graphics and data using approaches such as GRDDL. Although it is inevitable that data must be cast to pixels since humans perceive visual stimuli and make decisions from the observed patterns [19], this fact however does not preclude tool developers from persisting connections [20]. It is permissible to dive down to the mundane level so long as low-cost methods for obtaining alternate, semantic representations are available to analysts.

3.3.2. *Task 3, Application 2 ($\alpha_{3,2}$): Semantic Histogram Redux*

Amy chooses to reuse the semantic histogram tool in the previous application to generate a new visual showing the distribution of launched satellites by country, as iconified in the center lane of Figure 6. From the visual, she hopes to better understand which countries are polluting the most. Amy uses the selection data obtained from the previous application [12], supporting a kind of *inter-application* brushing and linking [21].

She first performs an alignment to classify her satellite classes of interest ($D_{3,1c}$) as histogram bins, resulting in the dataset $D_{3,2a}$ shown in the center lane of Figure 6. She uses the histogram tool to generate statistical RDF ($D_{3,2b}$) that is then represented as a gleanable histogram ($D_{3,2}$). Once again, the cast-then-glean inversion pattern will allow Amy to more easily reuse the data behind the histogram in subsequent applications.

[12] A tool's design dictates how much data to include by value when analysts make selections, possibly requiring them to perform dereferencing in cases when required values are not included.

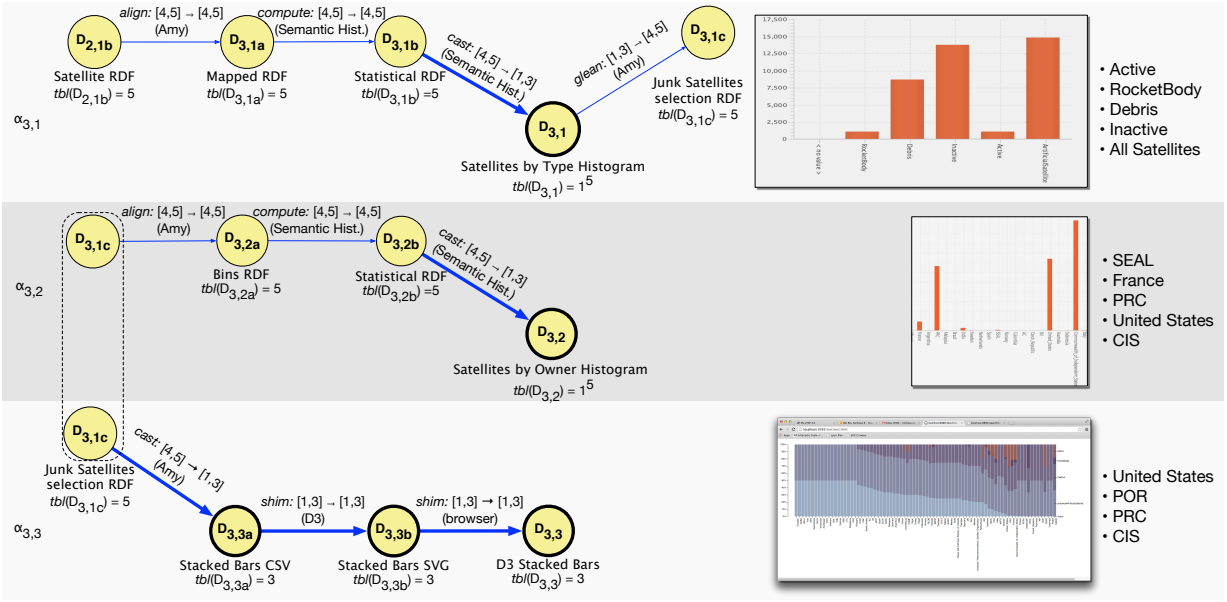

Fig. 6. Munging performed to support the applications during Task 3.

Amy notices that the Common Wealth of Independent States (CIS), United States, People's Republic of China (PRC), and France all launch large quantities of junk. Playing devils advocate, Amy believes it is unfair to use an absolute-valued histogram alone to criticize the four countries. After all, these countries launch more total satellites than any other country, so relatively more junk is to be expected. This insight leads Amy to think about relative values – could she seek a visual that would help her understand the amount of debris *relative* to active satellites launched for each country?

3.3.3. Task 3, Application 3 ($\alpha_{3,3}$): D3 Stacked Bars

To see if there is a correlation between the quantity of useful satellites and junk, Amy transitions to use a simple D3 stacked bars visualization[13]. The stacked bars visual is normalized and therefore conveys the relative active satellite-to-junk launches. This facilitates a more fair comparison among countries based on a sort of "junk efficiency". The resulting visualization is iconified in the bottom lane of Figure 6 and shows that the CIS, United States, China, and France all launch significant amounts of junk, even when compared to their number of active satellites.

Since the D3 implementation of stacked bars only accepts CSV, Amy has to cast her satellite RDF acquired from $\alpha_{3,1c}$ into a mundane form as shown in the bottom lane of Figure 6. The resulting CSV data $D_{3,3a}$ is eventually transformed to non-gleanable pixels ($D_{3,3}$) that Amy can perceive.

3.3.4. Reflection on Task 3's Three Applications

Amy set out to determine which countries launch space junk. Although the semantic histogram in application $\alpha_{3,2}$ showed which countries launch the most junk, the visual did not convey the "efficiency" of those launches which Amy became interested in. To accommodate, Amy transitioned to a stacked bars visualization showing efficiency regarding the relative proportion of junk to useful satellites.

Amy then turned to explore whether or not CIS, United States, China, and France have become "greener" over the years. All visualizations employed to this point lacked a temporal component.

3.4. Task 4: Is There a Trend Towards Launching Less Junk?

Amy has now acquired a non-trivial set of materials that she can consider to leverage to answer her newly inspired question:

- a KML file of satellite data
- an RDF version of that KML
- many intermediate datasets resulting from both mundane and semantic munging

[13] http://bl.ocks.org/mbostock/3886394

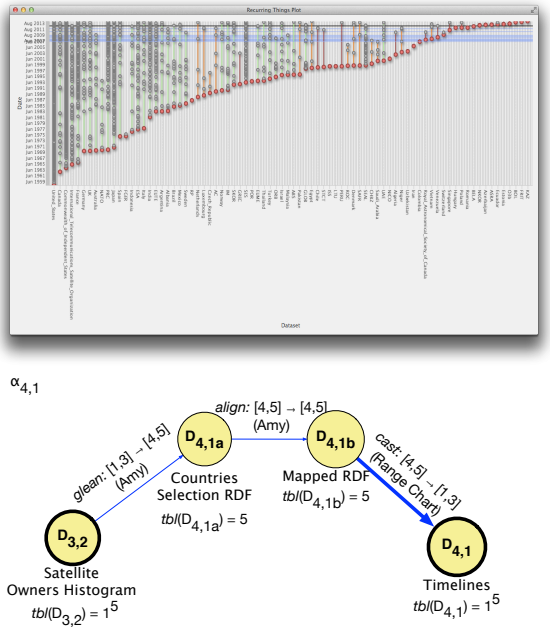

Fig. 7. Munges performed to support the applications during Task 4.

– a few visualizations, some of which are gleanable

She decides to reuse the gleanable histogram representation showing the distribution of junk by country ($D_{3,2}$) since she can cheaply access the set of satellites associated with the four countries of her interest.

3.4.1. Task 4, Application 1 ($\alpha_{4,1}$): Range Chart

Amy uses a timeline-based visualization called a Range Chart to understand the frequency of launches by CIS, United States, China, and France, as iconified in Figure 7. The visual shows Amy two important facets regarding satellite launches: entry into the space age and subsequent satellite launch dates. Amy notices that United States, CIS, China, and France continue to frequently launch amounts of junk into space, up until the 2014 boundary of the dataset's coverage.

The munge pattern associated with Amy's use of Range Chart can be seen in Figure 7. Amy gleans a set of junk-launching countries from the semantic histogram dataset $D_{3,2}$. She then aligns the set of countries to suit the input semantics of Range Chart, which in turn, generates another gleanable visualization $D_{4,1}$.

3.4.2. Reflection on Task 4's Only Application

Amy inquired whether the amount of junk launched by CIS, United States, China, and France had decreased over time. Having become aware that those countries maintain a steady rate of launching junk, Amy becomes interested in knowing if the United States allows any other countries to launch junk from its sovereign facilities. Is the United States fostering other countries' launching of junk and, if so, which countries and how much?

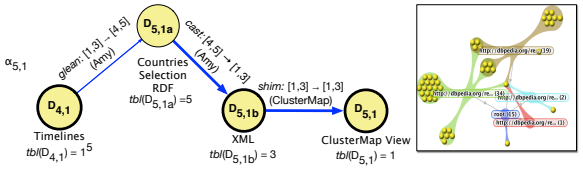

Fig. 8. Munges performed to support the applications during Task 5.

3.5. Task 5: What other Countries Launch Space Junk with the Help of the United States?

In addition to the materials available prior to the previous task, Amy has now acquired a gleanable representation of the Range Chart that shows when countries launched junk ($D_{4,1}$). She chooses to use this dataset to determine whether any of the junk-launching countries depend on United States' launch facilities.

3.5.1. Task 5, Application 1 ($\alpha_{5,1}$): Aduna ClusterMap

Amy employs Aduna ClusterMap [14] to visualize the correspondences between countries and different launch sites, as iconified in Figure 8. ClusterMap creates hierarchical visualizations showing the taxonomic relationship between objects and categories. Additionally, ClusterMap renders *intersection clusters* that represent objects that are cross-categorized. These intersection clusters allow Mary to see if countries (ClusterMap objects) share United States' launch sites (ClusterMap categories). From the visual, Amy can see that France launches from United States' sites; both countries occupy the ClusterMap intersection cluster associated with Cape Canaveral.

In terms of munging, Amy preforms a two-step approach to first glean RDF from the Range Chart visual and then cast the resulting semantic data ($D_{5,1a}$) into the mundane XML format required by Aduna ClusterMap ($D_{5,1b}$), resulting in another house-top anti-pattern.

3.5.2. Reflection on Task 5's Only Application

Amy realized that France, a junk launching country, launches from United States' sites. Once more, this gained insight caused Amy to transition into a new task

[14] http://www.aduna-software.com/technology/clustermap

with the purpose of identifying the locations of these shared launch sites.

3.6. Task 6: Where are the sites from which the United States Permits Other Countries to Launch Space Junk?

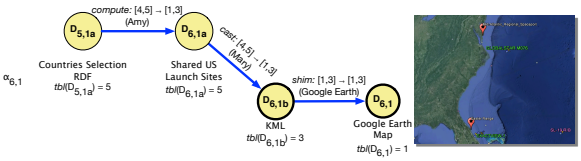

Fig. 9. Munges performed to support the applications during Task 6.

In addition to the many materials previously generated, Amy now has an Aduna ClusterMap PNG image showing which junk-launching countries used launch facilities owned by the United States. For this next task, Amy only wants to know the location of these shared facilities but is unable to access the list of relevant sites without employing expensive image processing or manual transcription activities. Amy therefore must circumvent the natural progression of her analytical results and fall back to an earlier form of the satellite data generated by the semantic Range Chart in Section 3.4.1, which contains the set of *all* junk launching countries. She must rely on her memory of the ClusterMap visual so she can generate a subset of Range Chart's data containing only the launch sites of interest, e.g., Cape Canaveral.

3.6.1. Task 6, Application 1 ($\alpha_{6,1}$): Google Earth

Coming full circle in her tool usage, Amy employs Google Earth to plot the location of United States' launch sites that are shared with other junk-launching countries. The resultant visualization is shown in Figure 9 and plots the launch sites for Easter Range and the Mid Atlantic Regional Spaceport.

Based on her memory of the Aduna ClusterMap visual, Amy computes a set of United States launch sites that are shared with other countries ($D_{6,1a}$). The launch site URIs are associated with a wgs:lat/long[15] coordinate that was asserted during her lift to RDF in application $\alpha_{2,1}$. Amy then casts $D_{6,1a}$ to a KML dataset and then uses Google Earth to shim the KML into the mundane geospatial visualization pixels $D_{6,1}$ satisfying her inquiry.

[15] www.w3.org/2003/01/geo/, prefix.cc/wgs

3.6.2. Reflection on Task 6's Only Application

Amy set out to find the location of launch facilities controlled by the United States using Google Earth. Considering the accumulated knowledge obtained from all applications influencing her analysis, Amy is now satisfied with her understanding of Earth's satellites. She therefore concludes her investigation.

What began as a simple task to understand satellite orbit positions, evolved into a focused analysis to understand a growing environmental issue. The following list outlines the different insights Amy obtained while performing her analysis.

1. There are many satellites populating Earth's low and geosynchronous orbits.
2. The overwhelming majority of satellites are junk.
3. The United States, France, China, and CIS launch much of this junk, even considering their ratio of active satellites to space junk.
4. The United States, France, China, and CIS have always launched junk and continue to do so
5. The United States lets France, a junk launching country, launch from its facilities
6. It is possible to determine the locations of sites that launch space junk by the U.S. and other countries.

Although this was a hypothetical scenario, the analysis is representative of the kinds of applications performed in both enterprise and secure sectors [9,22]. Analysts design and perform application chains that are linked by the materials they generate and reuse. To facilitate linking, analysts' ecosystems force them to perform anti-patterns such as house tops and hill slides that are associated with higher costs munges. Tools that generate gleanable datasets have the potential to reduce such costs.

4. A Metric for Application Seamlessness

In the previous section, Amy analyzed the environmental condition of Earth's orbit by performing a sequence of applications. Such a sequence of applications induces an *analytical ecosystem*. This section establishes two metrics to assess the *seamlessness* of such ecosystems based on the cost of munges performed.

The seamlessness metric is a formalization of the implicit munge-cost theory that is supported by evidence put forth by the visual analytics and linked data communities. The visual analytics community

has long described munging to be a ubiquitous and costly problem that reduces the efficiency of analyses [6,23,9]. Meanwhile, the linked data community has long been a proponent for the cost savings afforded by using more structured and interconnected data [24,1]. Our seamlessness metric embodies evidence from both communities into a single formalization that emits falsifiable predictions regarding how easy an analysis will be to perform.

Seamlessness is assessed using two scores, denoted S_1 and S_*, that reflect two complementary perspectives:

Current: the cost of a single ecosystem
Prospective: the cost of future ecosystems

Current cost is considered from a *presentist* standpoint, i.e., what is the current cost of an analyst's ecosystem? Prospective cost is considered from an *eternalist* standpoint, i.e., what is the future cost of an analyst's ecosystem? We introduce scoring methods to rate analytical ecosystems based on these characteristics.

4.1. Analytical Ecosystems

Let an analytical ecosystem E be the set of applications that influenced[16] a particular analysis:

$$E = \{\alpha_1, \alpha_2, ..., \alpha_n\}$$

Let an application α be a tuple comprising its set of munging activities M and the resulting dataset D_α:

$$\alpha = (M_\alpha = \{m_1, m_2, ..., m_m\}, D_\alpha)$$

The cost to perform an application is *at least* equal to the cost incurred by its munges. This inequality is based on existing work that shows that munging costs are a non-trivial portion of the overall analytical costs [6].

$$0 \ll cost(M_\alpha) = \sum_{m \in M_\alpha} cost(m) < cost(\alpha)$$

This application cost function is dependent on a munge cost function that maps munge activities to quantitative values. To bound our munge-level cost function, we first present a complete ordering of munge costs that embodies the tenets of the Linked Data paradigm and also aligns with the partial ternary ordering introduced in Section 3.

$$cost(shim) > cost(lift) + 2\,cost(align) + cost(cast)$$
$$cost(lift) > cost(cast)$$

$$cost(cast) > cost(align)$$
$$cost(align) > cost(comp)$$

$$cost(comp) > cost(glean)$$
$$cost(glean) > cost(conneg)$$

Analytical applications can range from the worst case, where an infinite number of high-cost shims were performed, to the ideal case where no munge was required. Because shimming is a composite of lifting, aligning, and casting, it incurs the highest cost because, during shimming, analysts must perform alignments mentally and without concrete intermediary models.

We use one such solution to the previous cost constraints to define a munge-level cost function, as shown below. The mappings can be adjusted depending upon specific cost measures, e.g., development hours, lines of code, commit frequencies, so long as the ordering constraints are satisfied. It is expected that any chosen metric conforms to this ordering and if it is found that these constraints do not hold, they should be revised.

$$cost(m) = \begin{cases} 19 = 6 + 2(4) + 5 : \text{ if } shim \\ 6 \ : \text{ if } lift \\ 5 \ : \text{ if } cast \\ 4 \ : \text{ if } align \\ 3 \ : \text{ if } comp \\ 2 \ : \text{ if } glean \\ 1 \ : \text{ if } conneg \end{cases}$$

4.2. Current Seamlessness

In practice, analysts are most concerned with their current cost, since their focus is on obtaining the next immediate result. To quantify this characteristic, we define a *seamlessness score* S_1 in terms of the actual munge costs analysts expend when performing a single

[16] http://www.w3.org/TR/prov-dm/#term-influence

thread of analysis:

$$S_1(E) = \frac{\sum_{\alpha \in E} \sum_{m \in M_\alpha} cost(shim)}{\sum_{\alpha \in E} \sum_{m \in M_\alpha} cost(m)} \quad (1)$$

The numerator reflects the hypothetical worst-case cost incurred by shimming exclusively, i.e., every munge performed in analysis E was a shim. This worst-case cost is divided by the actual cost, which is computed by summing the costs incurred by the munges described in the provenance. The S_1 score thus ranges from 1 to ∞, where a lower score reflects the least seamless ecosystem comprising only shim munges and a higher seamlessness score reflects lower difficulty in maintaining interconnections among data, analysts, and tools.

4.3. Prospective Seamlessness

Because communities benefit when individuals act in the interest of the broader community of subsequent costs, a measure of ecosystem seamlessness should consider the cost imposed on subsequent analysts when choosing which applications to perform. This *prospective* cost can be estimated by considering the type of materials generated from ecosystems. Applications that generate linked data, or datasets that can be trivially munged to yield linked data, reduce potential future costs during subsequent usage. Applications that generate mundane results, such as PowerPoint slides, impose higher costs on future analyses [9].

The function *pot* serves to discount the cost of any munge that results in a more reusable, semantic form otherwise, *pot* does not have an effect.

$$pot(D_\alpha) = \begin{cases} \frac{1}{cost(shim)} & : tbl(D_\alpha) > 3 \\ \frac{1}{cost(conneg)} & : conneg(D_\alpha) \supset \emptyset \\ \frac{1}{cost(glean)} & : glean(D_\alpha) \supset \emptyset \\ 1 & : \text{otherwise} \end{cases}$$

The *pot* function determines the scaling factor based on whether D_α:

- is RDF ($tbl(D_\alpha) > 3$),
- can be negotiated to RDF ($conneg(D_\alpha) \supset \emptyset$), or
- can be gleaned ($glean(D_\alpha) \supset \emptyset$)

The *prospective seamlessness score* S_* extends the seamlessness score S_1 using the *pot* function:

$$S_*(E) = \frac{\sum_{\alpha \in E} \sum_{m \in M_\alpha} cost(shim)}{\sum_{\alpha \in E} \sum_{m \in M_\alpha} pot(D_\alpha) cost(m)} \quad (2)$$

The discounted munging costs reflect future "returns on investments" enabled by generating semantic materials. Although lifting mundane data to semantic forms is expensive, multiple subsequent uses of the semantic content is relatively cheaper in the long run. When performing completely mundane analyses without regard for subsequent usage, the prospective seamlessness S_* is equal to S_1 as expressed in the inequality below:

$$1 \leq S_1(E) \leq S_*(E)$$

4.4. Amy's Scores

The following table enumerates the costs incurred by each application Amy performed during her analysis in Section 3. These values are computed from the provenance of Amy's analytical ecosystem, which encompassed 10 applications.

Table 1
Amy's current and prospective costs. The bold entries highlight applications that generated gleanable datasets.

	S_1		
	S_*		
App.	$cost(M_\alpha)$	$pot(D_\alpha)$	$cost \times pot$
$\alpha_{1,1}$	24	1	24
$\alpha_{1,2}$	19	1	19
$\alpha_{2,1}$	68	1	68
$\alpha_{2,2}$	25	1	25
$\alpha_{3,1}$	**12**	**1/2**	**6**
$\alpha_{3,2}$	**14**	**1/2**	**7**
$\alpha_{3,3}$	43	1	43
$\alpha_{4,1}$	**11**	**1/2**	**5.5**
$\alpha_{5,1}$	26	1	26
$\alpha_{6,1}$	27	1	27
Curr. Cost	269	Pros. Cost	250.5
S_1	2.04	S_*	2.19

Using the S_1 scoring method, the cost of each application is equal to the sum of its associated munges as shown in the table rows. For example, in $\alpha_{4,1}$ Amy

gleaned ($cost = 2$) RDF from the semantic histogram, *aligned* ($cost = 4$) the RDF to suit the requirements of Range Chart, and then *casted* ($cost = 5$) the aligned RDF to a gleanable Range Chart visualization, resulting in an application cost of 11. Considering the total cost of all her applications and the worst case munge cost of 551 (29 munges times the cost to shim 19), Amy's seamless score S_1 is 2.04.

Using the S_* scoring method, the cost of each application is equal to the sum of its *scaled* munges. For example, the same application $\alpha_{4,1}$ results with a scaled application cost of 5.5 since it generated a gleanable result. The table highlights such application rows with fractional scaling factors, i.e., $pot(D_\alpha) < 1$. Because the results of $\alpha_{3,1}$, $\alpha_{3,2}$, and $\alpha_{4,1}$ were in linked data form – or could be trivially transformed to linked data – the munging costs are discounted, resulting in a slightly higher prospective seamlessness S_* score of 2.19.

5. A Five-Star Theory of Seamless Analytics

We propose a simple Five-Star Theory of Seamless Analytics that embodies tenets of the semantic web in a form which emits falsifiable predictions. The theory combines the seamlessness measure S_* in the previous section with a proposed "5-star application rating scheme" which analysts and tool developers can use to design better applications that avoid expensive anti-patterns described in Section 3. The rating scheme is expressed in the form of ontological restrictions that progressively reduce the space of possible munge sequences, eventually resulting with only those sequences that can predictably reduce long term analytical costs.

We outline these ontology restrictions by extending the application ontology introduced in Section 2 to distinguish among five types of application subclasses that are illustrated in Figure 10. These subclasses are rated according to the cost predicted to be imposed upon an analyst, similarly to how different kinds of data is rated using Tim Berners-Lee's star ratings. Table 2 enumerates all five application star ratings and pairs each with their associated restriction(s). Figure 11 uses this table to rate the applications performed by Amy in her analysis presented in Section 3.

5.1. One-star applications

One-star applications satisfy our fundamental restriction that the sets of analysts, tool developers,

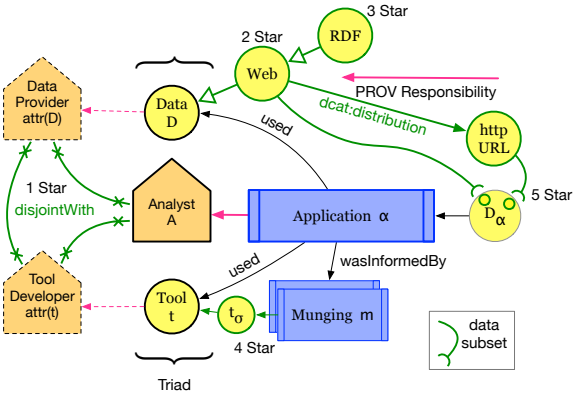

Fig. 10. An extension of the Application Ontology Core (Figure 2) to distinguish five subclasses of Application. The figure distinguishes between generic application concepts, shown in gray, and the extension concepts shown in bold-face.

Fig. 11. Star ratings for the ten applications in Amy's ecosystem.

and data providers (i.e., the application triad) associated with a particular application are disjoint, i.e., $attr(D) \cap A \cap attr(t) = \emptyset$, where A refers to the analyst and the function $attr$ supports the transitive discovery of responsibility for data providers and tools developers that can be practically achieved through SPARQL queries over analytical PROV traces. Figure 10 depicts this restriction towards the left-hand side, where cross-hatched lines connecting the triad of Data Providers, Analysts, and Tool Developers represent a disjoint relationship.

In the scenario presented in Section 3, Amy's use of Google Earth in application $\alpha_{1,2}$ earns one-star since Amy was neither the provider of the satellite data nor the developer of Google Earth. In contrast, the use

Table 2

A five-star rating scheme to assess analytical seamlessness of an individual application. The restrictions are specified in natural language and formally using functional and set notations. The function tbl maps the dataset D used in an application to its star rating as determined by Tim Berners-Lee's scale.

☆	Informal Restriction	Formally
1	data providers, analysts, and tool developers are disjoint	$attr(D) \cap A \cap attr(t) = \emptyset$
2	accept data (any format) via URL; cite that URL in the future	$tbl(D) >= 1 \wedge URL \in D_\alpha$
3	accept data (RDF format) via URL; cite that URL in the future	$tbl(D) >= 4$
4	use a tool's input semantics (OWL, SPARQL) when preforming munges	$used(m, t_\sigma) \wedge m \in M$
5	provide any information (RDF format) derived during use	$D \subset D_\alpha$

of government data mash-ups[17] and LOD metadata summaries such as Linked Open Vocabularies (LOV) [18]and SPARQL Endpoint Service (SPARQL-ES) [19] do not earn a star since the tool developers are also data providers. The LOV tree map view, for example, is immutably bound to LOV's underlying RDF store [20]. In these cases when the application triad is violated, the current and prospective costs associated with performing the application are degenerate and thus cannot be easily predicted.

When analysts are also data providers, they may have existing munging infrastructure specifically tailored for the data they publish. This infrastructure can be used to drive munging sequences that violate our munge cost inequality; consider a "stove pipe" workflow specialized for a specific analysis that is composed entirely of shims but is very easy to run. However, the same workflow may not be easily extended to do other analyses, preventing other analysts from easily reusing the infrastructure.

When analysts are also tool developers, they have intimate awareness of how their tool is designed and can easily modify it to suit ongoing needs, including tailoring the tool for a specific kind of novel data. Although current munging costs associated with the analyst/tool developer may be trivialized and reduced, subsequent usage of the tool in other applications using different data are likely to incur much steeper costs.

When tool developers are also data providers, it is regularly observed that they build tools that hard code data sources. For example, the emerging set of D3 visualizations usually couple the data handler callback function with a hard coded URL that references the originating dataset for which the visualization was designed. In order to reuse these visualizations, analysts must modify the source JavaScript which can incur significant costs depending upon the design and analysts' familiarity with the code.

With regards to munging patterns, one-star applications do not reduce the space of possible munge sequences; the triad restrictions to not constrain the structure of data involved nor the behavior of the tools used, for example capturing and preserving provenance information. One star-applications thus encompass the universe of munging that can range from an inexpensive case of only computing to a hypothetical worst case where analysts are unable to locate or verify quality of prior materials and must abandon analyses, as illustrated by the one-star munge graph in Figure 12. Without loss of generality, the munge sequences presented in the figure:

- show only two stages per application: one munge performed to get data into a tool and another performed using the tool itself
- assumes that data must *always* be munged to fit into a tool
- do not include trivial munges (i.e., gleaning and content negotiation); we assume the data entering the munge sequence has already been gleaned or content negotiated

The cost bounds associated with these worst/best-case munge sequences is also expressed as the following interval:

$$\begin{aligned} cost(\alpha_\star) < cost(\alpha_{0-star}) &= [0, \infty] \\ &= [2 \times cost(comp), \infty] \\ &= [6, \infty] \end{aligned}$$

[17] http://data-gov.tw.rpi.edu/demo/USForeignAid/demo-1554.html

[18] http://lov.okfn.org/dataset/lov/

[19] http://sparqles.okfn.org/

[20] http://lov.okfn.org/endpoint/lov

The lower bound results from the best case when two compute munges can be performed; the cost to perform a single computation is 3, as defined by the cost function $cost$ in Section 4. The upper cost bound ∞ represents the worst case when an analysis is abandoned due to unavailability or inability to verify quality of materials.

5.2. *Two-star applications*

Two-star applications accept data via URL and always cite that URL in the future. This restriction applies to any kind of data, i.e., $tbl(D) >= 1$; the function tbl maps a dataset D to its star rating as determined by Tim Berners-Lee's scale. Additionally, the restriction requires that applications maintain a simple and natural provenance since generated materials must cite the URL of the data from which they were derived. Like the previous rating, two-star applications also fulfill the requirement that data providers, analysts, and tool developers are disjoint. The two-star restriction is shown near the top of Figure 10, where the data D is available on the Web, has an associated dcat:Distribution leading to a URL for D's access, and the access URL is referenced within the generated dataset. Figure 2 introduces a data subset notation with a curved arrow tail and a small circle embedded within the larger circle D_α.

In Amy's analysis, her use of Google Earth in application $\alpha_{1,2}$ earns two stars; she used KML satellite data that is available on the Web. Additionally, Mary could copy away the URL of the KML file from the Google Earth visual, fulfilling the latter clause of the two-star requirement. In contrast, the use of LOV and SPARQL-ES earn two *conditional* stars. Conditionality refers to cases when an application fulfills a particular star level requirement but fails to fulfill the immediately-preceding level's requirement(s). Although LOV and SPARQL-ES fulfill two-star requirements by accepting URLs for data, these two tools violate the one-star condition by having that the tool developer and data provider as the same entity.

With regards to munging patterns, two-star restrictions provide a greater possibility that analysts can complete their applications since the provenance of used and generated materials is available. However, even with provenance and the availability of data on the Web, analysts may still incur significant costs since the data may be either mundane or semantic, providing for all munge sequence possibilities. The munging

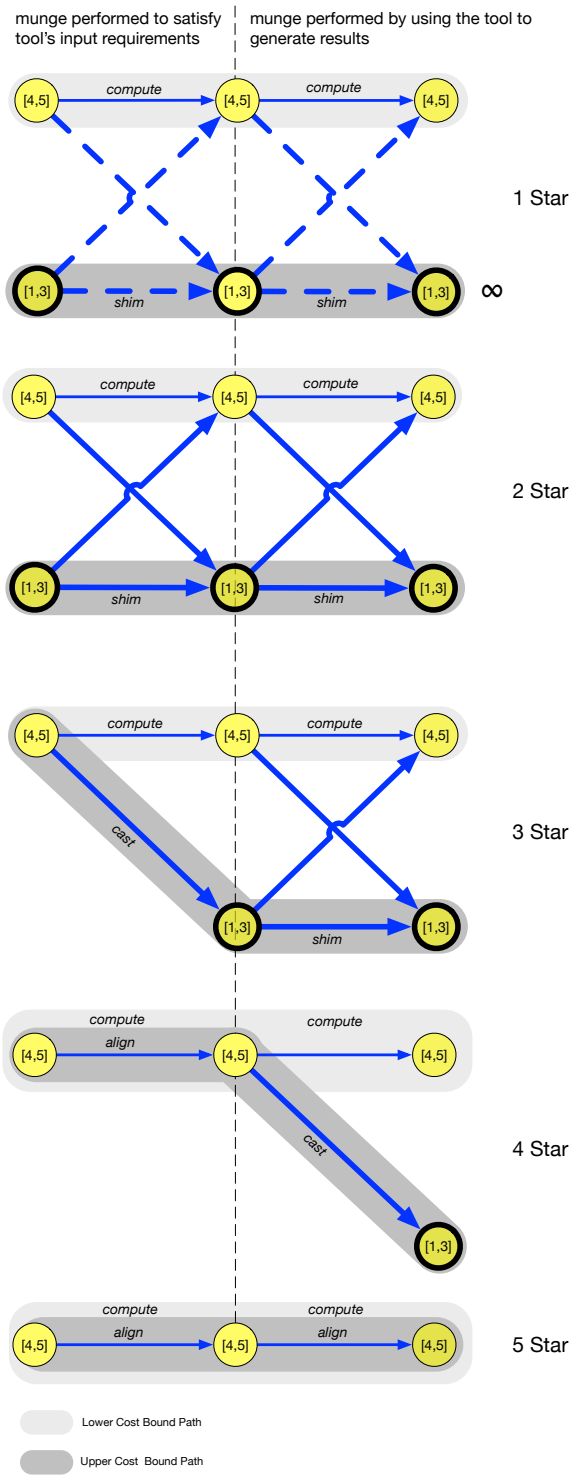

Fig. 12. Possible munge patterns associated with each application subclass. As the application restrictions increase, the space of possible munge sequences reduces. Dashed lines indicate that the munging may not be possible due to unavailability of data.

space therefore contains exclusive sequences of computes and shims.

The cost bounds associated with these worst/best-case munge sequences is also expressed as the following interval:

$$\begin{aligned} cost(\alpha_{\star\star}) < cost(\alpha_{\star}) &= [6, \infty] \\ &= [2 \times cost(comp), 2 \times cost(shim)] \\ &= [6, 38] \end{aligned}$$

The cost bound for two-star applications is not only *tighter* than one-star applications, but also *lower* since the upper cost is reduced from ∞ to the exclusive cost of shimming, 38.

5.3. Three-star applications

Three-star applications accept RDF data via URL, i.e., $tbl(D) >= 4$. The data can be "pure" RDF or embedded in a gleanable, mundane dataset. Like the previous rating, three-star applications must also use data available on the Web. The three-star restriction is shown at the top of Figure 10, where the Web accessible data is available as RDF. Web accessible RDF data is a subclass of the Web accessible data used by the two-star restriction and thus inherits the presence of a distribution URL.

In Amy's scenario, her final use of Google Earth in application $\alpha_{6,1}$ earns three stars; the application used an RDF dataset $D_{5,1a}$. Her use of the semantic histogram, Range Chart, and Aduna ClusterMap also earn at least three-stars since these applications either used RDF or were gleanable mundane datasets. Similarly, any applications that use generic linked data browsers [25,24,26] can earn at least three-stars as long as they also meet the one- and two- star requirements.

Three-star applications restrict the kind of data they use and therefore narrow the possible space of munge sequences that can be performed. Figure 12 presents the hypothetical minimum and maximum munge sequences associated with the reduced munge space induced by three-star applications. The munge space includes sequences that consist of only semantic munges to a more expensive case of "hill sliding" back down to the mundane level.

The cost bounds associated with these munge sequences is also expressed as the following interval:

$$\begin{aligned} cost(\alpha_{\star\star\star}) < cost(\alpha_{\star\star}) &= [6, 38] \\ &= [2 \times cost(comp), cost(cast) + cost(shim)] \\ &= [6, 24] \end{aligned}$$

The cost bound for three-star applications is not only *tighter* than one- and two-star applications, but also *lower* since the upper cost is reduced from 38 to 34. This reduction is a result of replacing a single shim that is possible in one- and two-star applications with a less expensive cast that is facilitated by using RDF.

5.4. Four-star applications

Four-star applications use a tool's input semantics (OWL, SPARQL) when performing munges, i.e., $used(m, t_\sigma) \wedge m \in M$. Like the previous rating, four-star applications also accept RDF via URL. Figure 10 depicts the four-star application restriction toward the center-bottom, where a munge m uses a tool t's input semantics t_σ during an application.

In the scenario, Amy's uses of the semantic histogram in applications $\alpha_{3,1}$ and $\alpha_{3,2}$ earn four stars since her input data was encoded in RDF and the tool's input semantics were available, allowing her to align data to forms expected by the semantic histogram tool. The semantic histogram tool's input semantics are shown below and encoded using Manchester Syntax [21]; the histogram tool accepts "things that are binned according to some category.[22]"

```
Class: hist:BinnedThing
EquivalentTo:
    (hist:binnedBy some hist:Category)
Class: hist:Category
```

Four-star applications are not in widespread use, but similar work in automated service orchestration relies on these kinds of applications. For example, the Semantic Automated Discovery and Integration (SADI) framework advocates that service providers publish input and output data requirements in the form of OWL, providing service consumers with an unambiguous representation of the service's I/O requirements. The SADI services, in turn, process RDF that complies

[21] http://www.w3.org/TR/owl2-manchester-syntax/

[22] DATACUBE

with the input semantics and generate additional RDF compliant with the output semantics. Bioinformatics applications that use SADI services to process gene sequence and protein data [27], therefore, are rated at least four stars.

When analysts perform four-star applications, we hypothesize that analytical seamlessness is increased. From the required use of RDF and the presence of input semantics, we can infer that tools employed in four-star applications accept RDF. Analysts can therefore perform low-cost computations and alignments to fit data into their tools. Four-star applications, however, do not restrict the kind of data generated and thus D_α can be mundane or semantic. Figure 12 presents the hypothetical minimum and maximum munge sequences, which range from the best case of performing only semantic computations to a more expensive case of "hill sliding" back down to the mundane level.

The cost bounds associated with these munge sequences is expressed as the following interval:

$$\begin{aligned} cost(\alpha_{\star\star\star\star}) < cost(\alpha_{\star\star\star}) &= [6, 24] \\ &= [2 \times cost(comp), cost(align) + cost(cast)] \\ &= [6, 9] \end{aligned}$$

29] do not earn five-stars. Although that work analyzed third party linked data, the results of the analyses are mundane and embedded in static imagery or plain text embedded in journal articles.

When analysts perform five-star applications, we hypothesize that analytical seamlessness is further increased. Analysts have the opportunity to perform semantic munges exclusively during the entire duration of a five-star application, yielding the tightest and lowest cost bounds of any other application subclass. Figure 12 presents the hypothetical minimum and maximum cost munge sequences associated with five-star applications, which ranges from the best case of performing only computations to a slightly more expensive case of performing only alignments. We deem applications that cast semantic data to gleanable, mundane representations equivalent to computing/aligning linked data directly. The figure therefore does not illustrate the former case of generated gleanable content.

Five-star munge sequences are thus predicted to fall within the following cost interval:

$$\begin{aligned} cost(\alpha_{\star\star\star\star\star}) < cost(\alpha_{\star\star\star\star}) &= [6, 9] \\ &= [2 \times cost(comp), 2 \times cost(align)] \\ &= [6, 8] \end{aligned}$$

5.5. Five-star applications

Five-star applications provide some information derived during use as linked data, i.e., $D \subset D_\alpha$. Like the previous rating, five-star applications also accept RDF via URL and use a tool's input semantics during munging. Figure 10 depicts the five-star application restriction toward the right, where the RDF data used in an application is a subset of the generated result D_α.

In the scenario, Amy's use of the Range Chart tool in application $\alpha_{4,1}$ earn five stars; the application used a gleanable dataset and generated a gleanable result. Amy's use of the semantic histogram in applications $\alpha_{3,1}$ and $\alpha_{3,2}$ similarly earn five stars. All these applications used and generated gleanable, mundane datasets, which embed easily obtainable semantic data.

Tim Berners-Lee's tabulator [24] can also be used by analysts to perform five-star applications. Tabulator emits RDF corresponding to edits that analysts make while browsing third party data. SADI can also support five-star applications, since they generate RDF that is a superset of their input. In contrast, the applications comprising the analysis of the linked data cloud [28,

5.6. Substantiation of the Five-Star Theory

Our Theory of Seamless Analytics explains how five-star applications constrain the space of possible munge sequences. We can therefore obtain reasonable seamlessness predictions by only considering the star ratings of applications that induced an ecosystem. To demonstrate, we adjust the star ratings of Amy's applications, modify the associated munges accordingly, and recalculate her seamlessness scores to illustrate the effects.

We reuse Amy's applications from Section 3 as a baseline ecosystem to compare with an alternate, hypothetical ecosystem induced from modifications of the baseline. In ecosystem E, Amy performed 1 zero-star application[23], 3 two-star applications, 3 three-star applications, and 3 five-star applications, resulting in seamlessness (S_1) and prospective seamlessness (S_*) scores of 2.04 and 2.19, respectively. We will craft the

[23] During application $\alpha_{2,2}$, Amy became a tool developer by configuring Sgvizler at the JavaScript level.

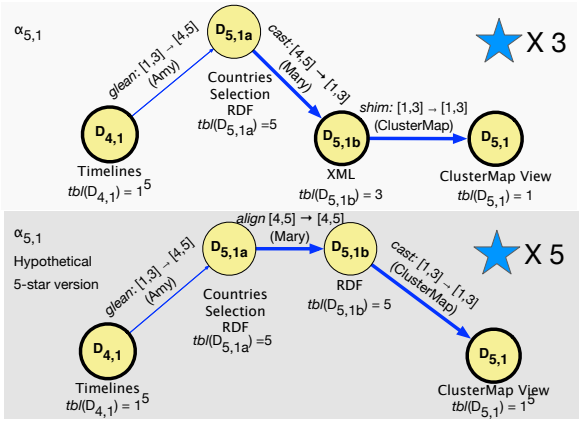

Fig. 13. Amy's three-star application using Aduna ClusterMap versus a hypothetical five-star variant.

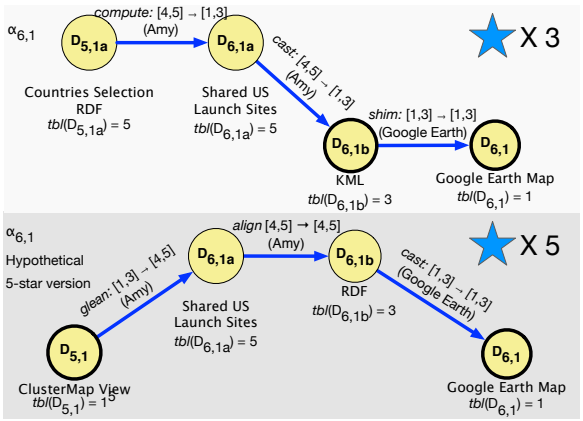

Fig. 14. Amy's three-star application using Google Earth versus a hypothetical five-star variant.

alternate ecosystem E_1 by substituting Amy's 3-star applications $\alpha_{5,1}$ and $\alpha_{6,1}$ with hypothetical five star variants.

Had Amy performed a five-star application during $\alpha_{5,1}$, the resulting hypothetical ecosystem E_1 would not have exhibited the expensive *house top* anti-pattern. Figure 13 shows Amy's three-star usage of ClusterMap juxtaposed with a hypothetical five-star version using the same tool. In the five-star variant, Amy performs a less costly align-cast combination rather than the cast-shim that was actually performed from $D_{5,1a}$ to $D_{5,1}$. The resultant dataset $D5,1$ in the hypothetical version is gleanable and therefore can be cheaply processed to extract the embedded semantic content. To accommodate, ClusterMap would need to be modified to accept hierarchical RDF and produce gleanable visualizations.

Had Amy also performed a five-star application during $\alpha_{6,1}$, the resulting hypothetical ecosystem E_1 would not have exhibited the expensive *hill slide* anti-pattern. Figure 14 shows Amy's three-star usage of Google Earth juxtaposed with a hypothetical five-star version using the same tool. In the five-star variant, Amy performs a less costly glean-align-cast combination rather than the compute-cast-shim that was actually performed from $D_{6,1a}$ to $D_{6,1}$. Additionally, Amy would have been able to use the result generated by Aduna ClusterMap in the previous application $\alpha_{5,1}$, rather than resorting to an earlier, less evolved dataset. To accommodate, Google Earth would need to be modified to accept geospatial RDF and produce gleanable geospatial visualizations.

Table 3 lays out the change in seamlessness scores between Amy's actual ecosystem E and the hypothetical variant E_1. Both scores S_1 and S_* when we incorporated a greater number of five-star applications. These applications resulted with less expensive, current and prospective munge sequences that reduced overall cost. The scores presented in this paper are merely used to exemplify the relative change in seamlessness rather than used as an absolute measure of reality. Practically, these deltas correlate with a variety of concrete measures such as time, money, effort, and cognitive load just to name a few.

Table 3
The seamlessness scores of E and the hypothetical variant E_1.

	E	E_1	△ score
S_1	2.04	2.31	0.27
S_*	2.19	2.64	0.45

6. Related Work

Early visualization researchers were focused on developing models to help them understand how data is transformed into views, rather than predicting costs incurred by those transformations [30,31]. Chi devised a visualization transform model that is useful for describing how data evolves from its raw state to a view state as it passes through a four-stage pipeline of *operators* [32]. Some of these operators, for example *data stage* operators, are broadly specified and could encompass munging activities. Chi's objectives, however, were centered on understanding and compar-

ing different visualization techniques in terms of their pipeline structure, not their imposed costs [33].

In parallel, the visual analytics community has continually developed and revised analytical cost models for decades [34,35]. These models mainly consider *cognitive* costs incurred by performing user interactions [36] and visual pattern recognition. In particular, Patterson presents a cognitive model of how analysts interpret and reason with visual stimuli in order to generate responses, e.g., decisions. He presents six leverage points that visual designers should employ in order to reduce the costs imposed on analysts when interpreting visualizations [19].

When analytics researchers do consider lower level transforms, the result has been models that are specified at too high a level to easily calculate quantifiable cost predictions. Wijk proposed an economic model that considers the ratio of value (i.e., knowledge gained) to the cost incurred to generate a visualization [37]. Wijk specifically highlighted the cost to *initialize* data, which can be equated to the cost of performing munging. It is not clear, however, which specific factors influence this cost, leaving analysts with little direction as to how to better quantify and mitigate those costs.

In contrast, Kandel's work provides a more detailed description about the different kinds of data munging analysts must perform and even outlines a set of research goals [6] that paved the way for a munging tool [23]. His hierarchy of munging types and research directions were elicited from actual enterprise analysts [9], the testimonies of which prove substantiating evidence that supports our simple theory of analytical seamlessness. Kandel begins to discuss the use of semantic data types to address the challenges of formatting, extracting, and converting data to fit input data requirements. He mentions that these data types should be shared and reused across analyses, similar to how the Linked Data community advocates the reuse of popular vocabularies [38]. However, we believe our seamlessness theory represents the next logical step of his work by articulating his analysts' testimonies in a form that can emit testable predictions.

Fink also outlined a set of challenges that were elicited from analysts in cyber-security settings [22]. He found that, like Kandel's enterprise subjects, analysts are limited by their capability to cheaply mitigate disparities among diverse data and tools. Some of the analysts interviewed by Fink believe that analytic environments should be as flexible as UNIX shells and allow arbitrary visualization tools to be piped together (e.g., application chains).

Meanwhile, the Linked Data community has long considered the potential costs and benefits associated with publishing and consuming linked structured data. Tim Berners-Lee is a proponent of Linked Data because of the potential benefits afforded to data consumers, whom can more easily discover, integrate, and reuse distributed RDF [24]. His scheme has been useful in understanding the affordances provided to data consumers in a *client-server* setting, where data is only generated by publishers. Our work, however, uses his scheme to understand the costs and benefits in a *peer-to-peer* setting, where consumers are also publishers that transform existing linked-data into new, additional RDF that can be ingested by other analysts.

Similarly, Janowicz and Hitzler [39] describe how the Semantic Web provides analysts with an opportunity to use third-party data in contexts not envisioned by the data provider. Analysts are able to quickly develop application-driven schema knowledge that can be used to align data into arbitrary, suitable forms required by tools. In the same spirit, Heath and Bizer describe an application architecture for linked data applications, citing data access (e.g., HTTP Get) and vocabulary mapping (i.e., a kind of munging) as major components [1]. The Data Integration layer that encompasses alignment-type munges is essential for supporting the more abstract and user-driven application layer.

When "cost" is mentioned in Linked Data literature, it is referred to in an abstract manner and not usually expressed in mathematical forms from which it can be calculated. The pieces of a theory are there, however they are not consolidated and formally articulated into a framework that can be used to test and predict the community's hypotheses regarding "ease-of-use". We believe our work embodies the community's assumptions, claims, and hypothesis as a simple theory that can be used to assess, predict, and refute the tenets of Linked Data that have been advertised for nearly a decade.

7. Future Work

Our future work can be grouped into three broad sections: refining our Theory of Seamless Analytics,

[24] http://5stardata.info

developing practical techniques for supporting seamlessness, and mapping out the Linked Data landscape. Our Theory of Analytical Seamlessness can be seen as a *class*, where the specific set of munge activities, seamlessness metric, and application ratings can collectively be seen as an *instance* of the theory. It may be possible to instantiate a Theory of Seamlessness using other factors such as cognitive load and coherence, so long as the instantiation is able to predict the cost to perform an analysis. The rest of this section describes future work of our specific theory instance and therefore includes alternative munge cost orderings and value bindings.

In terms of theory refinement, we can defer to organizations and communities to help us gather evidence that either supports or refutes our seamlessness theory. We would need to maintain a library of application chains that describe actual, realistic, and hypothetical uses cases. Similarly to VAST challenges [40], we can ask the Visual Analytics and Linked Data communities to implement the use cases using a variety of alternate technologies that challenge both the completeness and ordering of our munge ontology. When our theory fails to accurately predict seamlessness, we can accommodate by revising the membership and orderings of our munge hierarchy. Such an open testing framework allows the two communities to join forces and expose dominant cost factors using a common set of scenarios.

We can also elaborate on the distinction between mundane (1-3) and semantic (4-5) munges. Currently, our model stereotypes four- and five-star data into the same class, however, we observe significant cost differences in creating five-star data. Analysts must have experience in good URI design, popular vocabularies [25] and popular data repositories such as DBPedia [26] and datahub.io [27]. Additionally, analysts need to have some grasp of RDF patterns, such as PROV qualified associations and Semantic Science Integrated Ontology (SIO) [28], so they can understand how to more effectively anchor their RDF to existing linked data in more discoverable and recognizable ways. Considering this required experience, our theory should be extended to account specifically for the cost of generating high quality linked data. These extensions should draw strength from the ongoing work in the area of Linked Data quality, while at the same time inform future work in that area.

We can also strengthen our theory by considering the full space of cost bindings, which may be informed by an analyst's operational environment. Whereas the cost bindings presented in this paper are used only to exemplify relative cost savings, we expect that different organizations will specify cost bindings that better reflect their work environment, such as the maturity of their staff, their familiarity with different data formats, and the size of their software toolset. For example, an organization with a strong workforce of linked data researchers might further widen the cost between shimming and aligning to better emphasize the work force's efficiency with semantic technologies.

Along the same line, we can update our *pot* function to return the actual cost savings associated with subsequent usage of results. Our current function only returns a hard-coded, predicted discount based on semantic level of the data generated by applications. We can update the function to "follow up" on the actual gains by drawing from the provenance of downstream usage [41], providing analysts with additional feedback regarding their impact.

In terms of practical development, we can explore different representations for expressing a tool's inputs semantics. In addition to OWL, input semantics may be represented using SPARQL and even Java interfaces, provided the analyst's environment is supported by toolbox of Java APIs. An architecture that permits either representation will allow us to concretely compare among associated costs, eventually converging on the lowest cost option. We are currently developing an API that allows visualization widgets to self-describe their input semantics using either representation, providing us with the testing framework needed to make the best determination. Our hopes is to invert the current asymmetry; instead of analysts spending the majority of their time preparing views, we would rather they spend their time observing and vetting an abundance of automatically-generated visualization options – determined by matching data with the input semantics of a tool. We want to put the analyst into a "visualization zombie apocalypse" and change the paradigm to focus on *eliminating* visualizations that are not desirable from the abundance of what can be instantly available.

Analysts could also benefit if tools were associated with "output" and "parameter" semantics. Although input semantics help analysts get data into a tool, they

[25] Linked Open Vocabularies (LOV) maintains a listing of crowd sourced vocabularies `http://lov.okfn.org/dataset/lov/`

[26] `http://dbpedia.org`

[27] `http://datahub.io/`

[28] `http://semanticscience.org/`

do not help analysts understand the products generated by the tool. Additionally, input semantics do not help analysts understand how to tool can be configured to fine tune a result. Modeling the output and parameter semantics – not just the input semantics – could provide a useful information to determine whether or not such a tool *should* be used, beyond whether or not it *could* be used. Parameter semantics could provide useful information about the configuration of possible results, for example, orientation, coloring, and grouping.

There is also a need for clearly-defined and motivated knowledge-based software development methodologies that can help guide visualization tool developers. These new development methods would reconcile paradigms from both the software engineering and Linked Data communities, combining both sets of principles into a well-engineered process to build more seamless tools. This will allow Linked Data researchers to communicate the advantages of Linked Data to software engineers in the context of reducing development costs such as requirements elicitation and testing.

When cost constrains prohibit ground-up tool development, techniques for enabling Linked Data to work with legacy systems is also desirable. These legacy systems can exist anywhere on the spectrum from being developed in-house with a white-box knowledge of their inner-workings, to an open source version with limited knowledge of their inner-workings, to a full black box system such as Google Earth. To accommodate, we have begun developing a method for intelligently casting linked data to mundane representations required by legacy systems while at the same time providing the capability to return to semantic levels via cheap "gleans," both from concrete representations (i.e., a file) and their references (i.e. URIs).

We also need to culminate the expansive work on ontology mapping to firmly establish "alignments" as a truly cheaper alternative to shimming. The mapping solution space is large and finding the appropriate correspondences depends on the context of an analyst's task. We might therefore leverage the views generated by tools as targets that help guide analysts to the most beneficial mapping. As an analyst specifies a mapping and integrates data into a tool, they can inspect the visual generated by the tool and judge the usefulness of the mapping based on the utility of the visualization.

Another novel approach to the mapping challenge is to leverage cumulative social usage history. Rather than using mapping ontologies as "one-off" ephemeral bridges between data and tools, the resultant ontologies could have a more global effect on reducing munge costs of subsequent analyses. Mapping ontologies should be regarded as analytical products, and used by subsequent analysts as road maps for how to align their specific data.

In terms of the Linked Open Data cloud, we could support a new era of "massive ontology landscapes" laid out from the ground-up when analysts perform alignments. Considering the pairing of tools and input semantics, every time a new view enters an ecosystem its input semantics can be reconciled with all existing input semantics, either directly or indirectly through property chaining. Doing so allows us to evolve broader and deeper expression of visualization input semantics that currently only exists implicitly with visualization expert's tacit knowledge based on experience. Rather than a few organizations developing monolithic ontologies describing the linked data cloud, visualization tools that publish input semantics create a compelling reason for analysts to build these mapping ontologies from the ground up. The centralized effort of developing ontologies can now be crowd sourced to analysts who are just doing their job.

8. Conclusion

We forged a Theory of Seamless Analytics that predicts the cost analysts expend when performing non-trivial analyses that span across multiple applications. The theory attributes these global analytical costs to the underlying low-level munges that transform data into alternate representations required by tools. It suggests that analysts can more efficiently generate and use materials during analyses when they perform applications that require "semantic munging."

To describe our theory, we presented an extension of the W3C PROV Ontology that models analytic applications regardless of the type of data, tool, or objective involved. Our Application Ontology outlines three broad classes of munging that are distinguished by the five-star rating of the data from which they operate: mundane or semantic. If an analytical trace is described using our ontology, we can assess its seamlessness by applying a metric which rewards applications that process semantic data.

Our theory also describes how a particular munge sequence can be inferred from the structure of data and behavior of tools used to perform an application. From the combinations of possible data and tool behavior, we identify five configurations that are well differen-

tiated in terms of their associated munging and rate these configurations according to their predicted cost. The pinnacle "five-star application" uses a tool's input semantics and processes semantic data that leads to cheaper munge sequences. We demonstrate this association by comparing the seamlessness of a control analysis centered on understanding the environmental condition of Earth's orbit with an alternate version that employs five-star applications. We observed that the alternate version resulted with greater semantic munges that reduced the overall cost to perform the analysis.

References

[1] Tom Heath and Christian Bizer. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.

[2] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the linked data best practices in different topical domains. In *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014.

[3] James J Thomas and Kristin A Cook. *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press, 2005.

[4] Shixia Liu, Weiwei Cui, Yingcai Wu, and Mengchen Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, pages 1–21, 2014.

[5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D^3 data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[6] Sean Kandel, Jeffrey Heer, Catherine Plaisant, Jessie Kennedy, Frank van Ham, Nathalie Henry Riche, Chris Weaver, Bongshin Lee, Dominique Brodbeck, and Paolo Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011.

[7] Ghislain Auguste Atemezing and Raphael Troncy. Towards Interoperable Visualization Applications Over Linked Data. In *Talk Given at the 2nd European Data Forum (EDF), Dublin, Ireland (April 2013), http://goo. gl/JhVrax*.

[8] Timothy Lebo, Satya Sahoo, Deborah McGuinness, Khalid Belhajjame, James Cheney, David Corsar, Daniel Garijo, Stian Soiland-Reyes, Stephan Zednik, and Jun Zhao. Prov-o: The prov ontology. *W3C Recommendation, 30th April*, 2013.

[9] Sean Kandel, Andreas Paepcke, Joseph M Hellerstein, and Jeffrey Heer. Enterprise data analysis and visualization: An interview study. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2917–2926, 2012.

[10] Timothy Lebo and Gregory Todd Williams. Converting governmental datasets into linked data. In *Proceedings of the 6th International Conference on Semantic Systems*, page 38. ACM, 2010.

[11] LOD2 Collaborative Project. Report on Knowledge Extraction from Structured Sources. Technical report, 2010.

[12] RaphaÃńl Troncy Gabriel Kepeklian and Laurent Bihanic. Datalift: A platform for integrating big and linked data. In *In International Conference on Big Data from Space (BIDS'14), Rome, Italy, November 12-14, 2014 (to appear)*, 2014.

[13] Natalya F Noy. Semantic integration: a survey of ontology-based approaches. *ACM Sigmod Record*, 33(4):65–70, 2004.

[14] Carlos Buil-Aranda, Aidan Hogan, Jürgen Umbrich, and Pierre-Yves Vandenbussche. SPARQL Web-Querying Infrastructure: Ready for Action? In *The Semantic Web–ISWC 2013*. 2013.

[15] John R Anderson. Act: Asimple theory of ecoplex cognition. *American Psychologist*, 51(5):355, 1996.

[16] Manolis Savva, Nicholas Kong, Arti Chhajta, Li Fei-Fei, Maneesh Agrawala, and Jeffrey Heer. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 393–402. ACM, 2011.

[17] Ian Jacobs and Norman Walsh. Architecture of the world wide web. 2004.

[18] Martin G Skjæveland. Sgvizler: A javascript wrapper for easy visualization of sparql result sets. In *Extended Semantic Web Conference*, 2012.

[19] Robert E Patterson, Leslie M Blaha, Georges G Grinstein, Kristen K Liggett, David E Kaveney, Kathleen C Sheldon, Paul R Havig, and Jason A Moore. A human cognition framework for information visualization. *Computers & Graphics*, 42:42–58, 2014.

[20] Timothy Lebo, Alvaro Graves, and Deborah L McGuinness. Content-preserving graphics. In *COLD*, 2013.

[21] Marti A Hearst. User interfaces and visualization. *Modern information retrieval*, pages 257–323, 1999.

[22] Glenn A Fink, Christopher L North, Alex Endert, and Stuart Rose. Visualizing cyber security: Usable workspaces. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pages 45–56. IEEE, 2009.

[23] Sean Kandel, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3363–3372. ACM, 2011.

[24] Tim Berners-Lee, James Hollenbach, Kanghao Lu, Joe Presbrey, and Mc Schraefel. Tabulator redux: Browsing and writing linked data. 2008.

[25] Jans Aasman and Ken Cheetham. Rdf browser for data discovery and visual query building. In *Proceedings of the Workshop on Visual Interfaces to the Social and Semantic Web (VISSW 2011), Co-located with ACM IUI*, page 53, 2011.

[26] Tuukka Hastrup, Richard Cyganiak, and Uldis Bojars. Browsing linked data with fenfire. 2008.

[27] Mark D Wilkinson, Luke McCarthy, Benjamin Vandervalk, David Withers, Edward Kawas, and Soroush Samadian. Sadi, share, and the in silico scientific mehtod. *BMC bioinformatics*, 11:S7, 2010.

[28] Aidan Hogan and Jü Umbrich. An empirical survey of linked data conformance.

[29] Marko A Rodriguez. A graph analysis of the linked data cloud. *arXiv preprint arXiv:0903.0194*, 2009.

[30] Stuart K Card, Jock D Mackinlay, and Ben Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[31] Robert B Haber and David A McNabb. Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing*, 74:93, 1990.

[32] Ed Huai-hsin Chi and John T Riedl. An operator interaction framework for visualization systems. In *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 63–70. IEEE, 1998.

[33] Ed H Chi. A taxonomy of visualization techniques using the data state reference model. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pages 69–75. IEEE, 2000.

[34] Daniel M Russell, Mark J Stefik, Peter Pirolli, and Stuart K Card. The cost structure of sensemaking. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 269–276. ACM, 1993.

[35] Peter Pirolli and Stuart Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, volume 5, pages 2–4. Mitre McLean, VA, 2005.

[36] Heidi Lam. A framework of interaction costs in information visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1149–1156, 2008.

[37] Jarke J Van Wijk. The value of visualization. In *Visualization, 2005. VIS 05. IEEE*, pages 79–86. IEEE, 2005.

[38] Max Schmachtenberg, Christian Bizer, and Heiko Paulheim. Adoption of the linked data bset practices in different topical domains. In *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014.

[39] Krzysztof Janowicz and Pascal Hitzler. The digital earth as knowledge engine. *Semantic Web*, 2012.

[40] Kristin Cook, Georges Grinstein, Mark Whiting, Michael Cooper, Paul Havig, Kristen Liggett, Bohdan Nebesh, and Celeste Lyn Paul. Vast challenge 2012: Visual analytics for big data. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, pages 251–255. IEEE, 2012.

[41] Timothy Lebo, Patrick West, and Deborah L. McGuinness. Walking into the future with prov pingback: An application to opendap using prizms (in press). In Bertram Ludaescher and Beth Plale, editors, *Provenance and Annotation of Data and Processes*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014.