A Five-Star Rating Scheme to Assess Application Seamlessness

Editor(s): Aba-Sah Dadzie, The Open University, UK; Emmanuel Pietriga, INRIA France & INRIA Chile
Solicited review(s): Roberto Garcia Gonzales, Universitat de Lleida, Spain; Aba-Sah Dadzie, The Open University, UK; John Howse, University of Brighton, UK

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

Abstract.
Visual analytics is a costly endeavor in which analysts must coordinate the execution of incompatible visualization tools to derive coherent presentations from complex information. Distributed environments such as the Web pose additional costs since analysts must also establish logical connections among shared results, decode unfamiliar data formats, and engage with broader sets of tools that support the heterogeneity of different information sources. These ancillary activities are often limiting factors to our vision of *seamless analytics*, which we define as the low-cost generation and reuse of analytical resources. In this paper, we offer a theory of analytics that formally explains how analysts can employ Linked Data to maintain and leverage explicit connections across shared results as well as manage different representations of information required by visualization tools. Our theory builds on the well-known benefits of interconnected *data* and provides new metrics that quantify the utility of interconnected user- and task-centric, analytical *applications*. To describe our theory, we first 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 chain of applications described in our Application Ontology. Finally, we extend the ontology to distinguish five types of applications based on the structure of data involved and the behavior of the tools used. Together, our seamlessness measure and application ontology compose our Five-Star application theory that embodies tenets of Linked Data in a form that 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 (LD) is a large, decentralized, and loosely-coupled conglomerate covering a variety of topical domains and slowly converging to use well-known vocabularies [13,35]. To more fully reap the benefits of such diverse data, LD analysts must employ an equally diverse array of analytical tools. Meanwhile, the Visual Analytics community (VA) has been forging a science of analytical reasoning and interactive visual interfaces to facilitate analysis of "*overwhelming amounts of disparate, conflicting, and dynamic information* [9]." Although the VA community has produced a vast array of tools and techniques that could assist [28], these tools cannot be easily reused in evolving environments such as the world of LD analytics. The tools are typically developed to work with very particular non-semantic representations that make it difficult to establish and maintain connections across analyses. Regardless of which community's ap-

[*] Corresponding author. E-mail: Timothy.Lebo@us.af.mil

[**] DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. Case Number: 88ABW-2014-5577

proaches are adopted, the need to continually form explicit and well-defined interconnections among the triad consisting of *data*, *analyst*, and *tool* remains a costly endeavor – and to benefit from both VA and LD research, these costs need to be more clearly portrayed, assessed, and overcome.

We attribute a large portion of analytical costs to two major factors:

- the ability to easily apply software tools to arbitrary data
- the ability to easily reuse and repurpose prior analytical materials

With respect to using software, the flexibility afforded by new APIs such as D3 [4] has resulted in a proliferation of "one-off" visualization tools that inhibit low-cost reusability. These new visualizations often lack documentation describing the schema of input data and can cause analysts to spend 80% of their time uncovering hard-coded, hidden assumptions [17]. With respect to reusing prior results, even if analysts could easily use the near two-thousand cataloged D3 visualizations[1] for their own endeavors, each visualization is a sink from the standpoint of subsequent analysts. Derived results, including interactions and selections, are not often saved or exported in forms that can be easily used in new, subsequent analyses.

Given these cost factors, we formalize a "five-star theory of analytics" that formally explains analytical costs and describes how analysts can use Linked Data to mitigate these costs. The theory combines work from the VA and LD communities and explains analytical costs in terms of data evolution (i.e, VA theory) and data structuredness (i.e., LD theory). As data evolves into ordered forms that facilitate analytic reasoning, it oscillates between two levels: a high-cost, mundane level (i.e., non-semantic) and a low-cost, semantic level that maintains connections.

Figure 1 highlights that our five-star theory is just one instance in a class of possible analytical cost theories which all should contain: a model to represent analyses, a cost metric defined in terms of the model, and cost reduction strategies.

Our contributions and sectioning of this paper are also illustrated in Figure 1. At the bottom of the image, 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 (not shown) 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 progressively reduce the cost of analyses. Section 6 (not shown) describes past work in the area of analytical models and techniques for supporting interoperability in analytical environments. Finally, Section 7 (not shown) discusses future work before the conclusion in Section 8 (not shown).

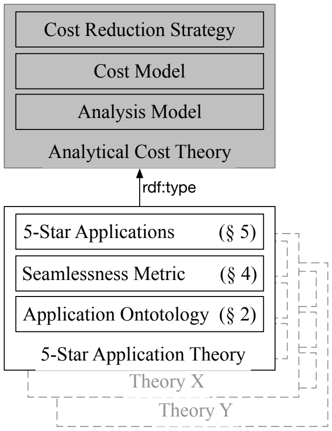

Cost Reduction Strategy
Cost Model
Analysis Model
Analytical Cost Theory
rdf:type
5-Star Applications (§ 5)
Seamlessness Metric (§ 4)
Application Ontotology (§ 2)
5-Star Application Theory
Theory X
Theory Y

Fig. 1. A theory of seamless analytics comprises three elements: a model, a cost model, and cost reduction strategies.

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**. An application refers to an analyst's contextualized use of some dataset within a tool to achieve some implicit objective, which contrasts with prior work of modeling applications as software components [2]. Ontologically speaking, an application is a kind of PROV Activity [25] and is therefore 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 also associates three key entities that we collectively refer to as the "application triad": 1) the input dataset, 2) the orchestrating analyst and 3)

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

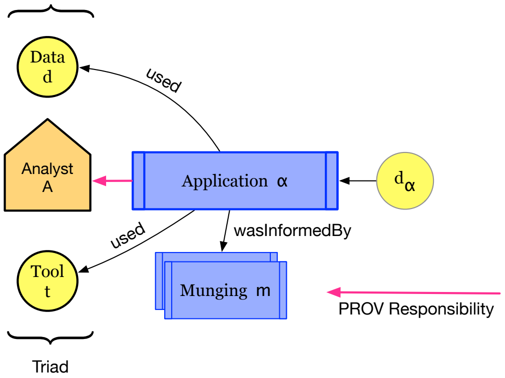


Fig. 2. Application Ontology Core is an extension of PROV. Applications use tools to generate new datasets which could include visualizations. Applications are informed by munging activities that transform data representations. Figure 9 illustrates an extension to further distinguish among our five types of applications.

the employed software tool. Figure 2 illustrates these relations using the PROV layout conventions[2] – analyst A used dataset d within tool t to derive d_α during application α. Application chains are formed when analysts use prior d_α's as input to new applications, as exemplified in Section 3. The cost of these application chains can be assessed using the seamlessness measure introduced in Section 4. Furthermore, application chains can be distinguished into five sub-types using the specifications introduced in Section 5.

The distinguishing aspect of our AO is the focus on **munging** activities that may be required to transform d into an alternate form that suits tool t's input requirements. Munging, also known as wrangling, is the imperfect manipulation of data into usable forms and has been recognized in the Visual Analytics (VA) field for decades, yet continues to be a ubiquitous and costly problem [19]. We focus on munging because it persists and dominates as a cost factor for applications.

The relationship between applications and munges is also shown in Figure 2 using PROV, but we further relate munging activities as also being *part of* the application[3].

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 [13].

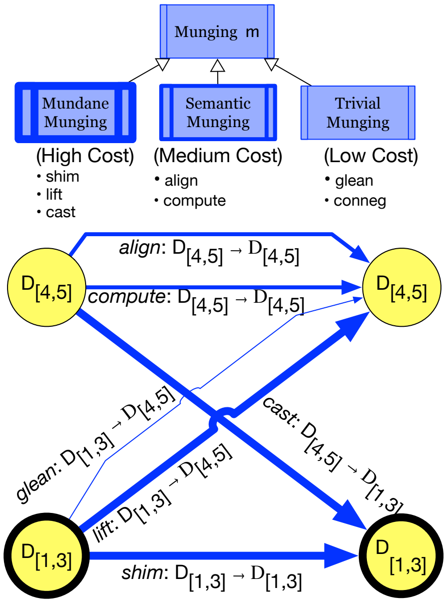


Fig. 3. Munging activities defined in terms of the Tim Berners-Lee's linked data scale. Not shown is content negotiation because it applies to all data types (an ideal situation).

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 LD communities that more highly rated data are easier to discover, reuse, and integrate. The seven sub-classes of munging (*shim*, *lift*, *cast*, *align*, *compute*, *glean*, and *conneg*) are defined in terms of using[4] 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. These abstract and coarse level costs are intended to reflect the ease at which data can be used within and across applications.

[2] http://www.w3.org/2011/prov/wiki/Diagrams

[3] Using Dublin Core `hasPart`, http://purl.org/dc/terms/hasPart

[4] We continue to follow PROV terminology to describe activities.

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 LD community is working to ameliorate.

Lifting ($lift$): generates $D_{[4,5]}$ from $D_{[1,3]}$; it creates RDF from non-RDF and has occupied the LD community's attention for most[5] of the past decade [27,38,21].

Casting ($cast$): generates $D_{[1,3]}$ from $D_{[4,5]}$; it creates mundane forms from RDF and, unfortunately, is regularly performed by many LD 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 [29].

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[6] and SPARQL-ES [5].

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[7] 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.*" [8]

3. An Analytical Scenario: Space Junk

This section presents two representative analyses modeled according to our application ontology presented in the previous section. Both analyses are centered on the broad topic of Earth's artificial satellites, e.g., their locations, type distribution, and associated launch sites. As our two analysts perform applications and inspect generated results, they will incrementally and serendipitously gain insight, formulate new questions, and perform subsequent applications to address their new inquiries. Collectively, the two analyses exemplify the "subsequent analyst" setting, where results of the first analyst are re-purposed by a second analyst with a different objective.

We use the scenario to unify the perspectives from the Visual Analytics (VA) and Linked Data (LD) communities. The VA community understands how information evolves into ordered *frames* that facilitate analytical reasoning [22,31,33]. The LD community understands how data structuredness (e.g., mundane or semantic) facilitates discovery, reuse, and integration [13,16]. We describe our representative analyses from both perspectives: as information evolves into ordered frames, it oscillates between mundane or semantic representations that affect how easily results can be repurposed.

We also use the scenario to highlight certain "anti-patterns," that can degrade an analyst's work performance [10,19] We posit that these anti-patterns create certain analytical "pain points" that have been well-documented by the VA community and which are paraphrased below:

[pp1] : understanding the structure and semantics of data
[pp2] : reusing prior application results
[pp3] : avoiding redundant work
[pp4] : obtaining different representations of data
[pp5] : understanding tools' input data requirements
[pp6] : obtaining the provenance of results

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

Finally, the applications described in this section are *instances* of the application class described in the previous section. To identify these application instances, we use subscripts, for example, α_1 denotes the first application an analyst performs. We also use subscripts to identify the result generated by a specific application, for example, d_{α_1} denotes the result generated by the first application. Finally, we use a pair of subscripts to identify an intermediate result, for example, $d_{1,2}$ denotes the dataset generated by the second munge of the first application. To disambiguate among applications and datasets across the two analyses, we will qualify the materials using the analyst's name, for example: Amy's α_1 or Bart's α_1.

3.1. Amy's Analysis

Amy, a student enrolled in a physics course, is learning about satellite launch trajectories and becomes curious about the amount of equipment launched into space. Although her professor states that over 2,000 functioning satellites have been launched from various countries, she remains curious about the satellites' location, classification, and ownership.

3.1.1. Application 1 (α_1): Where are the Satellites Located?

Amy begins her analysis with a URL of a Keyhole Markup Language (KML) dataset[9] that describes satellites':

- locations in orbit
- owning countries
- launch sites

Knowing that KML is a popular format for encoding geographical information, she uses an off-the-shelf Geographical Information System (GIS), such as Google Earth, to plot the location of the satellites.

Amy's activities are described by the provenance trace in Figure 4, which illustrates data transformations in terms of the seven types of munges defined in Section 2. In her first application, α_1, Amy shimmed the KML dataset, $d_{1,1}$, into a geospatial map, d_{α_1}.

The provenance trace for Amy's application α_1 exhibits a trivial case of the "flat-line" anti-pattern, which results when applications rely exclusively on shims. Figure 4 shows that Amy's first application was informed by a single shim operation: the transformation of a KML dataset into a set of set of pixels that represent a map. This resultant map, presented at the top of Figure 5, shows the location of over 50,000 satellites scattered throughout Earth's orbit. The map also provides an interactive legend with a set of checkboxes that allows Amy to toggle between the visibility of certain kinds of satellites which are classified as *Rocket Bodies*, *Debris*, *Active*, or *Inactive*.

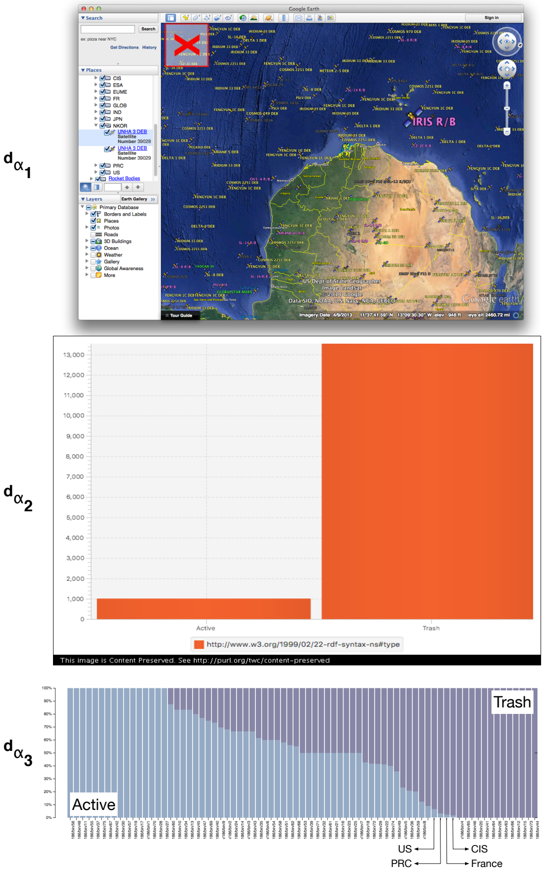


Fig. 5. Amy's application results.

Realizing that many satellites are inactive, Amy becomes interested in assessing launch efficiency by comparing the quantity of active, "useful" satellites to "space junk," which she defines as rocket bodies, debris, and inactive satellites. She clicks on the checkbox associated with active satellites and un-checks all other boxes, thus inducing a custom satellite grouping. She takes a screen shot of the map window and transitions into a new application, with a new objective.

[9] http://apps.agi.com/SatelliteViewer/

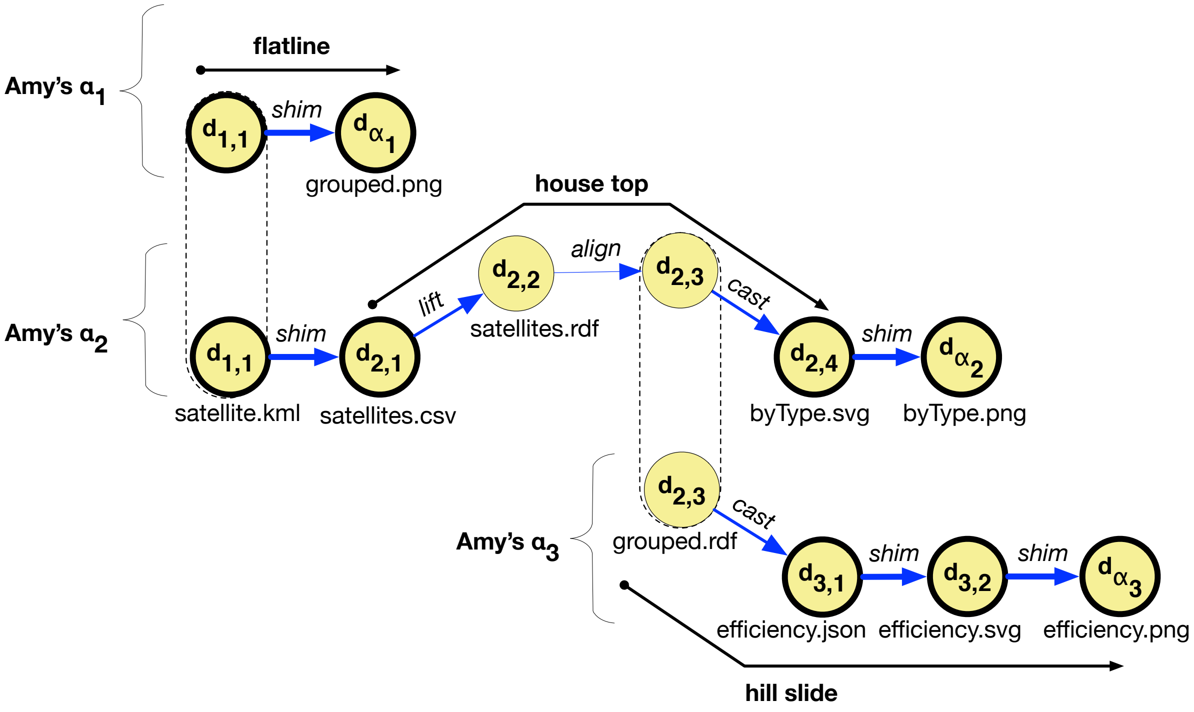


Fig. 4. Amy's analysis described using munge glyphs.

3.1.2. Application 2 (α_2): What is the Efficiency of Satellites Launches?

Amy can begin her second inquiry by building on materials generated in her first application:

$d_{1,1}$ URL to a KML satellite dataset
d_{α_1} map screenshot showing "useful" and "junk" satellites

The map screenshot, d_{α_1}, serves as Amy's analytical frame [22] and therefore most closely corresponds with her current understanding of satellites; there exists a set of visible satellites that Amy regards as useful and another set of hidden satellites that she regards as junk. She could use the map as a data source from which to calculate launch efficiency, but the information about the custom satellite groupings is embodied by pixels (and lack of pixels) and not explicitly linked to the underlying satellite KML. To extract the satellite grouping information into a more usable form, she would need to employ expensive **[pp2]** image processing [34].

Fortunately, the map screenshot displays the URL of the source KML dataset thereby supporting a kind of natural provenance **[pp6]** which arbitrary analysts could use to trivially retrieve the underlying satellite KML dataset. The KML dataset, although less in-line with Amy's current mental model of satellites, is at least structured. Unfortunately, Amy will have to reestablish her "useful" and "junk" satellite groupings from the KML file and thus redo work she performed while interacting with the geospatial map GUI **[pp3]**.

Using the KML dataset, Amy decides to generate a histogram showing the distribution of satellites by type. She first re-partitions satellites into her two groups, encodes these custom groupings using RDF, and uses an RDF visualization tool, such as Sgvizler [37], to generate a histogram.

This second application is described by the provenance trace labeled α_2 in Figure 4. The application reuses the satellite KML data, $d_{1,1}$, as indicated by the dashed lines in the figure. Amy first used a custom script to shim the satellite KML to a CSV file denoted as $d_{2,1}$. She then used a RDF converter[10] [38] to lift the CSV file into an equivalent RDF representation, $d_{2,2}$. A snippet of this RDF is shown below. Note that Amy created her own URIs for launch sites and countries instead of using existing DBPedia URIs.

```
<satellite-1> a pext:ActiveSatellite ;
```

[10] http://www.w3.org/wiki/ConverterToRdf

```
    prov:wasDerivedFrom
        <http://example.org/data/EasternRange> ;
    acl:owner <United_States> ;
    geo:lat    -100.000 ;
    geo:long    33 ;
    geo:alt     30000 .
<satellite-2> a :Debris;
    prov:wasDerivedFrom
        <http://example.org/data/EasternRange>
    acl:owner <UnitedStates>
    geo:lat    -130.000 ;
    geo:long   10 ;
    geo:alt    60000 .
```

Once she obtained RDF, Amy used an ontology mapping tool [29] to align her raw satellite RDF into a new dataset, $d_{2,3}$, which groups rocket bodies, debris, and inactive satellites as `nfo:Trash`. She controlled the mappings by specifying the following RDFS subclass axioms:

```
SubClassOf( :Debris        nfo:Trash )
SubClassOf( :RocketBody    nfo:Trash )
SubClassOf( :Inactive      nfo:Trash )
```

Amy finally used an RDF visualization tool to cast the grouped satellite data, $d_{2,3}$, into a SVG histogram, $d_{2,3}$. Supposing she used a tool, such as Sgvizler [37], she would have been required to to annotate HTML with instructions on how and where to execute a specific SPARQL query, essentially fixing the histogram to use only a single data source (see One-Star applications in Section 5). A web browser, in turn, shimmed the SVG graphics into a PNG image, d_{α_2}, which shows the distribution of satellites by type (i.e., useful or junk).

The segment of provenance from $d_{1,2}$ to $d_{2,2}$ is a stop-gap approach to obtain RDF, which is tolerable since standards for converting to linked data are relatively new[11]. Ideally, Amy would have obtained RDF using low cost techniques, such as content negotiation [15], GRDDL, and RDFa processors **[pp4]**. The more critical issue, however, is that Amy's derived semantic satellite groupings, $d_{2,3}$, fall back down to a mundane encoding of a histogram. Her effort to produce high-quality RDF [14], re-group satellites, and finally compute tallies on those groupings resulted in a set of SVG rectangles that are disconnected from any of the prior datasets.

We refer to these lift-then-cast sequences as the "house top" anti-pattern. With Amy's house top, information about the custom satellite groups and their corresponding member count (i.e., sio:count) became implicit in the SVG encoding; is the size of the bar graphic the membership size, some factor of the size, or is the graphic indicative of membership size at all? If the histogram labels are not informative or the provenance of histogram lost, it may be difficult for subsequent analysts to understand what the graphics represent **[pp1]**.

The resultant histogram, shown in the center of Figure 5, provides Amy with an easy, side-by-side comparison of relative bar lengths, which depict the number of useful and junk satellites. Amy can clearly see an order of magnitude difference between active satellites and junk, which leads her to believe that countries are inefficient when launching space materials. She does not know, however, which countries are most responsible for the resulting environmental condition. She performs the next application to explore launch efficiency on a per-country basis.

3.1.3. *Application 3 (α_3): What is the Efficiency of Satellite Launches per Country?*

Amy can begin her final inquiry using materials generated by her two previous applications:

$d_{1,1}$	URL to a KML satellite dataset
d_{α_1}	map screenshot showing "useful" and "junk" satellites
$d_{2,1}$	CSV representation of the KML satellite dataset
$d_{2,1}$	RDF representation of the KML satellite dataset
$d_{2,3}$	RDF representation of satellites grouped as useful or junk
$d_{2,4}$	SVG histogram showing satellite distribution by type
d_{α_2}	PNG image of a histogram depicting satellite distribution by type

Once again, Amy must choose between an analytical frame (i.e., the PNG or SVG of the histogram) encoded in some mundane format **[pp1,pp2]** or an earlier, intermediate result that is easier to reuse but less in-line with her current mental model **[pp3]**. She ultimately decides to reuse the RDF data containing her custom satellite groupings, $d_{2,3}$, to generate a normalized stacked bar chart showing launch efficiency on a per-country basis [12]

As presented by the provenance trace in Figure 4, Amy used a custom script to cast $d_{2,3}$ into a JSON file, $d_{3,1}$. A snippet of the JSON data is shown below:

[11] R2RML is a more recent standard for mapping relational data to RDF.

[12] In a distributed analytical environment without LD or provenance, a second analyst would unlikely be able to determine what intermediate result would be best to use.

```
{
  "owner" :"United States",
  "Active":259,
  "Trash" :3696
}
{
  "owner  :"France",
  "Active":114,
  "Trash" :5677
}
```

Widgets, such as D3 stacked bars[13], often impose custom input data requirements which are not explicitly or formally described. The lack of documentation forces analysts to inspect sample inputs and source code in order to infer the complete set of ingestion requirements. In Amy's scenario, the stacked bars tool only provided one example input CSV dataset such as the one show below:

```
State, 5 Years, 5 to 13 Years, Over
AL,    310504,  552339,        259034
AK,    52083,   85640,         42153
AZ,    515910,  828669,        362642
```

After tediously inspecting both the example dataset and the widget's JavaScript code, Amy realized that each row in the table specifies a single stacked bar. The first column specifies the label of the bar and the following columns specify the sizes of the sub-bars. By running some tests, she also realized that the input table can specify an arbitrary number of sub-bars, with the caveat that all stacked bars (i.e., rows) must have the same number of sub-bars (i.e., columns) **[pp5]**. Additionally, the widget can be easily modified to accept JSON versions of the CSV file with only minor tweaks to the data reader. With this knowledge, Amy was able to produce a JSON dataset, $d_{3,1}$ that is compliant with the stacked bars widget.

The stacked bars widget, in turn, cast the JSON data into a set of stacked bars encoded in SVG. Since the widget is web-based, Amy's third application also exhibits the "SVG to PNG" transformation pattern between $d_{3,2}$ and d_{α_3} and highlights another anti-pattern known as the "hill slide." Hill slides are a sub pattern of "house top"and thus result with a similar work-efficiency degradation for subsequent analysts.

From the stacked bar chart, shown at the bottom of Figure 5, Amy can see that most countries launch space junk to some degree. The bars are normalized and thus convey the relative efficiency of satellite launches. Amy notices that the Common Wealth of the Independent States (CIS), United States, China, and France all launch a large percentage of junk compared to other countries.

3.2. Bart's Analysis

Amy shows the normalized stacked bars to her classmate Bart and exclaims her concerns about the proliferation of space junk. She asks Bart to determine if the United States, her home country, allows any of other junk-producing countries to launch from its facilities and hands him all of her analytical materials including: source datasets, intermediate datasets, and application results. She points him to the normalized stacked bars where she left off, but also points out sources of information that were easiest for her to use, namely the KML file and her RDF that groups satellites as useful or junk.

3.2.1. Application 1 (α_1): What other Countries Launch Space Junk with the Help of the United States?

To complete his task, Bart needs to find information about:

- what kinds of satellites Amy considers junk
- which countries launch this junk
- what sites do these countries launch the junk from
- where are these sites geographically located

Reviewing a flat collection of Amy's materials without any context is a daunting task, even with pointers to the files she believed were easiest to work with. The relationships among source materials, intermediate datasets, and application results are not captured and preserved. Bart, therefore, is unable to easily determine what information each dataset captures, how the information overlaps[14], and what tools were used **[pp6]**. This challenge reflects the current state of analytics, where mundane data simply cannot stand on its own as an adequate interface between prior and subsequent analysts.

To save time and effort, Bart contacts Amy and asks for help addressing his aforementioned concerns, which can be impractical in some settings. From their interaction, both analysts determine that dataset $d_{2,3}$, which groups satellites as "useful" or "junk," serves as a suitable starting point for his analysis; the dataset contains satellite attribution information as well as Amy's perspective regarding the classification of space

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

[14] http://www.w3.org/TR/void/

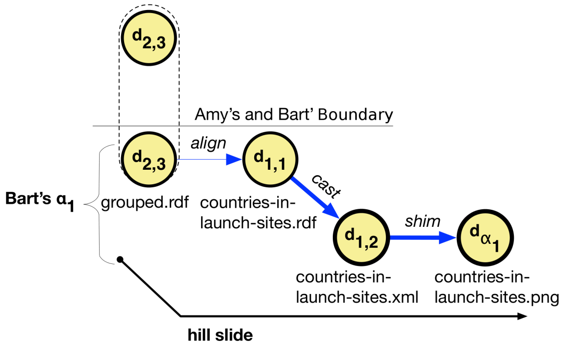


Fig. 6. Bart's analysis described using munge glyphs.

materials, although the dataset is missing the geographic locations of launch sites.

To support his task, Bart uses a categorical visualization tool, such as Aduna ClusterMap[15], to generate a cluster map that groups countries by the launch sites they use. Bart is particularly interested in identifying countries that are cross-categorized (i.e., countries that use multiple launch sites), which will be rendered as nodes within "intersection clusters", much like Venn Diagrams that illustrate intersection.

Bart's application is described by the provenance trace labeled α_1 in Figure 6. Bart first issued a SPARQL construct query to generate a new dataset, $d_{1,1}$, that categorizes countries by launch sites:

```
construct{
  ?country      vcard:hasCategory  ?launchSite    ;
                rdfs:label         ?countryName   .
  ?launchSite   rdfs:label         ?siteName
}
where {
  ?country    ^acl:owner           ?satellite  ;
               a                   foaf:Organization
                                   ;
               rdfs:label          ?countryName  .

  ?satellite   a                   nfo:Trash  ;
               prov:wasDerivedFrom ?launchSite  ;
               rdfs:label          ?siteName  .
}
```

The property `acl:owner` specifies the country that owns a satellite. The property `prov:wasDerivedFrom` specifies the site where a satellite was launched. Since dataset $d_{2,3}$ also included Amy's perspective on space junk, Bart was able to restrict his solution set to sites that launch `nfo:Trash` satellites. Unfortunately, Amy's data did not include geographic coordinates of launch sites, preventing Bart from restricting his solution set to only sites located in the United States.

Bart then used a custom script to cast the resulting dataset $d_{1,1}$ into an XML file, $d_{1,2}$, that conforms to the cluster map tool's input data requirements. Finally, he used the cluster map tool to generate the visualization, d_{α_1}, which resides as PNG image of a cluster map snapshot.

The resulting cluster visualization in the top of Figure 7 shows the global set of junk-launching sites and countries that use them. In the cluster map, launch sites are depicted as the shaded "octopus-like" figures and countries are depicted as nodes within them. Bart relies on his geographic expertise to identify launch sites that are located in the United States, namely the "Mid-Atlantic Regional Spaceport" and "Eastern Range." From these two clusters, expanded at the bottom in Figure 7, Bart can see that both France and CIS launch space junk from these facilities, as well as from Baikonur Cosmodrome located in Kazakhstan. He tries to save only the United States clusters, but the tool does not allow him to export selections made in the canvas.

As it stands, the cluster map is not immediately useful to Amy; the map is not focused on the United States and instead displays all launch sites from across the globe. To answer her question, Amy would first need to identify which launch sites are located in the United States, effectively re-establishing information already known to Bart. To reduce her workload, Bart can send Amy:

1. a zipped file that contains both the full visualization and a text file that lists the sites of interest
2. a manually cropped image, shown at the bottom of Figure 7, that contains only those clusters located in the United States

With option 1, Amy must reference a separate text file while she browses, interprets, and gleans information from the cluster map, essentially establishing cognitive links between the text file and the figures in the cluster map. Although this approach is high cost, Amy is provided a global information source about launch sites, which may be of interest to her in subsequent analyses. With option 2, Amy is provided with only the pertinent clusters relevant to her inquiry, but she loses information about the broader, global perspective on launch site usage.

Ideally, the information depicted in the cropped image would be physically and semantically linked with the larger, underlying information source from which

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

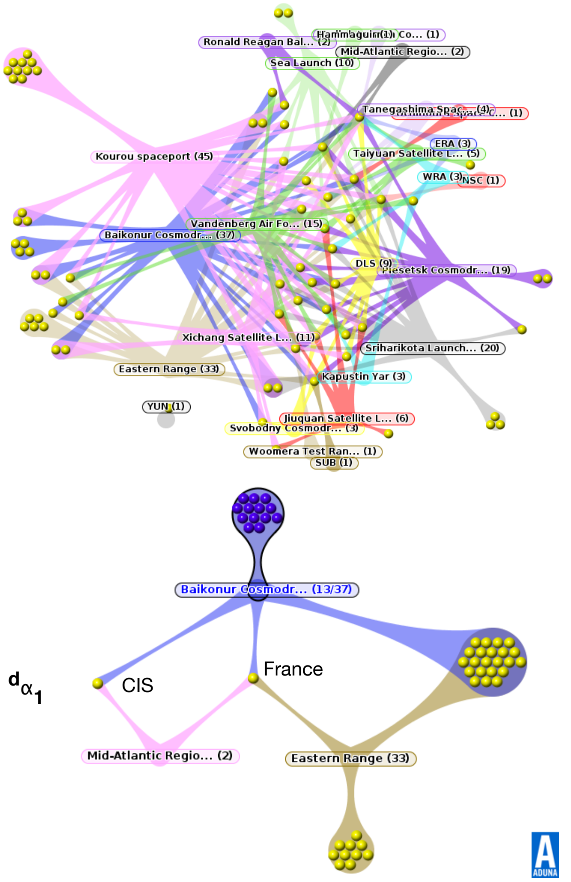


Fig. 7. Bart's application results. The top cluster map shows all launch sites. The bottom cluster map shows only sites associated with countries that launch from the United States.

the image was derived. Going even further, if Amy had referenced DBPedia launch sites in dataset $d_{2,3}$, for example Easter Range[16], Bart could have trivially acquired the missing location information and formulated a SPARQL query that matches only launch sites with a `dbp:country` of United States. He then could have easily generated a cluster map that shows only United States launch sites. However, to further capitalize on Amy's investment, the cluster map tool would need to allow users to obtain handles on the URIs of launch sites, so that additional information about these resources can be acquired in subsequent analyses.

3.3. Recap

Figure 8 provides an overview of Amy's and Bart's analysis that is juxtaposed with an ideal analysis, where every application outputs two results: a mundane dataset and an equivalent, semantic version. The dashed lines in the figure indicate that a dataset was reused in a subsequent application.

In the actual analysis shown at the top, the *final* result (i.e, d_α) of every application was mundane. Some applications generated intermediate semantic datasets but Amy did not directly draw insight from those intermediaries. Therefore, in every subsequent application, the analyst had to compromise between reusing materials that are more structured versus materials that more closely reflect the prior mental schema of the analyst. In practice, analysts usually choose the less evolved materials and reproduce prior work [19]. We see this pattern in Figure 8, where no dashed lines extend from the arrow tips corresponding to results of the application.

In the hypothetical analysis, every application uses semantic datasets and generates both mundane *and* equivalent semantic results. Humans rely on their broadband visual channel to receive information and, therefore, will always need mundane representations of information such as rendered graphics. However, when materials are passed to subsequent analysts, it may be more convenient for them to work with linked, machine readable representations. We can accommodate both settings if more tools would generate RDFa and GRDDL or publish results to content negotiable servers, for example.

4. A Metric for Application Seamlessness

In the previous section, Amy and Bart each composed unique application chains. Amy generated geospatial plots and histograms, while Bart generated a visualization that depicts categorical relationships between entities. Each unique sequence of applications induces a unique *analytical ecosystem*, E. Since Amy and Bart each performed a unique set of applications, they each induced a unique ecosystem, i.e., E_{Amy} and E_{Bart}.

Formally, an ecosystem E is defined as the set of applications that influenced[17] a particular analysis:

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

[16] http://dbpedia.org/page/Eastern_Range

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

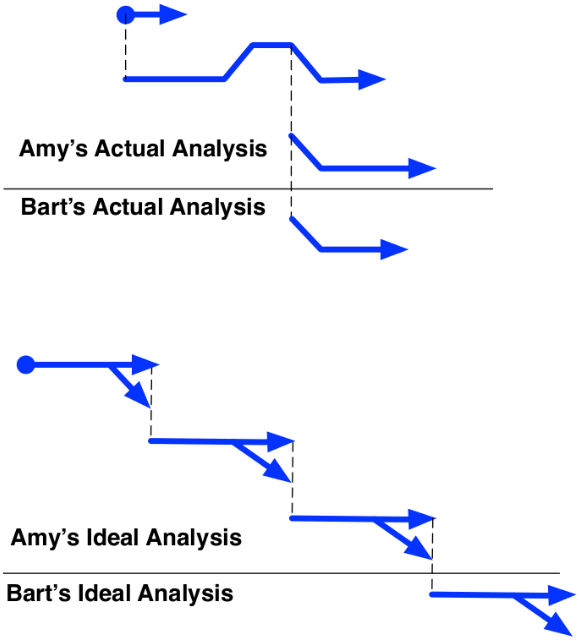


Fig. 8. Juxtaposition of an actual analysis vs. an ideal hypothetical analysis.

Each application α[18] , in turn, is formally defined as a tuple consisting of a non-empty set of munges and a resultant dataset d_α:

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

The remainder of this section defines a *seamlessness* metric, S, that can be used to assess the cost of ecosystems from two perspectives:

1. how easily can analysts generate materials
2. how easily can those materials be used by subsequent analysts

To capture the two perspectives, we first define a "result generation metric" that measures the cost for an analyst to generate results. We then define a "reuse potential" metric that predicts the ease by which future, subsequent analysts can reuse those results. We finally combine the generation and reuse potential metrics to formulate the analytical seamlessness metric S.

4.1. Generation Cost

We define a score, μ, that expresses how easily analysts were able to generate materials during a single application. Since we assume that munging dominates the cost of applications, the score is only a function of the kinds of munges performed during an application α.

$$\mu(\alpha) = \frac{\sum\limits_{m \in M_\alpha} cost(m)}{\sum\limits_{m \in M_\alpha} cost(shim)} \tag{1}$$

The numerator contains the actual cost of the application, which is calculated by summing the cost of each munge. The denominator reflects the hypothetical worst-case, where an application consists entirely of shims. Therefore, the equation has a range of $(0, 1]$, where lower values indicate a better score. Note that the lower bound is exclusive since we do not permit munges to have a zero cost and every application must have at least a single munge.

The generation score depends on a cost function that maps munge types to cost values. To bound our munge-level cost function, we first present a complete ordering of munge costs that aligns with the partial ternary ordering introduced in Section 3.

$$\begin{aligned} cost(\alpha) &> cost(shim) \\ cost(shim) &> cost(lift) + 2\ cost(align) + cost(cast) \\ cost(lift) &> cost(cast) \end{aligned}$$

———

$$\begin{aligned} cost(cast) &> cost(align) \\ cost(align) &> cost(comp) \end{aligned}$$

———

$$\begin{aligned} cost(comp) &> cost(glean) \\ cost(glean) &> cost(conneg) \\ cost(conneg) &> 0 \end{aligned}$$

The horizontal lines delimit the three munge groups shown in in Figure 3; the top group corresponds with mundane munges, the middle group corresponds with semantic munges, and the bottom group corresponds with trivial munges. The least expensive munge is a $conneg$ and the most expensive munge is a $shim$,

[18] The set-theoretic definition of an application is an alternate expression of the OWL ontology, described in Section 2, and is better suited for defining cost metrics.

which is a composite of lifting, aligning, and casting; shims incur the highest cost because they require analysts to perform mental data alignments without concrete intermediary models. Note, however, that the cost of a shim cannot equal the cost of the total application. This implied gap is filled by other costs, such as visualization interpretation costs, which are discussed in Section 7.

We use one such solution of the cost ordering constraints to define a munge-level cost function shown below:

$$cost(m) = \begin{cases} 20 : & \text{if } shim \\ 6 : & \text{if } lift \\ 5 : & \text{if } cast \\ 4 : & \text{if } align \\ 3 : & \text{if } comp \\ 2 : & \text{if } glean \\ 1 : & \text{if } conneg \end{cases}$$

Given these munge cost bindings, we see that μ favors applications that contain a larger proportion of trivial and semantic functions. For example, compare Amy's μ for her α_3 and Bart's μ for his α_1, both of which consist of three munges shown in Table 1. Amy's α_3 consists of one cast and two shims, which results in a μ of 0.75. Bart's α_1, on the other hand, consists of only a single shim, which results in a μ of 0.48.

Table 1

Amy's and Bart's μ for each application. The scores are broken down by actual and worst case cost.

An.	α	M_α	actual	worst	μ
Amy	α_1	shim	20	20	1
	α_2	shim, lift, align, cast, shim	55	100	0.55
	α_3	cast, shim, shim	45	60	0.75
Bart	α_1	align, cast, shim	29	60	0.48

In practice, analysts should assign munge costs that are based on different measures, e.g., man hours, lines of code, and commit frequencies. As long as the cost ordering constraints are satisfied, analysts can experiment with different cost valuations and obtain new μ scores that are consistent with previously computed rankings of their ecosystems. For example, given two ecosystems E_1 and E_2, where $S_1(E_1) < S_1(E_2)$ was established using munge cost function c, the ranking will hold under a different cost function c', so long as both c and c' respect the *same* cost order constraints. We can therefore consider c and c' as simple scaling factors.

4.2. Reuse Potential

We define a score that expresses how easily subsequent analysts can reuse materials generated by prior analyses. Since this score is looking at the seams (i.e., data) between different ecosystems, the score is a function of the kind of results that are generated by applications. We assume that LD, including data that can be trivially munged to yield LD, is easier for subsequent analysts to reuse. On the other hand, mundane results such as PowerPoint slides, CSV files, and raster images pose greater challenges [19] since these results are rarely explicitly connected to their source materials.

In the analysis described in Section 3, Bart made a strategic decision to reuse the intermediate and structured, albeit less evolved, satellite RDF dataset instead of the normalized histogram image. The histogram, although representative of Amy's analytical frame, is an island from a LD standpoint and is not linked to the source RDF information that Bart needed to complete his task.

To embody this idea, we define the potential (*pot*) function that returns a set of scaling factors whose values depend on whether D_α is mundane or semantic. Let the function *tbl* return the Berners-Lee star rating ([1,5]) of a dataset, i.e., $tbl(d) = s$.

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

The pot function is used to reward (i.e., reduce the value μ) applications that generate RDF. Therefore, if a resultant dataset is encoded in RDF, the pot function provides the greatest reward since subsequent analysts obtain RDF for free. If a result can be content negotiated to obtain RDF, the function provides less of an award since subsequent analysts would have to interact with a server to acquire RDF. If a dataset can be gleaned to obtain RDF, the function provides the smallest reward since subsequent analysts would have to perform a glean munge, which costs slightly more than content negotiation. Finally, if a dataset does not satisfy any of the above conditions, the function provides no reward; a scaling factor of 1 has no effect.

Since d_α can satisfy multiple conditions, the pot function returns a *set* of values, one for each bucket d_α satisfies. In these cases, subsequent analysts can choose which facet of the dataset they want to work with. For example, an analyst may be able to glean or content negotiate a single XML dataset, if the dataset contained embedded RDF and was referenced by a URL that could be content negotiated.

On the Web, capturing downstream usage of analytical results may be challenging for provenance systems. The pot function, therefore, provides some predictive measure about the efficiently at which downstream analysts can repurpose upstream materials. With continued work in the area of provenance [26], we may soon be able to follow up on the actual reported gains in downstream applications and report these gains to upstream analysts.

Also, the pot function is not expected to accurately reflect all work environments. Perhaps, for provenance concerns (i.e., `prov:alternateOf`), a downstream analyst prefers to keep the mundane and semantic results bundled together as gleanable datasets. In this case, the pot can be reconfigured to provide gleanable datasets with the greatest reward. Or perhaps file size is important and therefore gleanable datasets should return very little reward since they pack multiple representations into a single file.

4.3. Seamlessness Score S

We can now define the seamlessness score, S, that is built from the μ expression and pot:

$$S(E) = \frac{\sum\limits_{\alpha \in E} \sum\limits_{m \in M_\alpha} min(pot(d_\alpha))cost(m)}{\sum\limits_{\alpha \in E} \sum\limits_{m \in M_\alpha} cost(shim)} \tag{2}$$

Unlike μ, the seamlessness metric S computes scores for ecosystems, rather than single applications. The seamlessness score S sums up all *scaled* μ scores and normalizes these values by the hypothetical worst case: when an ecosystem is informed entirely by shims. As described in the previous subsection, the scale factors are computed by the pot function, which predicts the ease by which subsequent analysts can reuse d_α. Also, S uses the minimum value returned by pot in order to provide the greatest rewards.

4.4. Amy's and Bart's Scores

Table 2 presents the seamlessness scores for Amy's and Bart's ecosystem. The table breaks down the scores in terms of μ and pot and also computes the scaled value of μ, i.e., $cost \times pot$.

Table 2

Amy's and Bart's seamlessness score S. The scores are broken down into their constituent integration and reuse costs.

An.	App.	μ	$pot(d_\alpha)$	$cost \times pot$
Amy	α_1	20	1	20
	α_2	55	1	55
	α_3	45	1	45
			$S(E_{Amy}) =$	0.66
Bart	α_1	29	1	29
			$S(E_{Bart}) =$	0.48

From the table, we see that Bart's ecosystem, which scored 0.48, was more seamless than Amy's ecosystem which scored 0.66. Overall, Amy performed more shims that resulted with mundane datasets that degraded her work performance. Note, however, that neither analyst generated an RDF representation for any of their resultant visualizations, d_α, and thus no reductions were applied for any application, i.e., $pot(d_\alpha) = 1$. Also, note that Bart's ecosystem contained only a single application and thus his seamlessness score is equal to his only application's μ score presented in Table 1.

5. Reducing Analytical Costs with Five-Star Applications

We propose a "5-star application rating scheme" that analysts can use to design more efficient applications that avoid the anti-patterns and analytical pain points described in Section 3. The rating scheme is expressed in the form of ontological restrictions that progressively reduce the space of possible munge sequences. As the application ratings increase, the possibility of performing certain anti-patterns decrease.

We outline these ontology restrictions by extending the application ontology, presented in Section 2, to distinguish among five types of application subclasses that are illustrated in Figure 9. These sub-

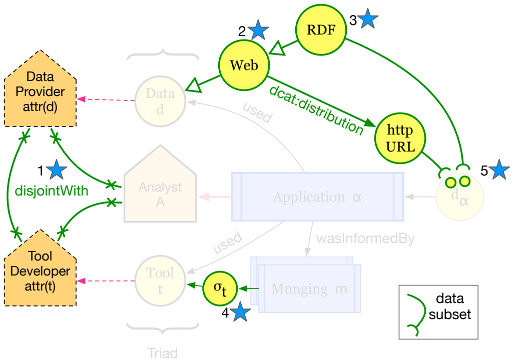
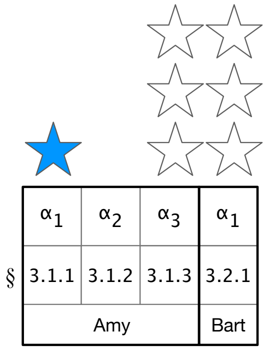


Fig. 9. An extension of the Application Ontology Core (Figure 2) to distinguish five subclasses of Application. The figure distinguishes between generic application concepts, shown in gray, and the extension concepts shown in bold-face.



	α1	α2	α3	α1
§	3.1.1	3.1.2	3.1.3	3.2.1
	Amy			Bart

Fig. 10. Star ratings for the applications in Bart's and Amy's ecosystems. White star indicate "conditional stars."

classes are rated according their predicted cost, which is expressed as an interval. We use intervals since we are describing *classes* of applications, each of which contains a set of different applications instances that have different predicted costs. The interval thus captures the min and max predicted cost of the application instances.

Table 3 enumerates all five application star ratings and pairs each with their associated restriction(s). Figure 10 uses this table to rate the applications performed by Amy and Bart in their analyses presented in Section 3.

5.1. *One-star applications*

One-star applications satisfy our fundamental restriction that the sets of analysts, tool developers, and data providers (i.e., the application triad) of a particular application are disjoint, i.e., $attr(d_\alpha) \cap A \cap attr(t) = \emptyset$, where A refers to the analyst and the function $attr$ supports the transitive discovery of data providers and tools developers that can be practically achieved through SPARQL queries over provenance traces described using our AO ontology (Section 2). Figure 9 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 the GIS tool in application α_1 earns one-star since Amy was neither the provider of the satellite data nor the developer of the employed GIS tool; when we state that an application instance, such as Amy's α_1, earns one star, this is equivalent to stating that the instance belongs to the one star application class. In contrast, Amy's use of the histogram tool in application α_2 does not earn one star since she was required to write HTML code. We therefore distinguish between developing munge scripts and developing the actual tool t that generates application results, where only the latter case precludes an application from earning one star.

Similarly to Amy's α_2, the use of government data mash-ups[19] and LOD metadata summaries such as Linked Open Vocabularies (LOV)[20] and SPARQL Endpoint Service (SPARQL-ES)[21] 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 [22].

The one-star application restriction speaks more from a tool developer perspective than from analysts who use those tools. If developers would design software with one-star applications in mind, they might refrain from hard-coding tools to accept only certain data sources (e.g., a particular quad store), and thus provide analysts with greater flexibility regarding which tools they can use.

The one-star application class also describes a set of possible munges sequence that we refer to as a *munge space*. Since the one-star application class does not place any restrictions on the structure of the data used and generated, the class describes applications that range from exclusive shims (i.e., flatlines described in Section 3) to exclusive computes, and every possible combination in between (i.e., housetops and hillsides).

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

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

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

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

Table 3

Five-star rating scheme to assess the seamlessness of a single application.

Rating	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	$used(m, \sigma_t) \wedge m \in M$
5	provide any information (RDF format) derived during use	$D \subset D_\alpha$

We depict the one-star munge space at the top in Figure 11. Without loss of generality, the munge spaces presented in the figure:

- assume that applications are composed of two non-trivial munges (see Section 2); one non-trivial munge to satisfy a tool's input requirements and another non-trivial munge performed by the tool itself.
- assume that applications that accept semantic data (i.e., $D_{[4,5]}$) may be preceded with an optional trivial munge, which is not considered in the total cost of the application. This relaxation will allow three-, four-, and five-star applications to ingest gleanable and content negotiable data without being penalized[23].

Because each application in this section describes a set of possible munge sequences, we describe their costs in terms of an interval. The lower bound specifies the cost the cheapest possible munge sequence while the upper bound specifies the cost of the most expensive possible sequence in the munge space. Therefore, the cost bounds for the one-star application class is expressed by the interval:

$$\begin{aligned} cost(\alpha_\star) &= [2 \times cost(comp), 2 \times cost(shim)] \\ &= [6, 40] \end{aligned}$$

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. Like the previous rating, two-star applications also fulfill the requirement that data providers, analysts, and tool developers are disjoint.

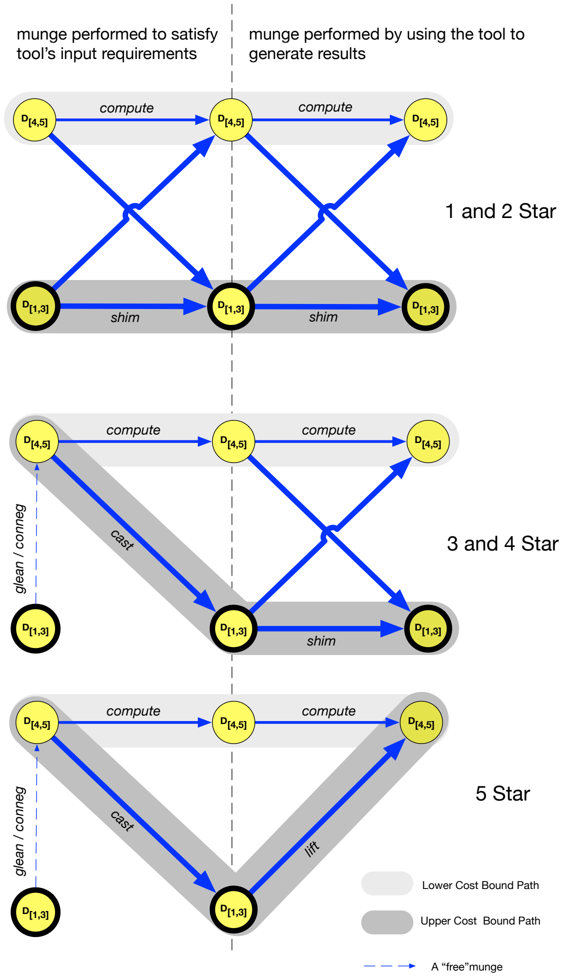


Fig. 11. Possible munge patterns associated with each application subclass. As the application restrictions increase, the space of possible munge sequences decreases.

Figure 9 depicts the two-star restriction near the top, where:

- data d is available on the Web

[23] Our μ score, presented in Section 4, assigns a negligible cost to trivial munges

- data d has an associated dcat:Distribution pointing to where d can be accessed
- the distribution URL is referenced by the output dataset, d_α; we depict this relationship using a line with a semi-circle endpoint that "hugs" a subcircle within d_α

Amy's application α_1, described in Section 3, earns two stars. The input to the application, $d_{1,1}$, was a KML satellite dataset that was available on the Web. Additionally, the resultant map, d_{α_1}, contained the URL of the input KML file, providing a simple and natural derivation provenance **[pp6]**.

We take Amy's α_1 as an opportunity to reinforce the definition of an application, which we define as a class of activities, not software entities. To earn a second star, an application must use a web accessible dataset and generate a result that cites that same dataset, regardless of the actual IO of the employed tool. For example, if Amy wanted to perform a two-star application, and her employed GIS tool did not generate a map citing the KML dataset, the burden would fall on Amy to somehow watermark the URL of the KML into the map, perhaps using a technique such as steganography [20].

In contrast, the use of LOV and SPARQL-ES would likely result with two *conditional* stars; both tools accept URLs (e.g., OWL files and SPARQL endpoints) and the HTML reports these tools produce reference those same input URLs. Conditionality refers to cases when an application fulfills a particular star level requirement but fails to fulfill the immediately-preceding requirement(s). Although LOV and SPARQL-ES accept URLs and thus implicitly encourage analysts to use URLs in their applications, these two tools violate the one-star condition since the tool maintainers have control over input data sources.

In terms of munge space, the two-star application class is equivalent to the one-star class since two-star applications do not restrict the structure of data consumed or generated.

5.3. Three-star applications

Three-star applications accept RDF data via URL, i.e., $tbl(d_\alpha) >= 4$. 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. Figure 9 presents the three-star restriction at the top, where d is an RDF dataset. The figure indicates that RDF is a subclass of Web, and thus inherits a `dcat:distribution` URL.

Both Amy's α_3 and Bart's α_1 earn three conditional stars. Both applications used an RDF dataset as input, yet the applications generated results that did not reference the URLs to those input RDF datasets; the applications did not fulfill the two-star requirement. These conditional star ratings are depicted as white stars in Figure 10.

Applications designed around linked data browsers [1,3,12] can earn at least three-stars *iff* the applications meet the one- and two- star requirements. These tools accept RDF and thus encourage analysts to use RDF in their applications.

The three-star application class defines a smaller munge space than one- and two-star application classes. If data d is encoded in RDF, it can only be computed, aligned, and cast. The three-star restriction thus removes the possibility for flat-line and house top munges, although hill slides are still possible. We depict the three-star munge space in Figure 11.

The cost bounds for the three-star application class is expressed by the interval:

$$\begin{aligned} cost(\alpha_{\star\star\star}) &= [2 \times cost(comp), cost(cast) + cost(shim)] \\ &= [6, 25] \end{aligned}$$

The cost bound for the three-star application class is not only *tighter* than one- and two-star application classes, but also *lower* since the upper cost is reduced from 40 to 25.

5.4. Four-star applications

Four-star applications use a tool's input semantics (OWL, SPARQL) to help guide munging, i.e., $used(m, \sigma_t) \wedge m \in M$. Like the previous rating, four-star applications also accept RDF via URL. Figure 9 depicts the four-star application restriction toward the bottom, where a munge m uses a tool t's input semantics σ_t during an application.

Amy and Bart did not use any tools that made their input semantics available and, therefore, did not perform any four-star applications. However, the Semantic Automated Discovery and Integration (SADI) framework [40] pairs services with OWL class definitions that describe the expected input and output graph patterns. These OWL classes provide service consumers with an unambiguous expression (**[pp5]**) of the service's I/O requirements, which allows agents to coordinate service execution sequences.

In terms of munge space, the four-star application class is equivalent to the three-star class; no restrictions are placed on the data consumed or generated.

5.5. Five-star applications

Five-star applications output results 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 9 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.

Amy and Bart did not perform any applications that generated Linked Data, and, therefore, neither of their ecosystems contains a five-star application. Similarly, some applications analyzing the Linked Data cloud [14,32] do not earn five-stars since the results are typically images or journal articles. On the other hand, Tim Berners-Lee's tabulator [3] can be used by analysts to perform five-star applications. As analysts make edits to third-party RDF, tabulator emits new RDF describing those edits. Analysts can also use SADI to perform five-star applications, since SADI services generate RDF graphs that expand on the inputs graphs.

When applications generate LD, they eliminate a number of analytical pain points. With LD, subsequent analysts can more easily determine how prior results are connected to source information and thus be better informed about meaning of those results **[pp1,pp2]**. Additionally, subsequent analysts can use results as a gateway from which to obtain more context that may be needed for their specific tasks. Finally, applications that generate gleanable LD or datasets that can be content negotiated to obtain LD allow subsequent analysts to easily work with their preferred data representation **[pp4]**.

The five-star application class defines the smallest munge space. Five-star applications use RDF and generate Linked Data, or results that can be trivial gleaned to yield Linked Data. Therefore the munge space includes a best case of exclusive computes and a worst case sequence exhibiting the"inverted house top" pattern (i.e., cast-lift combination), as shown in Figure 11. Essentially, five-star applications eliminate the possibility of anti-patterns described in Section 3.

The cost bounds for the five-star application class is expressed by the interval:

$$\begin{aligned} cost(\alpha_{\star\star\star\star\star}) &= [2 \times cost(comp), cost(cast) + cost(lift)] \\ &= [6, 11] \end{aligned}$$

The cost bound for five-star applications is not only *tighter* than the three- and four-star application classes, but also *lower* since the upper cost is reduced from 25 to 11.

5.6. Boosting Amy's Seamlessness Scores

In this section we will use Amy's ecosystem E_{Amy}, shown in Figure 4, as a baseline ecosystem to compare with an alternate, ideal ecosystem E_{ideal}, shown in Figure 12. The ideal ecosystem, E_{ideal}, contains only five-star applications and exemplifies a seamless analysis. Aside from calculating a better seamlessness score, we will also provide intuition as to why the ideal ecosystem alleviates certain analytical pain points.

Ecosystem E_{Amy} contained three applications that collectively spanned nine munges. To facilitate a more fair comparison, we retrofitted the applications in E_{Amy} with additional lifts and gleans to produce E_{ideal}, which is five-star compliant. Therefore, our ideal ecosystem, E_{ideal}, also contains three applications that are designed around the same tools and objectives as E_{Amy}.

Figure 12 shows the provenance for the applications comprising E_{ideal}. We assume that a LD version of the satellite dataset, d_{α_1}, existed prior to Amy's ideal analysis. Therefore, from a more global perspective, Amy is a subsequent analyst that reused the results of a prior, anonymous five-star application (we need this bootstrap in order to make this first application five-star compliant). A snippet of the satellite LD is shown below. Note the use of DBPedia URIs to refer to launch sites and countries.

```
<satellite-1> a pext:ActiveSatellite ;
prov:wasDerivedFrom
      <http://dbpedia.org/page/Eastern_Range> ;
    acl:owner
          <http://dbpedia.org/page/United_States> ;
    geo:lat     -100.000 ;
    geo:long    33 ;
    geo:alt     30000
    .
<satellite-2> a :Debris;
    prov:wasDerivedFrom
          <http://dbpedia.org/page/Mid-Atlantic...>
    acl:owner
          <http://dbpedia.org/page/United_States>
    geo:lat     -110.000 ;
    geo:long    50 ;
    geo:alt     60000
    .
```

Amy first used a script to cast the input satellite RDF dataset, $d_{1,1}$ to a satellite KML file, $d_{1,2}$. She then generated two, alternate representations of the satel-

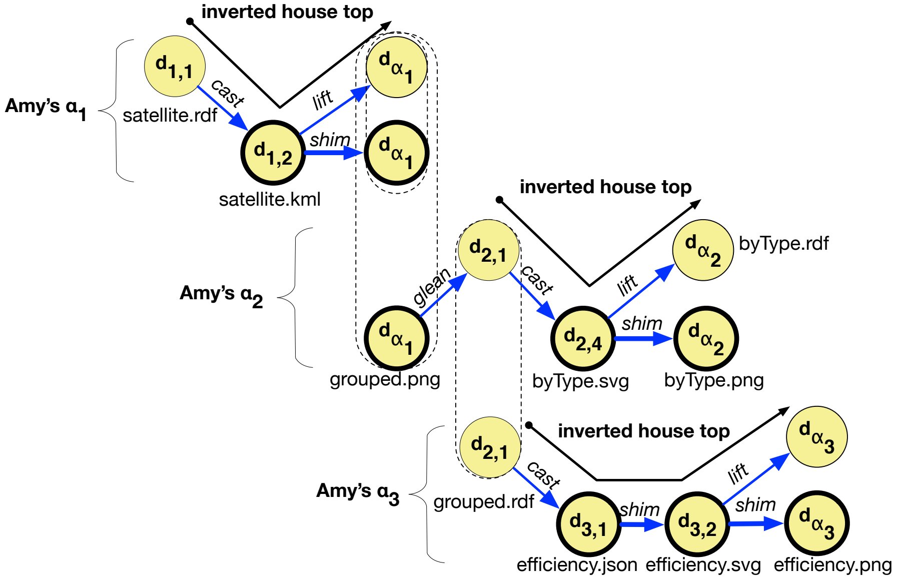


Fig. 12. Amy's ideal analysis supported entirely by five-star applications.

lite map, d_{α_1}, one semantic and one mundane. In practice, she could have used the GIS tool to generate the mundane version of d_{α_1} and then developed a separate script that "types" the satellites in $d_{1,1}$ as useful or junk. For the sake of this exercise, we'll assume that the GIS tool generated a gleanable image of the map that embeds LD describing her satellite groupings. We consider gleanable XML and any other RDF embedding mechanism, such as content preserved images [24], to be equivalent; these kinds of approaches all embed semantic content into mundane datasets.

Since this application is five-star, the GIS tool provided its input semantics in the form of the SPARQL query shown below:

```
select ?geonode ?lat ?long ?alt
where {
  ?geonode geo:lat   ?lat ;
           geo:long ?long ;
           geo:alt   ?alt .}
```

Although the SPARQL query does not include information about the particular KML format required by the GIS tool, the conceptual description, coupled with example KML dataset provided by tool, was enough information for Amy to produce the appropriate KML file, $d_{1,2}$.

Amy then used the gleanable map, d_{α_1}, as input for her next application, α_2. In E_{Amy}, Amy was not able to directly reuse the satellite map since it resided as a mundane PNG image that required expensive image processing to reuse. However, the hypothetical, gleanable version of the map contains an embedded LD dataset containing her custom satellite groupings. She performs a glean to extract the satellite groups, $d_{2,1}$, directly from the map and then uses the histogram tool to cast the dataset into an SVG histogram. Amy then lifts the SVG to generate the application's result, d_{α_2}, which is a LD representation of the SVG histogram [24]. Although the histogram provided its input semantics, we omit them in this text.

A snippet of the LD histogram is shown below[24]

```
:bin1  a  vsr:Bin;
       rdfs:label       "Useful␣Satellites" ;
       sio:count        "1000" ;
       owl:onProperty  <rdf:type> ;
       owl:hasValue    <pext:ActiveSatellite> .

:bin2  a  vsr:Bin ;
       rdfs:label       "Junk␣Satellites" ;
       sio:count        "14000" ;
       owl:onProperty  <rdf:type> ;
       owl:hasValue    <nfo:Trash> ;
```

In her final application, α_3, Amy uses the satellite groups, d_{2_1} to generate the normalized histogram. The dataset contains her custom groupings and references URIs of satellites published as LD, which Amy can dereference to obtain additional information, such as ownership and launch site information. She first cast $d_{2,1}$ into the JSON format required by the stacked bars

[24] The values of `owl:hasProperty` are references to the satellite types contained in dataset $d_{1,1}$ and thus enable subsequent analysts reconstitute the set of satellites that contribute to a bin count.

widget. During this cast, Amy was guided by the input semantics of the stacked bars widget, which were expressed using the SPARQL query below:

```
select ?bin ?binCount ?subBin ?subBinCount
where {
  ?bin a vsr:Bin;
       sio:count         ?binCount;
       dcterms:hasPart   ?subBin.

  ?subBin a vsr:Bin;
          sio:count      ?subBinCount.}
```

Once again, the input semantics coupled with an example dataset, provided by stacked bars, was enough information for Amy to produce the appropriate JSON file, $d_{3,1}$. Like applications α_1 and α_2, she generates both a mundane and semantic representation of the stacked bars.

Using the same mechanics in Section 4, we calculate the seamlessness score for E_{ideal} in Table 4. We also include the seamlessness score for the older ecosystem E_{ideal} for comparison purposes.

Table 4
The seamlessness scores for ecosystem E_{Amy} and E_{ideal}.

Analyst	App.	$cost(M_\alpha)$	$pot(D_\alpha)$	$cost \times pot$
Amy	α_1	20	1	20
	α_2	55	1	55
	α_3	45	1	45
			$S(E_{Amy}) =$	0.66
Amy′	α_1	31	0.5	15.5
	α_2	33	0.5	16.5
	α_3	51	0.5	25.5
			$S(E_{ideal}) =$	0.26

6. Related Work

Early visualization researchers developed a variety of models to help them understand the visualization process [6,11]. For example, Chi [8] devised a visualization transform model that describes of how data evolves from its "raw" state to a "view" state as it passes through a four-stage pipeline. Chi's intention was to establish a canonical way to describe any visualization technique, which would enable developers to compare and contrast different techniques as well as identify pipeline stages where techniques overlap [7]. Although Chi's effort was centered on data transformation, his model lacked a cost structure that could be used to establish metrics for rating or ranking visualizations.

In contrast, the Visual Analytics (VA) community has continually developed and revised analytical cost models for decades [33,31]. These models, however, mainly consider *cognitive* costs associated with user interactions [23] and visual pattern recognition. In particular, Patterson [30] described how analysts use visualizations to make decisions and suggested six leverage points that make visualizations easier to interpret.

Other VA researchers have taken a more data-centric perspective on visualization cost. Wijk, for example, proposed an economic model that considers the ratio of insight gained to the cost of generating a visualization [39]. Wijk specifically highlighted cost C_i, which captures cost of developing a visualization. It is not clear, however, which specific factors influence C_i (e.g., an analyst's familiarity with programming or ability to gather source information), leaving analysts with little direction as to how to better quantify and mitigate that cost.

Kandel [17], on the other hand, provided a detailed account of the challenges analysts face when generating visualizations and even developed a tool that can mitigate those challenges [18]. He discusses different classes of analysts with regards to their experience and tools they use. He also describes how each class of analyst approaches the problem of munging data, determining data quality, and reusing prior results. His work largely motivates our theory, which we believe is the next logical step in his work; formally articulate his analysts' testimonies. In addition to providing motivation, Kandel also touches on how semantic data can be used to address the challenges of formatting, extracting, and converting data to fit input data requirements. He even suggests that these data types should be shared and reused across analyses, similarly to how the Linked Data community advocates the reuse of popular vocabularies [36].

Similarly, Fink provided an account of the challenges faced in cyber-security settings [10]. He found that, much like Kandel's enterprise subjects, cyber security analysts are limited by their ability to cheaply mitigate disparities among diverse data and tools. Additionally, some analysts even noted the difficulty in linking applications and expressed their desire for environments that support result chaining.

The models from VA provide good explanations of how visualization quality, user experience, and workplace politics impact analytical costs, especially when results must flow from one analyst to the next. These models, however, do not emphasize how data structuredness and linkability impact cost; structure in VA refers to the conceptual schema of information rather than the physical format in which the information resides [22,31,33]. The Linked Data (LD) community, on the other hand, has long considered the potential costs and benefits associated with publishing and consuming structured, linked data, but not necessarily in analytical settings where results flow across analysts. For example, 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 linked RDF [25]. His scheme has been useful in understanding the affordances to data consumers in *client-server* settings where data is only generated by publishers, rather than *peer-to-peer* analytical settings, where consumers generate results and thus become publishers themselves.

Similarly, Janowicz and Hitzler [16] describe how the Semantic Web provides analysts with opportunities to use third-party data in contexts not envisioned by the data provider. Analysts can use OWL to formally articulate the input schema to their analytical applications, and then use those formal expressions as an alignment target, much like our notion of input semantics. In the same spirit, Heath and Bizer describe an application architecture for LD applications, citing data access (e.g., HTTP Get) and vocabulary mapping (i.e., a kind of munging) as major components [13].

7. Future Work

In terms of our seamlessness score described in Section 4, we can enhance our cost models to consider an analyst's experience. Different visualizations, d_α, are easier to interpret than others, depending upon the experience and biases of an analyst as well as how well the visual metaphor relates to the task at hand. Prior work [39] in VA defines a usage cost, C_e, that denotes the "perception and exploration cost" when analysts use visualization tools. We can include C_e in our application cost formula (i.e., the numerator in metric μ) to derive a new and more complete cost formula:

$$\sum_{m \in M_\alpha} cost(m) + C_e(d_\alpha)$$

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 quality five-star data [14,35] Analysts must have experience in good URI design and popular vocabularies[26]. Additionally, analysts need to have some grasp of RDF patterns, such as PROV qualified associations and Semantic Science Integrated Ontology (SIO)[27], so they can understand how to more effectively anchor their RDF to existing linked data in more recognizable and discoverable ways.

We also need engineered approaches for developing software tools that operate on Linked Data. Currently, most VA tools do not accept and generate RDF and thus it is up to analysts to employ munges that conform to the Five-Star requirements. We are working to provide the Software Engineering community with a suitable software abstraction and set of requirements that can guide the development of tools that better facilitate five-star usage. These new tools would expose their input semantics and generate linkages between source data and derived visualizations.

Ultimately, we believe our theory is a first step towards embodying the LD community's assumptions, claims, and hypothesis in a simple form that can be used to better understand the limitations and practical applications of LD. When our theory predicts a lower cost that what is observed, we may be able to locate high-cost applications and determine which munges contribute to the inflation; perhaps ontology alignment is still too expensive. In these cases, we may also be able to characterize work environments where the overhead of generating and maintaining LD is not outweighed by the prospective cost savings, for example, in settings where analysts do not share results and materials.

8. Conclusion

We forged a Theory of application seamlessness that predicts the cost of non-trivial analyses that span multiple applications. The theory is a conglomerate of theories from the Visualization Analytics and Linked Data communities and explains analytical costs in terms of data evolution (i.e, Visualization Analytics

[25] http://5stardata.info

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

[27] http://semanticscience.org/

theory) and data structuredness (i.e., Linked Data theory). As data evolves into ordered forms that facilitate analytic reasoning, it jumps within a dichotomous space of mundane and semantic formats. The theory suggests that when data occupies the mundane space, the cost to perform the analysis increases.

We described our theory in three parts: a Application Ontology (AO) that describes analytic applications regardless of the type of data, tool, or objective involved; a scoring metric to assess the cost of analyses described in AO; and a set of cost reduction strategies that are expressed in the form of restrictions on AO. We demonstrated the utility of the theory by comparing the actual cost and predicted cost of two analyses: one real-world example based on the current state of practice and an alternative, hypothetical analysis that employs the cost reduction strategies.

References

[1] J. Aasman and K. 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, 2 2011.

[2] G. A. Atemezing and R. Troncy. Towards interoperable visualization applications over linked data. In S. Auer and D. Fensel, editors, *EDF 2013, 2nd European Data Forum*, Dublin, Ireland, 04 2013. URL: http://www.eurecom.fr/publication/3964.

[3] T. Berners-Lee, J. Hollenbach, K. Lu, J. Presbrey, and M. Schraefel. Tabulator Redux: Browsing and Writing Linked Data. *Linked Data on the Web Workshop (LDOW 2008), Co-located with WWW 2008*, 4 2008.

[4] M. Bostock, V. Ogievetsky, and J. Heer. D^3 Data-Driven Documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[5] C. Buil-Aranda, A. Hogan, J. Umbrich, and P.-Y. Vandenbussche. SPARQL Web-Querying Infrastructure: Ready for Action? In *The Semantic Web–ISWC 2013*, pages 277–293. Springer, 2013. doi:10.1007/978-3-642-41338-4_18.

[6] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999. doi:10.1109/mmul.1999.809241.

[7] E. 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. doi:10.1109/infvis.2000.885092.

[8] E. H.-h. Chi and J. T. Riedl. An Operator Interaction Framework for Visualization Systems. In *Information Visualization, 1998. Proceedings. IEEE Symposium on*, pages 63–70. IEEE, 1998. doi:10.1109/infvis.1998.729560.

[9] K. A. Cook and J. J. Thomas. Illuminating the path: The Research and Development Agenda for Visual Analytics. Technical Report PNNL-SA-45230, Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2005.

[10] G. A. Fink, C. L. North, A. Endert, and S. Rose. Visualizing Cyber Security: Usable Workspaces. In *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, pages 45–56. IEEE, 2009. doi:10.1109/vizsec.2009.5375542.

[11] R. B. Haber and D. A. McNabb. Visualization idioms: A Conceptual Model for Scientific Visualization Systems. *Visualization in scientific computing*, 74:93, 1990.

[12] T. Hastrup, R. Cyganiak, and U. Bojars. Browsing Linked Data with Fenfire. *Linked Data on the Web Workshop (LDOW 2008) Co-located with WWW*, 4 2008.

[13] T. Heath and C. 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. doi:10.2200/s00334ed1v01y201102wbe001.

[14] A. Hogan, J. Umbrich, A. Harth, R. Cyganiak, A. Polleres, and S. Decker. An Empirical Survey of Linked Data Conformance. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:14–44, 2012. doi:10.1016/j.websem.2012.02.001.

[15] I. Jacobs and N. Walsh. Architecture of the World Wide Web. *W3C Recommendation*, Volume One, 2004. URL: http://www.w3.org/TR/webarch/.

[16] K. Janowicz and P. Hitzler. The Digital Earth as Knowledge Engine. *Semantic Web*, 3(3):213–221, 2012.

[17] S. Kandel, J. Heer, C. Plaisant, J. Kennedy, F. van Ham, N. H. Riche, C. Weaver, B. Lee, D. Brodbeck, and P. Buono. Research Directions in Data Wrangling: Visualizations and Transformations for Usable and Credible Data. *Information Visualization*, 10(4):271–288, 2011. doi:10.1177/1473871611415994.

[18] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive Visual Specification of Data Transformation Scripts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3363–3372, New York, NY, USA, 2011. ACM.

[19] S. Kandel, A. Paepcke, J. M. Hellerstein, and J. Heer. Enterprise Data Analysis and Visualization: An Interview Study. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2917–2926, 2012. doi:10.1109/tvcg.2012.219.

[20] S. Katzenbeisser and F. Petitcolas. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech house, 2000.

[21] G. Kepeklian, R. Troncy, and L. Bihanic. Datalift: A Platform for Integrating Big and Linked Data, 11 2014. URL: http://www.eurecom.fr/publication/4397.

[22] G. Klein, J. K. Phillips, E. L. Rall, and D. A. Peluso. A Data-Frame Theory of Sensemaking. In *Expertise out of context: Proceedings of the sixth international conference on naturalistic decision making*, pages 15–17. Psychology Press, 2007.

[23] H. Lam. A Framework of Interaction Costs in Information Visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1149–1156, 2008. doi:10.1109/tvcg.2008.109.

[24] T. Lebo, A. Graves, and D. L. McGuinness. Content-Preserving Graphics. In *Fourth International Workshop on Consuming Linked Data (COLD 2013), Co-located with ISWC 2013*, 2013.

[25] T. Lebo, S. Sahoo, D. McGuinness, K. Belhajjame, J. Cheney, D. Corsar, D. Garijo, S. Soiland-Reyes, S. Zednik, and J. Zhao.

PROV-O: The PROV Ontology. *W3C Recommendation, 30th April*, 2013.

[26] T. Lebo, P. West, and D. L. McGuinness. Walking into the Future with PROV Pingback: An Application to OPeNDAP using Prizms (in press). In B. Ludaescher and B. Plale, editors, *Provenance and Annotation of Data and Processes*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014. doi:10.1007/978-3-319-16462-5_3.

[27] T. Lebo and G. T. Williams. Converting Governmental Datasets into Linked Data. In *Proceedings of the 6th International Conference on Semantic Systems*, page 38, New York, NY, USA, 2010. ACM. doi:10.1145/1839707.1839755.

[28] S. Liu, W. Cui, Y. Wu, and M. Liu. A Survey on Information Visualization: Recent Advances and Challenges. *The Visual Computer*, pages 1–21, 2014. doi:10.1007/s00371-013-0892-3.

[29] N. F. Noy. Semantic Integration: A Survey of Ontology-Based Approaches. *ACM Sigmod Record*, 33(4):65–70, 6 2004.

[30] R. E. Patterson, L. M. Blaha, G. G. Grinstein, K. K. Liggett, D. E. Kaveney, K. C. Sheldon, P. R. Havig, and J. A. Moore. A Human Cognition Framework for Information Visualization. *Computers & Graphics*, 42:42–58, 2014. doi:10.1016/j.cag.2014.03.002.

[31] P. Pirolli and S. Card. The Sensemaking Process and Leverage Points for Analyst Technology as Identified Through Cognitive Task Analysis. In *Proceedings of International Conference on Intelligence Analysis*, volume 5, pages 2–4. Mitre McLean, VA, 2005.

[32] M. A. Rodriguez. A Graph Analysis of the Linked Data Cloud. *CoRR*, abs/0903.0194, 2009. URL: http://arxiv.org/abs/0903.0194.

[33] D. M. Russell, M. J. Stefik, P. Pirolli, and S. 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. doi:10.1145/169059.169209.

[34] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. 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. doi:10.1145/2047196.2047247.

[35] M. Schmachtenberg, C. Bizer, and H. Paulheim. Adoption of the Linked Data Best Practices in Different Topical Domains. In P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, editors, *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014. doi:10.1007/978-3-319-11964-9_16.

[36] M. Schmachtenberg, C. Bizer, and H. Paulheim. Adoption of the Linked Data Best Practices in Different Topical Domains. In P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, editors, *The Semantic Web–ISWC 2014*, pages 245–260. Springer, 2014. doi:10.1007/978-3-319-11964-9_16.

[37] M. G. Skjæveland. Sgvizler: A Javascript Wrapper for Easy Visualization of SPARQL Result Sets. In *The Semantic Web: ESWC 2012 Satellite Events*, pages 361–365. Springer, 5 2012. doi:10.1007/978-3-662-46641-4_27.

[38] J. Unbehauen, S. Hellmann, S. Auer, and C. Stadler. Knowledge extraction from structured sources. In S. Ceri and M. Brambilla, editors, *Search Computing*, pages 34–52. Springer-Verlag, Berlin, Heidelberg, 2012. URL: http://dl.acm.org/citation.cfm?id=2427336.2427340.

[39] J. J. Van Wijk. The Value of Visualization. In *Visualization, 2005. VIS 05. IEEE*, pages 79–86. IEEE, 2005. doi:10.1109/visual.2005.1532781.

[40] M. D. Wilkinson, L. McCarthy, B. Vandervalk, D. Withers, E. Kawas, and S. Samadian. SADI, SHARE, and The In Silico Scientific Mehtod. *BMC bioinformatics*, 11:S7, 2010.