Representing Interoperable Provenance Descriptions for ETL Workflows

André Freitas[1], Benedikt Kämpgen[2], João Gabriel Oliveira[1], Seán O'Riain[1], and Edward Curry[1]

[1]Digital Enterprise Research Institute (DERI)
National University of Ireland, Galway
[2]Institute AIFB
Karlsruhe Institute of Technology

Abstract. The increasing availability of data on the Web provided by the emergence of Web 2.0 applications and, more recently by Linked Data, brought additional complexity to data management tasks, where the number of available data sources and their associated heterogeneity drastically increases. In this scenario, where data is reused and repurposed on a new scale, the pattern expressed as Extract-Transform-Load (ETL) emerges as a fundamental and recurrent process for both producers and consumers of data on the Web. In addition to ETL, *provenance*, the representation of source artifacts, processes and agents behind data, becomes another cornerstone element for Web data management, playing a fundamental role in data quality assessment, data semantics and facilitating the reproducibility of data transformation processes. This paper proposes the convergence of these two Web data management concerns, introducing a principled provenance model for ETL processes in the form of a vocabulary based on the Open Provenance Model (OPM) standard and focusing on the provision of an interoperable provenance model for ETL environments. The proposed ETL provenance model is instantiated in a real-world sustainability reporting scenario.

Keywords: ETL, Data Transformation, Provenance, Linked Data, Web

1 Introduction

Extract-Transform-Load (ETL) is a fundamental process in data management environments. In Data Warehousing, data preprocessing is crucial for reliable analysis, e.g., reporting and OLAP; data coming from large databases or data derived using complex machine-learning algorithms may hide errors created in an earlier step of the analysis process. As a result, the design of ETL processes such as retrieving the data from distributed sources, cleaning it from outliers, and loading it in a consistent data warehouse demands up to 80 percent of data analysts' time [12].

The growing availability of data on the Web provided by Web 2.0 applications and, more recently through Linked Data, brought the computational pattern expressed as ETL to reemerge in a scenario with additional complexity, where the number of data sources and the data heterogeneity that needs to be supported by ETL drastically increases. In this scenario, issues with data quality and trustworthiness may strongly impact the data utility for end-users. The barriers involved in building an ETL infrastructure under the complexity and scale of the available Web-based data supply scenario demands the definition of strategies which can provide data quality warranties and also minimise the effort associated with data management.

In this context, *provenance*, the representation of *artifacts*, *processes* and *agents* behind a resource, becomes a fundamental element of the data infrastructure. Given the possibility to represent ETL workflows both at design time (*prospective provenance*), and after execution (*retrospective provenance*), provenance descriptions can overcome challenges of today's ETL scenarios in a large spectrum of applications including documentation for reproducibility and reuse, data quality assessment to improve trustworthiness as well as automatic consistency checking, debugging and semantic reconciliation [14]. Additionally, the frequency and generality of simple and recurrent processes such as contained in many data transformation workflows in an environment with increasing data availability justifies the importance of a provenance descriptions for ETL.

However, in an environment where data is produced and consumed by different systems, the representation of provenance should be made interoperable across systems. Interoperability represents the process of sharing the semantics of the provenance representation among different contexts. Although some systems in the areas of data transformation [1] and databases [20] provide a historical trail of data, those descriptions cannot be easily shared or integrated. *Provenance* and *interoperability* walk together: provenance becomes fundamental when the borders of a specific system or dataset are crossed, where a representation of a workflow abstraction of the computational processes can enable reproducibility, improve data semantics and restore data trustworthiness. Ultimately, provenance can make the computational processes behind applications interpretable at a certain level by external systems and users.

Standardisation efforts towards the convergence into a common provenance model generated the Open Provenance Model [11] (OPM). OPM provides a basic provenance description which allows interoperability on the level of workflow structure. The definition of this common provenance ground allows systems with different provenance representations to share at least a workflow-level semantics, i.e., the causal dependencies between artifacts, processes and the intervention of agents. OPM, however, is not intended to be a complete provenance model, but demands the complementary use of additional provenance models in order to enable applications of provenance that require higher level of semantic interoperability. The explicit trade-off between the semantic completeness of a provenance model and its level of interoperability imposes challenges in specifying a provenance model.

This paper focuses on the provision of a solution that allows the improvement of the semantic completeness and interoperability for provenance descriptors in complex data transformation/ETL scenarios. To achieve this goal, a vocabulary focused on modelling ETL workflows is proposed. This model is built upon the workflow structure of OPM, designed to extend the basic semantics and structure of OPM-based provenance workflows. In this work, the ETL acronym is used in a broader context, focusing on generic data transformation patterns, transcending the original Data Warehouse associated sense. The contributions of this work are summarised in the following: **(i)** analysis of requirements for an interoperable provenance model for ETL workflows, **(ii)** provision of a solution in the form of a Linked Data ETL vocabulary, **(iii)** application of the proposed model in a real-world ETL scenario.

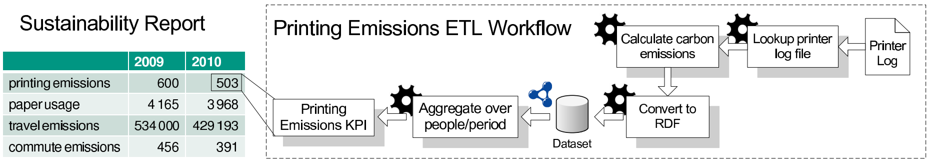
The paper is organised as follows: Section 2 presents an ETL motivational scenario, Section 3 analyses related work on the representation and formalisation of ETL provenance workflows; Section 4 provides a list of requirements for an ETL provenance model; Section 5 describes the construction of the ETL provenance model, describing *Cogs*, a provenance vocabulary for ETL. Section 6 describes the application of the ETL vocabulary in a case study for sustainable reporting. Section 7 finally provides conclusions and future work.

2 ETL Motivational Scenario

The ability to describe data transformation processes behind data resources plays a fundamental role while producing and consuming data, especially if done on heterogeneous data sources and by different parties. Applications need to become provenance-aware, i.e., attaching to the data the associated description of what has been done to generate the data. This brings provenance management as a key requirement for a wide spectrum of applications which publish and consume data on the Web and, in particular, to ETL activities.

As a concrete motivational scenario consider an organisation publishing a sustainability report on the Web (Figure 1). The sustainability report contains Key Performance Indicators (KPIs) related to the environmental impact of the organisation which can be audited by external regulators, reused by customers to calculate their indirect environmental impact or used internally to minimise the company environmental impact. One example KPI is the total volume of CO2 emissions/per time period which is calculated by collecting indicators of energy consumption emissions, travel emissions, printing emissions, etc. The data used to build the indicators is collected from distributed and heterogeneous sources which include spreadsheets, log files and RDF data, and is processed through distinct ETL workflows into data cubes. An application queries the final sustainability KPIs from the data cubes, publishing them as a report on the Web.

The problem that is specifically introduced in this scenario is the fact that different values might have been produced by independently developed and executed ETL workflows. For instance, the value indicating a printing emissions of 503 kg of carbon dioxide as indicated in Figure 1 is created by a lookup on the printer log file, a conversion to RDF, an aggregation over people and a filter on the year 2010. The printing emission for 2009, however, might have been produced by a crawl of RDFa from the organisation's website and a unit conversion by a constant factor. Each KPI should have an associated provenance trail describing the data processing steps from the original data sources, so that information consumers – both humans and machines – are able to make better sense of information generated by heterogeneous ETL processes.


Sustainability Report

	2009	2010
printing emissions	600	503
paper usage	4 165	3 968
travel emissions	534 000	429 193
commute emissions	456	391

Printing Emissions ETL Workflow
Calculate carbon emissions
Lookup printer log file
Printer Log
Printing Emissions KPI
Aggregate over people/period
Dataset
Convert to RDF

Fig. 1. Representing provenance behind the KPI of a sustainability report.

3 Related Work: The Gap of ETL Workflow Descriptions

Previous literature analysed and formalised conceptual models for ETL activities. In the center of these models is the concept of *data transformations*. This section describes previous data transformation models, analysing their suitability as interoperable provenance descriptions for our motivational scenario. Existing work can be grouped into two major perspectives: *ETL Conceptual Models*, which focus on the investigation of ontologies and serialisation formats for design, development, and management of ETL workflows, and *ETL Formal Models*, which concentrate on applications of ETL descriptions that require formal, logics- or algebra-based representations. Between these two groups we identify the gap of an interoperable ETL provenance model.

3.1 ETL Conceptual Models

Standardisation efforts by the W3C Provenance Incubator Group[1] and the later Provenance Working Group[2] have considered in-scope use cases of data integration and repeatable data analyses. Yet, their focus targets the determination of basic provenance descriptors allowing interoperability on an abstract level of workflow semantics and does not target more specific provenance descriptors from an ETL perspective.

Much work has been done in the usage of ontologies for automated design and standard descriptions of ETL tasks. Vassiliadis et al. [19] investigate generic properties present in ETL activities across different ETL implementations and, based on these properties, construct a taxonomy of ETL concepts. Vassiliadis et

[1] http://www.w3.org/2005/Incubator/prov/XGR-prov-20101214/

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

al. and Skoutas & Simitsis [19, 15] use a categorisation of operations from different systems to capture the relationship between input and output definitions. Trujillo and Luján-Mora [17] propose to use UML for the specification of ETL processes in terms of operations such as the transformation between source and target attributes and the generation of surrogate keys. Similarly, Akkaoui & Zimani [5] propose a conceptual language for modelling ETL workflows based on the Business Process Model Notation (BPMN). The artificiality of the solution lies on the fact that BPMN is not intended to be a universal data representation format, bringing questions on its suitability as an interoperable representation. In general, although the mentioned conceptual models introduce common terms and structures for ETL operations and help with ETL-related communication and discussions, they do not aim at providing a machine-readable representation of heterogeneous ETL processes to be shared by data consumers.

Becker & Ghedini [2] describe a system to document data mining projects, including the data preprocessing step. Descriptions are manually captured by the analysts in a Web 2.0 fashion. The ETL representations include tasks as an abstraction for various preprocessing activities, distinguishing between prospective task definitions and retrospective task executions. Tasks can be annotated using free text and can be tagged with predefined concepts. Although useful for reproducibility and reuse in terms of knowledge management between ETL designers, those descriptions lack a minimum of ontological commitment for interoperability between heterogeneous ETL applications.

Other works specifically aims at sharing descriptions between systems. The Common Warehouse Metamodel (CWM) is an open OMG standard for data warehousing which defines a metadata model and an XML-based exchange standard. In CWM, a *transformation* is a basic unit in the ETL process which can be combined into a set of tasks. Thi & Nguyen [16] propose a CWM compliant approach over an ontology-based foundation for modelling ETL processes for data from distributed and heterogeneous sources; their approach does not model the types of transformations explicitly but only provides a basic mapping infrastructure which can be used to reference external classes. Also, the approach lacks a concrete use case where its benefits are demonstrated.

Kietz et al. [9] introduce a cooperative planning approach for data mining workflows using the support of a data mining ontology (DMO). DMO covers artifacts (I/O-Objects), processes (Operators), as well as descriptors to describe artifacts in more detail (MetaData); as such, it provides ETL descriptions and allows semantic interoperability between systems. However, DMO was not designed to track the history of data and lacks retrospective provenance of ETL workflows.

3.2 ETL Formal Models

Formal models use logics or algebras to describe ETL descriptions. Davidson et al. [4] analyse the requirements for the construction of a formalism for modelling the semantics of database transformations and propose a declarative language for specifying and implementing database transformations and constraints. The motivation of their work is to generate a transformation formalism which can be used to verify the correctness of transformations. Galhardas et al. [8] propose another high-level declarative language for data transformations and describe the reasoning behind transformations. However, abstract languages for data cleaning and transformations are overly formal to be widely adopted for interoperable ETL descriptions.

Cui & Widom [3] formalise the lineage problem on general database environments proposing algorithms for lineage tracing. They restrict their model to specific transformation classes. The approach is not suited to describe general data transformation activities varying from simple filtering operations to complex procedural routines such as present in our motivational scenario.

Vassiliadis et al. [18] provide an abstract categorisation of frequently used ETL operations in order to introduce a benchmark of relational ETL systems. The benchmark documents measures such as data freshness and consistency, resilience to failures, and speed of workflows. In order to describe concrete ETL workflows such as given by our motivational scenario, both operations and measures are too abstract to help with problems of interpretability.

In summary, formal models of ETL workflows often explicitly limit their range of considered ETL workflows to fulfil specific tasks. Those models do not intend to provide interoperability across different ETL applications, but to achieve certain functionalities in their system, e.g., automatic debugging.

We have identified a gap regarding an interoperable ETL provenance model. Previous literature has either presented models with very high-level semantics lacking the ability to describe prospective and retrospective ETL provenance or presented rigorously formalised models that require to much ontological commitment for a broad adoption. As for ETL applications such as Kapow Software, Pentaho Data Integration, and Yahoo Pipes: currently, they either do not create and use provenance information or do not support sharing and integrating such provenance data with other applications.

4 Requirements of an Interoperable ETL Provenance Model

This section defines a list of requirements which summarises the core usability and model characteristics that should be present in an ETL provenance model. The requirements are defined to satisfy the two core demands which were found as gaps on the ETL literature (i) lack of a provenance representation from an ETL perspective and (ii) semantic interoperability across different ETL platforms and applications. An additional third demand is introduced: (iii) usability demand, i.e., the minimal effort and ontological commitment needed for an instantiation of a correct and consistent model. The requirements are described below:

1. *Prospective and retrospective descriptions*: Provenance descriptors represent both workflows specifications at design time (*prospective provenance*) and

workflows which were already executed (*retrospective provenance*). Impacts: i, ii and iii.

2. *Separation of concerns*: ETL-specific elements are separated from the provenance workflow structure, allowing at least a minimum level of interoperability between ETL and non-ETL provenance descriptors. This requirement is aligned with the OPM [11] compatibility. Impacts: ii.
3. *Terminological completeness*: Terminological completeness of the provenance descriptor is maximised; there is a large terminological coverage of ETL elements. Impacts: i and ii.
4. *Common terminology*: Descriptors allow a common denominator of representations of ETL elements. Elements present in different ETL platforms can be mapped. Impacts: i and ii.
5. *Lightweight ontology structure*: A lightweight provenance model is provided; complex structures bring barriers for the instantiation and consumption of models, including consistency problems, scalability issues, interpretability problems and additional effort in the model instantiation. Impacts: iii.
6. *Availability of different abstraction levels*: The vocabulary allows users to express multiple abstraction levels for both processes and artifacts, varying from fine grained to coarse grained descriptions. Users are able to express multiple levels of abstraction simultaneously. This requirement is also present in the OPM specification [11]. Impacts: ii and iii.
7. *Decentralisation*: ETL provenance descriptors may be deployed on distributed database platforms without requiring cooperation among all databases. Impacts: ii and iii.
8. *Data representation independency*: Descriptors are able to refer to any data representation format including relational, XML, text files, etc. Impacts: iii.
9. *Accessibility*: The generated provenance descriptors are easily accessible for data consumers. Both machines and humans are able to query and further process provenance descriptors. Impacts: ii and iii.

5 Provenance Model for ETL Workflows

The following high-level approach was used to provide an ETL provenance model addressing the requirements:

- Construction of the provenance model based on the Open Provenance Model workflow structure, extending OPM with a hierarchical workflow structure, facilitating the representation of nested workflows.
- Design of a complementary vocabulary for expressing the elements present in an ETL workflow. The vocabulary can be extended to describe domain-specific objects.
- Usage of the Linked Data principles for representing, publishing and linking provenance descriptors on the Web in a machine-readable format.

In the following, we describe in more detail the two main features of the ETL provenance model: the multi-layered design and the ETL vocabulary *Cogs*.

5.1 Multi-Layered Provenance Model

A three-layered approach is used, as depicted on the left side of Figure 2, to provide interoperable provenance representations of ETL and generic data transformation workflows. OPM is a technology agnostic specification: it can be implemented using different representations or serialisations. This work uses the OPM Vocabulary[3] (OPMV) as the representation of OPM. In this representation, the bottom layer represents the basic workflow semantics and structure provided by OPMV, the second layer represents the common data extraction, transformation and loading entities and the third layer represents a domain specific layer.

The ETL provenance model layer is built upon the *basic workflow structure* of the OPMV layer. The ETL provenance model layer is designed to include a set of common entities present across different ETL workflows, providing a terminologically-rich provenance model instantiated as the *Cogs* vocabulary. The third layer consists of a domain specific layer which extends the second layer, consisting of domain-specific schema and instance-level information, e.g., of domain-specific source and target datasets or operations. An example of domain specific elements are references to e-Science operations from biological experiments that would further specialise classes of Cogs operators.

This paper defines a conceptual model for the second layer and describes its interaction with the two complementary layers. The separation of the provenance model into the three-layered structure supports the requirement *(2) separation of concerns.*

5.2 Cogs: A Vocabulary for Representing ETL Provenance

In the construction of Cogs, the core relationships are provided by *object properties* on the OPMV layer. The Cogs model specialises the core OPMV entities, artifacts and processes, with a rich taxonomic structure. The approach used in Cogs focuses on the design of a Linked Data vocabulary, a lightweight ontology, which minimises the use of logical features such as transitive, inverse properties as well as the consistency/scalability problems associated with the reasoning process (impacts requirement *(5) lightweight ontology structure*).

The methodology for building the Cogs vocabulary considered the following dimensions: (i) the requirements analysis (ii) the core structural definition of modelling ETL workflows using the structure of OPMV workflows, (iii) an in depth analysis of concepts expressed in a set of analysed ETL/data transformation tools (Pentaho Data Integration,[4] Google Refine[5]) and (iv) concepts and structures identified from the ETL literature [3, 19, 10]. The core of the Cogs vocabulary captures typical operations, objects and concepts involved in ETL activities, at different phases of the workflow.

[3] `http://open-biomed.sourceforge.net/opmv/ns.html`

[4] `http://kettle.pentaho.com`

[5] `http://code.google.com/p/google-refine`

Cogs also extends the workflow structure of OPMV with additional object properties targeting the creation and navigation of hierarchical workflow structures. Hierarchical workflow structures allow the representation of both fine grained (important for machine interpretation and automated reproducibility) and coarse grained (important for human interpretation) provenance representation. This feature impacts both requirements *(6) availability of different abstraction levels and (1) prospective and retrospective descriptions*. Similar hierarchical features extending OPM were also targeted in [6]. Figure 2 depicts the core of the OPMV workflow model and the workflow extension of the Cogs vocabulary (marked with the cogs namespace).

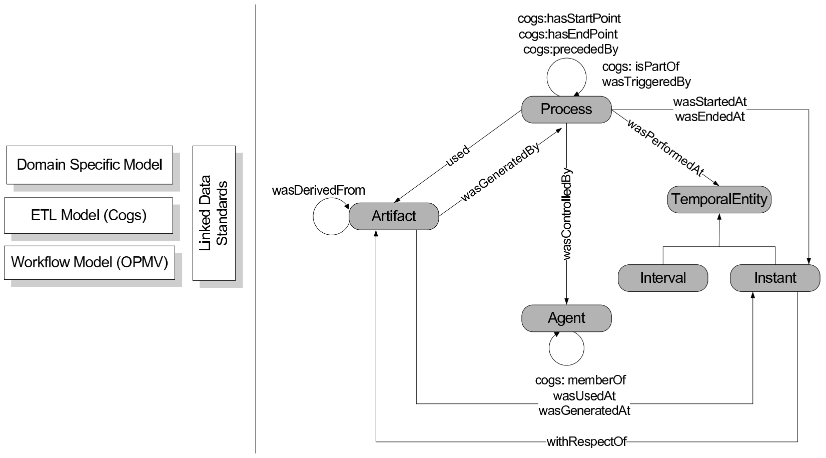


Fig. 2. OPMV workflow structure extended with additional Cogs properties.

The Cogs vocabulary defines a taxonomy of 151 classes. In addition, 15 object properties and 2 data properties are included in the vocabulary. The large number of classes allows a rich description of ETL elements supporting an expressive ETL representation (impacts requirements *(3) terminological completeness and (6) availability of different abstraction levels*). The classes, extracted from the ETL literature and from available tools also cover the *(4) common terminology* requirement. The vocabulary taxonomy is structured with 8 high-level classes which are described below:

- *Execution*: Represents the execution job (instance) of an ETL workflow. Examples of subclasses include *AutomatedAdHocProcess* and *ScheduledJob*.
- *State*: Represents an observation of an indicator or status of one particular execution of an ETL process. These can range from execution states such as *Running* or *Success* to execution statistics, captured by the subclasses of the *PerformanceIndicator* class.

- *Extraction*: Represents operations of the first phase of the ETL process, which involves extracting data from different types of sources. *Parsing* is a subclass example. cogs:Extraction is an opmv:Process.
- *Transformation*: Represents operations in the transformation phase. Typically this is the phase which encompasses most of the semantics of the workflow, which is reflected on its number of subclasses. Examples of classes are *RegexFilter*, *DeleteColumn*, *SplitColumn*, *MergeRow*, *Trim* and *Round*. cogs:Transformation is an opmv:Process.
- *Loading*: Represents the operations of the last phase of the ETL process, when the data is loaded into the end target. Example classes are *ConstructiveMerge* and *IncrementalLoad*. cogs:Loading is an opmv:Process.
- *Object*: Represents the sources and the results of the operations on the ETL workflow. These classes, such as *ObjectReference*, *Cube* or *File*, aim to give a more precise definition of opmv:Artifact (every cogs:Object is an opmv:Artifact) and, together with the types of the operations that are generating and consuming them, capture the semantics of the workflow steps.
- *Layer*: Represents the different layers where the data can reside during the ETL process. *PresentationArea* and *StagingArea* are some of the subclasses.

In practice, it is not always possible to capture all data transformation operations into a fine-grained provenance representation. One important feature of the Cogs vocabulary is the fact that program descriptions (i.e. source code) or executable code can be associated with the transformations using the *cogs:programUsed* property. This feature impacts the requirements *(3) terminological completeness, (6) availability of different abstraction levels and (1) prospective and retrospective descriptions.*

The use of Linked Data principles strongly supports requirement *(10) accessibility* by allowing a unified standards-based publication and access layer to data. In the proposed model, the standards-based provenance representation is separated from the database representation (a relational database record or an element inside an XML file can have its provenance information represented using Linked Data principles). The use of (provenance) URIs to associate provenance information to data items is a generic solution which can be directly implemented to every data representation format, supporting the requirement *(8) data representation independency.* Additionally, by using RDF(S), HTTP and URIs, provenance can be persisted in a decentralised way (requirement *(7) decentralisation*). Users can access provenance through SPARQL queries, faceted-search interfaces, and follow-your-nose Linked Data browsers over dereferenceable URIs.

Table 1 summarises the requirements coverage by the proposed provenance model. The current version of the Cogs vocabulary is available at `http://vocab.deri.ie/cogs` and complementary documentation is available at: `http://sites.google.com/site/cogsvocab/`.

Requirement	OPMV	Cogs	LD principles
Prospective and retrospective descriptions	+	+	-
Separation of concerns	+	+	-
Terminological completeness	+	+	+
Common terminology	+	+	-
Lightweight ontology structure	+	+	-
Availability of different abstraction levels	-	+	-
Decentralisation	-	-	+
Data representation independency	+	+	+
Accessibility	+	-	+

Table 1. Requirements coverage of each element of the provenance model: '+' represents an effective impact on the requirements dimension while '-' represents the lack of impact.

6 Vocabulary Instantiation

In order to analyse the suitability of the proposed vocabulary as a representation of ETL processes, we have implemented an instantiation of the Cogs vocabulary using as a case study a platform for collecting sustainability information at the Digital Enterprise Research Institute (DERI), similar to our motivational scenario. We first describe the application of the ETL provenance model in the use case and then discuss the results.

6.1 Use Case

The organisation-wide nature of sustainability indicators, reflecting the organisational environmental impact, means that potential information is scattered across the organisation within numerous existing systems. Since existing systems were not designed from the start to support sustainability analysis, heterogeneous data present in distributed sources need to be transformed into sustainability indicators following an ETL process. The correctness and consistency of each sustainability KPI needs to be auditable through the publication of the associated provenance information, which should be interpretable by different stakeholders.

The ETL process for the construction of sustainability indicators consists of four separate workflows, for printing emissions, paper usage, travel emissions and commute emissions. Data sources include RDF graphs for people, research units and different file formats containing raw data. The basic ETL workflow consists in a sequence of operations: file selection, filtering, transformation, CO2 emissions calculation and transformation into RDF conforming to the RDF Data Cube vocabulary. On the last step information in the data cubes is aggregated to generate a final report available on the Web. The ETL workflow is implemented in Java code. To make the ETL workflow provenance-aware, the Prov4J-Light framework was used, a lightweight version of [7], which is a Java framework for provenance management, that uses Semantic Web tools and standards to address the core challenges for capturing and consuming provenance information in generic Java-based applications. Core Java objects are mapped to artifacts and processes in the OPMV + Cogs provenance model. The set of generated instances is persisted in a separate provenance dataset. The connection between the final data, which is available in HTML format, and its provenance descriptor is given by a provenance URI (provURI) which is a reflection of the annotated artifact in the provenance store, pointing to its associated retrospective provenance workflow. Each element in the provenance store is represented by a dereferenceable provenance URI. Applications and users can navigate through the workflow structure by following the graph links or by executing SPARQL queries. Figure 3 depicts the high-level components of the provenance capture and storage mechanism.

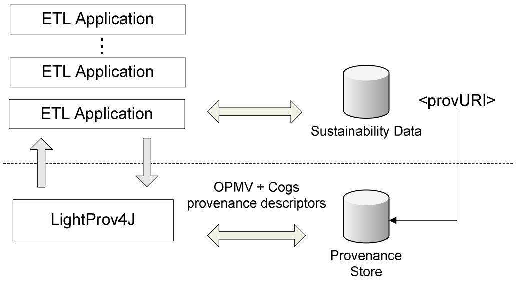


Fig. 3. High-level architecture of the provenance capture and storage mechanism.

6.2 Discussion

The purpose of the workflow usage should be determined in advance, where coarse grained data transformation representations are more suitable for human consumption (in particular, in the determination of human-based quality assessment) while fine grained representations provide a higher level of semantic interoperability which is more suitable for enabling automatic reproducibility. The proposed provenance model for ETL can serve both granularity scenarios. For our case study, since the main goal is to provide a human auditable provenance trail, a coarse grained implementation was chosen. Figure 4 depicts a short excerpt of the workflow in the provenance visualisation interface with both OPMV and Cogs descriptors. The user reaches the provenance visualisation interface by clicking in a value on an online financial report. Readers can navigate through a workflow descriptor for the printing CO2 emissions on the Web.[6] The final average linear workflow size of the high-level workflow consisted of 4 processes and 5 artifacts.

[6] http://treo.deri.ie/cogs/example/swpm2012.htm

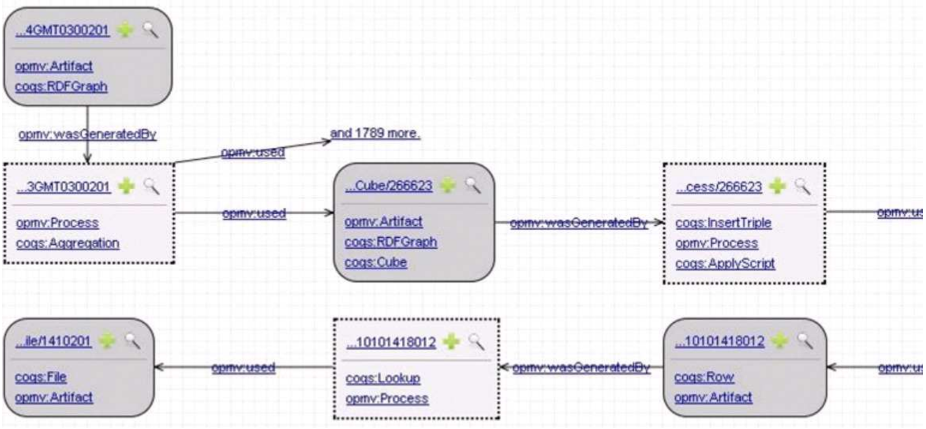


Fig. 4. Visualisation interface for the retrospective provenance of the implemented ETL workflow.

One important aspect for a provenance model is the expressivity of the queries supported by it. The OPMV layer allows queries over the basic workflow structure behind the data, such as *What are the data artifacts, processes and agents behind this data value?*, *When were the processes executed?*, *How long does each process take?*. By adding a Cogs layer to the OPMV layer it is possible to define queries referring to specific classes within the ETL environment, such as *What are the RDF data sources used to generate this data value?*, *Which extractors are used in this workflow?*, *What are the schema transformation operations?*, *Which formulas were used to calculate this indicator?*, *Which is the source code artifact behind this data transformation?*. More specifically to the use case, queries such as *How long did all lookups take?*, *What scripts have been used to transform the data into RDF?*, *To which values constant factors have been applied?*, *Which aggregation functions were used to calculate this indicator?* could be answered to support the interpretation of different ETL executions. The third layer contains information which is domain-specific (not likely to be directly interoperable with other systems). It consists of specific operations (e.g., reference to specific data mining algorithms), schema-level information (such as table names and column names) and program code references (as in the example instantiation). This third layer specialises the classes of the Cogs layer: the presence of the Cogs classes can be used to facilitate the entity resolution among domain-specific layers of different contexts. The use of the Cogs vocabulary allows an increase of the query expressivity in relation to OPMV, allowing queries over the ETL elements. In addition to the direct interoperability increase provided by Cogs-compatible systems, the additional semantics of Cogs can facilitate knowledge discovery between provenance workflows, facilitating the inductive learning and semantic reconciliation of entities in the domain-specific layer.

Compared to previous works, the proposed provenance model focuses on providing a standards-based solution to the interoperability problem, relying on the structure of a community-driven provenance model (OPM) to build a provenance model for ETL. Linked Data standards are used for leveraging the accessibility of provenance descriptors. The proposed provenance model is able to provide a terminology-based semantic description of ETL workflows both in the prospective and retrospective provenance scenarios. The model is targeted towards a pay-as-you-go semantic interoperability scenario: the semantics of each workflow activity can be described with either a partial or a complete provenance descriptor.

7 Conclusion & Future Work

This work presented a provenance model for ETL workflows, introducing *Cogs*,[7] a vocabulary for modelling ETL workflows based on the Open Provenance Model (OPM). The proposed vocabulary was built aiming towards the provision of a semantically interoperable provenance model for ETL environments. The vocabulary fills a representation gap of providing an ETL provenance model, a fundamental element for increasingly complex ETL environments. The construction of the vocabulary is based on the determination of a set of requirements for modelling provenance on ETL workflows. The proposed provenance model presents a high coverage of the set of requirements and was applied to a realistic ETL workflow scenario. The model relies on the use of Linked Data standards.

A more thorough evaluation of the interoperability gained when using Cogs is planned. Future work include the refinement of the vocabulary based on feedback from users. The provenance model proposed in this paper was already implemented to describe interactive data transformations from the Google Refine platform [13]. The verification of the interoperability between Google Refine and an open source ETL platform is planned.

8 Acknowledgements

The work presented in this paper has been funded by Science Foundation Ireland under Grant No. SFI/08/CE/I1380 (Lion-2), by the German Ministry of Education and Research (BMBF) within the SMART project (Ref. 02WM0800) and the European Community's Seventh Framework Programme FP7/2007-2013 (PlanetData, Grant 257641).

References

1. M. Altinel, P. Brown, S. Cline, R. Kartha, E. Louie, V. Markl, L. Mau, Y.-H. Ng, D. Simmen, and A. Singh. Damia: a data mashup fabric for intranet applications. In *Proceedings of the 33rd international conference on Very large data bases*, 2007.

[7] http://vocab.deri.ie/cogs

2. K. Becker and C. Ghedini. A documentation infrastructure for the management of data mining projects. *Information & Software Technology*, 47, 2005.
3. Y. Cui and J. Widom. Lineage tracing for general data warehouse transformations. *The VLDB Journal*, 12, 2003.
4. S. Davidson and P. Buneman. Semantics of database transformations. *Semantics in Databases*, 1998.
5. Z. El Akkaoui and E. Zimanyi. Defining ETL worfklows using BPMN and BPEL. In *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP*, DOLAP '09, pages 41–48, New York, NY, USA, 2009.
6. A. Freitas, T. Knap, S. O'Riain, and E. Curry. W3P: Building an OPM based provenance model for the Web. In *In Future Generation Computer Systems*, 2010.
7. A. Freitas, A. Legendre, S. O'Riain, and E. Curry. Prov4J: A Semantic Web Framework for Generic Provenance Management. In *Second International Workshop on Role of Semantic Web in Provenance Management (SWPM 2010)*, 2010.
8. H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. In *Proceedings of the 27th International Conference on Very Large Data Bases*, 2001.
9. J.-U. Kietz, F. Serban, A. Bernstein, and S. Fischer. Towards cooperative planning of data mining workflows. In *Proc of the ECML/PKDD09 Workshop on Third Generation Data Mining(SoKD-09)*, 2009.
10. R. Kimball and J. Caserta. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning.* John Wiley & Sons, 2004.
11. L. Moreau. The open provenance model core specification (v1.1). *Future Gener. Comput. Syst.*, 27, 2011.
12. K. Morik and M. Scholz. The miningmart approach to knowledge discovery in databases. In *In Ning Zhong and Jiming Liu, editors, Intelligent Technologies for Information Analysis*, 2003.
13. T. Omitola, A. Freitas, S. O'Riain, E. Curry, N. Gibbins, and N. Shadbolt. Capturing Interactive Data Transformation Operations using Provenance Workflows. In *In Proceedings of the 3rd International Workshop on Role of Semantic Web in Provenance Management (SWPM 2012)*, 2012.
14. Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance in e-science. *SIGMOD Rec.*, 34, 2005.
15. D. Skoutas and A. Simitsis. Designing ETL processes using semantic web technologies. In *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP*, 2006.
16. A. Thi and B. T. Nguyen. A Semantic approach towards CWM-based ETL processes. In *Proceedings of I-SEMANTICS 08*, 2008.
17. J. Trujillo and S. Luján-Mora. A UML based approach for modeling ETL processes in data warehouses. In I.-Y. Song, S. W. Liddle, T. W. Ling, and P. Scheuermann, editors, *ER*, volume 2813 of *Lecture Notes in Computer Science*, pages 307–320. Springer, 2003.
18. P. Vassiliadis, A. Karagiannis, V. Tziovara, and A. Simitsis. Towards a benchmark for etl workflows. In V. Ganti and F. Naumann, editors, *QDB*, pages 49–60, 2007.
19. P. Vassiliadis, A. Simitsis, and S. Skiadopoulos. Conceptual modeling for ETL processes. In *Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP*, 2002.
20. J. Widom. Trio : A System for Integrated Management of Data , Accuracy , and Lineage. *Innovative Data Systems Research (CIDR 2005)*, 2005.