Introducing Context and Context-awareness in Data Integration: Identifying the Problem and a Preliminary Case Study on Informed Consent

Christophe Debruyne
Trinity College Dublin
Dublin 2, Ireland
debruync@tcd.ie

ABSTRACT
Data integration is the process of selecting, preprocessing, and transforming data from heterogeneous sources in data-driven projects. This process also requires the most time, effort, and resources. Data integration is such an involved process due to the many informed decisions one has to make. These decisions are influenced by the complex context of a data-driven project. We argue that using said context could facilitate the decision-making processes and even automate some integration steps. However, the problem we identify in this paper is that the context of a data-driven project is tacit and, therefore, not easily accessible by humans and certainly not by software agents. From the SotA, however, we observe that current context models approach context in crude and simplistic terms. Context-aware data integration, proposals are furthermore often built for specific tasks or application domains such as query optimization or a smart home. The current state of affairs is thus not fit for intelligent data integration. Next to identifying the problem, we postulate that solving this problem requires two steps: formalizing context and using that context for building context-aware agents. We illustrate this notion of “context-aware data integration” with preliminary results obtained with a use case in the domain of GDPR, more specifically the generation of datasets that takes into account informed consent.

CCS CONCEPTS
-Information systems-Data management systems-Information integration
-Information systems-World Wide Web-Web data description languages

KEYWORDS
Context-aware Data Integration, Context, Ontology Engineering

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ACM Reference format:

1 With "practitioner", we refer to all the stakeholders actively involved in a data-driven project. This to distinguish practitioners from other stakeholders such as clients, users, etc.
In this paper, we will look at the State-of-the-Art on context-awareness in data integration and identify the problem that context is often regarded in too simplistic terms (Section 2). In Section 3, we elaborate on context in data-driven projects and how one should divide the creation of context-aware data integration agents into 1) the creation of context ontologies, and 2) the creation of those agents using those ontologies. In Section 4, we illustrate our approach in the context of GDPR and the generation of GDPR compliant datasets. This is thus an example of policy-compliant data integration. The algorithms used in this illustration were already described in prior work from a technical perspective. Their integration into this paper’s narrative as an illustration of context-aware data integration is novel. And finally, in Section 5 we conclude the paper.

2 Context-awareness and Data Integration

Data integration is traditionally concerned with combining data from various heterogeneous source representations to a target representation. One thus needs to prescribe how one transforms the source into the target with mappings [3], [4]. Various (standardized) representations exist for mappings such as SQL for views [3] in a data warehousing setting and R2RML2, a W3C Recommendation, for declaring mappings from relational data to the Resource Description Framework (RDF). The scientific community has looked into facilitating data integration with ontology matching and aligning [4], rules to link entities in different datasets [5], and even the use of AI [6] and ML [7] to find correspondences in datasets. [8] recently demonstrated how they used NLP to extract knowledge from unstructured text to facilitate data integration. So while the community has made significant strides in representing and executing mappings, and even automating data integration in terms of ontology/schema matching and data interlinking, we have yet to make a big leap forward by using context for driving the whole data integration process. In this section, we will present and assess related work from two perspectives: the context-model and the techniques availing of said model for context-aware data integration.

2.1 Related Work on a Context Model

Context is defined as “any information that can be used to characterize the situation of entities [...] that are considered relevant to the interaction between a user and an application, including the user and the application themselves.” [9] The authors of [9] furthermore state that context is typically the location, state, and identity of people and objects. As far as capturing context goes, researchers have tried modeling context for their uses: surveys include [10] for IoT, [11] for mobile environments, and [12] for pervasive computing. In [13] and [14], the authors provide examples of context modeling for personalization and recommender systems, respectively.

The State-of-the-Art often limits itself to a limited number of so-called sensors, which are often crude and simple. Those sensors are used to distill information from facts that can be sensed, either via hardware or software, and usually limited to the identity of agents, location, time, and environment [13]. This limitation is still actual after two decades of research in context-aware computing [10]. What we can observe is that one is often concerned with the integration of (specific types of) sensor data and systems are closed – i.e., the applications are in control what (types of) data is processed. Few context modeling initiatives take into account events, e.g. [15] or activities, e.g. [16]. While vague, [17] did introduce the notion of “application context”, albeit in a network embedded system. This application context is a group of rules that fire messages to applications. This effort is worth mentioning as it is one of the few considering ‘rules’ as part of the model next to using rules for using the model.

Context has also been formalized to capture knowledge that is true in different scenarios’ or so-called “possible worlds” as introduced by [18] and more recently by [19]. Since such efforts are concerned with formalizing “containers” of knowledge and the relationships between these containers, we do not consider those relevant for the related work described later on.

2.2 Beyond the “traditional” notion of context

Looking beyond traditional sensors that could constitute ‘context’, we can look at initiatives describing datasets. Within the scientific community, however, strides have been made to capture aspects of datasets. A data value vocabulary proposed in [20] aims to assess and quantify a dataset’s value for a particular purpose. Their work was informed by an organization’s need to assess their assets’ value. Not only is that need part of the context, but the so-called data value dimensions are also valuable for subsequent data-integration steps. Luzzu [21] allows one to declare data quality metrics to assess RDF datasets, which can be used to rank datasets in a personalized manner. These vocabularies need to be combined with data description vocabularies such as DCAT [22] and VOiD [23] for future interrogation.

We also consider vocabularies for describing activities beyond traditional sensors. The scientific community has looked into formalizing processes and activities for specific projects. In bioinformatics, [24] proposed an ontology for detailing workflows in that domain. The P-PLAN ontology [25], extending PROV-O3 was created to prescribe scientific workflows. [26] proposed an ontology for modeling ontology engineering workflows in the Protégé4 ontology development environment. While we can deem the work of [26] as “domain agnostic”, since one can apply such projects in any domain, the ontology is created for ontology-engineering projects and used to capture the provenance information of each activity, e.g., to inspect its compliance or reproduce steps. More “generic” initiatives for process models also exist. We have, for instance, sBPMN for business process models [27], but their purpose is to provide semantic interoperability for such models. In [28], the authors

1 https://www.w3.org/TR/r2rml, last accessed November 4, 2019.
2 https://www.w3.org/TR/prov-o/ last accessed November 4, 2019. PROV-O a W3C Recommendation for relating activities, entities, agents, and their relationships.
proposed annotating the manipulation and analysis of data in data processing "pipelines" for the creation of meaningful logs—i.e., annotated programs and code. Here, the authors also availed of PROV-O to representing activities, which the authors have used to render data analysis traceable and reproducible. In that sense, they avail of PROV-O to represent and store "sensor data" but do not use it to model the context.

While [17] provided an anecdotal example of grouping knowledge, the State-of-the-Art does not group concepts and relations for specific purposes. This is likely due to the simplicity of the context and the controlled environments of the middleware; there are no two different interpretations of a concept, for instance.

2.3 Related Work on Context-aware Data Integration

We argue that it does not matter what formalism is used to represent the context model, but that formalism needs to be expressive enough to be rendered actionable. The chosen formalism, however, may affect the reasoning capabilities of the overall solution. Rule-based approaches may avail of rule- or inference engines. The HYDRA Middleware Project [17], for instance, relies on the Drools [29] rule engine. Others, such as [15], use the OWL to model context, which allows them to avail of OWL reasoning to infer additional facts. If the chosen formalism has no support for "third-party" reasoning engines, then the "reasoning" needs to be implemented in the solution. It seems, from the aforementioned surveys in different domains (CFR. Section 2.1), that rule languages are more widely adopted. Indeed, OWL reasoners are built for specific reasoning tasks such as classification, inferring subclasses, and satisfiability checking.

One can make a distinction between solutions that integrate context-awareness in the application or the middleware. An example of the former is presented [30] for managing intrusive push notifications on mobile phones. In [17], the authors presented a middleware solution for integrating wireless devices combining a domain ontology with rules. In [31], the authors reported a middleware for a Smart City environment in which stationary sensors are combined with "mobile" sensors (such as a smartphone) to provide more intelligent data services. They achieve this by combining bespoke services with a rule base. As the application domains are specific, the literature does not look into the practice of data integration (and data analysis). Such platforms provide a well-defined and well-scope environment for data curation and integration activities that limit the complexity of data integration.

In [32], the authors present an approach to optimize query-execution in data integration by examining the network and inspecting metadata of the data sources. Their work was published a decade before any of the open description standards emerging. More recent are [33] and [34], who both recognize that formalizing the UoD of context should be kept separate from the systems. In [33], the authors addressed the problem of defining views over relational databases in a context-aware manner. They proposed to model context as so-called "context dimension trees" (CTDs). CTDs have one root node under which one can find context-dimensions nodes. Context dimensions roughly correspond with classes and relations in an ontology. Under these nodes, one must have one or more context-dimension-value nodes that represent specific instances or values for a context dimension. A context dimension value can, themselves, have context dimensions nodes. This allows one to represent complex paths such as: <CD,Medium>→<CDV.Paper>→<CD,Size>→<CD,A4>. Nodes in the tree are mapped onto partial views, which are then used to compute views based on a user's context. The work in [34] aimed to address context management by separating the systems from the context model and proposed the use for a more expressive ontology language—a combination of OWL and rules. While [33] and [34] recognized the importance of modelling context separately, and proposed formalisms to do so, their contributions focused on the (use of) their formalism and not on the knowledge engineering activities required to formalize context.

In [35], the authors proposed a context knowledge-base for integrating data. More specifically, however, the context knowledgebase contains ontology alignments that are used to create SPARQL CONSTRUCT queries to transform triples in one ontology to another. They thus only focused on the transformation of data, and the alignment had no link with the rest of a data-driven project's context. Finally, [36] introduced various types of context (user, application, environment, and "other") for rewriting queries that users might discover more information. In [37], the paper sheds more light on how context is modeled and stored (using relational tables that are tailored for a particular domain—in this case, “software engineering”). We argue that these are anecdotal examples in which the ‘context’ captured is specific to one task (e.g., query optimization or query-rewriting).

2.4 Discussion

Context-aware computing (in general), ‘Context’ in the SoA is mostly concerned with these ‘crude’ and simple sensors, and these are too simplistic to capture the context of data integration in data-driven projects. Outside initiatives concerned with context, there are efforts that have (tried to) formalize(d) aspects relevant to it.

We observe that context-aware platforms built for data analysis are made for specific domains (e.g., healthcare, and smart cities) and siloed in middleware. While this suggests that context-awareness exists, it is only in a limited, non-transferable scope. Those silos propose a controlled environment for data integration, thus reducing the challenges. The context is more involved in data integration activities of data-driven projects and will require various context-aware techniques to drive decision-making processes in those. We thus need a more generic approach for the activities of a data integration process, which are not known beforehand. In conclusion, we have not yet truly explored context-awareness in data integration. There have been

---

anecdotal attempts for specific tasks (e.g., query optimization), and related work in IoT benefited from environments with limited types of data and specific applications/purposes.

3 What is Context in Data Integration?

Data integration is stated to be concerned with “combining data residing at different sources, and providing the user with a unified view of these data” [3]. We refine the definition as “selecting, preprocessing and transforming data from different data sources [in order] to create a unified version of that data for data processing” as to emphasize the various types of decision points of “combining” and leaving both the result of the integration process and whether consumer—human or software—open.

In context-aware data integration, one has to consider: 1) conceptualizing and formalizing the model, and 2) propose agents that use that model for facilitating or automating specific data integration processes. The first is considered an ontology engineering activity. The latter is a development activity, as we have to build agents—in some way, shape, or form—that use the model to achieve certain tasks. Granted, depending on the form, the second can be regarded as a knowledge engineering activity as well. Ideally, however, the context-aware techniques should avail of techniques built on semantic technologies to make them as declarative as possible. We illustrate our approach in the next section.

4 Case-study: Consent-aware Dataset Generation

The problem we want to tackle is to generate datasets that took into account the informed consent of users. By ensuring that such datasets are compliant, we can reduce the overhead of post hoc compliance analysis. Data processing, in general, is increasingly the subject of various regulations, such as the General Data Protection Regulation\(^5\) (GDPR). In GDPR, data subjects give their explicit informed consent for their data to be used for specific data processing purposes. Data subjects also have the right to revoke their consent at any given time. One of the challenges here is to identify what is personal or sensitive data, and which (parts) of an organization’s data sources contains said data. An example of a purpose is to analyze previous purchases to send a list of recommendations (i.e., a recommender system).

Rather than relying on a team of practitioners to ensure that the data integration steps necessary are compliant, we can adopt semantic technologies to facilitate the context and the generation for datasets for the recommender system. In terms of the project, the context includes the data processing purpose (i.e., the purpose of the service) with the goal being to increase sales, GDPR as the regulation (and the organization’s compliance

---

Introducing Context and Context-awareness in Data Integration (with GDPR), and the various data sources used. In terms of disciplines, the context includes data governance and law. Data governance is concerned with data management, roles, and accountability. Important for data governance is knowing where to find information, how it is classified or regarded, and how it should be used. The last touches upon Law and how to conduct activities lawfully. While GDPR is an external “entity” affecting the data-driven project, knowledge of how to interpret, use, and implement that regulation is the discipline. Finally, in terms of the practitioner, the context includes the data engineer or data scientist who needs to compile a dataset for their recommender system.

To solve this problem, we proposed the architecture shown in Fig. 2. Parts of the architecture is based on algorithms published in [41] and [42]. The former is concerned with generating data integration mappings from annotated schemas, and the latter is concerned with “filtering” the resulting data for ensuring compliance. We use green to show the various processes and artifacts for the former, and pink for the latter.

No matter the structure of the resulting dataset (XML, CSV, or a graph), the resulting dataset has a schema. The data governance challenge here is to relate these schemas with references to an organization’s data. Those annotations are used to generate, on the fly, data integration mappings that fetch information from those databases to populate a schema. The generated dataset needs to consider informed consent that the organization has gathered. This means annotating the schemas with their data processing purposes and formalizing parts of GDPR to achieve this.

GDPR is thus the context and meant we had to conceptualize and formalize (i.e., create an ontology) for the following concepts: “data processing purpose”, the “policy” which lists these purposes, and “consent” instances. All these concepts have been described in our consent ontology [42]. The schemas are, in addition to the references to existing sources, annotated with instances of policies and purposes, and instances of the concept Consent contain information on the policy-purpose pairs users have consented to (or withdrawn). The resulting consent information base is then used to filter the generated datasets with the goal to exclude information of those who have not given their consent or whose consent is expired. All processes furthermore generate provenance information, which facilitates the transparency and reproducibility of the pipeline. While we omit the Web Ontology Language (OWL) formalization of the ontology, the important concepts and relations are shown in Fig. 3.

![Diagram](image_url)

**Fig. 2.** The various steps involved in generating a policy-compliant dataset taking into account the informed consent of users.
The “implementation” of the context-aware technique for dataset generation has been entirely written in SPARQL [41] lists the various SPARQL queries for generating executable mappings, and [42] the queries for filtering for compliance. While bespoke code could have been written to build these agents, we believe that the declarative nature of these queries improves transparency for future audits. The listing below illustrates how we use the consent knowledge base to obtain the latest consent information for each user. That information is subsequently used to find all users that have given consent for a particular purpose that has not yet expired. While we cannot provide all queries due to the limited space, we refer the reader to [42].

1. DESCRIBE ?consent WHERE {
3.   { # GET THE LATEST INCLUSION FOR A POLICY
4.     SELECT ?inclusion WHERE {
5.       ?inclusion ont:ofPurpose <.../purpose> .
7.       <.../policy> dcterms:created ?dt .
8.     } ORDER BY DESC(?dt) LIMIT 1 }
11.  # GET THE LATEST CONSENT FOR EACH USER BY FILTERING
12.  # THOSE THAT HAVE BEEN SUCCEEDED BY ANOTHER CONSENT
13.  FILTER NOT EXISTS {
14.     [ ont:forInclusion ?inclusion ;
15.         ont:givenBy ?user ;
16.         ont:registeredOn ?datetime2 ]
17.     FILTER(?datetime2 > ?datetime) }
}

Listing 1: Retrieving consent information for a purpose and policy. All consent information is returned by the DESCRIBE query. The query first looks for the latest inclusion of a service. The second part of the query seeks the latest consent information for each user. For brevity, we omitted prefixes, and use <.../purpose> and <.../policy> for IRI’s of a purpose and a policy.

In this section, we presented a case study on generating datasets that are compliant with the explicit and informed consent given by users. While this section did not contain many technical details, those details have been published and demonstrated in [41] and [42]. The goal of this section was to illustrate how we had to formalize a particular aspect of a data-driven project’s context (GDPR and informed consent) to automatically select and transform the data for a particular purpose. The various SPARQL queries, executed in a specific sequence, thus constitutes our GDPR-aware dataset generation.

5 Conclusions

In this paper, we presented the problem of context-aware data integration. Context-awareness in Computer Science is all too often focused on simple sensors and approaches context as “things” that can be sensed. Related work on context-aware data integration is sparse and often specific and siloed environments such as IoT and Smart Homes. The SotA on context-awareness for data integration in data-driven projects is, in our opinion, not sufficient. Achieving this, however, would open opportunities for facilitating data integration practices in organizations.

To achieve context-aware data integration, one has to formalize context into ontologies and build agents that use these ontologies to make them context-aware. Ideally, those agents are—to the extent possible—written in terms of semantic technologies. We illustrated this approach with a preliminary study on generating GDPR-compliant datasets. The case study uses algorithms published and demonstrated in [41]. In this paper, we integrated both into the narrative for context-aware data integration.

ACKNOWLEDGMENTS

This research was conducted with the financial support of Science Foundation Ireland under Grant Agreement No. #13/RC/2106 at the ADAPT SFI Research Centre at Trinity College Dublin. The ADAPT SFI Centre for Digital Media Technology is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant #13/RC/2106.

REFERENCES
Introducing Context and Context-awareness in Data Integration


[22] F. Maali and J. Erickson, Data Catalog Vocabulary (DCAT) 2014.


