

Bridging Context and Data Warehouses through Ontologies

Okba Barkat, Selma Khouri, Ladjel
Bellatreche
LIAS/ISAE-ENSMA, France
(lastname.firstname)@ensma.fr

Narhimene Boustia
Blida University, Algeria
nboustia@gmail.com

ABSTRACT

Nowadays, we are assisting to three continuously demands from companies: (i) developing analytical applications around Data Warehouse systems (DW) from numerous data sources, (ii) the explicitation the semantic of these sources to reduce heterogeneities and (iii) contextualization of sources. By examining the literature, we identify the existence of several efforts attempting to offer solutions merging these three issues. The merging has been performed partially. To be more concrete, we have identified that the two first demands have been merged. Similarly, the second and the third ones gave raise to contextual ontologies. Unfortunately, all three are not well merged. This paper proposes a comprehensive methodology to design multi-contextual semantic DWs. Our approach consists first in merging context and ontologies and then with DWs. Firstly, a connection between ontologies and context model is built at meta model level. Secondly, a formalization of multi-contextual semantic data warehouse is given, followed by a deep description of the most important steps of the data warehouse design. Finally, a case tool and experiments are conducted using a contextualized hospital ontology to show the effectiveness of our approach.

CCS Concepts

•Information systems → Extraction, transformation and loading;

Keywords

Context, Ontology, Data warehouse, Modeling, ETL

1. INTRODUCTION

The development of context-aware applications is an intense issue that has been challenged in different domains such as smart cities, Internet of things, Machine Learning, Databases and Data Warehousing, etc. In this last field, The ability to take a view on concepts in different ways when put into various contexts is key for

Permission to make digital or hard copies of all or part 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. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC 2017, April 03-07, 2017, Marrakech, Morocco

Copyright 2017 ACM 978-1-4503-4486-9/17/04...\$15.00

DOI: <http://dx.doi.org/10.1145/3019612.3019838>

many Database (DB) and DW users. We can resume Context definitions provided in DB literature as follows "Context is any internal or external information to the application can be used to characterize its data". In this way, context provides a local definition and a *precise semantic* to stored data according to different contextual situations.

In the other hand, assigning *semantic* to data is usually achieved through the use of Ontologies. The conventional notion of domain ontologies describes a given domain in a consensual way from a single vision of the reality described by the ontology. However, many studies have proved that there is not a single and unique representation of this described reality, which allowed the proposition of *contextual ontologies* for managing multiple representations [1]. These studies connected Ontologies to Context management issues in order to define contextual information with precise semantics. An area 'par excellence' where ontologies have widely contributed in developing analytical applications around DW systems. The connection between ontologies and DW definition issues emerged easily where ontologies have been used in different design steps for specifying the used concepts and for eliminating conflicts occurring in a large number of sources with high variety. In DW systems, where information is stored in various forms and scattered around different repositories having their local context definitions for concepts, contextualized information processing has heavy demand for a *semantic layer* on top of the data layer. However, the connection between these three issues has not been fully achieved. The main shortcomings of proposed studies are: the absence of a context model covering all DW resources (concepts, properties and instances) and the lack of a complete DW design approach analyzing the impact of context information during each design phase.

This paper proposes a complete and comprehensive methodology to design multi-contextual semantic DWs. Our first contribution in this study was to provide a context meta model (language) which unifies the definitions provided in Database literature.

This language is then connected to the ontology language in order to provide a contextual definition of the used concepts. A main feature of this connection is that it offers the externalization of the context from the ontology side. This offers several advantages for ontology design, since the designers are free to choose a context language differently from the ontology language.

We then claim that initiatives for designing semantic DW systems should be revisited to incorporate this ontology vision including contextual information. We illustrate in this study that the inclusion of context have an impact on different design phases and mainly on the conceptual design phase, the logical phase and the ETL process

definition.

This paper is organized as follows: section 2 reviews some studies related to context management and ontology-based DW design approaches. Section 3 presents the context language we propose. Section 4 proposes a contextual ontology-based approach for designing DW systems. Section 5 presents a tool and experimentations illustrating the relevance of our approach. Section 6 concludes the paper and sketches some perspectives to this study.

2. RELATED WORK

In this section, we present the main studies related to: (i) the contributions of ontologies in DW systems, (ii) contextual ontologies and (iii) Context management in DW systems.

Ontologies have largely contributed in building DW systems during DW *Exploitation* phase [6] and all *Design* phases including: requirements definition [3], conceptual, logical and physical Design [7, 12] and ETL design [13]. The ontology can reference the used concepts or can be stored in the final data storage system. This architecture is called semantic database (SDB) [7].

In parallel to these studies, an important research area has been advocated to the definition of contextual ontologies, that support the idea that there is no unique way to look at data and to represent them, and manage multiple representations of concepts. Benslimane et al. [1] defines a contextual ontology as an explicit specification of a conceptualization. Context-OWL (C-OWL) is proposed as an extension of OWL¹ language, which is the standard proposed by W3C consortium to define ontologies. This situation faces DW systems to a new challenge where ontologies are not mono contextual but can be multi-contextual. Some studies have provided multi-contextual DW systems, but few attempts tried to related the three issues for defining DW systems using multi-contextual ontologies. The main studies are: *Perez et al.* [10] proposed an approach for building a contextual DW by combining a classical DW with a document warehouse. *Garrigos et al.* [5] proposed an approach for personalizing OLAP schema for each decision maker by taking into account the user characteristics, context, requirements and behavior. *Pitarch et al.* [11] proposed an approach that considers the context for enhancing the flexibility and the expressiveness of hierarchies to overcome generalization problem in DW systems. *Oukid et al.* [9] proposed a contextual text cube model (CXT-Cube) in which each dimension is related to a contextual factor.

A couple of studies used ontologies for defining multi contextual DWs: *Calvanese et al.* [2] proposed an approach for the integration of sources including context information (eg, the currency). In this approach, the target conceptual model and sources models are formalized using \mathcal{DLR} (a description logic language, which can be used to define a specific ontology). The approach focuses on the specification of reconciliation correspondences to be used in order to solve conflicts among sources. *Khoury et al.* [8] proposed a DW design approach using an ontology extended by contextual information. The context concerned functional dependencies between properties only, the evaluation of properties and the definition of contextual concepts was not considered.

We compare between these approaches in table 1 w.r.t the following criteria: (1) *Approach*: it distinguishes between *conventional* and *ontology based* approaches for DW generation. (2) *Model*: it provides details about the *context model* that the approach proposes

¹<https://www.w3.org/TR/owl-features/>

| Ref | Approach | Context Model | Phases | Context Info. | | |
|-----------|--------------|----------------|---------|---------------|-------|------|
| | | | | Def. | Eval. | |
| | | | | | Int. | Ext. |
| [10] | Conventional | Document-based | C | ✓ | | |
| [5] | Conventional | Graphical | C | | | ✓ |
| [11] | Conventional | Graphical | E | | ✓ | |
| [9] | Conventional | Graphical | C | ✓ | | ✓ |
| [2] | Semantic | Key-value | C,L | | | ✓ |
| [8] | Semantic | Ontology-based | L,ETL | | ✓ | |
| Our study | Semantic | Ontology-based | C,L,ETL | ✓ | ✓ | ✓ |

Table 1: Related Work Comparison

or relies on. We distinguish from the literature analysis: Key-Value models (Eg. Time-10h00), tag-based models (Eg. XML models), Graphical Models (Eg. UML or graph models) and Ontology-based models. (3) *DW definition phase*: it identifies the DW definition phases covered by the approach (Conceptual Design (C), Logical Design (L) and ETL design or Exploitation phase (E)). (4) Context information: we distinguish information that incorporate the context in the *definition* of the DW concepts (*Def.*), from information that incorporate context for the *evaluation* of DW properties (*Eval.*). Concerning the evaluation context, it may concern internal contexts (*Inter.*) depending on local concepts information or external (*Ext.*) or surrounding information contexts like Space and Time. Note that in this study, we use an ontology-based model, we defined a general context language which covers different context definition situations (definition and evaluation situations, internal and external) and we cover the main phases of DW design cycle.

3. BACKGROUND: ONTOLOGIES AND SDB

In the Description Logics (DL) formalism, underlying OWL standard language, a knowledge base in DL is composed of two components: the *TBOX* (Terminological Box) stating the *intensional* knowledge and the *ABOX* (Assertion Box) stating the *extensional* knowledge or the instances. In our context, the domain ontology is defined by its TBOX, the instances are stored in the local sources. Based on DL formalism, the TBOX can be formalized as follows $O: \langle \mathcal{C}, \mathcal{R}, \text{Ref}(C), \text{Ref}'(R), \text{Formalism} \rangle$:

- \mathcal{C} : denotes *Concepts* of the model (atomic concepts and concept descriptions).
- \mathcal{R} : denotes *Roles* (or properties) of the model. Roles can be relationships relating concepts to other concepts, or relationships relating concepts to data-values (like Integers, Floats, etc).
- Ref : $C \rightarrow (\text{Operator}, \text{Exp}(C,R))$. Ref is a *function* defining terminological axioms of a DL TBOX. Operators can be inclusion (\sqsubseteq) or equality (\equiv). $\text{Exp}(C,R)$ is an expression over concepts and roles of CM using constructors of description logics such as union, restriction, etc. (e.g., $\text{Ref}(\text{Patient}) \rightarrow (\sqsubseteq, \text{Person} \sqcap (\exists \text{hasSojourn}.\text{Sojourn}))$).
- Ref' : $R \rightarrow (\text{Operator}, \text{Exp}(C,R))$. Ref' is a similar function as Ref , but is defined for roles.
- *Formalism* is the *formalism* followed by the ontology model like OWL-Lite, OWL-DL, RDF, etc.

The ABOX is formed by the set of instances \mathcal{I} . The ontology (usually a relevant par of it) can be stored in a Semantic Database (SDB), which can be formalized as follows: $\langle \mathcal{O}_i, \mathcal{I}, \text{Pop}, \text{SL}_{\mathcal{O}}, \text{SL}_{\mathcal{I}} \rangle$ where: \mathcal{O}_i is the ontology schema $\langle \mathcal{C}, \mathcal{R}, \text{Ref}(C), \text{Ref}'(R), \text{Formalism} \rangle$ of the source S_i , \mathcal{I} represents the instances of the source S_i , Pop : $C \rightarrow 2^{\mathcal{I}}$ is a function that relates each concept to its instances, $\text{SL}_{\mathcal{O}_i}$: is the Storage Layout of the ontology model

and SL_I is the Storage Layout of the instances I , where three main layouts are usually defined: triple, horizontal or binary [7].

4. CONTEXTUAL ONTOLOGIES

Our goal in this study is to provide a complete context language that allows the definition of contextual concepts and contextual roles. The Context language is a domain specific language (DSL) dedicated to the definition of contextual ontological resources. A DSL language is defined by three elements: (i) *Abstract syntax*: it is the language structure based on elements and their relationships (the meta-model). We used the UML diagram class formalism to design our meta-model. (ii) *Concrete syntaxes*: they correspond to specific representations of the design language in order to instantiate its meta-model. (iii) *Semantic*: the meaning of meta-model concepts and how they can be represented on the instantiation.

Our Context model (figure 1) is linked to the ontology meta model, which extends the definition of each ontological resource by its contextual information. By instantiating the Context model, the designer can provide a contextual definition of any resource. In figure 1, the root entity of the model is *Context*. It is characterized with a code: used to give a unique identifier and a name: used to describe the context with a linguistic term. The *Context* entity is divided into two sub-entities that represent our categorization of the context: Definition Context and Evaluation context.

Definition Context. It represents the context in which each *concept* is defined, and is provided by an expression over resources (concepts and roles). The *Statement* entity models such expressions. For example, the concept *Patient* is defined in context c_1 as a person who made at least a sojourn in the hospital, while in context c_2 , a patient is defined as a person who has at least a medical monitoring. $Patient[c_1] = Person \sqcap (\exists \text{hasSojourn.Sojourn})$; $Patient[c_2] = Person \sqcap (\exists \text{hasMonitoring.MedicalMonitoring})$. The entity *Global Definition* of the model represents the global definition in which each concept can be defined, i.e. a definition not referencing a local context, but regrouping all contextual definitions. For example, the global definition of *Patient* regroups its two contextual definitions: $Patient = Patient[c_1] \cup Patient[c_2]$.

Evaluation Context. It represents the context in which *roles* can be evaluated. We define three categories:

- **Local Context:** in which the contextualized roles are evaluated. We differentiate between two sub-categories:
 - *Mathematical Function Context:* the relationship between the concerned role and the context parameters takes the form of a mathematical function. For example, the role *Average Length of Stay (ALOS)* depends on the nature of sojourns: (1) for long stay sojourns, the following formula is used: “ $ALOS = Total\ Inpatient\ Days\ Of\ Care\ (TIDC) \div Total\ Admissions\ (TA)$ ”, (2) for short stay sojourns, the following formula is used: “ $ALOS = Total\ Discharge\ Days\ (TDD) \div Total\ Discharge\ (TD)$ ”.
 - Model.** The entities *Math Function Context*, *Math Formula*, *Context Parameter* and *Context Condition* model this category of context. Taking the example of the *long stay* context: *TIDC* and *TA* represent the context Parameters, (ward = psychiatry) represents the context condition and $(TIDC \div TA)$ represents the Math formula.
 - *Functional Dependency Context:* the context information here is represented by a functional-dependency-like relationship between the concerned roles. For example, the evaluation of the

blood pressure category (Low, Normal or High) functionally depends on the following parameters: *age*, *smoker* and *blood pressure (BP)*.

Model. the entities *Functional Dependency Context*, *Context Parameter* (*age*, *smoker* and *BP*), *Context Condition* (*BPCategory* = Normal) and *Context result* (*BPCategory* = Normal) model this category of context information.

- **Surrounding Context:** is the external context in which roles can be evaluated. We consider two sub categories: (i) Temporal context and (ii) Spatial context which are the most frequent context parameters in literature. They are respectively designed in our model by the entities *Temporal Context* and *Spatial Context*.
- **Unit Of Measure Context:** represents the explicitation of the units of measure including currencies which are used to scale concerned roles. For example, the evaluation of the role *price* depends on the used currency (e.g., Euro, Dollar). The *Unit Of Measure Context* models this category.

In order to strengthen the formal and consensual aspects, some attributes of our model are defined by external domain ontologies: (i) *EngMath*² which covers all aspects of mathematical modeling, (ii) *Time Ontology*³ which formally defines temporal concepts, (iii) *Space Ontology*⁴ which defines a complete vocabulary for spatial concepts and (vi) *Ontology of Units of Measure and Related Concepts*⁵ which formally defines the units of measure and currencies.

Our context model can be linked to any existing domain ontology. This link is materialized by: (i) *has_definition_context* association: which attaches any ontological concept represented by the entity *Concept* to its corresponding definition contexts; (ii) *has_evaluation_context*: which attaches any ontological role represented by the entity *Role* to its corresponding evaluation contexts.

Based on the context language we proposed, we can extend the ontology formalization $O: \langle C, R, Ref(C), Ref'(R), Formalism \rangle$, where *Ref* function does not provide a single definition but can provide context aware definitions: Contextual Global Ontology (CGO): $\langle C, R, Formalism, Context, Cxt_C(C, Cx), Cxt_R(R, Cx) \rangle$, where:

- *Context* is the set of defined contexts Cx_1, \dots, Cx_n ,
- $Cxt_C(C, Cx)$: $C \times Context \rightarrow expression(C, R)$ provides the definition of the concept C in a given context Cx
- $Cxt_R(R, Cx)$: $R \times Context \rightarrow expression(C, R)$ provides, for the role R the details about the given context Cx . Note that *expression(C, R)* can use any DL operator and can be used to evaluate roles using Data vales (included in the set R)

A contextual SDB is also formalized by extending SDB formalization as follows: $\langle Oi_{Context}, \mathcal{I}, Pop, SL_O, SL_I \rangle$, where:

- Pop_i : $C_i \times Context \rightarrow 2^{I_i}$ is a *function* that relates each contextual concept to its instances. Note that *Pop* function will be provided by the ETL algorithm, which integrates data from sources to the target DW schema.

5. PROJECTION OF CONTEXTUAL ONTOLOGIES ON DW PHASES

²<http://www-ksl.stanford.edu/knowledge-sharing/papers/engmath.html>

³<http://www.w3.org/2006/time#>

⁴<http://sweet.jpl.nasa.gov/1.2/space.owl>

⁵<http://www.wurvoc.org/vocabularies/om-1.8/>

CGO and schemas of sources (S). The discovering of mappings can be done using different integration scenarios: (i) *a priori* during the design process, where correspondences between global and local ontologies are defined a priori at the design time of the sources. In such case, the mappings are already available. Designers agree to make efforts when designing the sources in order to get a “free” ETL process when data would need to be merged. (ii) *a posteriori* either manually or automatically using mining algorithms. This issue is related to the domain of contextual ontology matching [4], and is out of the scope of this study. Once the mappings discovered, the integration process resembles to the first scenario.

Before presenting the ETL algorithm, we first discuss its execution according to the defined mappings where two main scenarios can be identified :

(1) The mappings are defined between a concept C in CGO and S according to each defined context of C (cx_1, cx_2, \dots, cx_n) of CGO. In this case, each concept (of CGO) in a given context Cx_j is mapped with the sources. In this case, the ETL process can be executed in order to populate each contextual concept. The global concept can be populated directly from CGO using an ontology reasoner. For example, the concept *Patient* in CGO is defined according to two contexts (*Patient* [cx_1] \sqcup *Patient* [cx_2]) such that: *Patient* [cx_1] = *Person* \sqcap (\exists hasSojourn. Sojourn) and *Patient* [cx_2] = *Person* \sqcap (\exists hasMonitoring. MedicalMonitoring). Assuming that we have two sources S_1 and S_2 , the following mappings can be defined: *Patient* [cx_1]_{CGO} \equiv *Retrieve*(*Person* [S_1] \sqcap (\exists hasSojourn [S_1]. Sojourn [S_1])) \cup *Retrieve*(*Person* [S_2] \sqcap (\exists hasHospitalization [S_2]. Hospitalization [S_2])) and *Patient* [cx_2]_{CGO} \equiv *Retrieve*(*Person* [S_1] \sqcap (\exists hasMonitoring [S_1]. MedicalMonitoring [S_1])) \cup *Retrieve*(*Person* [S_2] \sqcap (\exists hasMonitoring [S_2]. Monitoring [S_2])). These mappings will be used to populate the concepts *Patient* [cx_1] and *Patient* [cx_2]. The instances of the global concept *Patient*_{CGO} can be deduced from the instances of both concepts using a reasoner.

(2) The mappings are defined between a concept C in CGO and S using its global context definition (not according to each defined context). In this case, the ETL process can be executed in order to populate each global concept using its global definition. The instances of contextual concepts can be populated in the case where their definition is provided. For example, *Patient* [cx_1] defined previously can be populated using its definition (*Person* \sqcap \exists hasSojourn.Sojourn) if concepts *Person* and *Sojourn* are populated (using mappings). An ontology reasoner can be used to populate these concepts based on their definition. The ETL algorithm will iterate until all mappings concepts are populated.

These scenarios are applicable either the sources are contextualized or not. In both scenarios, the algorithm relies first on the mappings to populate the concepts. If some mappings are missing (either concerning the global definition or the contextual ones), the reasoner is used to populate concepts when possible. For the concepts that cannot be populated, the target schema will include these informations, the designer can choose either to exclude these concepts from CGO, or to include them and then to manage null values. Also, if the reasoner identifies inconsistent instances, the designer can manage them similarly. Our ETL algorithm considers the two defined context dimensions: concepts and roles. For the roles dimensions, we introduced a new operator *Contextualize* which evaluates the concerned roles according to their corresponding contexts:

- *Math Function Context*: the defined formula is applied for evaluating the role.

- *Functional Dependency Context*: the concerned role is evaluated according to the defined FD. Example: for the context *NormalBP*, the role *BPCategory* takes the value *Normal*.

- *Time Context* or *Space Context*: the concerned role is evaluated according to defined parameters. Example, for the context *Tariff 2013*, the role *Price* takes the defined value 25.

- *Units Of Measure Context*: the concerned role is evaluated by applying the defined conversion function.

The formalization of the ETL algorithm is presented in the link: <https://sites.google.com/site/conticdw/>, it contains the following steps:

- For each mapping assertion, the ETL process **extracts** instances from the source concepts in a data staging area (temporary location).
- A set of **transformations** are achieved. Some transformations are explicitly defined in the mappings (constraints of conversion, aggregation or filtering) and additional transformations can be required by the context. For these ones, the algorithm evaluates the concerned roles according to their contexts (the operator *Contextualize*).
- Finally, the transformed data are **retrieved** and **stored** in the target system using respectively the operators *Retrieve* and *Store*.

In order to ensure the storage of contextual data, *Extract*, *Retrieve* and *Store* are redefined as follows: Assuming that (in figure 1), *Context*(C) defines the instances of the class Context linked to the class C using *has_evaluation_context* role: (i) **Extract**_{context}(S,C) = *Join* (*Extract* (S,C), *Context*(C), *has_evaluation_context*) : this new operator extracts the instances of the concept C as well as the instances of its corresponding contexts (context code and name). (ii) **Retrieve**_{context}(S,C) = *Join* (*Retrieve*(S,C), *Context*(C), *has_evaluation_context*): similarly, this new operator retrieves the instances of the concept C and those of its corresponding contexts. (iii) **Store**_{context}(S,C,I) = *Store*(S,C,I) \cup *Store*(S, *Context*(C), *Pop*(*Context*(C))) : this new operator stores the instances of the concept C and those of its corresponding contexts.

6. EXPERIMENTATION AND TOOL

In order to illustrate the feasibility and relevance of our proposal, we have conducted experiments and implemented the approach in a case tool for CONTEXTual semantic Data Warehouse Design (CONTIC-DW tool). CONTIC-DW is implemented using: (1) *Java language* using *Eclipse* development environment. (2) *Jena API* that provides a collection of tools and Java libraries for the development of Semantic Web applications. (3) *GraphViz API* that provides graph visualization tools. Our tool is composed of five modules that correspond to the different construction steps of the contextual semantic DW, for the identification of data sources, identification of the Global Ontology, Requirements Definition and multi-dimensional model Generation, Mappings definition and the ETL process. The demonstration video is available at: <https://youtu.be/vAbuAqHWn1E>.

We used the following experimentation scenario: it consists of three locally populated semantic sources (*Hospital1*, *Hospital2* and *Hospital3*) referencing a shared ontology (*Healthcare Ontology* presented in [8]). We assume that the same vocabulary is imposed for all hospitals by a director organism (for example the department of health) which needs to perform some analysis using a set of performance indicators (*ALOS*, *Bed Utilisation Rate* (*BUR*), *Turnover Interval* (*TOI*), etc ...) concerning the different hospitals. For this purpose, the input data sources need to be integrated into a DW

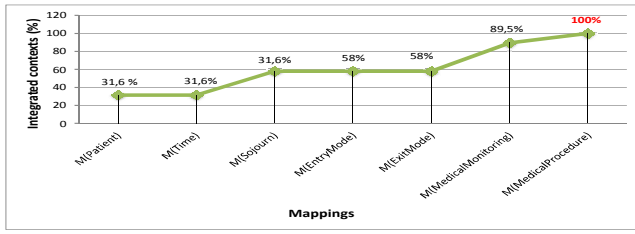


Figure 2: Results of the ETL process: Integrated contexts

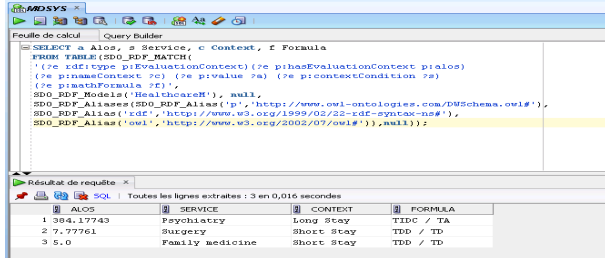


Figure 3: OLAP queries: ALOS per service

system. Our data sources are defined as fragments of *Healthcare* ontology using simple mappings. Our study case encounters the two types of context situations: Definition contexts and Evaluation contexts. In total, 23 contexts have been defined in each source.

The result of the design process is a DW whose schema corresponds to the projection of the set of requirements on the *Healthcare* ontology and is populated by instances selected from our contextualized data sources. Figure 2 shows the percentage of integrated contexts based on defined mappings after the execution of the ETL process. We note that each algorithm iteration corresponds to the population of a DW class according to a mapping (GaV approach). The figure illustrates that contextual resources have been fully integrated in the last iteration. Once the DW schema is populated, it can be loaded into a semantic DBMS. We used Oracle semantic database. This will provide a contextual SDBs, and following our previous formalization: $\langle \mathcal{O}_{Context}, \mathcal{I}, \mathcal{Pop}, \mathcal{SLO}, \mathcal{SLI} \rangle$, $\mathcal{O}_{Context}$ is the conceptual DW schema defined, \mathcal{I} is the set of instances loaded, \mathcal{Pop} is defined by the ETL process execution, \mathcal{SLO} and \mathcal{SLI} in Oracle is the triple storage layout. On this contextual DW, contextual analysis can be performed using OLAP queries. For example, figure 3 illustrates the *SPARQL* query corresponding to the OLAP query: *finding ALOS values per service*, and shows its results as executed in Oracle. The query is executed on the resulted schema. The obtained values of ALOS are calculated using the right formula depending on contexts: (*Long Stay* context for *Psychiatry* service and *Short Stay* context for *Surgery* and *Family medicine* services). The obtained values of ALOS reveals that our ETL algorithm has used, depending on context, the right formula to calculate it.

7. CONCLUSION

In this paper, we have proposed a comprehensive methodology to design multi-contextual semantic data warehouses. Our proposal is motivated by the existence of only pairwise connection between context, data warehouses and ontologies. As a consequence, we proposed a cyclic connection between them. Another important issue of this paper is externalization of the context from the ontology

side. This simplifies the design of contextual applications and may significantly reduce the design complexity, since the designers are free to choose a context language that may be different from the ontology language. To do so, we proposed a context model covering the main context situations and we connected it to the ontology model to explicit its concepts and properties. The main steps of DW design are then revisited by considering our external contextual ontologies. Our methodology is validated through a case study tool, called *CONTIC-DW*, by considering a contextualized hospital ontology. Currently, we are working on hardness study of externalization of the context from ontologies, to evaluate its design and reasoning complexity. Very large ontologies are considered to evaluate the scalability of our proposal.

8. REFERENCES

- [1] D. Benslimane, A. Arara, G. Falquet, Z. Maamar, P. Thiran, and F. Gargouri. Contextual ontologies. In *ADBIS*, pages 168–176, 2006.
- [2] D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. Data integration in data warehousing. *International Journal of Cooperative Information Systems*, 10(03):237–271, 2001.
- [3] Z. Djilani and S. Khouri. Understanding user requirements iceberg: Semantic based approach. In *MEDI Conference, Springer*, pages 297–310, 2015.
- [4] J. Euzenat, J. David, A. Locoro, and A. Inants. *Context-based ontology matching and data interlinking*. PhD thesis, Lindicle, 2015.
- [5] I. Garrigós, J. Pardillo, J.-N. Mazón, and J. Trujillo. A conceptual modeling approach for olap personalization. In *ER*, pages 401–414. Springer, 2009.
- [6] B. Kämpgen, S. O’Riain, and A. Harth. Interacting with statistical linked data via olap operations. In *Extended Semantic Web Conference*, pages 87–101. Springer, 2012.
- [7] S. Khouri, I. Boukhari, L. Bellatreche, S. Jean, E. Sardet, and M. Baron. Ontology-based structured web data warehouses for sustainable interoperability: requirement modeling, design methodology and tool. *Computers in Industry*, pages 799–812, 2012.
- [8] S. Khouri, L. El Saraj, L. Bellatreche, B. Espinasse, N. Berkani, S. Rodier, and T. Libourel. Cidhouse: contextual semantic data warehouses. In *DEXA*, pages 458–465. Springer, 2013.
- [9] L. Oukid, O. Asfari, F. Bentayeb, N. Benblidia, and O. Boussaid. Cxt-cube: contextual text cube model and aggregation operator for text olap. In *DOLAP*, pages 27–32. ACM, 2013.
- [10] J. M. Pérez, R. Berlanga, M. J. Aramburu, and T. B. Pedersen. A relevance-extended multi-dimensional model for a data warehouse contextualized with documents. In *DOLAP*, pages 19–28. ACM, 2005.
- [11] Y. Pitarch, C. Favre, A. Laurent, and P. Poncelet. Enhancing flexibility and expressivity of contextual hierarchies. In *IEEE ICFS*, pages 1–8, 2012.
- [12] O. Romero, D. Calvanese, A. Abelló, and M. Rodríguez-Muro. Discovering functional dependencies for multidimensional design. In *DOLAP*, pages 1–8, 2009.
- [13] D. Skoutas and A. Simitis. Ontology-based conceptual design of etl processes for both structured and semi-structured data. *IJSWIS*, 3(4):1–24, 2007.