

# Towards Distributed Contextualized Knowledge Repositories for Analysis of Large-Scale Knowledge Graphs

Loris Bozzato<sup>1</sup> and Christoph G. Schuetz<sup>2</sup>

<sup>1</sup> Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy

<sup>2</sup> Johannes Kepler University Linz, Altenberger Str. 69, 4040 Linz, Austria

bozzato@fbk.eu, christoph.schuetz@jku.at

**Abstract.** A knowledge graph (KG) represents real-world entities as well as their properties and relationships in a structured and often logic-based formalism. Given the large amount of information and the diversity of data stored in KGs, operations for analysis of such data akin to traditional OLAP operations are useful to understand the contents of KGs along different dimensions. In this direction, we recently proposed Knowledge Graph OLAP (KG-OLAP), a framework based on contextualized description logics that allows to organize knowledge graphs in a multi-dimensional structure – a KG-OLAP cube. For KG-OLAP cubes, we defined operations for combination of knowledge from different cells and for abstraction of knowledge within cells. Experiments with a proof-of-concept prototype, however, revealed that the management of a centralized KG-OLAP cube is impractical for large KGs. In this paper, we extend KG-OLAP in order to formalize the case in which knowledge is distributed across different repositories. We hence formalize a distributed version of the multidimensional cube structure, and we show how the operations can be adapted to this scenario.

## 1 Introduction

A *knowledge graph* (KG) represents real-world entities and their properties as well as relationships between entities in a structured and often logic-based formalism KGs [8]. In recent years, many organizations aimed for the development of KGs for practical purposes, necessitating the development of appropriate *Knowledge Graph Management Systems (KGMS)* to leverage KGs. In particular, given the diversity of data stored in KGs, operations for analysis of such data akin to traditional OLAP operations can be useful to understand the contents of graphs and abstract their information along different dimensions (see e.g., [1,7]).

In this direction, we recently proposed *Knowledge Graph OLAP (KG-OLAP)* [9], a framework based on contextualized description logics that allows to organize KGs in a multi-dimensional structure - a *KG-OLAP cube*. The main feature of the KG-OLAP model, in contrast to traditional OLAP, where numerical measures are aggregated, is that the cube cells in a KG-OLAP cube comprise knowledge facts. In this regard, each cell is identified by the combination of different *dimensions* (e.g., *time*, *location*, *importance*) with the members being hierarchically organized objects (e.g., *7-july-20* <

---

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

*july-2020, austria  $\prec$  europe*). The multi-dimensional model of KG-OLAP cubes is an adaptation of the *Contextualized Knowledge Repository (CKR)* framework [11,3], a formal approach for representing and reasoning with contexts in knowledge graphs based on a description logic. The multi-dimensional structure of KG-OLAP cubes allows us to formulate OLAP-like operations for the analysis of a cube’s contents: We defined sets of operations for the combination of knowledge from different cells (*contextual operations*) and for the abstraction of knowledge within cells (*graph operations*). The KG-OLAP model and operations were implemented in a proof-of-concept prototype and evaluated with respect to the real-world scenario of *Air Traffic Management (ATM)* knowledge graphs [10].

A limitation of the model, especially when applied to large-scale KGs, lies in the centralized view of the cube: in practical cases (like in the ATM scenario [10]), the information may be distributed across different repositories and only part of it should be integrated when applying analysis operations. Thus, the centralized view of KG-OLAP might not be applicable to represent cases where single “nodes” in a distributed model cannot have the entire vision of the knowledge and structure of the cube. Moreover, experiments with the proof-of-concept prototype of KG-OLAP framework revealed that the management of operations over a centralized cube is impractical for KG-OLAP cubes with a large amount of data contents or a large number of cells.

In response to these needs, in this paper we introduce our ongoing work to extend the KG-OLAP model towards a distributed version of the framework, with the aim of managing scenarios where knowledge from the model is partitioned in different repositories. The benefit of this decentralized view is not only to be able to model the integration and analysis of a distributed KG, but also to allow for larger contextualized KGs to be divided into smaller units that can be managed by different processors. Thus, while in this paper we focus on the formalization of the modeling and data analysis in a distributed KG-OLAP model, we envision that such model can be useful also in implementations in order to increase the performance of reasoning (e.g., by instance-level materialization as in [9]) and analysis over large-scale KGs.

We present the distributed KG-OLAP model in Section 2. We define the syntax and semantics of the distributed model by extending the KG-OLAP model presented in [9]. The proposed model can be seen as a formal specialization in the context of KGs of the multi-dimensional model presented in [10] for the management of ATM information. In Section 3 we adapt the contextual OLAP operations presented for single KG-OLAP cubes to the distributed model, considering operations over the node structure and the local organization of cells.

## 2 Distributed KG-OLAP Framework

As in the original KG-OLAP model [9], we build our extended KG-OLAP model on definitions of the CKR framework [5,3] for a generic description logic language [2]. Intuitively, in our model we distribute the information of a KG-OLAP cube to different *nodes*. Nodes are organized in a hierarchical structure (akin to the coverage structure of contextual dimensions). Each node will contain a partial view of the KG-OLAP cube (representing its local information, e.g., in ATM, for a specific airport or geographic zone) and nodes from lower levels will inherit knowledge from the higher levels.

The multidimensional structure is expressed using a DL signature  $\Omega$ , that we call *cube vocabulary*, composed of the mutually disjoint sets  $\text{NR}_\Omega$  of atomic roles,  $\text{NC}_\Omega$  of atomic concepts, and  $\text{NI}_\Omega$  of individual names. The vocabulary further specifies a set  $\mathbf{F} \subseteq \text{NI}_\Omega$  of *cell names*, a set  $\mathbf{D} \subseteq \text{NR}_\Omega$  of *dimensions*, a set  $\mathbf{I} \subseteq \text{NI}_\Omega$  of *dimension members*, and for every dimension  $E \in \mathbf{D}$ , a set  $\text{D}_E \subseteq \mathbf{I}$  of dimension members of  $E$  (cf. dimensional structure in [11]). The *cube language*  $\mathcal{L}_\Omega$  is a DL language over cube vocabulary  $\Omega$ . For every dimension  $A \in \mathbf{D}$ , we define the role  $\prec_A$  (the *dimensional ordering* for  $A$ ) as a strict partial order relation over dimension members  $\text{D}_A$ . In the following, we may also use non-strict orderings  $\preceq_A$  and direct successor relations  $\dot{\prec}_A$ .

In order to represent the points of the distributed structure of repositories, we consider a set of *nodes*  $\text{S} \subseteq \text{NI}_\Omega$ . Similarly to dimensions, we consider a role  $\prec_S$  which defines a partial order over the nodes of  $\text{S}$  and describes the (hierarchical) relationships across repositories. In the rest of the paper, we assume that each dimension and the node structure  $\text{S}$  are ordered in a simple hierarchy (in particular supporting a notion of level, see e.g., *ranked hierarchies* in [6,4]). We thus consider a set  $\mathbf{L} \subseteq \text{NI}_\Omega$  of *levels* and a relation *lev* associating each dimensional value to its level as defined in [9].

The notion of dimensional vector provides a local identifier for cells inside a given repository. Let  $|\mathbf{D}| = k$ , we define a *dimensional vector* as the set  $\mathbf{d} = \{A_1 := d_1, \dots, A_k := d_k\}$  where every  $d_j$  is in the set of dimension members  $\text{D}_{A_j}$  of  $A_j$ , with  $j \in \{1, \dots, k\}$ . We refer to the set of all dimensional vectors of cube vocabulary  $\Omega$  as the *multidimensional space*  $\mathfrak{D}_\Omega$ .

The distribution of the cube into different nodes affects the way in which we identify and order cells inside the overall structure: intuitively, these need to depend both on the node and local dimensional coordinates. Given a node and a dimensional vector, we associate with such coordinates a *cell name* using the function  $\text{cn} : \text{S} \times \mathfrak{D}_\Omega \rightarrow \mathbf{F}$ . We require  $\text{cn}$  to be bijective, i.e., each cell name  $c$  is associated with a couple  $(s, \mathbf{d})$  identifying the repository  $s$  and the point  $\mathbf{d}$  in the dimensional space; conversely,  $c$  can be interpreted as the unique identifier of the corresponding dimensional point  $\mathbf{d}$  in node  $s$ . Cells coverage now depends both on the ordering of nodes and the coverage of dimensional vectors. Let  $\mathbf{d}, \mathbf{e} \in \mathfrak{D}_\Omega$ , we say that  $\mathbf{d} \preceq \mathbf{e}$  iff  $d_A \preceq e_A$  for each  $A \in \mathbf{D}$ . In the case of cells, given  $c_1, c_2 \in \mathbf{F}$ , we say that  $c_2$  *covers*  $c_1$  and we write  $c_1 \preceq c_2$  iff  $\text{cn}(s_1, \mathbf{d}) = c_1$  and  $\text{cn}(s_2, \mathbf{e}) = c_2$ ,  $s_1 \preceq_S s_2$  and for every  $A \in \mathbf{D}$ ,  $d_A \preceq e_A$ .

Knowledge represented inside cells is expressed in a DL language  $\mathcal{L}_\Sigma$ , called the *object language*, which is based on a DL object vocabulary  $\Sigma = \text{NC}_\Sigma \uplus \text{NR}_\Sigma \uplus \text{NI}_\Sigma$ .

The distributed KG-OLAP model can be represented by a distributed extension of the CKR framework [3,5]. The contextual structure of a CKR is expressed by a *meta-vocabulary*  $\Gamma$ , a DL vocabulary that contains a set of *context names*  $\mathbf{N} \subseteq \text{NI}_\Gamma$ , a set of *module names*  $\mathbf{M} \subseteq \text{NI}_\Gamma$ , a set of *context classes*  $\mathbf{C} \subseteq \text{NC}_\Gamma$  including class  $\text{Ctx}$ , a set of *contextual relations*  $\mathbf{R} \subseteq \text{NR}_\Gamma$ , a set of *contextual attributes*  $\mathbf{A} \subseteq \text{NR}_\Gamma$ , and for every attribute  $A \in \mathbf{A}$ , a set  $\text{D}_A \subseteq \text{NI}_\Gamma$  of *attribute values* of  $A$ . The role *mod* defined over  $\mathbf{N} \times \mathbf{M}$  expresses associations between contexts and modules. Modules represent pieces of knowledge specific to a context or context class; attributes describe contextual properties (e.g., time, location, provenance) identifying a context (or class); the context class  $\text{Ctx}$  defines the class of all contexts. It is then easy to relate the KG-OLAP cube language  $\mathcal{L}_\Omega$  to the CKR meta-language  $\mathcal{L}_\Gamma$ : we have that  $\mathbf{F} \subseteq \mathbf{N}$  (i.e. cells are a kind of context),  $\mathbf{D} \subseteq \mathbf{A}$  (i.e. dimensions are a kind of contextual attributes) and cell coverage is a partial order relation in  $\mathbf{R}$ .

We can define as follows the extension of the CKR with distribution on nodes.

**Definition 1 (distributed Contextualized Knowledge Repository).** A distributed Contextualized Knowledge Repository (dCKR) over  $\langle \Gamma, \Sigma \rangle$  is a family  $\mathfrak{K}_S = \{\mathfrak{K}_s\}_{s \in S}$  where each  $\mathfrak{K}_s$  is a structure  $\mathfrak{K}_s = \langle B_s, \mathfrak{G}_s, \mathbf{K}_{\mathbf{M}_s} \rangle$  such that:

- $B_s$  is a set of direct ordering relations relative to node  $s$  of the kind  $z \prec_{Ss}$  or  $s \prec_{Sz}$ ;
- $\mathfrak{G}_s$  is a DL knowledge base over  $\mathcal{L}_\Gamma \cup \mathcal{L}_\Sigma$ ;
- $\mathbf{K}_{\mathbf{M}_s} = \{\mathbf{K}_m\}_{m \in \mathbf{M}_s}$  where every  $\mathbf{K}_m$  is a DL knowledge base over  $\mathcal{L}_\Sigma$ , for each module name  $m \in \mathbf{M}_s \subseteq \mathbf{M}$ .

Intuitively, this definition extends the definition of local CKRs with a set  $B_s$ , containing the direct relations of the node with other higher or lower nodes. In the following we call  $\mathfrak{K}_s$  a *distributed KG-OLAP cube* (or simply *distributed cube*) if its metaknowledge is based on a cube language  $\mathcal{L}_\Omega$ , following the above relations.

A dCKR interpretation is composed by a set of CKR interpretations for each node of the distributed structure; in turn, each CKR interpretation, has a DL interpretation for the global context and a DL interpretation for every local context.

**Definition 2 (dCKR interpretation).** A dCKR interpretation  $\mathfrak{I}_S$  for  $\langle \Gamma, \Sigma \rangle$  is a family  $\{\mathfrak{I}_s\}_{s \in S}$  where each  $\mathfrak{I}_s$  is a structure  $\mathfrak{I}_s = \langle \mathcal{M}_s, \mathcal{I}_s \rangle$  such that: (i)  $\mathcal{M}_s = \langle \Delta^{\mathcal{M}_s}, \cdot^{\mathcal{M}_s} \rangle$  is a DL interpretation of  $\Gamma \cup \Sigma$ ; (ii) for every  $x \in \text{Ctx}^{\mathcal{M}_s}$ ,  $\mathcal{I}_s(x) = \langle \Delta^{\mathcal{I}_s(x)}, \cdot^{\mathcal{I}_s(x)} \rangle$  is a DL interpretation over  $\Sigma$ . Moreover, the following conditions hold:

- (i). for  $s \in S$ , for every  $c \in \mathbf{N}$ ,  $c^{\mathcal{M}_s} \in \text{Ctx}^{\mathcal{M}_s}$  and, for every  $C \in \mathbf{C}$ ,  $C^{\mathcal{M}_s} \subseteq \text{Ctx}^{\mathcal{M}_s}$ ;
- (ii). for  $s, z \in S$ ,  $\Delta^{\mathcal{M}_s} = \Delta^{\mathcal{M}_z}$ ; moreover, for every  $x \in \text{Ctx}^{\mathcal{M}_s}$ ,  $\Delta^{\mathcal{I}_s(x)} = \Delta^{\mathcal{M}_s}$ ;
- (iii). for  $s, z \in S$  and  $a \in \text{NI}_\Sigma$ ,  $a^{\mathcal{M}_s} = a^{\mathcal{M}_z}$ ; moreover, for every  $x \in \text{Ctx}^{\mathcal{M}_s}$ ,  $a^{\mathcal{I}_s(x)} = a^{\mathcal{M}_s}$ .

The conditions in this definition ensure that the interpretation of domains is consistent across contexts and nodes. The interpretation of ordinary DL expressions in each DL interpretation is defined, as in CKR, by the language of choice for the object language (e.g., *SRQIQ-RL* in [3]).

Given a distributed cube  $\mathfrak{K}_S$ , let us consider the set  $B_S$  of all bridge conditions defined as  $B_S = \bigcup_{s \in S} B_s$ . We consider  $B_S^*$  to be the closure of  $B_S$  with respect to the node ordering relation. We then extend as follows the definition of KG-OLAP model [9] with new conditions for the intended interpretation of the node structure.

**Definition 3 (distributed KG-OLAP cube model).** A dCKR interpretation  $\mathfrak{I}_S$  is a distributed KG-OLAP cube model of  $\mathfrak{K}_S$  iff the following conditions hold:

- **Local conditions:** for each  $s \in S$ 
  1. for  $\alpha \in \mathcal{L}_\Sigma \cup \mathcal{L}_\Gamma$  in  $\mathfrak{G}_s$ ,  $\mathcal{M}_s \models \alpha$ ;
  2. for  $\langle x, y \rangle \in \text{mod}^{\mathcal{M}_s}$  with  $y = m^{\mathcal{M}_s}$ ,  $\mathcal{I}_s(x) \models \mathbf{K}_m$ ;
  3. for  $\alpha \in \mathfrak{G}_s \cap \mathcal{L}_\Sigma$  and  $x \in \text{Ctx}^{\mathcal{M}_s}$ ,  $\mathcal{I}_s(x) \models \alpha$ .
  4. if  $c_1, c_2 \in \mathbf{F}$ , and for every  $A \in \mathbf{D}$  with  $d \in D_A$ ,  $\mathcal{M}_s \models A(c_1, d)$  and  $\mathcal{M}_s \models A(c_2, d)$  then  $c_1 = c_2$ .
  5. for  $\mathbf{d} \in \mathcal{D}_\Omega$  and  $\text{cn}(s, \mathbf{d}) = c \in \mathbf{F}$ , then  $\mathcal{M}_s \models A(c, d_A)$  for each  $A \in \mathbf{D}$  with  $d_A \in D_A$ .
- **Global conditions:**

6. if  $s \prec_S z \in B_S^*$ , then for each  $\alpha \in \mathcal{L}_S \cup \mathcal{L}_T$  in  $\mathfrak{G}_z$ ,  $\mathcal{M}_s \models \alpha$ ;
7. given  $c_1, c_2 \in \mathbf{F}$  with  $\text{cn}(s, \mathbf{d}) = c_1$  and  $\text{cn}(z, \mathbf{e}) = c_2$  where  $s \preceq_S z \in B_S^*$ , if  $\mathcal{M}_s \models c_1 \preceq c_2$  and  $\mathcal{M}_s \models \text{mod}(c_2, m)$  with  $m \in \mathbf{M}$ , then  $\mathcal{M}_s \models \text{mod}(c_1, m)$ ;

Intuitively, global conditions require that knowledge from the higher nodes is propagated to nodes at the lower levels: if  $s \prec_S z$ , condition (6) requires that axioms of the global context  $\mathfrak{G}_z$  are propagated to  $\mathfrak{G}_s$ ; condition (7) ensures that knowledge modules associated to a higher cell  $c_2$  (inside the same or higher node) are also associated to cell  $c_1$  in lower levels of the multi-dimensional structure.

Given  $\mathfrak{R}_S$  and  $c \in \mathbf{N}$  with  $\text{cn}(s, \mathbf{d}) = c$ , an axiom  $\alpha \in \mathcal{L}_S$  is *c-entailed* by  $\mathfrak{R}_S$  (denoted  $\mathfrak{R}_S \models c : \alpha$ ) if  $\mathcal{I}_s(c^{\mathcal{M}_s}) \models \alpha$  for every model  $\mathcal{I}_s$  of  $\mathfrak{R}_S$ . We say that an axiom  $\alpha$  is *globally entailed* by  $\mathfrak{R}_S$  (denoted  $\mathfrak{R}_S \models \alpha$ ) if: (i)  $\alpha \in \mathcal{L}_S$  and  $\mathfrak{R}_S \models c : \alpha$  for every  $c \in \mathbf{N}$ , or (ii)  $\alpha \in \mathcal{L}_T$  and  $\mathcal{M}_s \models \alpha$  for every cube model  $\mathcal{I}_s$  in  $\mathcal{I}_S$  of  $\mathfrak{R}_S$ .

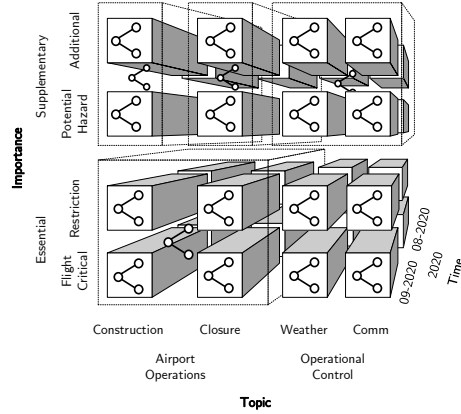
*Example 1.* We demonstrate the idea of distributed KG-OLAP in the Air Traffic Management scenario. Figure 1 shows a “monolithic” KG-OLAP cube for the representation of a contextualized KG in ATM: each cell contains the information relative to a specific context, e.g., describing Flight Critical knowledge about Construction in 08-2020. The dimensions are hierarchically ordered, e.g., Flight Critical and Restriction are part of the Essential package of knowledge. Cells exist at multiple granularities within the cube, e.g., the KG-OLAP cube contains Additional knowledge about Weather in 08-2020 but also Supplementary knowledge about Operational Control in 2020.

Figure 2 illustrates the distribution of such KG-OLAP cube for ATM in multiple nodes. Instead of a single monolithic KG-OLAP cube for the entire relevant knowledge, the knowledge is distributed across nodes for the knowledge of the Europe area, the LOVV (Austria) flight information region and the LOWW (Vienna) airport. The presented framework also allows for the redundant allocation of knowledge on different nodes, e.g., LOVV and LOVV'. Knowledge propagates, on the one hand, from more general to more specific cells. For example, the knowledge from the Essential-Closure-2020 context propagates to the Restriction-Closure-2020 context. On the other hand, knowledge also propagates from more general to more specific nodes, e.g., the knowledge from the Essential-Closure-2020 cell at the Europe node propagate to the Essential-Closure-2020 cell at the LOWW node.  $\diamond$

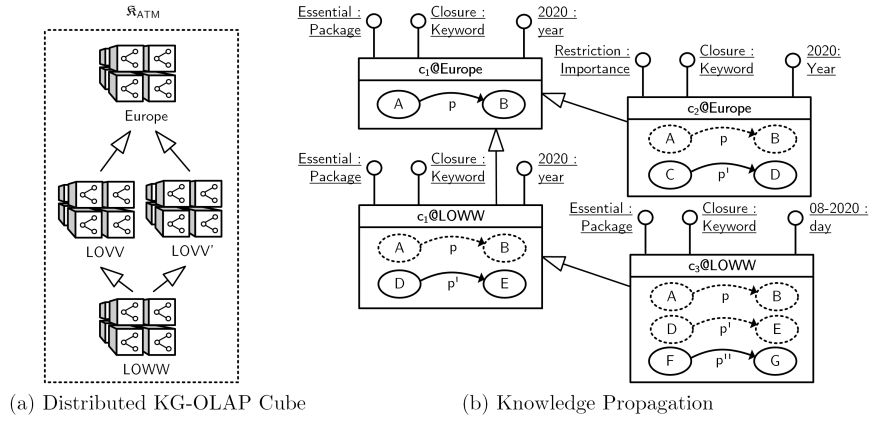
### 3 Query operations for distributed KG-OLAP cubes

In this section we show how OLAP query operations can be defined for analyzing the contents of distributed KG-OLAP cubes. In the following we only consider *contextual operations*, namely operations that manipulate the structure of distributed cubes, while *graph operations*, that is operations that modify the graph inside modules of a cell (and thus do not depend on the organization of cells), can be defined similarly to the original formulation in [9].

**Slice and dice.** The slice and dice operations allows for the selection of a set of facts (i.e. a “slice” of the distributed cube structure) with specific coordinates for subsequent manipulation. In the case of distributed cubes, we can now consider two distinct operations for the selection of “slices” of the multi-dimensional structure provided by the node structure and dimensional space.



**Fig. 1.** An individual three-dimensional KG-OLAP cube with the hierarchically ordered dimensions Importance, Topic, and Time and cells at multiple levels of granularity



**Fig. 2.** An example of (a) the distributed KG-OLAP cube  $\mathfrak{R}_{ATM}$ , which corresponds to the federation of the individual cubes Europe, LOVV, LOVV', and LOWW; (b) propagation of knowledge along the hierarchy of nodes and the hierarchy of cells. The arrows represent the partial order between nodes and contexts, respectively, e.g.,  $LOVV \preceq_{ATM} Europe$  and  $c_2 \preceq c_1$ .

A first operation, the *node slice and dice*, operates on the node structure and restricts the structure to local cubes related to an input node.

**Definition 4 (Node slice and dice).** Given a distributed cube  $\mathfrak{R}_S = \{\mathfrak{R}_s\}_{s \in S}$  and an element  $z \in S$ , we define the node slice and dice operation  $\delta(\mathfrak{R}_S, z)$  as a new distributed cube  $\mathfrak{R}_{S'} = \{\mathfrak{R}_s\}_{s \in S'}$  such that  $S' = \{s \in S \mid s \preceq_S z \text{ or } z \preceq_S s \in B_S^*\}$ .

The selection of cells inside a local cube (once the node has been fixed), defining a *local slice and dice*, can be determined by the values of their dimensions, analogously to the slice operation in [9]. Let us define the set of cells relative to a node  $s \in S$  as  $F_s = \{c \in \mathbf{F} \mid cn^-(c) = (s, e)\}$ .

**Definition 5 (Local slice and dice).** Given a cube  $\mathfrak{K}_s = \langle B_s, \mathfrak{G}_s, K_{M_s} \rangle$  and a dimensional vector  $\bar{\mathbf{d}}$  which defines the dice coordinates, we define the local slice and dice operation  $\delta(\mathfrak{K}_s, \bar{\mathbf{d}})$  as a new cube  $\mathfrak{K}'_s = \langle B_s, \mathfrak{G}'_s, K_{M_s} \rangle$  over  $\langle \Gamma', \Sigma \rangle$ , such that:

- For each  $A \in \mathbf{D}$ ,  $D'_A = \{e \in D_A \mid e \preceq d_A \text{ or } d_A \preceq e, \text{ with } d_A \in \bar{\mathbf{d}}\}$ ;
- $\mathbf{F}'_s = \{c \in \mathbf{F}_s \mid \text{cn}^-(c) = (s, \mathbf{e}), \text{ for each } e_A \in \mathbf{e}, e_A \in D'_A\}$  and other components of  $\Gamma'$  are defined analogously to  $\Gamma$ ;
- $\mathfrak{G}' = \mathfrak{G}_\Sigma \cup \mathfrak{G}_{\Gamma'}$  (i.e., metaknowledge in  $\mathfrak{G}'$  is equal to the formulas in  $\mathfrak{G}_\Gamma$  that have only symbols in  $\Gamma'$ ).

Note that while this dimension based slice is defined on a single node, it can be easily extended to all the cubes in  $\mathfrak{K}_S$  if the operation is applied over each  $s \in S$ .

**Merge.** Similarly, we can provide operations for merging information considering the whole distribution structure and local cubes. The merge (or roll-up) operations combine knowledge at a certain granularity by merging knowledge from a lower granularity. In the case of distributed cubes, we can distinguish a *node merge* operation, that adds to a given node the combination of knowledge coming from its more specific nodes, and a *local merge*, which adds to cells of a given “dimensional level”  $l$  the roll-up of knowledge from cells in lower levels (inside a local cube).

Similarly to [9], we can parametrize both operations with respect to the method *met* that is used to combine the rolled-up knowledge. In the following definitions we consider the cases in which *met* can be the union ( $\cup$ ) or intersection ( $\cap$ ) of such knowledge.

**Definition 6 (Node merge).** Given a distributed cube  $\mathfrak{K}_S = \{\mathfrak{K}_s\}_{s \in S}$  and an element  $z \in S$ , we define the node merge operation  $\bar{\rho}^{\text{met}}(\mathfrak{K}_S, z)$  as a new distributed cube  $\mathfrak{K}_{S'} = \{\mathfrak{K}_s\}_{s \in S'}$  such that:

- $S' = S \setminus \{s \in S \mid s \prec_S z \in B_S^*\}$ ;
- $\mathbf{M}'_z = \mathbf{M}_z \cup \{\text{mg}(c) \mid c \in \mathbf{F}_z\}$  with each  $\text{mg}(c)$  a new module name;
- for each  $c \in \mathbf{F}_z$ ,  $\text{mod}(c, \text{mg}(c))$  is added to  $\mathfrak{G}'_z$
- **Union merge** ( $\text{met} = \cup$ ):
  - $\mathfrak{G}_z \cup \bigcup_{s \prec_S z} \mathfrak{G}_s$  is added to  $\mathfrak{G}'_z$ ;
  - for every cell  $c \in \mathbf{F}_z$  with  $\text{cn}^-(c) = (z, \mathbf{d})$ , add module  $K_{\text{mg}(c)}$  to  $K_{\mathbf{M}'_z}$  with union of every module for  $c'$  s.t.  $\text{cn}^-(c') = (s, \mathbf{d})$  for  $s \prec_S z \in B_S^*$ .
- **Intersection merge** ( $\text{met} = \cap$ ):
  - $\mathfrak{G}_z \cup \bigcap_{s \prec_S z} \mathfrak{G}_s$  is added to  $\mathfrak{G}'_z$ ;
  - for every cell  $c \in \mathbf{F}_z$  with  $\text{cn}^-(c) = (z, \mathbf{d})$ , add module  $K_{\text{mg}(c)}$  to  $K_{\mathbf{M}'_z}$  with intersection of every module for  $c'$  s.t.  $\text{cn}^-(c') = (s, \mathbf{d})$  for  $s \prec_S z \in B_S^*$ .

Let us now consider the local merge operation, which is defined as a merge on the local cells at a specific level of the dimensional structure. We define a *level vector* as a set  $\mathbf{l} = \{l_1, \dots, l_k\}$  s.t. for  $j \in \{1, \dots, k\}$ ,  $l_j \in L_{A_j}$ . We define restrictions of dimensional space  $\mathfrak{D}_\Omega$  given w.r.t. a level vector  $\mathbf{l}$  as follows. The subspace  $\mathfrak{D}_\Omega^{\mathbf{l}}$  identifies all the vectors *exactly* at the level specified by the level vector  $\mathbf{l}$ :  $\mathfrak{D}_\Omega^{\mathbf{l}} = \{\mathbf{d} \in \mathfrak{D}_\Omega \mid \text{for } d \in D_A, \text{ lev}(d, \mathbf{l}) \text{ with } l \in \mathbf{l}\}$ .  $\mathfrak{D}_\Omega^{\mathbf{l} \succeq}$  defines the vectors *above* (or equal to) the specified level vector:  $\mathfrak{D}_\Omega^{\mathbf{l} \succeq} = \{\mathbf{d} \in \mathfrak{D}_\Omega \mid \mathbf{e} \preceq \mathbf{d}, \text{ with } \mathbf{e} \in \mathfrak{D}_\Omega^{\mathbf{l}}\}$ . Let  $\mu_s(c) = \{m \in \mathbf{M} \mid \mathfrak{G}_s \models \text{mod}(c', m), c' \prec c\}$ . The set  $\mu_s(c)$  then contains all module names of the initial cube associated to contexts  $c'$  that are more specific than the input context  $c$ .

**Definition 7 (Local merge).** Given a local cube  $\mathfrak{K}_s = \langle B_s, \mathfrak{G}_s, \mathcal{K}_{\mathbf{M}_s} \rangle$  and a level vector  $\mathbf{l}$ , we define the merge operation  $\rho^{\text{met}}(\mathfrak{K}_s, \mathbf{l})$  as a new cube  $\mathfrak{K}'_s = \langle B_s, \mathfrak{G}'_s, \mathcal{K}'_{\mathbf{M}_s} \rangle$  over  $\langle \Gamma', \Sigma \rangle$  such that:

- $\mathbf{F}'_s = \{c \in \mathbf{F}_s \mid \text{cn}^-(c) = (s, \mathbf{d}), \mathbf{d} \in \mathfrak{D}_{\Omega}^{\mathbf{l}_c}\}$ ;
- $\mathbf{M}'_s = \mathbf{M}_s \cup \{\text{mg}(c) \mid c \in \mathbf{F}'_s \text{ with } \text{cn}(c)^- = (s, \mathbf{d}), \mathbf{d} \in \mathfrak{D}_{\Omega}^{\mathbf{l}}\}$  with each  $\text{mg}(c)$  a new module name;
- Metaknowledge of  $\mathfrak{G}'_s$  is restricted to  $\Gamma'$  and  $\text{mod}(c, \text{mg}(c))$  for each  $c \in \mathbf{F}'_s$  with  $\text{cn}(c)^- = (s, \mathbf{d}), \mathbf{d} \in \mathfrak{D}_{\Omega}^{\mathbf{l}}$  is added to  $\mathfrak{G}'_s$ ;
- **Union merge** ( $\text{met} = \cup$ ): knowledge module  $\mathcal{K}_{\text{mg}(c)}$  for  $c$  is added to  $\mathcal{K}'_{\mathbf{M}_s}$  with:  $\mathcal{K}_{\text{mg}(c)} = \bigcup_{m \in \mu(c)} \mathcal{K}_m$
- **Intersection merge** ( $\text{met} = \cap$ ): knowledge module  $\mathcal{K}_{\text{mg}(c)}$  for  $c$  is added to  $\mathcal{K}'_{\mathbf{M}_s}$  with:  $\mathcal{K}_{\text{mg}(c)} = \bigcap_{m \in \mu(c)} \mathcal{K}_m$

We remark that the operations over the distribution and dimensional structure have been separated for a more clear presentation: however, in practice they could be easily combined in a single operation that manipulates both levels of the KG-OLAP structure.

## 4 Summary and Future Work

In this paper we introduced a distributed version of the KG-OLAP model for the analysis of contextualized knowledge graphs. We first extended the KG-OLAP framework presented in [9] to provide a decentralized definition for the multi-dimensional framework structure. On the base of this, we then demonstrated how analytic operations on KG-OLAP cubes can be extended to manipulate the newly defined distributed structure of the framework. This paper represents a first step in the study of distributed KG-OLAP cubes. As a next step, we need to develop reasoning methods for computing inferences over the distributed structure, possibly by extending the materialization methods proposed for KG-OLAP. In order to evaluate the possibilities provided by the distributed framework, we will develop a proof-of-concept implementation of KG-OLAP cubes in [9]. This will allow us to compare the current performance and scalability results over centralized KG-OLAP cubes to the case of a distributed scenario. As previously mentioned, we conjecture that distribution of a KG over multiple processors can be used to enhance the performance of reasoning over large-scale KGs. We also plan to verify the applicability of our model and operations to the real scenario of ATM information, possibly by further developing our initial definitions.

## References

1. Abelló, A., Darmont, J., Etcheverry, L., Golfarelli, M., Mazón, J., Naumann, F., Pedersen, T.B., Rizzi, S., Trujillo, J., Vassiliadis, P., Vossen, G.: Fusion cubes: Towards self-service business intelligence. *Int. J. of Data Warehousing and Mining* 9(2), 66–88 (2013)
2. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): *The Description Logic Handbook*. Cambridge University Press (2003)
3. Bozzato, L., Eiter, T., Serafini, L.: Enhancing context knowledge repositories with justifiable exceptions. *Artificial Intelligence* 257, 72–126 (2018)



4. Bozzato, L., Eiter, T., Serafini, L.: Justifiable exceptions in general contextual hierarchies. In: CONTEXT 2019. Lecture Notes in Computer Science, vol. 11939, pp. 26–39. Springer (2019)
5. Bozzato, L., Serafini, L.: Materialization calculus for contexts in the Semantic Web. In: DL 2013. CEUR Workshop Proceedings, vol. 1014. CEUR-WS.org (2013)
6. Bozzato, L., Serafini, L., Eiter, T.: Reasoning with justifiable exceptions in contextual hierarchies. In: KR 2018. pp. 329–338. AAAI Press (2018)
7. Colazzo, D., Goasdoué, F., Manolescu, I., Roatiş, A.: RDF Analytics: Lenses over Semantic Graphs. In: WWW'14. pp. 467–478 (2014)
8. Krötzsch, M., Weikum, G.: Editorial for special section on knowledge graphs. *Journal of Web Semantics* 37-38, 53–54 (2016)
9. Schuetz, C.G., Bozzato, L., Neumayr, B., Schrefl, M., Serafini, L.: Knowledge Graph OLAP: A Multidimensional Model and Query Operations for Contextualized Knowledge Graphs. *Semantic Web Journal* (2020), in press. <http://www.semantic-web-journal.net/content/knowledge-graph-olap-multidimensional-model-and-query-operations-contextualized-knowledge-0>
10. Schuetz, C.G., Neumayr, B., Schrefl, M., Gringinger, E., Wilson, S.: Semantics-based summarisation of ATM information: Managing information overload in pilot briefings using semantic data containers. *The Aeronautical Journal* 123(1268), 1639–1665 (2019)
11. Serafini, L., Homola, M.: Contextualized Knowledge Repositories for the Semantic Web. *Journal of Web Semantics* 12, 64–87 (2012)