HAL
open science

Enterprise Knowledge Graph : définition et contribution à un système 'Team of Teams'

Bastien Vidé, Max Chevalier, Franck Ravat

To cite this version:

Bastien Vidé, Max Chevalier, Franck Ravat. Enterprise Knowledge Graph : définition et contribution à un système 'Team of Teams'. Revue des Nouvelles Technologies de l'Information, 2020, RNTI-B-16, pp.60-68. hal-03123181

HAL Id: hal-03123181

https://hal.science/hal-03123181v1

Submitted on 27 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Enterprise Knowledge Graph : définition et contribution à un système 'Team of Teams'

Bastien Vidé*,**, Max Chevalier*, Franck Ravat*

*IRIT, 118 route de Narbonne, 31400 Toulouse
max.chevalier, franck.ravat, bastien.vide@irit.fr
**umlaut, 3 Bd Henri Ziegler, 31700 Blagnac
bastien.vide@umlaut.com

Résumé. Actuellement, le concept de 'Knowledge Graph (KG)' est populaire dans le monde de l'entreprise. Un *KG* est vu comme une potentielle solution aux 'problèmes des silos de données' en proposant une vue unifiée de celles-ci. Dans cet article nous comparons les *KG* d'entreprises avec les autres solutions proposant une vue unifiée des données. Dans un second temps, nous proposons une définition du concept de '*Enterprise Knowledge Graph* (EKG)'. Enfin, nous illustrons le concept d'*EKG* au travers d'un projet 'Team of Teams' d'une entreprise réelle et effectuons une expérimentation d'implantation de cet EKG selon deux implantations : relationnelle et graphe.

1 Introduction

Les entreprises aujourd'hui organisent leur large volume de données en 'silos de données'. Ce type de stockage permet aux entreprises d'organiser les données selon différents critères propres à leur fonctionnement (par projet, par unité structurelle – Business Unit...). Cela permet également de gérer localement les données avec un système de stockage adapté. Cependant, une difficulté engendrée par ces silos réside dans le fait que les données qu'ils contiennent sont isolées. Cela peut amener à une redondance voire à des incohérences fortes entre les données contenues dans les différents silos. De plus, le pilotage de l'entreprise ne dispose alors pas d'une vision unifiée des données pour l'aide à la prise de décision.

L'objectif de ce papier est de présenter le concept 'd'Enterprise Knowledge Graph (EKG)' qui fournit entre autres une telle vue unifiée. Cette vue a pour but d'interconnecter les informations pertinentes pour l'entreprise tout en favorisant l'extraction de connaissances par les utilisateurs. Dans la section 2, nous présentons les différentes approches existantes permettant de construire une vue unifiée pour ensuite définir le concept d'EKG. Enfin, nous étudions et expérimentons les modèles de stockage pour un tel EKG (modèle relationnel et graphe) sur un jeu de données réel issu d'une entreprise dans un périmètre de service Team of Teams.

2 Vue unifiée des données : approches existantes

L'objectif général de notre travail d'offrir une vision unifiée de l'ensemble des informations pertinentes pour l'entreprise (e.g. issues des données de production et des documents circulant dans l'entreprise) afin d'en faciliter l'exploitation ainsi que la construction de nouvelles connaissances. Si nous nous référons à l'état de l'art, nous pouvons identifier trois concepts menant à cet objectif : les *Entrepôts de Données*, les *Lacs de Données* et les *Knowledge Graph*.

2.1 Entrepôt et Lac de Données

Dans le cadre de la **Business Intelligence** (BI), un **Entrepôt de Données** (ED) est une base de données décisionnelles centralisant et historisant un extrait des sources pertinentes pour les décideurs ; ces données sont non volatiles et disponibles pour l'interrogation décisionnelle. Un ED ne contient qu'un extrait des données de production, déterminé et modélisé pour répondre à des besoins clairement explicités préalablement à son développement. Cette phase d'intégration s'effectue au travers de processus *ETL* (*Extract, Transform, Load*) et n'intègre donc pas toutes les données circulant dans une organisation.

Un **Lac de Données** (LD) est une solution de Big Data Analytics (Ravat et Zhao, 2019) permettant : (i) d'ingérer les données brutes provenant de diverses sources (données structurées, semi ou non structurées), (ii) de stocker les données dans leur format natif, (iii) de préparer les données uniquement lorsqu'elles sont analysées afin de fournir des accès à différents utilisateurs, (iv) de gouverner les données pour gérer la qualité, la sécurité, et le cycle de vie des données. Une des grandes différences avec un ED est l'intégration de la totalité des données sans processus de transformation initiale. L'intérêt d'un LD par rapport à un ED réside également dans le fait qu'aucun modèle de données préalable n'est nécessaire. Les données d'un LD sont aussi disponibles à un plus grand nombre d'utilisateurs ayant des profils différents.

2.2 Knowledge Graph

Les deux solutions précédentes ne répondent que partiellement aux objectifs que nous visons et qui sont importants pour toute organisation. Le concept de **Knowledge Graph** (KG) semble en effet plus prometteur. Le terme de KG a été popularisé par Google en 2012 lorsque cette société l'a implanté dans son moteur de recherche. Les informations du KG de Google sont extraites depuis de nombreuses bases d'informations dont Wikidata et Freebase. Le KG de Google supporte des interrogations en langage naturel et restitue le résultat au travers d'une '*Knowledge Card*'. Cette Knowledge Card fournit un ensemble d'informations et de pointeurs (e.g. sources) décrivant l'objet retrouvé (Ehrlinger et Wöß, 2016).

Dans le milieu industriel, le KG apparaît de plus en plus populaire (Kendall Clark, 2017) et présente l'avantage d'unifier les données contenues dans les silos dans un seul système. Aussi, dans le milieu académique, le *Knowledge Graph* dispose de nombreuses définitions. Dans la communauté scientifique du Web Semantic (Färber et al., 2017), il est associé à un graphe RDF (Resource Description Framework). Dans (Paulheim, 2016), il est défini comme un graphe contenant des entités ainsi que leurs relations. Nous pouvons trouver une définition plus générale dans Blumauer (2014) : "*Knowledge graphs could be envisaged as a network of all kind things which are relevant to a specific domain or to an organization. They are not limited to abstract concepts and relations but can also contain instances of things like documents*

and datasets." Cette définition est bien plus générale car elle précise que le KG peut contenir des éléments abstraits et concrets. Ehrlinger et Wöß (2016) précisent que le terme *knowledge graph* est souvent plus utilisé comme un 'Buzzword' adopté par les entreprises afin de parler des applications dédiées à la représentation de la connaissance. En réponse à cela, ils proposent une définition plus précise que les précédentes intégrant une dimension d'objectif du KG : '*A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge.*' Cette définition souligne au delà de l'intégration des données, la notion d'inférence importante pour le support à la détermination de nouvelles connaissances.

2.3 Synthèse

Au regard des sections précédentes nous pouvons dire que, bien que ces approches visent à supporter la prise de décision, les *ED* manipulent des données préalablement traîtées (information), alors que les *LDs* permettent de manipuler les données brutes. Enfin, le *KG* offre plutôt une vision conceptuelle et/ou factuelle des données extraites depuis internet (Pujara et al., 2013) sans réelle aide à la prise de décision.

	Contenu privilégié			Expression des besoins					
	Data	Info	Knowledge	Préalable	À la volée	Évolutivité	Liaison DIK*	Prise de décision	Démarche de construction
ED		Oui		Oui		Possible[a]	Non	Oui	Oui
LD	Oui	Oui			Oui	Oui	Non[b]	Oui	Oui
KG			Oui		Oui	Oui	Oui	Non	Non
EKG	Lien	Oui	Conceptuel	Oui	Oui	Oui	Oui	Oui	Oui

* DIK : Data/Information/Knowledge (Fig. 1)

[a] Demande de revoir le modèle du magasin à chaque changement.

[b] Aucun lien puisque les données sont stockées de manière indépendante et non structurée.

TAB. 1 – *Récapitulatif des différents concepts.*

La Table 1 présente la synthèse des travaux discutés précédemment ainsi que le positionnement de l'EKG que nous souhaitons proposer. La table indique le **Contenu privilégié**, c'est-à-dire le(s) type(s) de contenu(s) stocké(s) prioritairement dans les différentes approches. Elle indique si l'**expression des besoins** est faite préalablement au développement de la solution, ou si elle n'est prise en compte qu'à la volée (e.g. au moment de l'interrogation) et qu'il peut suivre 'facilement' [1] l'évolution des besoins de l'entreprise et/ou des utilisateurs. La **Liaison DIK** indique si les Données, les Informations et les Connaissances sont reliées explicitement dans ces systèmes. Enfin, la table indique si la solution aide à la **Prise de décision** et s'il existe à notre connaissance une **Démarche de construction** pour celle-ci.

3 Un modèle d'Enterprise Knowledge Graph

L'objectif de notre contribution est de proposer une vue unifiée des données favorisant la construction de nouvelles connaissances dans l'entreprise par les utilisateurs : l'**Enterprise**

1. Sans remettre en cause le modèle sous-jacent.

Knowledge Graph (EKG). Cette contribution tend à tirer parti ‘du meilleur des ED et des LD‘. Elle doit pouvoir intégrer tous types de sources de données (structurées ou non, depuis des LD et des ED par exemple).

Cette vue unifiée contient des informations disponibles dans l'entreprise tout en conservant des relations explicites entre ces informations afin de les ‘contextualiser‘ avec le besoin de l'entreprise. Les informations ainsi que leurs relations seront historisées pour faciliter leur interprétation et le suivi de leurs évolutions. La relation entre les données et les informations seront également conservées. L'EKG pourra, sur la base de ce contenu, être interrogé à la volée ou à partir d'un ensemble de Vues répondant à des besoins utilisateurs préalablement définis (dans la même logique que les magasins de données) en facilitant l'exploration des données à l'aide de traitements et de visualisations adaptées voire d'une fonction de navigation permettant un parcours libre du contenu (c-à-d sans requête préalable, pouvant se baser sur OLAP (Beheshti et al.)).

3.1 EKG : notre définition

Nous définissons un EKG comme *un modèle de représentation d'entités informationnelles d'intérêt pour l'entreprise. Les données et informations sur lesquelles sont construites ces entités sont conservées et laissées dans leurs sources originales. Un EKG doit permettre la recherche et l'analyse des entités et de leurs relations ainsi qu'un retour aux données brutes si nécessaire.*

Un EKG est modélisé via un graphe représentant l'ensemble des entités informationnelles (c-à-d des ensembles de données, de biens matériels ou de concepts dans le contexte de l'entreprise) de l'entreprise liées entre elles par des relations. Les entités et les relations peuvent posséder des propriétés extraites des données sources (données structurées issues des sources de production ou documents non structurés). Nous présentons ci-dessous une définition formelle d'un d'EKG reposant sur celle des graphes.

L'EKG (eq. 1) est défini comme un graphe orienté, c'est-à-dire comme un ensemble de relations et d'entités. Dans un EKG, un noeud du graphe représente une **Entité** et un lien du graphe représente une **Relation**. Les entités $e_m \in E$ et les relations $r_n \in R$ possèdent un type T_{r_n} et T_{e_m} ainsi que des propriétés P_{r_n} et P_{e_m}. Les noeuds et les liens comportent au moins une ou plusieurs *Références vers une source* S_{r_n}, S_{e_m} et un *ID* I_{r_n} et I_{e_m}. Les noeuds disposent en plus d'un *nom* N_{e_m}. Les *références vers la source* sont définies par un couple (U_{r_n}, D_{r_n}) pour les relations et (U_{e_m}, D_{e_m}) pour les entitées, où D_{r_n} et D_{e_m} sont des horodatages de l'extraction depuis la source, et U_{r_n} et U_{e_m} représentent les liens vers la source. Les **Vues** V (eq. 6) sont définies comme étant un sous-graphe de l'EKG.

$$EKG = (E, R) \tag{1}$$

$$r_n = \{I_{r_n}, T_{r_n}, S_{r_n\,0}, S_{r_n\,1}, \cdots, P_{r_n\,0}, P_{r_n\,1}, \cdots\} : n \in \mathbb{N} \wedge r_n \in R \tag{2}$$

$$e_m = \{I_{e_m}, T_{e_m}, N_{e_m}, S_{e_m\,0}, S_{e_m\,1}, \cdots, P_{e_m\,0}, P_{e_m\,1}, \cdots\} : m \in \mathbb{N} \wedge e_m \in E \tag{3}$$

$$S_{r_n\,\alpha} = (U_{r_n\,\alpha}, D_{r_n\,\alpha}) : \alpha \in \mathbb{N} \tag{4}$$

$$S_{e_m\,\beta} = \left(U_{e_m\,\beta}, D_{e_m\,\beta}\right) : \beta \in \mathbb{N} \tag{5}$$

$$V = (E_s, R_s) : E_s \subseteq E \wedge R_s = \{(v_x, v_y) \in R : \{x, y\} \subset E_s\} \tag{6}$$

3.2 Construction de notre EKG

Notre EKG, repose sur les niveaux de la pyramide Data/Information/Knowledge (Baskarada et Koronios, 2013). Notre méthode est divisée en trois principales étapes. La première intègre dans l'EKG toutes les données utiles depuis les sources. Ensuite, ces données sont transformées en informations en y ajoutant un contexte, une signification, particulièrement en reliant les différentes sources entre elles. Cette étape dépend des besoins de l'entreprise, selon les connaissances qu'elle souhaite mettre en évidence depuis ses données. Enfin, la dernière étape consiste à conceptualiser l'information. Nous ajoutons des informations plus générales aux entités et aux relations de notre graphe. Cela peut notamment passer par la liaison entre les informations et des concepts de plus haut niveau provenant par exemple d'ontologies adaptées aux informations de l'entreprise telles que par exemple SKOS, SIOC ou une ontologie créée par l'entreprise. Cela nous permet une meilleure généralisation de son contenu.

3.3 Modèles de stockage de l'EKG et contribution à un système Team of Teams

Un point essentiel dans notre méthode de construction d'un EKG est de choisir le modèle de stockage le plus adapté à l'EKG. Intuitivement, nous pourrions envisager l'utilisation d'un datastore NoSQL de type graphe. Cependant, rien n'assure que ce modèle de stockage soit plus pertinent qu'un modèle relationnel par exemple. C'est la question que nous étudions dans cette section en prenant pour exemple un EKG intégrant des données réelles liées aux compétences des employés d'une entreprise. L'EKG illustré ici se situe au coeur de la mise en place d'un Team of Teams au sein de l'entreprise.

Nous avons donc développé pour les expérimentations 11 requêtes (simples ou complexes). Ces requêtes sont présentées pour chaque modèle de stockage sur le site compagnon associé à cet article [2].

3.3.1 Les données intégrées à l'EKG

Les sources de notre EKG sont les réponses au questionnaire envoyé à 35 collaborateurs de l'entreprise. Ce questionnaire recense 98 caractéristiques organisé en 11 types liées aux personnes : compétences, traits de caractères, langues parlées, préférences, hobbies et sports pratiqués, etc. Ce questionnaire est un tableau csv anonymisé. Les différents attributs ainsi que leur type sont présentés sur le site compagnon[2].

3.3.2 Expérimentations liées au modèle de stockage de l'EKG

Comme première étape, afin d'étudier le comportement des modèles (relationnel et graphe) nous avons étudié la complexité et les performances des requêtes pour chacun de ces modèles utiles dans le cas d'un Team of Teams. Pour cela, nous utilisons le langage SQL pour une BD relationnelles Sqlite et Cypher le langage d'interrogation de Neo4J. Les deux stockages ont la même volumétrie, les mêmes données, et les requêtes rendent exactement les mêmes résultats.

2. https ://tinyurl.com/ekg-expe

Ingestion des données : Il semble qu'utiliser une base de données de type graphe soit plus 'simple' et plus adaptée qu'une base de données relationnelle. De plus, cette dernière a demandé plus de réflexion et de temps afin d'aboutir à un résultat pertinent, là où la base de données graphe nous a demandé moins d'étapes préalables. Nous avons également pu réaliser l'ingestion des données de manière native dans la base de données graphes alors qu'il fallait utiliser des outils externes pour la BDD relationnelle.

Protocole de test : Nous avons laissé le 'Page Cache' de Neo4J activé pour travailler à égalité avec SQLite. Nous avons désactivé la mise en cache du plan d'exécution afin de mesurer la vitesse réelle d'exécution. Dans les deux cas, nous utiliserons les *CLI* (clients prompt) des deux différents systèmes pour les interroger et exécuter les requêtes. Chacune des requêtes a été executée 10 fois et une moyenne a été calculée en ne prenant en compte que le temps d'exécution, sur une même machine, avec exactement les mêmes processus lancés tout au long de l'expérimentation.

Comparaison des performances : Pour effectuer les comparaisons, nous avons exécuté des requêtes de complexité différente sur chacun des systèmes de stockage. Pour chaque requête, nous avons comparé le nombre d'éléments retournés, sa complexité (i.e le nombre d'instructions) et le temps d'exécution. Les résultats sont donnés dans la Table 2.

3.3.3 Récapitulatif des résultats

Lors de notre expérimentation, nous avons donc confronté SQL et Cypher. Bien que les résultats de performance tendent à montrer que SQL soit un meilleur choix, il ne semble pas efficace lorsque le nombre de relations à traiter augmente. De plus, la facilité d'écriture d'un système graphe le rend beaucoup plus facile à utiliser (notamment via la visualisation de graphe 'native' de Neo4J). La base de données graphe a donc l'avantage sur une quantité de données importantes et sur sa simplicité d'utilisation et d'intégration.

Un autre avantage non cité dans l'expérimentation est la simplicité d'implantation des algorithmes spécifiques aux graphes, comme ceux de similarité (ressemblance entre deux noeuds), de centralité (noeuds les plus connectés/influents) et de communauté (determination des différents groupes d'entités) qui peuvent être intéressantes à appliquer lors de l'exploitation des données. Nous avons ainsi testé l'algorithme de similarité (Jaccard) sur des Vues (voir eq. 6) de notre graphe en comparant les personnes et en grâce à une propriété contenue dans les relations. Les resultats sont présentés dans le site compagnon [3]. Ces algorithmes spécifiques aux graphes sont difficilement implémentables avec une base de données relationnelle. Pourtant, leur résultat est important afin de classer les entités et de découvrir de nouvelles relations dans un graphe.

4 Conclusion

Les entreprises ont encore plus besoin aujourd'hui d'une vue unifiée de leurs données, et d'un moyen de les mettre en valeur. Pour répondre à ce besoin, nous avons comparé différentes

3. https ://tinyurl.com/ekg-expe

Requête	Résultat	Complexité (plan d'exec)			Temps		
		SQL	Cypher	Ecart	SQL	Cypher	Ecart
Sélections simples de Noeuds							
Toutes les propriétés de toutes les personnes	34	1	1	0%	0.5ms	12.6ms	2422%
Toutes les compétences techniques	11	1	2	103%	0.2ms	11.3ms	5553%
Les dernières personnes intégrées dans le graphe	2	3	4	37%	0.1ms	19ms	18904%
Sélection avec des relations du premier degré							
Toutes les relations avec les noms des personnes et des compétences	2512	33	2	-93%	17.6ms	11.5ms	-34.65%
Les expertises de toutes les personnes	1246	33	3	-91%	5.9ms	9.9ms	74.79%
Calculs et groupements							
Personnes qui ont toutes les compétences techniques	5	7	7	0%	1ms	22.8ms	2189%
Personnes qui ont le plus de relations	3	91	6	-93%	5ms	7.5ms	60%
Calculs et groupements avec des relations du second degré et plus							
Personnes ayant des compétences manquantes à X	26	9	8	-11%	1.3ms	18.4ms	1327%
Personnes ayant toutes les compétences techniques manquantes à X	10	15	10	-33%	1.9ms	25.9ms	1276%
Selections avec multiple jointures							
Toutes les personnes et leurs domaines ayant des expertises techniques là où d'autres n'en ont pas	148	8	10	40%	9.1ms	35ms	299%

TAB. 2 – *Résultats de l'expérimentation.*

solutions : ED, LD et Knowledge Graph. Nous avons proposé le concept d'**Enterprise Knowledge Graph (EKG)** visant à intégrer le meilleur des autres modèles (particulièrement ED et LD) permettant l'acquisition et la création de nouvelles connaissances ainsi que leur interrogation et leur parcours à la volée ou au travers de vues. Nous avons comparé les modèles de stockage (relationnel et graphe) sous-jacents à un tel EKG en prenant pour exemple une application de 'Team of Teams'. Les données utilisées sont des données réelles provenant d'une entreprise. Les résultats obtenus nous confortent dans l'idée que les idées 'intuitives' ne sont pas nécessairement les meilleures dans toutes les configurations.

Nous comptons poursuivre nos activités de recherche en mettant l'accent sur la définition d'une méthode d'alimentation d'un EKG ainsi que sur les opérateurs permettant de le manipuler. Nous souhaitons également poursuivre nos expérimentations quant aux problématiques de stockage en intégrant notamment de nouveaux 'besoins d'analyse' et les besoins d'architecture récents, type micro-services.

Remerciements

Les auteurs souhaitent associer à cet article, Joan MARTY, de la société **umlaut** avec qui nous collaborons sur ces travaux et qui a permis de mener les expérimentations sur le 'Team of Teams'.

Références

Baskarada, S. et A. Koronios (2013). Data, Information, Knowledge, Wisdom (DIKW) : A Semiotic Theoretical and Empirical Exploration of the Hierarchy and its Quality Dimension. *Australasian Journal of Information Systems 18*(1), 20.

Beheshti, S.-M.-R., B. Benatallah, H. R. Motahari-Nezhad, et M. Allahbakhsh. A framework and a language for on-line analytical processing on graphs. In X. S. Wang, I. Cruz, A. Delis, et G. Huang (Eds.), *Web Information Systems Engineering - WISE 2012*, Volume 7651, pp. 213–227. Springer Berlin Heidelberg. Series Title : Lecture Notes in Computer Science.

Blumauer, A. (2014). From Taxonomies over Ontologies to Knowledge Graphs.

Ehrlinger, L. et W. Wöß (2016). Towards a Definition of Knowledge Graphs. In *Joint Proceedings of the Posters and Demos Track of 12th International Conference on Semantic Systems*, pp. 4.

Färber, M., F. Bartscherer, C. Menne, et A. Rettinger (2017). Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web 9*(1), 77–129.

Kendall Clark (2017). What is a Knowledge Graph.

Paulheim, H. (2016). Knowledge graph refinement : A survey of approaches and evaluation methods. *Semantic Web 8*(3), 489–508.

Pujara, J., H. Miao, L. Getoor, et W. Cohen (2013). Knowledge Graph Identification. In H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, et K. Janowicz (Eds.), *The Semantic Web – ISWC 2013*, Lecture Notes in Computer Science, Berlin, Heidelberg, pp. 542–557. Springer.

Ravat, F. et Y. Zhao (2019). Data Lakes : Trends and Perspectives. In S. Hartmann, J. Küng, S. Chakravarthy, G. Anderst-Kotsis, A. M. Tjoa, et I. Khalil (Eds.), *Database and Expert Systems Applications*, Volume 11706, pp. 304–313. Cham : Springer International Publishing.

Summary

Currently, the concept of 'Knowledge Graph (KG)' is popular in the business world. A *KG* is seen as a potential solution to the 'data silo problem', by providing a unified view of data. In this article we compare enterprise *Knowledge Graphs* with other solutions offering a unified view of data. In a second step, we propose a definition for the concept of '*Enterprise Knowledge Graph* (EKG)'. Finally, we illustrate the *EKG* concept through a 'Team of Teams' project of a real company and carry out an implementation experimentation of this EKG using two systems: using relational and graph databases.