

---

## Vers une approche efficace de gestion d'évolution des données graphes

**Landy Andriamampianina<sup>1,2</sup>, Franck Ravat<sup>1</sup>, Jiefu Song<sup>1,2</sup>,  
Nathalie Vallès-Parlangeau<sup>1</sup>**

1. IRIT-CNRS (UMR 5505), Université Toulouse 1 Capitole  
2 Rue du Doyen-Gabriel-Marty 31042 Toulouse, France  
*prenom.nom@irit.fr*

2. Activus Group,  
1 Chemin du Pigeonnier de la Cépière 31100 Toulouse, France  
*prenom.nom@activus-group.fr*

---

*Cet article est un résumé de l'article : Landy Andriamampianina, Franck Ravat, Jiefu Song, Nathalie Vallès-Parlangeau, Towards an Efficient Approach to Manage Graph Data Evolution: Conceptual Modelling and Experimental Assessments. RCIS 2021: 471-488.*

*MOTS-CLES : Graphe temporel, Snapshots, Evolution temporelle, Bases de données graphes.*

*KEYWORDS: Temporal graph, Snapshots, Temporal evolution, Graph data stores.*

---

Les données graphes modélisent naturellement les relations complexes entre des entités dans de nombreux domaines. Les changements au cours du temps de ces données graphes fournissent des informations contextuelles à un décideur, et ainsi nécessitent d'être intégrées aux données graphes de base. L'évolution temporelle des données graphes peut être classée en trois types, à savoir (i) *l'évolution de la topologie* (l'ajout et la suppression de nouvelles entités et relations), (ii) *l'évolution des attributs* (l'ajout et la suppression des attributs décrivant une entité ou une relation) et (iii) *l'évolution de la valeur des attributs* (changement de valeur des attributs décrivant une entité ou une relation) (Zaki et al., 2016). Nous présentons dans cet article une proposition de gestion de tous les types d'évolution des données graphes.

Les solutions existantes, nommées *graphes temporels*, ne capturent que partiellement l'évolution dans les graphes en se limitant à deux types d'évolution : la topologie et la valeur des attributs (Zaki et al., 2016). De plus, l'approche classique pour historiser les changements est la séquence de snapshots. Cette dernière ne capture pas réellement les changements dans le graphe et crée de la redondance des données qui ne changent pas (Moffitt et Stoyanovich, 2017). Afin de répondre à ces limites, notre objectif est de proposer une nouvelle approche de gestion de graphes temporels qui gère non seulement l'évolution de la topologie et de la valeur des

attributs (types (i) et (iii) mentionnés ci-dessus) mais également l'ajout et la suppression des attributs descriptifs (type (ii)). De plus, il est nécessaire de définir des règles d'implémentation qui garantissent la faisabilité et l'efficacité de la solution proposée.

La solution que nous proposons intègre un modèle conceptuel de graphe temporel composé d'*entités* et de *relations temporelles*. Une entité (ou relation) temporelle est composée d'un ensemble d'états. Chaque état contient une liste d'attributs ainsi que leurs valeurs présentes pendant une période de validité. A travers ces concepts, nous modélisons, d'une part, l'évolution de la topologie en comparant deux ensembles d'états d'entités (ou de relations) correspondant à des périodes de validité différentes, et d'autre part, l'évolution des attributs et de leurs valeurs en comparant deux états d'une même entité (ou d'une relation). Gérer les changements au niveau de chaque entité et relation permet d'éviter la redondance des données des snapshots. Au niveau logique, notre solution inclut des règles d'implémentation de notre modèle conceptuel dans un graphe de propriétés.

Notre proposition a été évaluée à travers une série d'expérimentations afin d'illustrer sa faisabilité (implémentable) et son efficacité (éviter la redondance des données et produire des temps d'exécution raisonnables). Pour ce faire, nous avons utilisé un jeu de données d'un benchmark de référence contenant les trois types d'évolution précédents. Nous l'avons implanté à l'aide du système de gestion de bases de données graphe Neo4j. Nous avons créé 28 requêtes de référence avec une couverture complète des différents types d'analyse possibles.

Notre étude de faisabilité a permis de montrer que notre modèle temporel conceptuel est implantable sans difficulté sous Neo4j. Dans notre étude d'efficacité, nous avons implanté notre jeu de données dans Neo4j selon trois approches de modélisation : l'approche classique des snapshots, une approche optimisée des snapshots (réduisant partiellement la redondance) et notre modèle. Nous avons comparé l'efficacité de notre proposition aux snapshots en mesurant l'utilisation du disque et le temps d'exécution des requêtes. Notre proposition permet de diviser par 12 l'utilisation du disque des implémentations basées sur les snapshots. De plus, sur la base des 28 requêtes, notre proposition permet de réduire le temps d'exécution des requêtes entre 58% et 99% suivant le type de requêtes.

En résumé, nous proposons une approche efficace de gestion de tous les types d'évolution des données graphes. La prochaine étape de nos travaux se concentre sur la manipulation des graphes temporels. L'objectif de ces travaux est d'enrichir les analyses décisionnelles via l'ajout de la temporalité dans les données graphes.

## Bibliographie

- Moffitt V. Z., Stoyanovich J. (2017). Towards sequenced semantics for evolving graphs. EDBT.
- Zaki A., Attia M., Hegazy D., Amin S. (2016). Comprehensive Survey on Dynamic Graph Models. International Journal of Advanced Computer Science and Applications. vol. 7, n°2, p. 573--582.