



HAL
open science

Information Systems: from Model to Data Driven Decision Support

Faten Atigui

► **To cite this version:**

Faten Atigui. Information Systems: from Model to Data Driven Decision Support. Computer Science [cs]. Conservation National des Arts et Métiers, 2023. tel-04754027

HAL Id: tel-04754027

<https://hal.science/tel-04754027v1>

Submitted on 25 Oct 2024

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.



Conservatoire National des Arts et Métiers

Habilitation à Diriger des Recherches

(Spécialité Informatique)

Ecole doctorale SMI - Sciences des Métiers de l'Ingénieur

Information Systems: from Model to Data Driven Decision Support

Présentée par : Faten Atigui

Rapporteurs:

KÁTHIA MARÇAL DE OLIVEIRA, Professeure, Université Polytechnique Hauts-de-France

SELMIN NURCAN, Professeure, Paris 1 Panthéon-Sorbonne

OSCAR PASTOR, Professeur, Université polytechnique de Valencia, Espagne

Examineurs:

KAMEL BARKAOUI, Professeur, Conservatoire National des Arts et Métiers

ELHADJ BENKHELIFA, Professeur, Université Staffordshire, Royaume-Uni

SAMIRA CHERFI, Professeure, Conservatoire National des Arts et Métiers (Garante)

RAJA CHIKY, Professeure, Institut d'Ingénierie Informatique de Limoges

Soutenue le : 19 décembre 2023

Acknowledgements

First of all, I would like to thank my HDR reviewers Prof. Káthia Marçal de Oliveira, Prof. Selmin Nurcan, and Prof. Oscar Pastor who accepted to review and evaluate this dissertation.

Many thanks to Prof. Kamel Barkaoui, Dr. Raja Chiky, and Prof. Elhadj Benkhelifa for being a part of my examining committee.

I am deeply grateful to Prof. Samira Cherfi for supporting me in the realization of this HDR. I would like to express my sincere gratitude for her guidance, encouragement and ongoing support. Her mentoring has contributed to both my professional and personal maturity.

I want to express my deep gratitude and sincere thanks to Prof. Gilles Zurfluh for the invaluable guidance and support he has provided me with over the years.

My sincere thanks go to my colleagues in the CEDRIC Laboratory, and the computer science Department (EPN5) of the Conservatoire National des Arts et Métiers.

Many thanks to my PhD students for several years of constructive and enriching discussions and to my colleagues with whom I collaborated on various projects.

A special feeling of gratitude goes to my sisters and friends who generously dedicated their time and efforts to proofread this thesis.

Last but not least, I am thankful to my wonderful family for their unlimited support and encouragement.

Résumé

Ce mémoire d'HDR présente mes activités de recherche au cours des neuf dernières années. Il expose nos contributions dans le domaine des Systèmes d'Information (SI), de la sécurité dans les organisations de santé, et de la Business Intelligence & Analytics (BI&A). Nous nous sommes concentrés sur les enjeux de modélisation et de développement des SI exploitant les bases de données NoSQL. Nous nous sommes également penchés sur le problème de l'intégration des données NoSQL dans les systèmes BI&A. De plus, dans le cadre du projet H2020 SAFECARE, nous avons travaillé sur la modélisation de la sécurité dans les systèmes cyber-physiques liés aux organisations de santé. Nos contributions visent à l'aide à la décision basée sur les modèles et/ou les données. Pour faire face à l'évolution des SI et à la gestion de la sécurité des systèmes de santé, nous avons proposé des approches basées sur les modèles, alors que l'intégration des données NoSQL dans les systèmes BI&A est principalement basée sur les données.

Approches basées sur les modèles

- **Développement dirigé par les modèles des systèmes d'information big data.** Depuis plusieurs décennies, le stockage et l'exploitation des données reposent principalement sur des Bases de Données Relationnelles (BDR). Avec l'avènement du big data, le volume de données a explosé, la variété s'est accrue provoquant plusieurs enjeux liés à la transformation digitale que ce soit en termes de stockage, d'exploitation de données, de coût ou de performance. Pour répondre à ces enjeux, de nouveaux systèmes appelés systèmes NoSQL sont apparus. Dans ces systèmes, les données sont organisées selon 4 familles, à savoir, **clé-valeur**, **orientée-documents**, **orientée-colonnes** et **orientée-graphes**. Aujourd'hui, en plus des solutions relationnelles, il existe plus de 225 systèmes NoSQL¹. Ainsi, face à la panoplie de modèles et de solutions existantes, le choix de la structure et du (ou des) système(s) en adéquation avec les besoins fonctionnels et techniques est une tâche coûteuse, complexe et irréalisable de façon manuelle. Des choix inadéquats peuvent entraîner des problèmes d'évolutivité, de cohérence et de coût. Il est judicieux de suivre une démarche qui accompagne le concepteur avec des choix fondés afin de mener à bien le projet SI.

Dans ce contexte, nous proposons une démarche globale qui facilite et automatise le processus de transformation d'un modèle conceptuel en modèles physiques relatifs aux 4 familles NoSQL, mais aussi au modèle relationnel. Cette démarche permet également de guider le choix des modèles et des solutions techniques les plus adaptés aux besoins métiers. Nous adoptons l'architecture dirigée par les modèles (MDA) qui fournit un cadre formel pour la modélisation et la transformation de modèles. En se basant sur des règles de raffinement de modèles, notre approche vise aussi à guider le choix au niveau logique (quelle famille, imbrication partielle, totale ou normalisation du schéma) et au niveau physique considérant des critères techniques (les coûts, les performances, etc.). Ces contributions ont été proposées dans le cadre de 2 thèses de doctorat (1 thèse soutenue : A. Ait Brahim, 2015-2018 ; une thèse en cours : J. Mali, 2020-2023) et 1 M2R avec publication (A. Mokrani, 2017). Nos travaux ont été publiés dans plusieurs conférences nationales et internationales (INFORSID'16, KMIS'16, DaWak'17, AICSSA'17, ICEIS'17, CIBSE'17, PDPTA'18, DEXA'20, EGC'20, BDA'20, RCIS'22).

- **Modélisation de la sécurité dans les systèmes de santé cyber-physiques.** Les hôpitaux et les organisations de santé font partie des infrastructures cyber-physiques les plus critiques et les plus vulnérables. Ces organisations utilisent de plus en plus de nouvelles technologies, notamment les capteurs portables ainsi que la surveillance à distance des patients prenant appui sur des interfaces et des normes de communication communes. Ceci renforce les failles de sécurité et ouvre la porte aux malfaiteurs exposant les organisations de santé à plusieurs menaces. Dans ce contexte, nous proposons une ontologie de domaine qui vise à modéliser la sécurité cyber et physique dans les systèmes de santé. L'ontologie est conçue pour supporter un modèle de propagation d'impacts et met en évidence les interactions cyber-physiques entre les actifs hospitaliers et les conséquences de ces relations hybrides sur les éventuels incidents. Ces travaux ont été menés dans le contexte du projet H2020 SAFECARE et ont été publiés dans 1 journal international (IEEE ACCESS'22), 2 conférences internationales (CAISE'21, RCIS'20), 1 chapitre d'ouvrage et 2 livrables.

¹<https://hostingdata.co.uk/nosql-database/>

Approche dirigée par les données pour l'intégration de big data dans les systèmes BI

Les systèmes NoSQL sont couramment utilisés pour stocker et gérer les big data grâce à des modèles de données flexibles et basés sur des systèmes distribués. Cependant, afin de résoudre les défis des big data, les systèmes NoSQL ont abandonné les caractéristiques fondamentales des BDR, essentiellement : un schéma et des contraintes d'intégrité définis au préalable. Ces systèmes sont, soit à schémas flexibles, soit sans schéma. Ainsi, l'exploitation de données NoSQL pour la BI nécessite de revoir l'ensemble de l'architecture afin de tenir compte de l'hétérogénéité de ces données. De plus, l'extraction des données pertinentes pour la prise de décision, requiert souvent l'accès à plusieurs sources (en l'occurrence à plusieurs collections dans les BD orientées documents) ; il est donc indispensable d'assurer le lien approprié entre les collections sources. Alors que la jointure entre tables dans les BDR est facilitée grâce à la présence de clés, dans les BD orientés documents (et de manière générale dans les BD NoSQL), les collections sont loin d'avoir une clé exacte en raison de l'absence de contraintes d'intégrité. Dans ce contexte, nous proposons une approche dirigée par les données qui permet d'intégrer des sources orientées documents, pour la BI. Notre approche part de plusieurs collections sources et vise à automatiser la détection des attributs de jointure entre ces collections en l'absence des contraintes d'intégrité. Ces travaux de recherche ont été menés dans le cadre de la thèse de doctorat de M. Souibgui (2018-2022) et ont été publiés dans 1 journal international (DKE'22) et 3 conférences internationales (KES'19, RCIS'20, HICSS'21).

Mots clés : Système d'information, Données massives, Aide à la décision, BD NoSQL, Développement dirigé par les modèles, Transformation de modèles, Raffinement de modèles, Systèmes de santé cyber-physiques, Sécurité cyber et physique, Assets cyber et physiques, Ontologie, Business Intelligence, ETL, Data driven, BD orientées documents, Découverte d'identifiants et de références.

Abstract

This HDR thesis summarizes approaches and tools developed during the last nine years of my research activities as well as future directions of my research projects. It outlines our contributions in the field of Information Systems (IS), security in cyber-physical systems, and Business Intelligence & Analytics (BI&A). We have focused on the modeling and development challenges of IS leveraging NoSQL data stores. We have also considered the problem of integrating NoSQL data into BI&A systems. Additionally, within the SAFECARE, H2020 project, we have worked on modeling security in healthcare systems. Our contributions aim to strengthen decision-making based either on models, data, or both. To deal with the development of IS and managing security in healthcare cyber-physical systems, we proposed model-based approaches, whereas the integration of NoSQL data into BI&A systems is mainly data driven.

Model-Based Approaches

- **Model Driven Development of Big Data Information Systems.** For many decades, the storage and the use of data have mainly relied on Relational Databases (RDB). With the advent of big data, the volume of data has exploded; the variety has increased causing several issues related to digital transformation, whether in terms of storage, data exploitation, cost or performance. Hence, new data management systems highly efficient, and easy to scale called NoSQL systems have appeared. In these systems, the data is organized according to 4 different families, namely, **key-value**, **document-oriented**, **column-oriented**, and **graph-oriented**. Today, in addition to the different existing relational solutions, there are more than 225 different NoSQL solutions². It is difficult to determine the most suitable solution that meets both functional and technical requirements. Besides, transferring the database from one solution to another is a heavy and costly process. Inadequate choices can lead to scalability, data consistency, and cost issues. Following an approach that supports the user with well-founded decision in order to carry out the IS project would be very helpful. To this aim, we propose a new approach that guides and automates the process of transforming a conceptual model into physical models related to the 4 NoSQL families, and to the relational model. This approach guides the choice of most suited models, and platforms, to meet business and functional requirements. We adopt the Model Driven Architecture (MDA) that provides a formal framework for metamodeling, and model transformation. Based on model refinement rules, our approach also aims to guide the selection at the logical level (which family, partial or complete nesting, schema normalization) and at the physical level considering technical criteria (costs, performance, etc.).

This work is carried out as part of 2 PhD (a defended PhD: A. Ait Brahim, 2015-2018; a PhD in progress: J. Mali, 2020-2023), and a research master thesis (A. Mokrani, 2017). Our contributions were published in many international, and national conferences (INFORSID'16, KMIS'16, DaWak'17, AICSSA'17, ICEIS'17, CIBSE'17, PDPTA'18, DEXA'20, EGC'20, BDA'20, RCIS'22).

- **Modeling Security in Healthcare Cyber-Physical Systems.** Hospitals and health organizations are among the most critical and vulnerable cyber-physical infrastructures. Healthcare organizations can now leverage recent technological advancements like wearable sensors and remote patient monitoring in order to offer more personalized services. Additionally, they can utilize IS to provide patients and partners with updates on health services, resource availability (such as beds and medical personnel), and controlled access to patients' data through open platforms. Unfortunately, these new technologies that rely on common communication interfaces and standards, enhance security breaches and open the door to hackers exposing hospitals to several threats. In this context, we provide domain ontology for modeling security in healthcare cyber-physical systems. This ontology is designed to support an impact propagation model, and highlights cyber-physical interactions among hospital assets and the consequence of these hybrid relationships on incidents they may encounter. This work is part of the H2020 SAFECARE project, and was published in an international journal and conferences (RCIS'20, CAISE'21, IEEE Access'22), 1 book chapter, and 2 deliverables.

²<https://hostingdata.co.uk/nosql-database/>

Data Driven Approach for Big Data Integration in BI Systems

NoSQL systems are commonly used to store and manage big data through flexible data models, and distributed systems. However, in the rush to solve big data challenges, NoSQL systems have abandoned some of the core features of relational databases, basically, predefined schema and integrity constraints. Exploiting NoSQL data for BI requires reviewing the entire BI architecture to deal with the heterogeneity of big data. In fact, fetching relevant data that meets the decision-maker requirements, often needs to access more than one data store, thereby needs to join data stores. While joining tables in relational data sources is straightforwardly owed to the availability of a precise join key, in NoSQL sources, like, document stores, collections are the furthest from having an exact join key due to the absence of integrity constraints. In this context, we propose a new approach that aims to extract, transform, and load document-oriented data sources. We provide a multi-source approach that enables automatic detection of join attributes between multiple collections despite the lack of integrity constraints. This work was carried out as part of M. Souibgui's PhD (2018-2022) and was published in an international journal and conferences (KES'19, RCIS'20, HICSS'21, DKE'22).

Key words: Information Systems, Big data, Decision Support, NoSQL Data Stores, Model Driven Development, Model Transformation, Model Refinement, Health Cyber-Physical Systems, Cyber and Physical Security, Cyber and Physical Assets, Ontology, Business Intelligence, ETL, Data Driven, Document Stores, Identifiers and References Discovery.

1	Introduction	13
1.1	General Context	13
1.2	Contributions	14
1.3	Manuscript Outline	17
2	Model Driven Development of Big Data Information Systems	19
2.1	Related Work	19
2.1.1	From Relational to NoSQL DB	20
2.1.2	Transforming a Conceptual Model to NoSQL DB	20
2.1.3	NoSQL DB Choice Orientation	21
2.1.4	Discussion	21
2.2	Overview of our Approach	22
2.3	Model Driven Approach for IS Development	22
2.3.1	Conceptual PIM to Logical PIM Model Transformation	23
2.3.2	Logical PIM to PSM Model Transformation	29
2.4	ModelDrivenGuide: Guiding NoSQL-based IS Development	33
2.4.1	Overview of our Approach	34
2.4.2	A Common Metamodel to Unify the 5 Families of Models	34
2.4.3	Common Model Refinement	36
2.4.4	Completeness of our Approach	38
2.4.5	Smart Data Model Search Optimizer	39
2.5	Experimental Study	43
2.5.1	Metamodels and Transformation Rules Implementation	43
2.5.2	ModelDrivenGuide Implementation	45
2.6	Conclusion	48
3	Modeling Security in Healthcare Cyber-Physical Systems	49
3.1	Problem Statement	49
3.2	Related Work	50
3.2.1	Risk & Threat Modeling Framework	50
3.2.2	Attack & Incident Modeling Framework	51
3.2.3	Discussion	51
3.3	An Ontology for Security Management in Healthcare CPS	52
3.3.1	Knowledge Elicitation	52
3.3.2	SafecareOnto	54
3.3.3	Core Ontology Conceptualization	55
3.4	Implementation	59
3.4.1	SafecareOnto Implementation	59
3.4.2	Monitoring Cyber-Physical System	59
3.4.3	Incident Impact Propagation	60
3.5	Conclusion	60
4	Data Driven Approach for Document Integration in BI Systems	63
4.1	Related Work	63
4.1.1	NoSQL Data Integration	64
4.1.2	Data Discovery	65
4.1.3	Ontology and Schema Matching Approaches	66

4.1.4	OLAP Analysis over Document Stores	68
4.1.5	Discussion	68
4.2	Document Stores Integration Approach	69
4.2.1	Overview of our Approach	69
4.2.2	Schema Extraction	71
4.2.3	Schema and Data Integration	72
4.2.4	OLAP Analysis	73
4.3	IRIS-DS: an Approach for Identifiers and References DIScovery in Document Stores	75
4.3.1	Discovery of Candidate Identifiers	75
4.3.2	Identifying Candidate Pairs of Identifiers and References	78
4.3.3	The IRIS-DS Algorithm	80
4.3.4	Case Study	82
4.4	Experimental Study	83
4.4.1	Technical Architecture of our Prototype	84
4.4.2	Data Collection and Preparation	84
4.4.3	Evaluation Protocol	86
4.4.4	Experimental Results	86
4.5	Conclusion	88
5	Conclusion and Research Perspectives	89
5.1	Summary	89
5.1.1	Model-Based Approaches	89
5.1.2	Data Driven Approach for Big Data Integration in BI Systems	90
5.2	Perspectives	90
5.3	Future Research Projects	91
A	Curriculum Vitae of Faten Atigui	105

2.1	Types of field reference	33
2.2	Generated models	46
3.1	Primary asset identification	53
3.2	Assets description	54
3.3	Structural patterns	56
4.1	The main ETL operations for document stores	74
4.2	Initial list of candidate <i>identifiers</i> with their scores	84
4.3	Minimal and maximal depth within the considered datasets	85
4.4	IRIS-DS results for candidate <i>identifiers</i> discovery in TPC-H, TPC-E, and Twitter collections	87
4.5	Importance of the depth feature: CustomersAccounts collection	88
4.6	IRIS-DS results for candidate pairs discovery in TPC-H, TPC-E, and Twitter collections	88

1.1	Summary of my main contributions	16
2.1	Model driven approach for modeling, guiding, and implementing NoSQL-based IS	23
2.2	Conceptual PIM metamodel	25
2.3	N-ary link at the logical level	26
2.4	Logical PIM metamodel	28
2.5	Conceptual PIM to logical PIM QVT mapping rules - «Main» relation	29
2.6	Conceptual PIM to logical PIM QVT mapping rules - «ClassToTable» relation	30
2.7	Conceptual PIM to logical PIM QVT mapping rules - «N-aryLinkToTable» relation	30
2.8	PSM MongoDB metamodel	31
2.9	ModelDrivenGuide: a heuristic to generate data models	34
2.10	Driving Example	35
2.11	Merge: $m(C, O, ref_{C \rightarrow O})$	35
2.12	Split: $s(O, price)$	35
2.13	Logical PIM - common metamodel for the 5Families of data models	35
2.14	5Families model refinement - Concepts' rows merge QVT rule	37
2.15	Particular case: cycle	37
2.16	5Families model refinement - Row's split QVT rule	38
2.17	Two levels of the graph of possible solutions	40
2.18	Ecore metamodels: conceptual PIM, logical PIM, and MongoDB PSM	44
2.19	Class diagram, 5Families & MongoDB XMI models	44
2.20	ConceptualMMToLogicalMM QVT transformation	45
2.21	LogicalMMToMongoDBMM QVT transformation	45
2.22	Number of data models	47
2.23	5FMHeuristic - Number of data models	47
2.24	5FMHeuristic - Number of data models wrt. nb of concepts varying keys (5 queries)	48
3.1	Data acquisition methodology phases	53
3.2	Assets relationships extracted from cyber and physical architectures	54
3.3	The conceptual view of SafecareOnto	54
3.4	SafecareOnto implementation in Protégé	59
3.5	Query answer result	60
3.6	Incident's impact propagation on scenario assets	61
4.1	Our approach general architecture	70
4.2	Example of document flat schema generation	71
4.3	IRIS-DS general architecture	75
4.4	Example of the <i>Cliff</i> method applied to the ranked scores of candidate <i>identifiers</i>	77
4.5	Example of the input graph for <i>node2vec</i>	79
4.6	Example of JSON documents from the TPC-H benchmark	82
4.7	Node2vec input	83
4.8	Technical architecture of our prototype	85
4.9	Impact of the dataset size: <i>Orders</i> collection	87

1.1 General Context

The digital transformation of companies and society as a whole has led to a significant increase in the generation and accumulation of data from various computing devices. This has led to an increase in data volume, variety, and velocity, giving rise to the concept of big data. The evolution of these data trends has necessitated the integration of big data techniques, analytics, and operations into existing systems. Information Systems (IS) have been impacted by this evolution and must be capable of effectively managing the challenges posed by data volume, variety, velocity, as well as ensuring data quality and security.

Indeed, for many decades, the storage and management of data in IS has primarily relied on Relational DataBases (RDB), which have demonstrated their limitations in handling big data. Relational Database Management Systems (RDBMS) face two significant challenges when it comes to accommodating big data features. Firstly, horizontal scaling poses a challenge as RDBMS were originally designed for single-server configurations [86]. Scaling a relational database involves distributing it across multiple servers, which brings about financial and technical constraints. Managing tables across different servers remains complex. Secondly, the rigid model definition required by RDBMS during database creation does not align with the dynamic nature of big data. In big data scenarios, users need the flexibility to easily integrate new data, which is not guaranteed by RDBMS. Modifying the relational model incrementally without impacting performance or disrupting the database operation would be a difficult task.

To address these limitations, IS storage technologies have evolved, introducing highly efficient database management systems capable of handling the volume, variety, and velocity of big data. These new systems are commonly known as "Not Only SQL" (NoSQL) systems. NoSQL refers to a category of database management systems that go beyond traditional relational systems associated with the SQL language by accommodating more complex data structures. NoSQL systems are highly scalable, schema-free, and built on distributed systems, making them suitable for scaling and sharding. The NoSQL models encompass four main families: **key-value oriented**, **column-oriented**, **document-oriented**, and **graph-oriented**.

A crucial stage in the development of IS is to provide a conceptual model that accurately reflects user requirements. This conceptual model serves as the foundation for implementing an efficient system at the physical level. The widespread adoption of NoSQL DBMS has presented a challenge in IS projects, as it requires leveraging NoSQL systems while still considering relational systems and the importance of data models. Data models play a crucial role in enabling efficient and accurate data exploitation, including querying IS data and integrating it into Business Intelligence and Analytics (BI&A) systems.

BI&A systems aim to leverage the data stored in a company's internal IS to support decision-making processes. BI&A refers to the technologies, processes, and practices used to gather, store, analyze, and present data in order to support decision-making and provide insights for business operations [74, 135]. Data Warehouses (DW) play a central role in BI&A systems. Inmon's definition [74] characterizes a data warehouse as "a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision-making process". In BI&A systems, data is collected from both internal operational databases within the company and external data sources. This data is often heterogeneous, meaning it comes in different types and formats. Before it can be loaded into the DW, it undergoes a process called Extract, Transform, and Load (ETL) [130]. The latter performs three basic functions: (i) extraction from data source; (ii) extracted data undergoes transformations (data cleansing and reformatting, data matching, aggregation, etc.) to ensure it is in the appropriate format and structure for querying and analysis; (iii) the resulting data set is loaded into the target system, typically the DW.

With the rise of NoSQL databases, there has been a question regarding their relationship with BI and their applicability in data warehousing. Due to their schema-free nature and distributed systems, NoSQL systems offer scalability and ease of scaling and sharding. However, in the pursuit of addressing the challenges posed by big data and high volumes of concurrent users, NoSQL databases have abandoned some of the fundamental features of relational databases, such as predefined schema and integrity constraints.

These features are crucial for ensuring data consistency and reliability in traditional BI systems. The schema-less nature of NoSQL data stores requires careful consideration of data modeling, data quality, and data integration techniques to ensure that meaningful insights can be derived from the stored data. To effectively utilize NoSQL databases for decision-making, it is necessary to reevaluate and adapt the entire BI&A architecture. Therefore, leveraging NoSQL databases in BI&A requires a holistic approach that addresses the specific challenges posed by schema-free and distributed data stores, while still maintaining the principles of data quality, integration, and analysis that are essential for effective decision-making..

In the context of IS and BI&A, three significant issues have captured our attention. The first issue revolves around the development of big data IS, specifically the modeling stage, which is critical in the overall development process. Developing an IS that effectively handles big data requires significant efforts of how to model and structure the data. Therefore, new modeling techniques and methodologies need to be explored to ensure the efficient and effective management of big data in IS.

The second issue pertains to the integration of NoSQL data into BI&A systems. Integrating NoSQL data into these systems requires the development of new approaches and tools to handle the characteristics of NoSQL data, such as its schema-less nature and distributed architecture. To effectively exploit NoSQL data stores for decision-making, it is necessary to review and adapt the architecture of BI&A systems. This includes rethinking ETL processes, metadata management strategies, and analytical techniques to accommodate the diverse and evolving nature of NoSQL data. By addressing these challenges, organizations can leverage the full potential of NoSQL systems in their BI&A initiatives and gain valuable insights from their big data assets.

Moreover, in the context of the H2020 SAFECARE project, our focus is on addressing the critical issue of security in healthcare cyber-physical systems. As healthcare organizations increasingly adopt digital technologies such as electronic patient records, wearable sensors, and remote patient monitoring, the integration of cyber and physical infrastructure becomes crucial. They need to share information about health services, resource availability, and patient data through open and controlled platforms.

These advancements enable healthcare providers to deliver more personalized services and improve patient care. However, they also introduce new challenges related to security and privacy. The reliance on common communication interfaces and standards make hospitals vulnerable to security breaches and expose them to various threats. Given the nature of healthcare services, where human lives and well-being are at stake, the potential damages from security breaches can be severe. According to the IBM data breach report¹, healthcare organizations have the highest costs associated with data breaches, with an average of \$6.45 million. To address these challenges and enhance the efficiency of security solutions, it is crucial to examine all facets of the problem.

This HDR thesis outlines my contributions in the field of information systems, security in healthcare systems, and business intelligence & analytics. The research presented in this thesis encompasses the period from 2014 to early 2023 and reflects my work as an Associate Professor at CNAM Paris (CEDRIC Laboratory, EA 4629, ISID Team). Throughout this period, I have collaborated with colleagues, supervised five PhD students, and mentored over twenty master students. These collaborations have been instrumental in conducting research and generating the results presented in this thesis. The thesis includes a Curriculum Vitae in Appendix A, providing additional information about my academic and research background.

1.2 Contributions

During my PhD thesis [15], I worked on model driven development of BI systems. We have proposed a model driven approach that aims to formalize and automate the process of developing BI systems from conceptual modeling to physical implementation. Our unified approach integrates both multidimensional data and ETL processes modeling. Also, we have proposed an approach that reduces the stored data over time. After that, I focused my research on issues related to model driven development of NoSQL-based IS,

¹<https://www.ibm.com/downloads/cas/ZBZLY7KL>

and modeling security in healthcare cyber-physical systems. Besides, I work on the problem of integrating NoSQL data in BI systems.

As depicted in Figure 1.1, my contributions are primarily focused on providing decision support. To achieve this goal, we have adopted both model driven and data driven approaches. The bidirectional link shown in the figure highlights the integral connection between these methods: models utilize data in various ways, while data influences and informs model creation. Model driven approaches rely on defined structures to interpret, analyze, and manipulate data. These approaches also facilitate user interactions, enabling structured queries and exploration of data for meaningful insights. On the other hand, data driven methods leverage models as foundational instruments for comprehending data, extracting insights, and facilitating decision-making. They contribute to standardizing data, ensuring consistency, and managing complexities within data. Furthermore, these models play a crucial role in maintaining data quality, integrating disparate sources, and harmonizing them into a coherent framework.

Our main goal is to facilitate and enhance the processes of modeling, developing, and managing security in IS, as well as the integration of big data in BI systems. These contributions form the foundation of my research and aim to advance the fields of IS, security in healthcare systems, and BI, and can be summarized as follows:

- A global model driven approach for the development of information systems: based on model transformation and refinement, this approach facilitates, automates, and guides the process of transforming a conceptual model into physical models. Our framework supports the mapping and adaptation of models across different DB, including the four major NoSQL families and the relational model.
- An ontology-based model for security in healthcare cyber-physical systems: to address the challenges of security, we have developed a domain ontology specifically tailored for modeling cyber and physical security in healthcare systems. The starting point was a set of real-world threat scenarios representing domain experts knowledge. An iterative process going through elicitation, formalisation and validation steps lead to an ontology for security in cyber-physical systems. This highlights the bidirectional relationship between model driven and data driven methodologies shown in Figure 1.1.
- A data driven approach for document integration in BI systems: we introduce a new approach that encompasses techniques for extracting, transforming, and loading data from various documents stores, enabling organizations to leverage their data for decision-making. The clear boundaries between data and models, which are explicit in RDB, are blurred in NoSQL DB. While RDBs require creating a schema before data insertion, NoSQL DB accommodate gradual data insertion alongside schema. Indeed, NoSQL sources may encompass a schema, or part of it. Within our approach, the first step involves schema extraction (see Section 4.2) which allows for a better characterization of these sources for integration purposes, further concretizing the bidirectional connection between data and models.

Model Driven Development of Big Data Information Systems. For more than a decade, NoSQL systems have been extensively utilized for storing and managing big data. Presently, there are over 225 diverse NoSQL solutions² in addition to the various existing relational options. With the multitude of systems, methodologies, and software development life cycles available, it is crucial to adopt a comprehensive approach that assists the IT team in making well-informed decisions to successfully carry out information system projects. For this, we propose a novel approach that guides and automates the process of transforming a conceptual model into physical models corresponding to the four NoSQL families and the relational model. We employ the Model Driven Architecture (MDA)³ [80] as it provides a formal framework for metamodeling and model transformation. First, we propose a logical metamodel that offers a unified representation of the five families of data models, encompassing the four NoSQL families and the relational model. Subsequently, we present a set of Query-View-Transformation (QVT) rules that automates the entire transformation process. The first set of rules aims to map the conceptual model (represented by a

²<https://hostingdata.co.uk/nosql-database/>

³<https://www.omg.org/mda/index.htm>

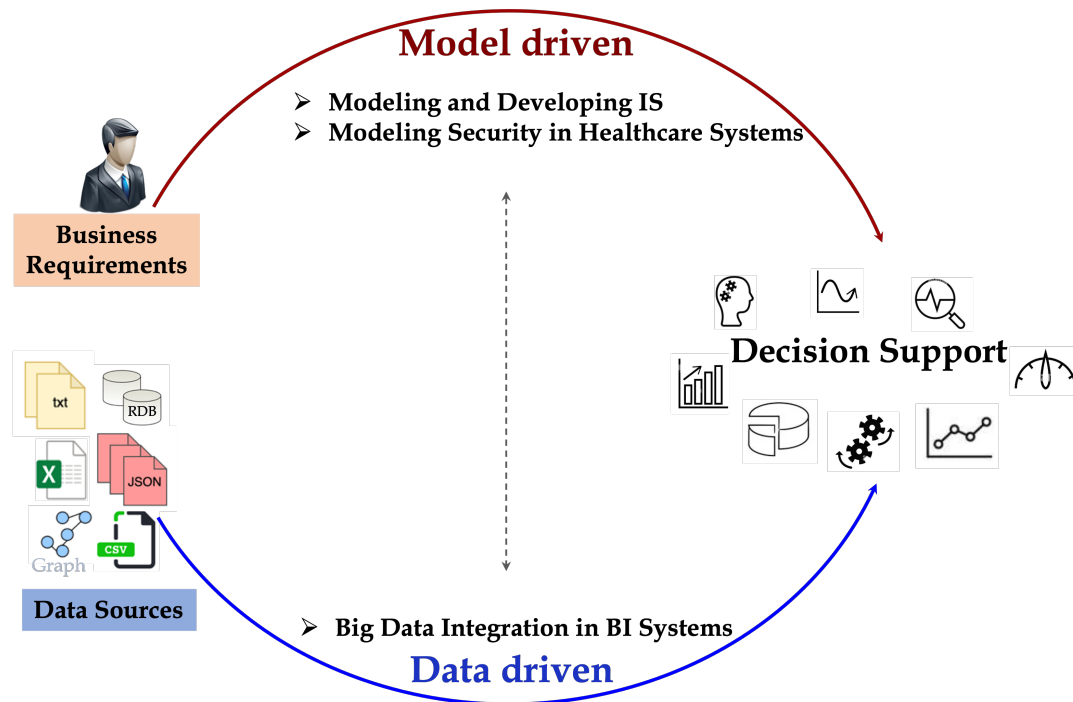


Figure 1.1: Summary of my main contributions

UML class diagram) into the logical model, which adheres to the common **5 Families** metamodel. After that, this metamodel is further transformed into physical models tailored to the respective platforms associated with the different families.

Furthermore, we propose a new approach that facilitates the selection of the most suitable models and platforms to fulfill business and functional requirements. Leveraging the **5 Families** metamodel, our approach, called **ModelDrivenGuide**, offers logical modeling capabilities that enable the refinement of models for generating optimized model variants. By utilizing refinement rules, our approach integrates a functional decision-making process that incorporates the specific use case, thereby guiding the choice of relational and NoSQL solution(s) that best align with the business requirements. We propose a formalism for addressing the denormalization problem, providing complexity bounds along with merge and split rules. Additionally, we introduce a heuristic technique that effectively reduces the search space when generating data models. Our approach subtly combines the aspects of conceptualization and optimization within the IS transformation process.

Modeling Security in Healthcare Cyber-Physical Systems. Healthcare infrastructures are frequently targeted by various threats that exploit their vulnerabilities, resulting in high-impact incidents. These incidents arise from a combination of cyber and physical attacks. To address the need for innovative solutions that integrate both cyber and physical security features, we propose an ontology-based solution as part of the H2020 SAFECARE project. The SAFECARE ontology is specifically designed to support an impact propagation model and emphasize the interdependence between cyber and physical aspects of hospital assets. It aims to capture the consequences of these hybrid relationships on the incidents that healthcare infrastructures may encounter. Given the project’s requirements, we engaged in extensive discussions with experts to incrementally develop the ontology. Our approach involved constructing a modular ontology centered around the asset management module, which then extended to encompass protection and impact modules. This modular structure proved valuable as it accommodated the gradual acquisition of domain knowledge, considering the diverse stakeholders involved in different locations.

Data Driven Approach for Big Data Integration in BI Systems. In the rush to address the challenge

posed by big data and large number of concurrent users, NoSQL databases have abandoned certain fundamental features of relational databases, specifically predefined schemas and integrity constraints. However, leveraging NoSQL data for BI necessitates a reevaluation of the BI architecture to effectively handle the heterogeneity inherent in big data. To tackle this challenge, we propose an approach that considers both schema-less data sources and the specific requirements of decision-makers, enabling efficient exploration, integration, and analysis of document stores.

Our particular focus lies on the ETL stage, which serves as the foundation of our approach. During this stage, one of our primary objectives is to automatically join multiple document stores. In practice, retrieving relevant data that fulfills the requirements of decision-makers often necessitates accessing data from more than one document store, requiring a dedicated join operation. While joining tables in relational data sources is relatively straightforward due to the availability of precise join keys, the situation is more complex in document stores. Due to the absence of integrity constraints, collections in document stores lack exact join keys. Consequently, identifying the "joinable" fields to establish connections between two document stores poses a challenging task. To address this challenge, we introduce IRIS-DS (Identifiers and References DIScovery in Document Stores), a novel approach designed to automatically discover pairs of join keys (*identifier*, *reference*) from multiple document stores.

1.3 Manuscript Outline

The manuscript is structured as follows:

In **Chapter 2 - Model Driven Development of Big Data Information Systems**, I present our contributions dedicated to the development of NoSQL-based IS. After the reports on related work about the storage and modeling problems of data using NoSQL systems (Section 2.1), I present an overview of our model driven development approach (Section 2.2). Then, Section 2.3 details the different metamodels, and model transformation rules. Section 2.4 provides the ModelDrivenGuide metamodel and the model refinement rules. In Section 2.5, I present the experimentation before concluding in Section 2.6.

In **Chapter 3 - Modeling Security in Healthcare Cyber-Physical Systems**, I present the problem of cyber and physical security in healthcare organizations and our contributions in this context. Section 3.1 shows an example of cyber-physical scenario attack. Then, Section 3.2 reports on related works. In Section 3.3, I present the knowledge acquisition methodology, and our ontology for cyber-physical security in healthcare systems. Finally, I present the implementation of the ontology in Section 3.4 and I conclude in Section 3.5.

In **Chapter 4 - Data Driven Approach for Document Integration in Business Intelligence Systems**, I present our contributions in the context of BI, and how to integrate NoSQL document stores in BI systems. In section 4.1, I report on related works that have dealt with the problem of NoSQL data integration and their analysis in BI systems. In Section 4.2, I present our approach for document stores integration in BI. Then, Section 4.3 details the core stage of our approach, which is the identifier and references automatic discovery. Finally, I present our experimental study in Section 4.4 and I conclude in Section 4.5.

Chapter 5 - Conclusion summarizes my contributions and highlights some future directions.

This chapter presents the research conducted as part of two PhD thesis. The first one is A. Ait-Brahim's PhD, defended in October 31, 2018 that I co-supervised with Pr. G. Zurfluh (IRIT - Institut de Recherche en Informatique de Toulouse) and Dr. F. Abdelhedi (CBI² - Trimane company). The second one is J. Mali's PhD (in progress) that I co-supervise with Dr. S. Ahvar (ISEP - Institut Supérieur d'Electronique de Paris), Dr. N. Travers (DVRC - De Vinci Research Center & CEDRIC), and Dr. A. Azough (DVRC). The result of this work was published in [2]/12, [3]/18, [4]/20, [5]/15, [6]/13, [7]/14, [20]/23, [35]/16, [93]/5, [94]/8, [95]/24.

With the advent of big data, the volume of data has exploded; the variety has increased causing several issues related to digital transformation, whether in terms of storage, data exploitation, cost or performance. For this, new data management systems, highly efficient, and easy to scale called NoSQL systems have appeared. In these systems, the data is organized according to 4 different families of structures, namely, **Key-Value (KV)**, **Document-Oriented (DO)**, **Column-Oriented (CO)**, and **Graph-Oriented (GO)**. Today, with the multitude of existing models and solutions, it is difficult to determine the most suitable model and solution that meet both functional and technical needs. Besides, transferring the database from one solution to another is a heavy and costly process. Inadequate choices can lead to scalability, data consistency, and cost issues. Following an approach that supports the user with well-founded decision in order to carry out the IS project would be very helpful.

This chapter presents the two contributions that we proposed to address these problems:

- A model driven approach that formalizes and automates the process of transforming a conceptual model into physical models related to the 4 NoSQL families. Our approach is based on:
 - Three levels of metamodels : UML class diagram metamodel at the conceptual level, a common 4 families metamodel at a logical, and physical metamodels dedicated to different implementation platforms,
 - A set of QVT¹ rules that automates the transformation process from the conceptual level to the logical one, and from the logical level to the different physical models.
- A model driven guide that completes the first approach and subtly combines conceptual modeling, with optimization in databases transformation process. This contribution is based on:
 - A common metamodel dedicated to support all families (relational model, and the 4 NoSQL families),
 - A set of QVT model refinement rules that merge or split concepts to ensure normalization, and partial or total nesting of schema,
 - A functional decision-making process that integrates the use case to guide the choice of most suited models, and platforms, to meet business and functional needs,
 - A heuristic that reduces the search space of the generated models.

This chapter is organized as follows: Section 2.1 shows the related work. Section 2.2 presents our approach. Section 2.3 details our model driven approach for NoSQL based IS development. Section 2.4 is dedicated to the model driven guide. Section 2.5 shows experiments before concluding in Section 2.6.

2.1 Related Work

In the literature, several works have focused on the storage and modeling problems of data using NoSQL systems. Most of the studies carried out on NoSQL DB have proposed either (i) a comparative

¹<https://www.omg.org/spec/QVT/About-QVT/>

study between RDB and NoSQL DB and/or how to transform relational data into NoSQL data or (ii) how to transform a conceptual schema into a specific NoSQL DB; (iii) while very few studies have considered the problem of how to guide the choice of the adequate model and/or platforms.

2.1.1 From Relational to NoSQL DB

Existing approaches, like [76], have proposed comparison between relational and NoSQL DB, other contributions have studied the problem of data migration from a RDB into a NoSQL DB. In [89], the authors propose a set of rules that transform a relational model into HBase model. The foreign keys are transformed using column family of references. In [13], the authors propose to transform a relational model into MongoDB document-oriented model. They provide a semi-automatic process, in which, the user chose a list of MySQL tables to be transformed into MangoDB collections. Then, the process is executed to automatically generate text files in Pentaho Data Integration² format in order to load the data into MongoDB. In [103], the authors present a process to migrate a relational model to a column-oriented DB. This process consists of two steps: (i) building a model of a RDB and (ii) transforming this model into a column-oriented model.

In [67, 87], the authors propose to denormalize RDB into NoSQL DB in order to optimize queries. They provide an algorithm that automatically selects the optimal NoSQL DB based on a RDB and a set of queries provided as inputs.

In [113], the authors provide an automatic framework that aims to migrate MySQL RDB to a MongoDB database. The main features of this framework is to preserve the way the data is modeled in the RDB, then to use an equivalent data model to store this data in MongoDB . Similar works [129, 26, 87] provide a set of mapping rules between the relational model and a column-oriented or document-oriented models.

2.1.2 Transforming a Conceptual Model to NoSQL DB

Some studies have focused on the transformation of a conceptual model into a NoSQL DB. Mainly, the conceptual model, is either the Entity-Relationship (ER) model, or UML class diagram, or domain-specific model like the multidimensional model in BI systems.

In [66], the authors propose to transform an ER model into a column-oriented model, based on the definition of a CO schema using primary and foreign keys, and on transformation rules. Similarly, [40] suggests a query-driven approach for modeling *Cassandra* starting from an ER model. They define dedicated logical and physical models, as well as transformation rules between ER and Cassandra models. The work in [48], presents a conceptual transformation approach that maps an ER model into one of the 4 NoSQL families with an abstract formalization of the mapping rules.

Few authors have studied the transformation of UML class diagram to a physical NoSQL model. In [90], the authors propose an MDA approach that transforms UML class diagram into HBase model. They present UML class diagram metamodel, HBase column-oriented metamodel, and define the mapping rules between the two metamodels. These rules aim to transform a class diagram directly into a HBase-dedicated model. Similar work is shown in [56], and proposes a model driven approach to transform a UML class diagram into Cassandra-dedicated model. In [46] the authors propose to transform a UML class diagram, and associated OCL constraints into a graph-oriented model, the proposed mapping rules are specific to graph-oriented family only.

In BI systems, existing work propose to store data cube using a NoSQL DB. In [42], the authors have studied how to transform a multidimensional schema (fact and dimension) into columns-oriented and document-oriented models. They propose 3 main transformation processes: (i) a total denormalization of the multidimensional model where the fact and the dimension are stored in the same target

²https://help.pentaho.com/Documentation/8.3/Products/Pentaho_Data_Integration

concept, (ii) transforming the fact and the dimension, into a target concept each one, (iii) normalizing dimension where the fact and each level of dimension is transformed into a target concept. In [50], the authors deal with the implementation of big data warehouses using column-oriented systems. They propose 3 levels of abstraction models, i.e., conceptual, logical, and physical, and 2 main ways to transform the multidimensional model into a CO logical model: (i) normalized transformation where the fact and the dimensions are transformed into column families, separately, and (ii) denormalized transformation where both the fact and the dimensions are transformed to one column family. The authors in [115] propose to denormalize the multidimensional schema and store it using one HBase table that contains 2 column family, the first one stores the fact and the second stores the dimensions.

2.1.3 NoSQL DB Choice Orientation

Few works have proposed to guide the choice of the NoSQL DB, and most of them are based on technical performance criteria. In [102], the authors present a comparative study of NoSQL DB. They present the features and the benefits of each family, a classification, comparison, and evaluation (based on design, integrity, indexing, distribution, and system) of different families of NoSQL DB. In [63], the authors present a comparative classification model and propose to link functional and non-functional requirements to the techniques of each family. The result is presented as a decision tree that guides the choice of the NoSQL system based on sharding, volume of data, and CAP properties [59].

2.1.4 Discussion

Existing approaches, like [89, 76, 13, 113] have discussed how to transform a relational model into a NoSQL physical models. These approaches respond, certainly, well to the concrete expectations of companies who want to store their large databases in NoSQL systems. But, the source of the transformation process, here a relational model, does not present the semantic richness that we find in big data and that we can express in a class diagram (in particular thanks to the different types of links between classes: aggregation, composition, inheritance, etc.). Regarding the conceptual to NoSQL transformation contributions, in the context of BI systems, [42, 50, 48] have proposed to transform a multidimensional model to a NoSQL model. Mainly, 2 NoSQL DBMS were considered: HBase column-oriented system and MongoDB document-oriented system. Although the starting point of the process (a multidimensional model) is a conceptual model, this model does not have the same features of class diagram in terms of complexity as it is composed of 2 two concepts (Facts and Dimensions) and a single type of link. In class diagram, we have to consider atomic and multivalued attributes, association, composition, aggregation, and inheritance relationships as well as association classes. On the other hand, the works presented in [90, 56, 46] propose an MDA transformation process of class diagram to a NoSQL physical model. Mainly, these works propose a dedicated platform-model that target, one single NoSQL family (column-oriented [90, 56] or graph oriented [46]).

Besides, most of existing model transformation approaches provide rules that transform a source concept into a target concept (for instance, a class is transformed to a column family, in CO DB or a collection in DO DB). Regrettably, these approaches fail to leverage the overall schema flexibility provided by NoSQL databases and do not enable seamless transitions between different NoSQL families through transformations. Also, they do not consider splitting a source concept into several target concepts, or merging several source concepts into one target concept. Thus, the flexibility of models has not been sufficiently exploited to facilitate transformations and improve performances. Finally, the few existing works on NoSQL DB choice orientation are basically based on technical criteria and do not consider data model nor functional requirements.

To overcome these limitations, we propose an MDA approach that transforms a conceptual model (UML class diagram), and considers all the possible target data models: the relational model as

well as the 4 families of NoSQL DB (columns, documents, graphs and key-value). Automating the transformation process by considering all the possible data models would be of great help for the IS designer. However, the question of choosing the appropriate family model, whether it's normalized, denormalized, or something in between, and which platform would be the most suitable, still remains. For this, we propose a model driven guide that helps the designer to choose the most suited target SQL/NoSQL solution(s). The decision criteria are based on the context of use, functional, and technical requirements.

2.2 Overview of our Approach

We propose a new approach for modeling, guiding, and implementing NoSQL-based IS. Our approach is based on transformation rules starting from the conceptual model, then going from one logical model to another by refinement. In order to formalize the model and their transformation, we adopt the model driven architecture that offers 3 types of models, namely 1) the *Computation Independent Model* (CIM) which is the basic analysis model of the field of application that allows to describe the requirements; 2) the *Platform Independent Model* (PIM), called the design model, and describes the components of the system independently of platforms; 3) and the *Platform Specific Model* (PSM) which describes the components of the system using a precise technical platform. MDA also recommends transforming models by formalizing the transformation (or refinement) rules in language such as QVT, which is the OMG standard language for models transformation. Our approach provides a modeling framework based on those multiple dimensions of choice as illustrated in Figure 2.1.

The **PIM1** (*Platform Independent Model*) of the first level integrates the functional needs of the IS, both in terms of data and queries. This traditional UML class diagram serves as a basis for modeling the user requirements.

The **PIM2** is the second level independent model, common to the five families of models (NoSQL & relational). It allows to carry out *refinements* by generating all possible denormalized models through transformation rules. To reduce the search space, a heuristic keeps only the effective solutions by simplification based on the use case.

The **PSM_x** (*Platform Specific Model*) are obtained by the transformation of compatible PIM2 data models with the target data family (e.g. nesting for D0, rows for C0, edges for G0, etc.). Choices of sharding and indexing strategies are obtained based on a generic cost model for the 5 families. All solutions can be proposed and sorted in relevance order.

In this approach, as part of A. Ait Brahim's PhD, we have proposed an MDA approach which aims to transform a UML class diagram into a logical metamodel that covers the 4 NoSQL families. We have also proposed a set of transformations from the logical level towards different physical models related to different platforms. These contributions are detailed in Section 2.3. To complete these contributions, and as part of J. Mali's PhD, we propose to consider the relational model and the 4 NoSQL families. So we have adapted the logical metamodel to integrate all the 5 families. Also, we propose a set of refinement rules that aims to split or merge source concept to one or more target concept. Split and Merge operations enable partial and complete normalization and denormalization of the model, ensuring coverage of all possible modeling scenarios. Moreover, we propose, to help the designer throughout the IS project by providing a decision tree that aims to find the most suited models and platforms. These contributions are detailed in Section 2.4.

2.3 Model Driven Approach for IS Development

In this section, we present the transformation process which is based on metamodeling and model transformation. This process operates in 2 main steps: the first transforms the conceptual PIM into a logical PIM, the second transforms the logical PIM into different platform models (PSMs).

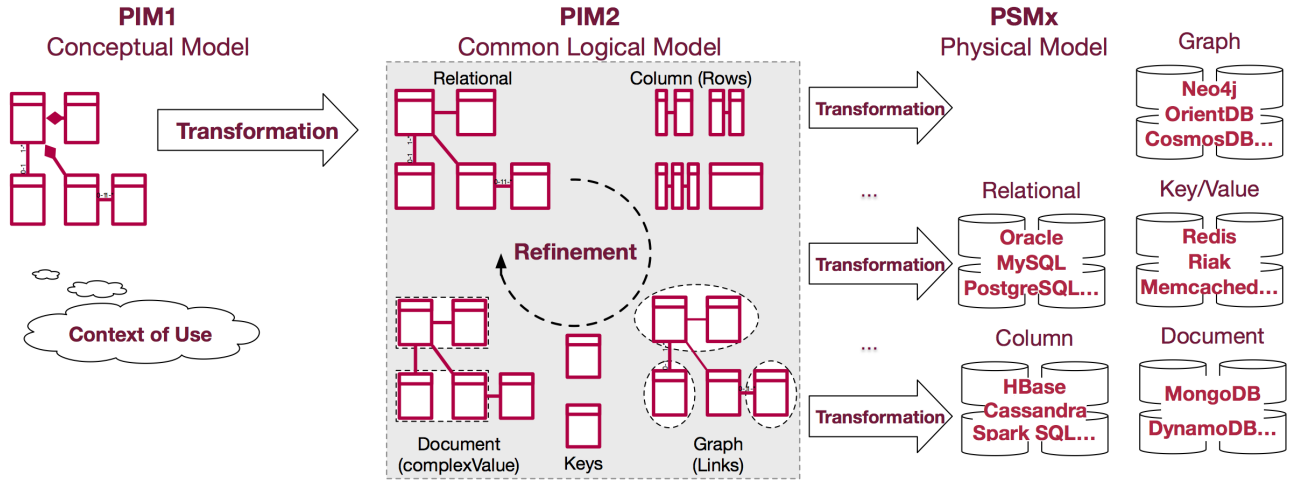


Figure 2.1: Model driven approach for modeling, guiding, and implementing NoSQL-based IS

2.3.1 Conceptual PIM to Logical PIM Model Transformation

In this section, we present the first transformation process that maps the conceptual PIM (UML class diagram) into the logical PIM (generic NoSQL model). We start by defining the source and the target models of this transformation, then we detail the transformation rules.

2.3.1.1 The source Metamodel: Conceptual PIM

First, we formalize the concepts present in the UML class data model as follows.

Definition 1: A class diagram is defined as (N, C, L, C^{asso}) where:

- N is the class diagram name,
- $C = \{c_1, \dots, c_n\}$ is a set of classes,
- $L = \{l_1, \dots, l_m\}$ is a set of links,
- $C^{asso} = \{C_1^{asso}, \dots, C_k^{asso}\}$ is a set of association classes.

We remind that a class defines the structure (attributes) and the behavior (operations) of a set of objects having common semantics and properties. As shown in Definition 2, our approach consider the structural features of a class only.

Definition 2: $\forall i \in [1..n]$, a class $c_i \in C$ is defined as (N, A^c) where:

- $c_i.N$ is the name that identifies the class,
- $c_i.A^c = \{a_1^c, \dots, a_q^c\} \cup \{Id^c\}$ is the set of the class attributes with $q \geq 1$, where:
 - * $\forall j \in [1..q]$, the attribute schema $\{a_j^c \in A^c\}$ is a couple (N, c) where:
 - $a_j^c.N$ is the name that identifies the attribute,
 - $a_j^c.c$ is the class that identifies the attribute type, c could be a predefined class as common data type classes like **String**, **Integer**, **Date**, etc., or a class defined by the user.
 - * Id^c is the object identifier, which is a particular attribute of c that we have defined in order to create references to access the objects of c . As all the attributes, Id^c has a name $Id^c.N$ and a type "OID".

A link relates two or more classes of objects. It expresses semantic connections between objects. Definition 3 presents this concept by considering the the four most common types of links, i.e., association, composition, aggregation, and inheritance.

Definition 3: $\forall i \in [1..m]$, a link $l_i \in L$ is defined as (N, T_y, CP^l) where:

- $l_i.N$ is the name that identifies the link,
- $l_i.T_y$ is the link type that could be association, composition, aggregation, and inheritance.
- $l_i.CP^l = \{cp_1^l, \dots, cp_f^l\}$ is a set of couples, with $f \geq 2$ is the degree of l_i . $\forall j \in [1..f], cp_j^l = (c, cr^c)$ where:
 - * $cp_j^l.c$ is a linked class,
 - * $cp_j^l.cr^c$ is the multiplicity of the side c . Note that, if no multiplicity indicated, $cp_j^l.cr^c$ will contain Null value. This is the case of the inheritance link and the association of 3 or more classes where the multiplicities are quite difficult to interpret and are mostly not precised.

In UML class diagrams, an association class has both the features of a class and those of a link. Thus, the definition of an association class (Definition 4) is made up of two parts: the first refers to the definition of a class (Definition 2) and the second refers to the definition of a link (Definition 3).

Definition 4: $\forall i \in [1..k]$, an association class $c_i^{asso} \in C^{asso}$ is defined as $(N, A^{C^{asso}}, CP^{c^{asso}})$ where:

- $c_i^{asso}.N$ is the name that identifies the association class,
- $c_i^{asso}.A^{C^{asso}} = \{a_1^{c^{asso}}, \dots, a_p^{c^{asso}}\} \cup \{Id^{c^{asso}}\}$ is the set of the association class' attributes with $p \geq 1$, where:
 - * $\forall j \in [1..p]$, an attribute $a_j^{c^{asso}} \in A^{c^{asso}}$ is defined by (N, c) where:
 - $a_j^{c^{asso}}.N$ is the name that identifies the attribute,
 - $a_j^{c^{asso}}.c$ is the class that identifies the attribute.
- $Id^{c^{asso}}$ is the link identifier, which is a particular attribute that identifies the different link of c^{asso} . As all the attributes, $Id^{c^{asso}}$ has a name $Id^{c^{asso}}.N$ and a type "OID",
- $c_i^{asso}.CP^{c^{asso}} = \{cp_1^{c^{asso}}, \dots, cp_l^{c^{asso}}\}$ is a set of couples, with $l \geq 2$. $\forall r \in [1..l], cp_r^{c^{asso}} = (c, cr^c)$ where:
 - * $cp_r^{c^{asso}}.c$ is a linked class,
 - * $cp_r^{c^{asso}}.cr^c$ is the multiplicity of the side c . Note that, if no multiplicity indicated, $cp_r^{c^{asso}}.cr^c$ will contain Null value. This is the case of the inheritance link and the association of 3 or more classes where the multiplicities are quite difficult to interpret and are mostly not provided.

We present all these concepts through the metamodel shown in Figure 2.2 proposed by the OMG³, and that we have adapted to our conceptual PIM. This metamodel as well as all the proposed metamodels are implemented and validated using the Eclipse Modeling Framework (EMF) platform as presented in Section 2.5.

As shown in Figure 2.2, the conceptual PIM metamodel shows the main elements of a UML class metamodel, specially, their structural features. A UML class diagram consists of **Classes**, **Links**, and **Association Classes**. A **Class** is composed of **Attributes** which are structural features. A link connects two or more classes, and has at least two ends (**LinkEnd**), each one represents a connection between the link and a class. Therefore, the structural part of a link is defined by its ends. As we presented in Definition 4, an Association Class is an element having both the features of a class and those of a link. So, an association class is considered as a link that has the same features of a class.

2.3.1.2 The Target Metamodel: Logical PIM

The target metamodel is shown in Figure 2.4. We present in Definitions 5, 6, and 7 the main concepts of this metamodel.

Definition 5: A database DB is defined as (N, T, R) where:

³<https://www.omg.org/spec/UML/2.4.1/About-UML/>

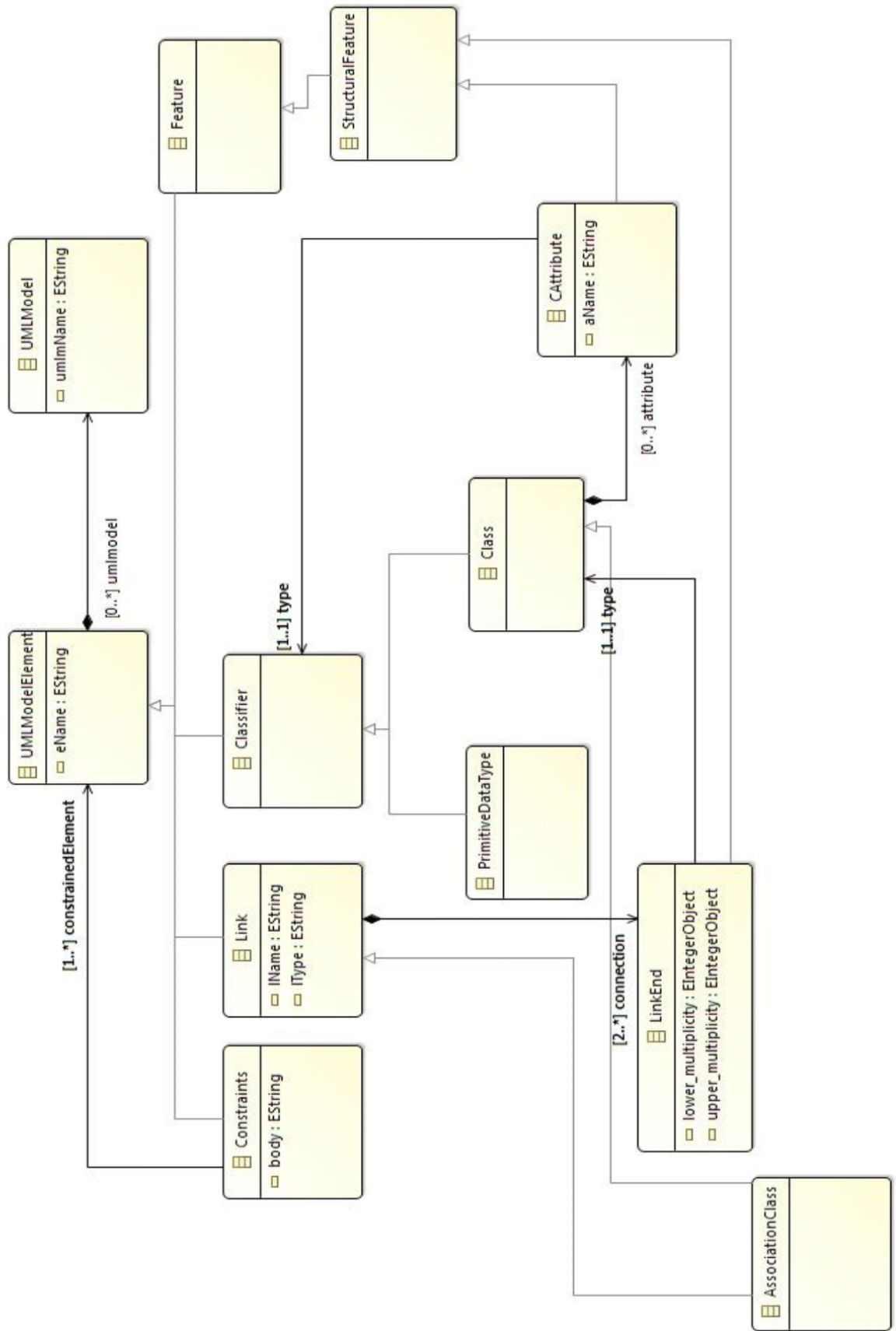


Figure 2.2: Conceptual PIM metamodel

- N is the database name,
- $T = \{t_1, \dots, t_n\}$ is a set of tables,
- $R = \{r_1, \dots, r_m\}$ is a set of binary relations.

A table is a grouping of rows that do not necessarily have the same schema. Each row is a couple (Identifier: Value), the value of a row is a set of couples (attribute: value). The attributes are either atomic of standard type like Integer, Float, String, etc. or complex, composed of a set of other attributes.

Definition 6: $\forall i \in [1..n]$, a table $t_i \in T$ is defined as (N, A^t) where:

- $t_i.N$ is the name that identifies the table,
- $c_i.A^t = \{a_1^c, \dots, a_q^c\} \cup \{Id^t\}$ is the set of attributes used to define the rows of t , where:
 - * $\forall j \in [1..q]$, the attribute schema $\{a_j^t \in A^t\}$ is a couple (N, T_y) where:
 - $a_j^t.N$ is the name that identifies the attribute,
 - $a_j^t.T_y$ is the attribute type,
 - * Id^t is the row's identifier, which is a particular attribute of t called *RID*. We will use this attribute later in the transformation (see Section 2.3.1.3) in order to create links between tables at the physical level.

We mention that at the logical level, we consider binary relationships only. Thus, a conceptual link of degree n (with $n > 2$) is transformed to a new table and n binary relations emanating from this table. For example, in Figure 2.3, the conceptual link L that relates classes A , B , and, C is transformed into a new table T , and three binary relations: R_A that relates T and A , R_B that relates T and B , and R_C that relates T and C . If L includes attributes, these are reported in table T .

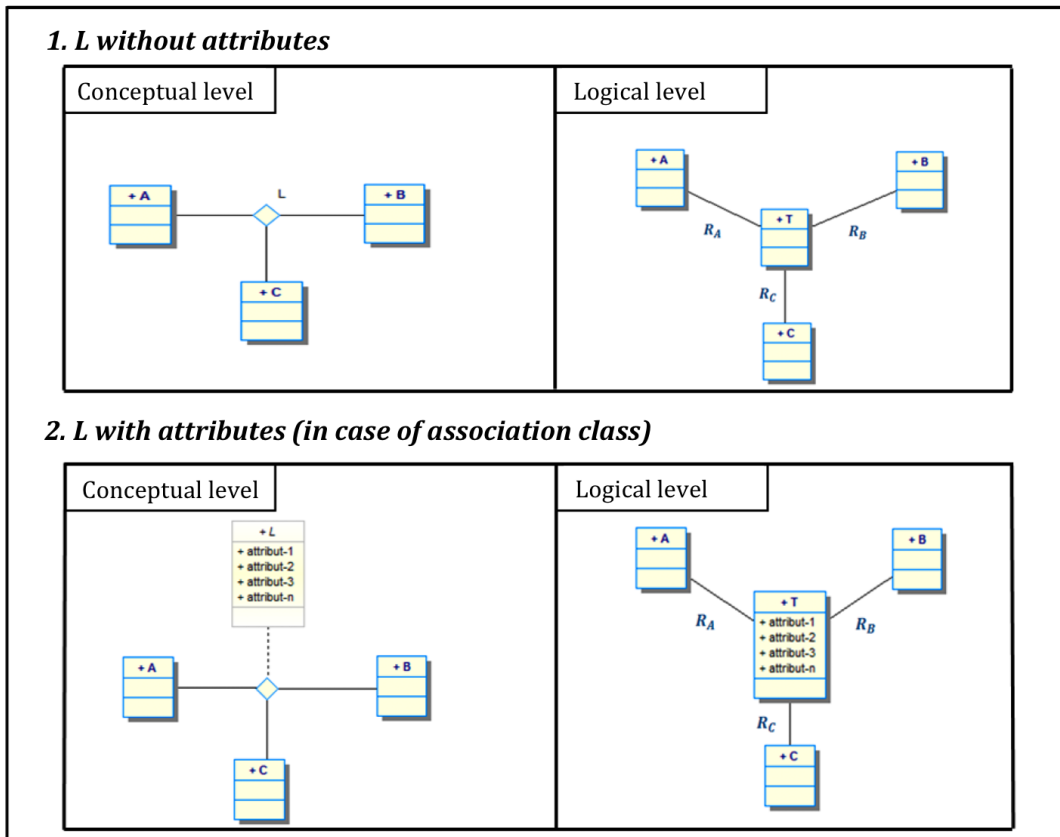


Figure 2.3: N-ary link at the logical level

We propose a formalization of this conceptual-to-logical mapping of n-ary links in Section 2.3.1.3.

Definition 7: $\forall j \in [1..h]$, a relation $r_j \in R$ is defined as (N, CP^r) where:

- $r_j.N$ is the name that identifies the relation,
- $r_j.CP^r = \{cp_1^r, cp_2^r\}$ is a set of 2 couples, where $\forall i \in 1, 2, cp_i = (t_i, cr^{t_i})$ where:
 - * $cp_i.t_i$ is a linked table,
 - * $cp_i.cr^{t_i}$ is the multiplicity of the side t_i .

We describe the concepts of our logical PIM presented in this section through the metamodel depicted in Figure 2.4. As shown in Definition 5, a database is made up of **Tables** and **Binary Relations**. A table is a grouping of **Rows**; each row contains an **Identifier** and a **Value**. A value is composed of one or more couples (**Attribute: Value**). An **Attribute** has a **Name** and a **Type**, and it has a **Value** which can be either **Atomic** or **Complex** (i.e., composed of other attributes), this is insured using the **{XOR}** constraint. Binary relationships allow linking the tables. A binary relationship have two ends (**RelationshipEnd**).

2.3.1.3 Transformation Rules: from Conceptual PIM to Logical PIM

In this section, we present the set of rules that automatically transforms the conceptual PIM (UML Class Diagram), i.e., the source model to the logical PIM, i.e., the target model.

- **Rule R1.** A Class Diagram CD is transformed into a database DB , where $DB.N = CD.N$
- **Rule R2.** A Class $c \in C$ is transformed into a table $t \in T$, where:
 - * $t.N = c.N$,
 - * Each attribute $a^c \in c.A$ is transformed into an attribute a^t , where $a^t.N = a^c.N$, $a^t.Ty = a^c.C$, and added to the attribute list of its transformed container t such as $a^t \in t.A$,
 - * The object identifier of c is transformed into a row identifier in the target table t where $Id^t.N = Id^c.N$ and $Id^t.Ty = Rid$ and added to the attribute list of its transformed container t such as $id^t \in t.A$.
- **Rule R3.** A binary link $l \in L$ (regardless of its type: Association, Composition or Generalization) between 2 classes c_1 and c_2 is transformed into a relationship $r \in R$ that associates 2 tables t_1 and t_2 that correspond to c_1 and c_2 respectively, where $r.N = l.N$, $r.Pr^r = \{(t_1, cr^{c_1}), (t_2, cr^{c_2})\}$.
- **Rule R4.** A n-airy link $l \in L$ with $n > 2$ is transformed to both:
 - * new table t^l with an identification attribute Id^{t^l} , where $t^l.N = l.N$ and $t^l.A = \{Id^{t^l}\}$, and
 - * n binary relations $\{r_1, \dots, r_n\}$, $\forall i \in [1..n]$, r_i associates t^l to table t_i that corresponds to a linked class c_i , where $r_i.N = (t^l.N) + ' _ ' + (t_i.N)$ and $r_i.Cp^r = \{(t^l, null), (t_i, null)\}$.
- **Rule R5.** An association class c^{asso} that relates n classes $\{c_1, \dots, c_n\}$ with $n \geq 2$ is considered as n-airy link l , with $n > 2$, and transformed to:
 - * new table t^{asso} , where $t^{asso}.N = l.N$, and $t^{asso}.A = c^{asso}.A^{asso}$, and
 - * n binary relations $\{r_1, \dots, r_n\}$, $\forall i \in [1..n]$, r_i associates t^{asso} to table t_i that corresponds to a linked class c_i , where: $r_i.N = (t^{asso}.N) + ' _ ' + (t_i.N)$, and $r_i.Cp^r = \{(t^{asso}, null), (t_i, null)\}$.

We have formalized these transformation rules using the graphical formalism of the QVT language. A QVT transformation between a source model and target one, is specified through a set of relationships. Each relationship is made up of the following elements:

- **«Domain»:** shows the source model concept and its corresponding concepts in the target model.
- **«Relation Domain»:** presents the type of relationship between domains, it has tow possible values, **Checkonly** noted C or **Enforced** noted E. A **Checkonly** domain verifies whether there is a valid match that satisfies the relationship. The **Enforced** domain creates the corresponding concept in the target model, even if the mapping is not verified.

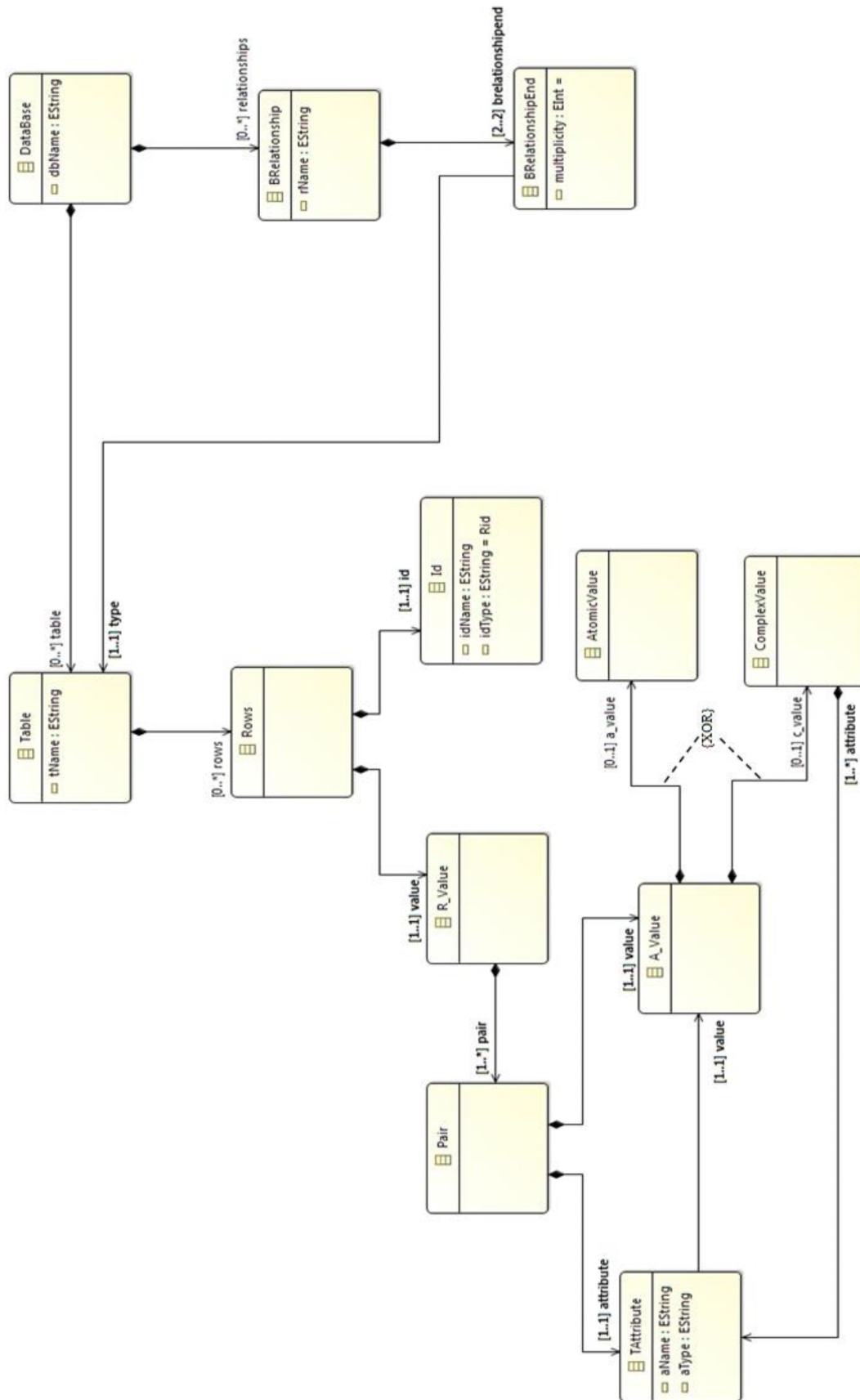


Figure 2.4: Logical PIM metamodel

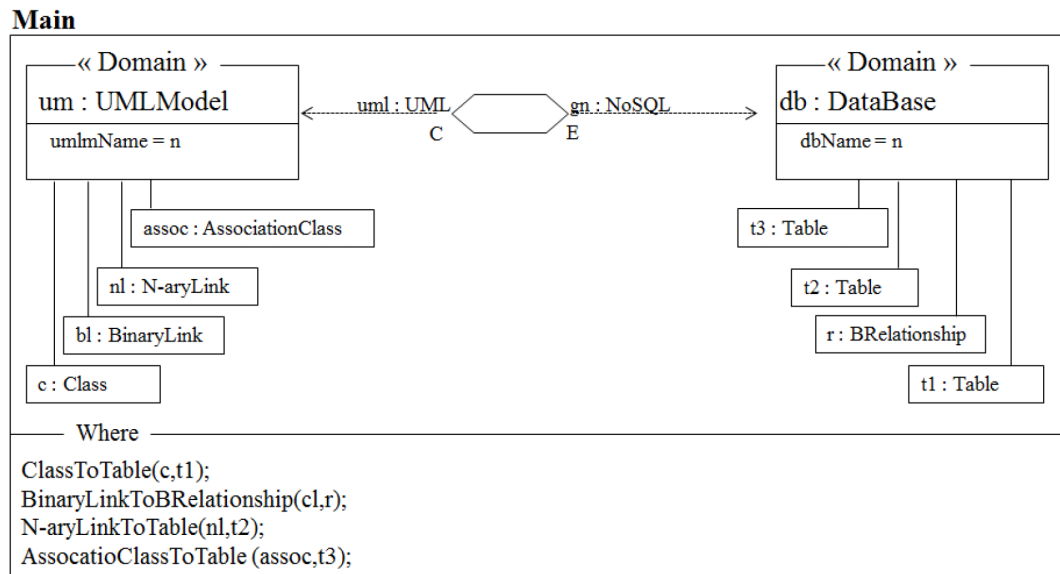


Figure 2.5: Conceptual PIM to logical PIM QVT mapping rules - «Main» relation

- «When» clause: shows the preconditions that have to be satisfied before executing the transformation.
- «Where» clause: shows the postconditions that have to be satisfied after executing the transformation.

Hereafter, we introduce the key transformation rules that map the conceptual PIM into the logical PIM, employing the QVT graphical formalism.

Relation «Main». This relation is the entry point to the transformation process. The left side of Figure 2.5 shows the elements of the source model (uml : UML) that will be transformed into elements of the target model (gn: NoSQL) shown in the right side of the figure. The UML model is transformed into a database that will have the same name. The "where" clause specifies that classes, n-ary links and association classes are transformed into tables, and that binary links are transformed into binary relationships.

Relation «ClassToTable». This relation shows that a class is transformed into a table that will have the same name. All the attributes of the class are transformed into attributes of the table, this will be ensured using the relation CAttributeToTAttribute shown in the Where clause of Figure 2.6. The binary and n-ary links related to the class are transformed into binary relations and tables, respectively.

Relation «N-aryLinkToTable». Once the classes are transformed into tables (the "ClassToTable" precondition of the "When" clause), the link is transformed into:

- A new table that gets the link name (see Figure 2.7),
- A set of binary relationships that link this table to the other linked tables ("ClassToBRelationship" relation of the "Where" clause).

2.3.2 Logical PIM to PSM Model Transformation

In this section, we first remind the source model, then we present the target NoSQL platform that we have chosen to illustrate our work as well as the target metamodel. Finally, we describe the transformation of the generic model to the physical model. In our work, we have proposed transformation rules to 4 different platforms (MongoDB, Cassandra, Neo4j, and Redis) that implements

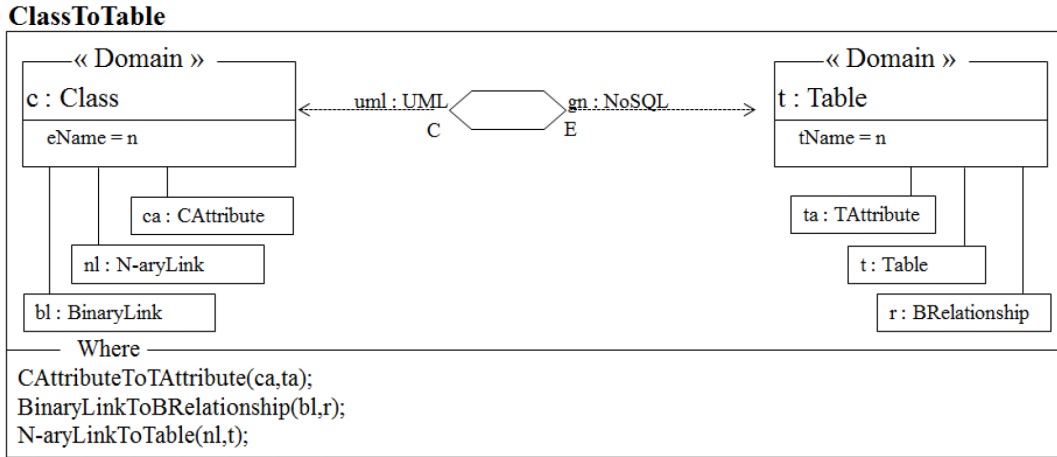


Figure 2.6: Conceptual PIM to logical PIM QVT mapping rules - «ClassToTable» relation

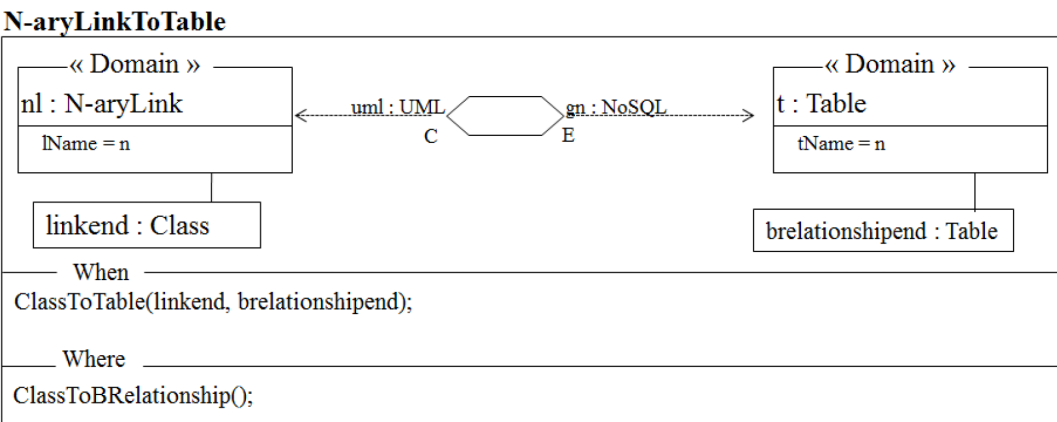


Figure 2.7: Conceptual PIM to logical PIM QVT mapping rules - «N-aryLinkToTable» relation

the 4 families of NoSQL models. We present here, the metamodel and the transformation rules to MongoDB. For more details, and all the transformation rules Cassandra, Neo4j, and Redis, we refer the reader to A. Ait Brahim PhD thesis [34, 6].

2.3.2.1 Source Metamodel: Logical PIM

The input of the `GenericModel2PhysicalModel` transformation is the output of the previous transformation (`Object2GenericModel` transformation) detailed in Section 2.3.1, which is the generic NoSQL model that covers the 4 types of NoSQL DBMS depicted in Figure 2.4.

2.3.2.2 Target Metamodel: MongoDB PSM

As outputs, the `GenericModel2PhysicalModel` transformation generates NoSQL physical models (PSMs). To illustrate our approach, we present here, our generic model implementation to MongoDB. MongoDB is a document-oriented DBMS designed to provide high read and write performance as well as automatic database scaling. This system stores data in BSON (Binary JSON) format which is a textual representation of Key-Value data. The key is a unique identifier associated with an aggregate of fields called a Document. In a MongoDB database, a row is a document in BSON format. A document is always identified by a key, noted `_id`, and is a set of fields composed of a name and a value. The value of a field can be atomic or complex (i.e., it includes other documents). The documents are grouped in Collections. A collection is composed of documents which can have

different structures (i.e., documents belonging to the same collection do not necessarily have the same fields). The collections belong to database that we call a MongoDB.

We define MongoDB database and collections in Definition 8 and 9 respectively. The MongoDB metamodel is shown in Figure 2.8.

Definition 8: A MongoDB database DB^{md} is defined as (N, CLL) where:

- N is the database name,
- $CLL = \{coll_1, \dots, coll_n\}$ is a set of collections.

Definition 9: $\forall i \in [1..n]$, the schema of a collection $coll_i \in CLL$ is a couple (N, FL) where:

- $coll_i.N$ is the name that identifies the collection,
- $coll_i.FL = FL^a \cup FL^{cx} \cup Id^{coll}$ is the set of atomic field $FL^a = \{fl_1^a, \dots, fl_r^a\}$ and complex field $FL^{cx} = \{fl_1^{cx}, \dots, fl_s^{cx}\}$ used to define $coll_i$ where:
 - * $\forall i \in [1..r]$, the schema of an atomic field $fl_i^a \in FL^a$ is a couple (N, Ty) where:
 - $fl_i^a.N$ is the field name,
 - $fl_i^a.Ty$ is the field type.
 - * $\forall j \in [1..s]$, the schema of complex field $fl_j^{cx} \in FL^{cx}$ is a couple (N, FL') where:
 - $FL_j^{cx}.N$ is the name that identifies the field,
 - $FL_j^{cx}.FL$ is the set of fields embedded within FL_j^{cx} where $FL' \subset FL$.
 - * Id^{coll} is a specific field that identifies every document of the collection $coll_i$. This field has always the name $_id$.

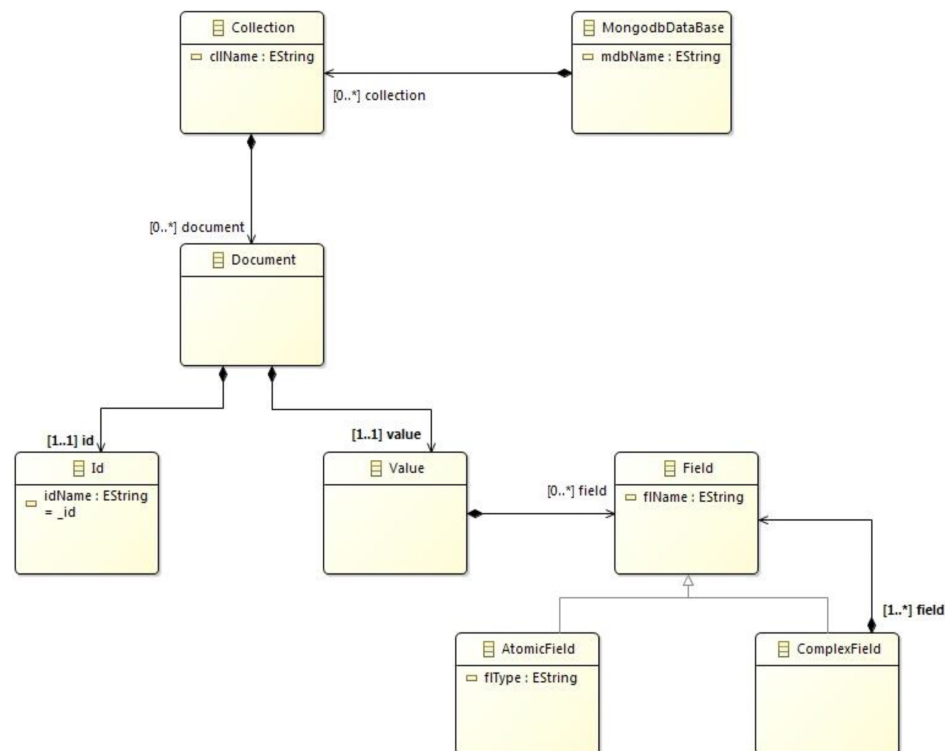


Figure 2.8: PSM MongoDB metamodel

2.3.2.3 Transformation Rules: from Logical PIM to MongoDB PSM

In this section, we present the automatic transformation of the logical PIM to MongoDB PSM presented in the previous section. Two types of transformations are considered: (1) the transformations

which generate the elements necessary for the implementation of the database, these are the elements that must be declared beforehand, before inserting data, and (2) the transformations that create the guidelines of assistance useful for the implementation of the treatments; these guidelines show the use of attributes and the implementation of relationships previously presented.

In MongoDB, the database name and collections have to be declared before inserting data, so required for the DB implementation. The names of the fields and the way to implement the relationships are specified in the assistance guidelines.

(1) The required elements for the database implementation:

- **R1.** A database DB is transformed to a MongoDB database DB^{MD} , where $DB^{MD}.N = DB.N$,
- **R2.** A table $t \in DB$ is transformed to a collection $cll \in DB^{MD}$, where $cll.N = t.N$.

(2) Assistance guidelines:

- **R3.** A Table's attribute $a^t \in t$. A^t is transformed to an atomic field fl^a , where $fl^a.N = a^t.N$, $fl^a.Ty = a^t.Ty$, and then added to the list of field of its transformed container cll as, $fl^a \in cll.FL^a$, where cll is the collection that corresponds to the table t .
- **R4.** A row identifier of a table t is transformed into a document identifier of the cll collection that corresponds to t , where $Id^{cll}.FL$.
- **R5.** The MongoDB system expresses objects relationships by either using references fields or by nesting documents. A reference field is a kind of "foreign key" field that have the value of a document identifier ($_id$), to which it refers. The values of a reference field must exist in the $_id$ field of the referenced document. This constraint is not automatically handled by the MongoDB system, the user have to check it by himself.

Note that with the MongoDB system, relationships can exist between documents in the same collection or between documents belonging to different collections. In our case, all the relationships that we have to implement exist between different collections. In fact, a collection can contain documents representing objects of different types. For example, the same collection can stores patients documents, doctors documents, and consultations documents. In our case of study, a collection (physical level) corresponds to a class (conceptual level), therefore it contains documents with the same semantics. So, we mainly, handle relationships between separate collections.

We define the rule R3 that transforms the relations of the logical level as follows: for each relation r that links two tables t_1 and t_2 , 5 transformation solutions are possible:

- **Solution 1.** r is transformed to a field fl that refers to a document in the collection ccl_2 (originally, t_2 in the logical PIM), where $fl.N = ccl_2.N + _Ref$ and $fl.Ty = Id^{ccl_2}.Ty$, and then added to the list of field of ccl_1 (originally, t_1 in the logical PIM), as $fl \in ccl_1.FL$. When instantiating ccl_1 , depending on the cardinalities of r (see Table 2.1), the reference field fl will have one or more values of an existing document's identifier in ccl_2 .
- **Solution 2.** r is transformed to a field fl that refers to a document in the collection ccl_1 (originally, t_1 in the logical PIM), where $fl.N = ccl_1.N + _Ref$ and $fl.Ty = Id^{ccl_1}.Ty$, and then added to the list of field of ccl_2 (originally, t_2 in the logical PIM), as $fl \in ccl_2.FL$. When instantiating ccl_2 , depending on the cardinalities of r (see Table 2.1), the reference field fl will have one or more values of an existing document's identifier in ccl_1 .
- **Solution 3.** r is transformed by nesting a document $d \in ccl_2$ (originally, t_2 in the logical PIM) in ccl_1 (originally, t_1 in the logical PIM) where, $d \in ccl_1.FL^{cx}$.
- **Solution 4.** r is transformed by nesting a document $d \in ccl_1$ (originally, t_1 in the logical PIM) in ccl_2 (originally, t_2 in the logical PIM) where, $d \in ccl_2.FL^{cx}$.
- **Solution 5.** r is transformed to a new collection ccl , $ccl.N = r.N$, $ccl.Fl = \{fl_1, fl_2\}$, $fl_1.N =$

Table 2.1: Types of field reference

Relation	Solution	Type of the field reference
$r = (N, \{(t_1, *), (t_2, 1)\})$	Solution 1	Mono-valued
	Solution 2	Multi-valued
	Solution 5	Mono-valued
$r = (N, \{(t_1, 1), (t_2, *)\})$	Solution 1	Multi-valued
	Solution 2	Mono-valued
	Solution 5	Mono-valued
$r = (N, \{(t_1, 1), (t_2, 1)\})$	Solution 1	Mono-valued
	Solution 2	Mono-valued
	Solution 5	Mono-valued
$r = (N, \{(t_1, *), (t_2, *)\})$	Solution 1	Multi-valued
	Solution 2	Multi-valued
	Solution 5	Mono-valued

$(cll_1.N)^+ _Ref'$, $fl_1.Ty = Id^{cll_1}.Ty$, $fl_2.N = (cll_2.N)^+ _Ref'$, $fl_2.Ty = Id^{cll_2}.Ty$. cll_1 and cll_2 are the collections that correspond to tables t_1 and t_2 respectively.

Depending on the cardinalities of r , the reference fields used in solutions 1, 2, and 5 can be mono-valued or multi-valued. Table 2.1 indicates the type of the reference field according to both the cardinalities of the relation and of the chosen transformation solution.

2.4 ModelDrivenGuide: Guiding NoSQL-based IS Development

In Section 2.3, we have presented our approach that aims to formalize and automate the development process of big data IS. We have presented a logical metamodel that describes the 4 NoSQL families as well as the transformation rules that map the concepts of the class diagram (conceptual level - PIM1) into concepts of the logical level (PIM2), then from PIM2 to Mongo DB PSM. In this section, we extend our previous contributions in order to consider both the relational model and the 4 NoSQL families, and to provide a guide that helps the designer to make well-founded decisions when choosing the most suited model(s) and platform(s) according to functional and technical requirements. This guide is based on the following contributions:

- We adapt and extend the logical metamodel to include, in addition to the 4 NoSQL families, the relational model, so we provide the 5Families metamodel (5FMM),
- When mapping the conceptual model (PIM1) to the logical model (PIM2), it is possible to transform a source concept into a target concept, but also to split a concept into several target concepts or conversely merge several source concepts into one only target concept. For this, in addition to the transformation rules which automate the transformation process by mapping a source concept into a target concept (presented in Section 2.3), we propose refinement rules that split a source concept into several target concept or inversely, merge several source concepts into a single target concept,
- A complete formalization of the search space of generated data models with transformation rules, allowing to get *all* the possible models going from one logical model to another,
- A full formalism of the data model refinement problem with complexity bounds for both merge and split rules,
- A recursive heuristic to reduce the search space of generated data models, based on redundancies and use cases,
- A cost model that gives the execution time of queries, the environmental and financial costs for the different possible models. The cost model helps the designer to choose one model or another,

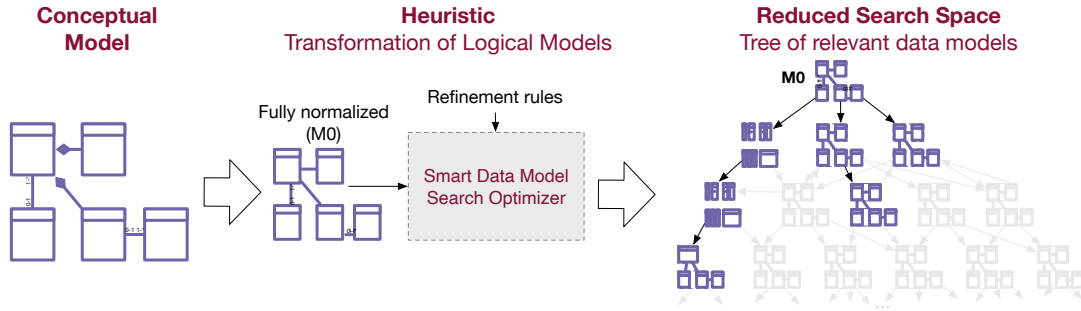


Figure 2.9: ModelDrivenGuide: a heuristic to generate data models

one platform or another according to functional and technical requirements. This contribution is in progress and will not be detailed here.

2.4.1 Overview of our Approach

As presented in Section 2.2, we propose an approach based on different modeling levels and factors in order to guide the choice of the target SQL/NoSQL solution(s): context of use, data model, functional, and technical needs. We provide a common logical metamodel favoring refinements to move from one target model to another which allows to find optimal solution(s). This approach is based on logical models and refinement rules.

In this section, we focus especially on the PIM2 5Families metamodel used to generate all possible data models, the transformation and refinement rules (between and inside metamodels) and also the heuristic that reduces the search space of PIM2 data models by removing unuseful solutions. The choice of proper data models determined by a cost model which is a work in progress. As shown in Figure 2.9, ModelDrivenGuide begins with the fully normalized data model (\mathcal{M}_0) (i.e., the relational model). Then, the *Smart Data Model Search Optimizer* relies on data models' denormalization with refinement rules applied recursively on logical models (Definition 10) to obtain all possible solutions. It is based on a heuristic that optimizes the denormalization process. This process applies refinement rules recursively without conditions to generate all models and is called *Naïve* in the following. The heuristic reduces the search space to relevant data models since some of the generated models can be redundant and even useless for the associated use case. So far, no other approach combines merge and split rules to generate *all data models* suitable for a given context of use.

Definition 10: Let \mathcal{M} be a data model conform to the 5Families metamodel where $\mathcal{M} = (\mathcal{C}, \mathcal{R}, \mathcal{L}, \mathcal{K}, \mathcal{E}, \kappa)$ is composed of **concepts** $c(r_1, \dots, r_m) \in \mathcal{C} | r_1, \dots, r_m \in \mathcal{R}$, **rows** $r(k_1, \dots, k_n) \in \mathcal{R} | k_1, \dots, k_n \in \mathcal{K}$, **key values** \mathcal{K} (Atomic Values or Complex Values), **references** $ref_{i \rightarrow j} \in \mathcal{L} | k_i, k_j \in \mathcal{K}$, **edges** $\mathcal{E} : \mathcal{C} \times \mathcal{C}$ and **constraints** $cons(k) \in \kappa | k \in \mathcal{K}$.

To simplify the formalism of manipulations on data models \mathcal{M} , we will focus mostly on rows' transformations and references. Thus, a data model \mathcal{M} can be denoted as a graph $\mathcal{M}(\mathcal{R}, \mathcal{L})$ where nodes are rows such that: $\forall i \in \{1, \dots, n\}, r_i \in \mathcal{R}$, and edges are references such that: $\forall j, k \in \{1, \dots, n\}, ref_{j \rightarrow k} \in \mathcal{L} | r_j, r_k \in \mathcal{R}, r_k = ref_{j \rightarrow k}(r_j)$. Our goal is to start from an initial $\mathcal{M}_0(\mathcal{R}, \mathcal{L})$ and to generate a set of optimal models \mathcal{M}^{opt} .

Example 1: Figure 2.10 gives a logical representation of a data model \mathcal{M} with 4 rows **Warehouse**, **Customer**, **StoredIn** and **Order** (resp. W, C, S, and O) of 4 corresponding concepts. Four references link keys from source row (i.e., primary key - bold red) to the target row (i.e., foreign key).

2.4.2 A Common Metamodel to Unify the 5 Families of Models

The common metamodel (PIM2 level) is the cornerstone of our approach. This metamodel integrates all possibilities of data schema *refinement*. It seeks to produce different schemas compatible with the

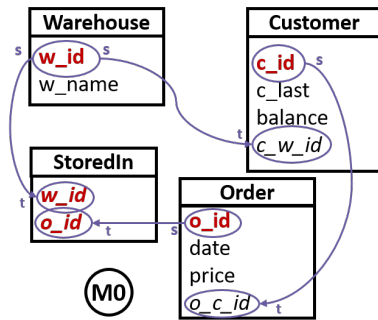


Figure 2.10: Driving Example

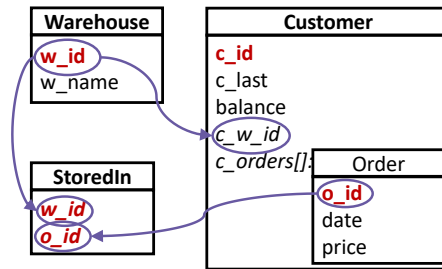


Figure 2.11: Merge: $m(C, O, ref_{C \rightarrow O})$

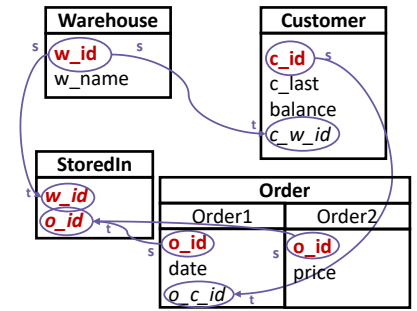


Figure 2.12: Split: $s(O, price)$

PSM constraints while remaining independent. The subtlety of the common logical PIM is to integrate the 5 families of data models. A major advantage of this metamodel is that if a denormalization (or normalization) solution proves to be well suited to a case of study using a family, this solution can also be the suited one to the four other families. Using *refinement* rules, it will be possible to merge or split concepts to adapt them to a relational, key-value, column, document or graph data model.

Figure 2.13 shows the PIM2 5Families metamodel integrating all the concepts used in the 5 families of data models: the **concepts** contain **rows** (for column-oriented models), **key-values** with **atomic** or **complex** values. Concepts can also be linked by **edges** to facilitate the integration of a graph database. **Complex Values** are represented by the composition of **rows**. A loop occurs between rows and complex values in the metamodel, it shows that a value can be multiple and its schema is structurally identical to a concept. The key point of this orientation is to enable the application of refinement rules without iteration limits, allowing the production of new schemas. Meta-constraints are associated with the Complex Value to prevent the mutual composition of instances. **Constraints** and **references** are associated with **keys**, which can belong to distinct concepts. The idea is to be able to transform these constraints into new representations using refinement rules. For example, a **Reference** can be transformed into an **Edge** for graph-oriented models or into a **Foreign Key** in RDB. UML classes from the PIM1 are transformed to the PIM2 using traditional mappings between concepts, keys and their values. The transformation rules are expressed in QVT as presented in Section 2.3.

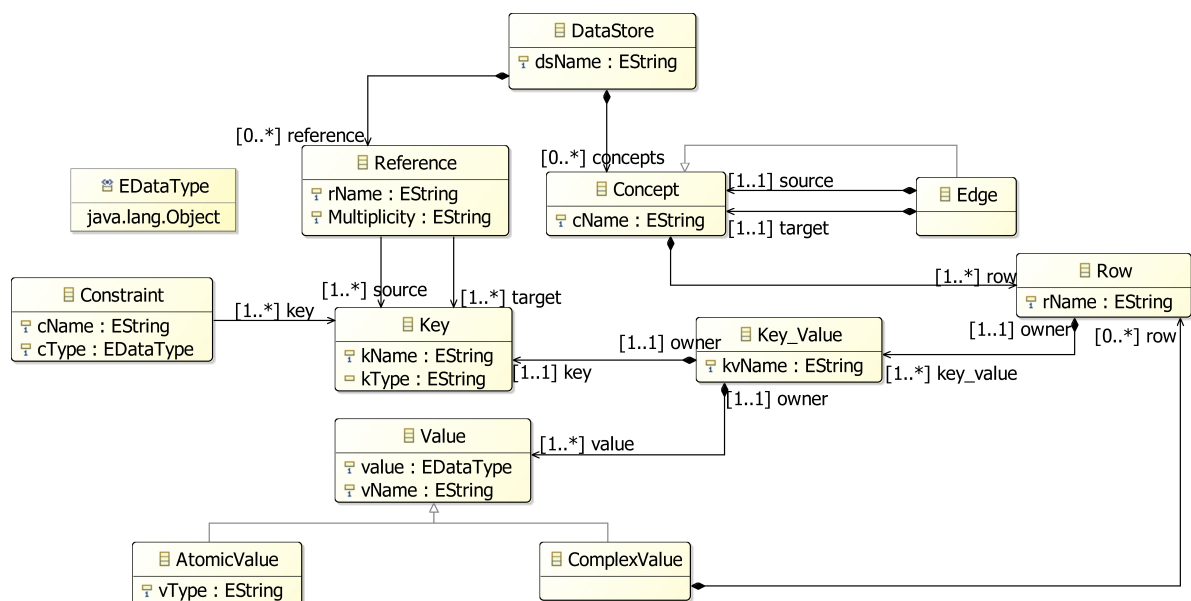


Figure 2.13: Logical PIM - common metamodel for the 5Families of data models

2.4.3 Common Model Refinement

The refinement can be applied iteratively with respect to the metamodel. The PIM2 refinement is based on three transformation rules:

- **Merge** rows between concepts to produce *complex values* for nesting (CO, DO) or to merge keys for values concatenation (KVO),
- **Split** rows to produce new columns in a same target *concept* (CO),
- Transform *references* into the equivalent *edges* (GO).

At the logical PIM level, models' refinement uses endogenous transformations where both the source and the target models are conform to the same metamodel, here the 5Families metamodel.

As detailed in the following, the generation of data models will rely on those rules by applying them recursively on various combinations of transformations. Each new schema produced by QVT rules can be transformed to target PSMs. We detail especially *merge* and *split* refinement rules which are the basis of the data model generation framework.

2.4.3.1 Merge

Merge is a refinement rule applied between two rows referring to each other (*i.e.*, associated classes), where the result is a single row with a complex value.

Definition 11: Let $m: \mathcal{M} \rightarrow \mathcal{M}$ be an endogenous function that merges rows from a 5Families model \mathcal{M} . The merge function $m(r_i, r_j, ref_{i \rightarrow j})$ is applied on two rows $r_i, r_j \in \mathcal{R}$ (source and target rows) linked by a reference $ref_{i \rightarrow j} \in \mathcal{L}$ with corresponding keys $k_i, k_j \in \mathcal{K}$. The merge function produces a new model \mathcal{M}' where r_j is a complex value of r_i , and removes $ref_{i \rightarrow j}$, denoted by:

$$m(r_i, r_j, ref_{i \rightarrow j}) = r_i\{r_j\}$$

The merge function is a bijective function $m^{-1}(r_i\{r_j\}) = (r_i, r_j, ref_{i \rightarrow j})$ which rebuilds $ref_{i \rightarrow j}$ and non-nested rows.

This rule corresponds to the merge of two nodes in graph \mathcal{M} where node j (row) is embedded in node i . Notice that $m(r_i, r_j, ref_{i \rightarrow j}) \neq m(r_j, r_i, ref_{i \rightarrow j})$ since the nested row is done in the opposite way. From reference $ref_{i \rightarrow j}$, when the target row r_j is nested into the source r_i , r_j is embedded into a list of values.

We present in Figure 2.14 the RowConceptsToNestedRowConcept QVT rule that merges rows belonging to two different concepts. Thus, two rows from two different concepts referring to each other (initially classes associated at the PIM1 level) can be merged into a single row with a complex value. Row $r2$ of concept $co2$ is linked by reference ref to concept $co1$ through row $r1$. To merge rows, row $r2$ is then nested into a complex value cv which corresponds to the transformation of the reference into a new key value k . The latter corresponds to the transformation of the reference key $k1$. This rule can be applied in both directions by switching $co1$ and $co2$ (generating a list of complex values in k).

Example 2: Applying a merge on the driving example, like $m(Customer, Order, ref_{C \rightarrow O})$ results in the data model shown in Figure 2.11. We notice that after applying this merge rule, the reference $ref_{C \rightarrow O}$, linking Customer row and Order row was transformed into `c_orders` containing a list of nested Order [].

Cycles issue. By recursive application of merges, a problem occurs in case of data models containing cycles through references like in our driving example. Figure 2.15 illustrates recursive application of four merges to obtain successively M_0 (of Figure 2.10) to M_4 data models. Since the merge function removes references (Definition 11), the M_4 data model doesn't contain references anymore.

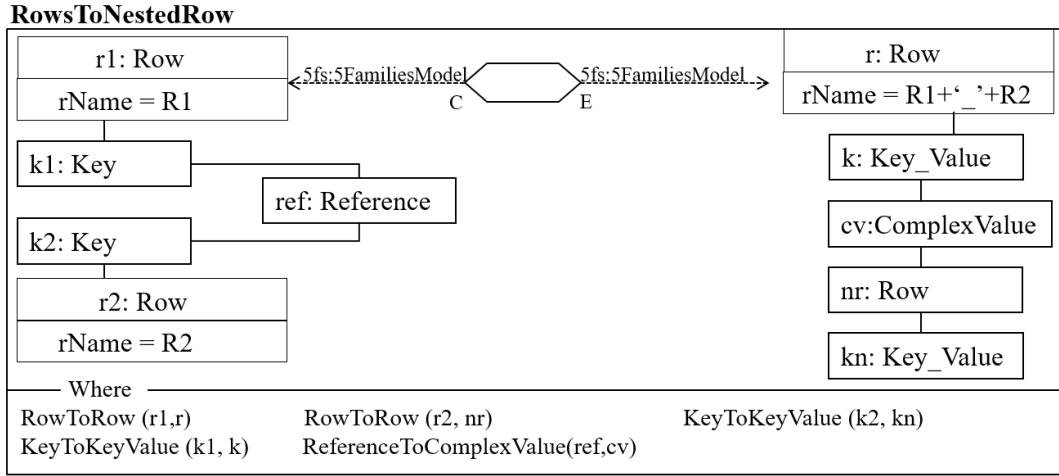


Figure 2.14: 5Families model refinement - Concepts' rows merge QVT rule

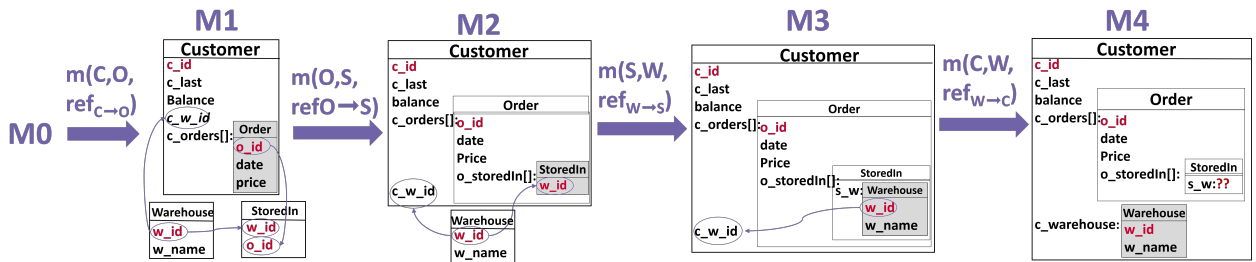


Figure 2.15: Particular case: cycle

However, we must notice that the M_0 contains a cycle with a reference $ref_{W \rightarrow S}$ between *Warehouse* and *StoredIn*. During the 4th merge, this reference has been transformed into a *ComplexValue* into *StoredIn*[]. But the remaining reference $ref_{C \rightarrow W}$ between *Customer* and *Warehouse* is used to merge them (data model M_4) which gets rid of the information $ref_{W \rightarrow S}$. Due to this removal it is impossible to get backward in the data model's generation process. This case occurs during merges on nested rows. We apply m^{-1} on the nested row to rebuild the reference, and apply m on the other reference.

2.4.3.2 Split

Split is a refinement rule applied on a row containing several keys, associating several rows for a same concept (*i.e.*, CO's column family).

Definition 12: Let $s: \mathcal{M} \rightarrow \mathcal{M}$ be an endogenous function that splits rows from a 5Families model \mathcal{M} . The split function $s(r_i, k)$ is applied on a row $r_i \in \mathcal{R}$ and a key value $k \in keys(r_i)$ not linked to a constraint. The split function produces a new model \mathcal{M}' with two rows $\overline{r_{i_k}}$ and r_{i_k} with the same constraint key $pk \in keys(r_i)$ (*i.e.*, primary key) where $\overline{r_{i_k}} = (pk, k_i) | \forall k_i \in keys(r_i) \wedge k_i \neq k$, and $r_{i_k} = (pk, k)$. The split function s is denoted by:

$$s(r_i, k) = (\overline{r_{i_k}}, r_{i_k})$$

s is bijective $s^{-1}(\overline{r_{i_k}}, r_{i_k}) = (r_i)$ and merges common constraints and keys.

Example 3: Let's take the example of the model depicted in Figure 2.10, if we apply a split on *Warehouse* using key *price*, we obtain the model in Figure 2.12. Notice that rows *Order1* and *Order2* are linked to the same concept *Order*.

Figure 2.16 illustrates the *RowConceptToRowsConcept* split refinement rule. It shows that a key value k linked to a row r can be separated from it and placed in a new row $r2$ which is linked to

the same concept. This split represents the fact that several rows can occur for a same concept (*i.e.*, a column family in CO). We notice that the primary key constraint c is duplicated in the new row $r2$ to preserve uniqueness and to allow instances reconstructions when instances will be physically separated. The content of row r remains unmodified except for the moved key value k , which means that remaining key values from r are still linked to r (called $r1$ in the target data model). The rule `RowConceptToRowsConcept` can also be applied on a set of key values to move this set into $r2$.

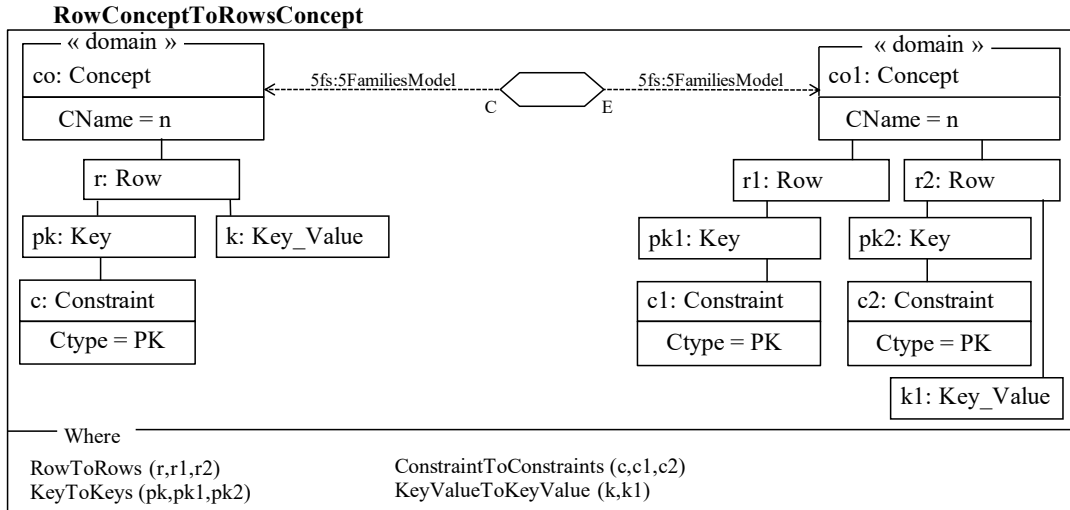


Figure 2.16: 5Families model refinement - Row's split QVT rule

2.4.3.3 Inverse Functions

We should notice that *merge* and *split* refinement rules can be considered inverse functions for specific cases. In fact, if a split is applied on a key k_j as a *complex value* r_j , it rebuilds a row with a single key containing the reference information $ref_{i \rightarrow j}$ and keys from row r_j . Then:

$$s(r_i\{r_j\}, k_j) = (r_i, r_j) = m^{-1}(r_i\{r_j\}) \tag{2.1}$$

At the opposite, a *merge* between two rows $\overline{r_{i_k}}$ and r_{i_k} with a same constraint pk_i can be considered to be the inverse of a *split*. Then:

$$m(\overline{r_{i_k}}, r_{i_k}, pk_i) = (r_i) = s^{-1}(\overline{r_{i_k}}, r_{i_k}) \tag{2.2}$$

2.4.4 Completeness of our Approach

Our approach based on the previously defined refinement rules generates all possible data models by combining recursively several merges and splits. Before defining a generation strategy of data models, it is necessary to prove the completeness of our approach. It states that splits and merges are sufficient to generate all possibilities without modifying keys (*e.g.*, materialization, replication). Moreover, there is always a path between two data models.

Theorem 2.4.1: Completeness. *Merge* and *Split* refinement rules are sufficient to generate all denormalized data models \mathcal{M}^* without key modification, conformed to the 5Families metamodel, beginning with the fully normalized data model.

Proof 2.4.1: We want to prove that there is at least one *path* (sequence of rules) between two denormalized models, which means there is a path between a denormalized model M_1 and the fully normalized data model M_0 .

Let $step(M_1, M_2, r)$ be a rule applied between two models M_1 and M_2 , where $r \in \{merge, split\}$ and

$M_2 = r(M_1)$. And let $path(M_1, M_n)$ be a sequence of steps between two models M_1 and M_n , such that: $path(M_1, M_n) = [step(M_1, M_2, r_1), \dots, step(M_{n-1}, M_n, r_{n-1})]$

Suppose that M_0 and M_n share the same set of rows and keys and there isn't any $path$ between a denormalized model M_n and the normalized one M_0 .

$$\implies path(M_0, M_n) = \emptyset$$

$$\implies \exists M, path(M, M_n) \neq \emptyset, path(M_0, M) = \emptyset$$

This implies that: 1) this model M is not conformed to the **5Families** metamodel (Absurd since rules are endogenous), or 2) the applied rules can't be inverse (Absurd, from Definitions 11 and 12).

Of course, it exists more than one path from a data model to another since there are several possible compositions of functions that produce a given data model. Thus, we obtain a *lattice* that represents the generated solutions where nodes are data models, and edges are applied refinement rules. The lattice reduction problem is discussed in the following section.

2.4.5 Smart Data Model Search Optimizer

Models refinement leads to the production of plenty of data models which forms a graph. The number of possibilities explodes as splits on \mathcal{M} can be applied on each key and merges can be done in both ways. Starting from \mathcal{M} as $\mathcal{M}(\mathcal{R}, \mathcal{L})$, applying the i^{th} transformation rule on the rows in \mathcal{R} , leads to \mathcal{M}_i . If the rule is split, it divides one row to several columns. If the rule is merge, it merges 2 rows which are a set of columns themselves.

Considering \mathcal{M} as a set of rows, all generated \mathcal{M}_i by the naïve process, form a graph $G(\mathcal{M}^*, \mathcal{T}^*)$ where \mathcal{M}^* is the set of generated data models and \mathcal{T}^* is the transformations that link data models. This problem is a reduced problem of generating all partitioning of a set. The total number of partitions of the set \mathcal{M} can be calculated as the *Bell* number. *Bell* number grows exponentially by increasing the size of the set, and set partitioning has been proved to be NP-hard [88]. Moreover, splits on data models are combined with merges making the growth far more computational. Therefore, we need a heuristic to explore only an optimal subset of state space. Considering the initial \mathcal{M} as a state space graph of $\mathcal{M}_0(\mathcal{R}, \mathcal{L})$, and transformed graph after applying the i^{th} transformation rule as \mathcal{M}_i , $T(\mathcal{M}^{opt}, \mathcal{T}^{opt})$ is the reduced search space tree growing through transformations done by our heuristic. To obtain T we apply 1) a Depth First Search (*DFS*) strategy to traverse G at each step i and 2) a reduction of the graph based on the use case.

In this section, we first present the application of the naïve refinement approach which turns out to be complex. Then, we present the rules to follow to apply the model refinement. These rules will be used by the heuristic to reduce the number of unuseful generated models.

2.4.5.1 Complexity of the Naïve Refinement Process

The generation of data models \mathcal{M}^{opt} by applying refinement rules recursively on various combinations of model transformations \mathcal{M}_i , produces a hierarchy of target data models T . At the top level of this hierarchy is the fully normalized relational model \mathcal{M}_0 and at the lowest level are completely denormalized models. This hierarchy's size $|T|$ depends on the size of the input data model with the number of rows $|\mathcal{R}|$ and references $|\mathcal{L}|$. Figure 2.17 shows the first two levels of the data models' generation graph (G) for our driving example, where W , C , O , and S stand for *Warehouse*, *Customer*, *Order* and *StoredIn* respectively. The dashed edges show the edges that were deleted by the heuristic to avoid redundancies, cycles and transformations that do not consider the use case. Consequently, dashed squares represent the lonely nodes (models) which cannot be produced since no more transformation can reach them.

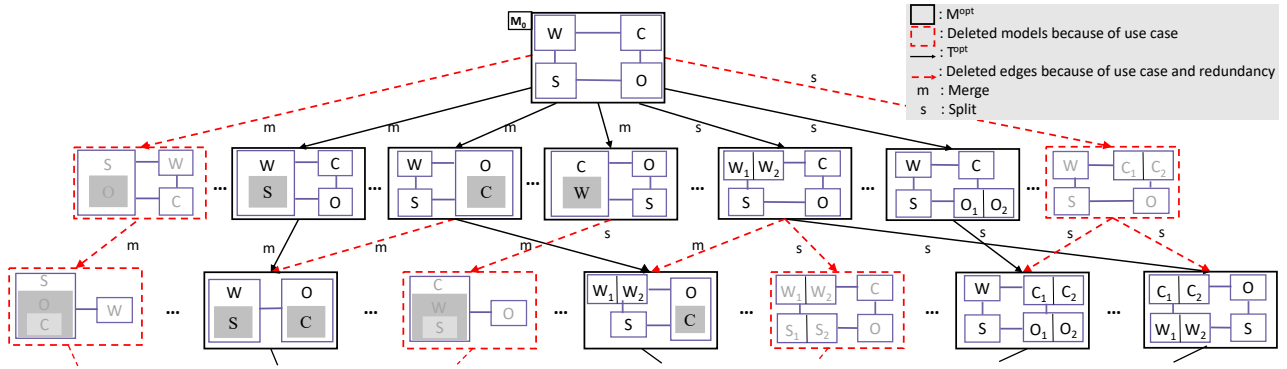


Figure 2.17: Two levels of the graph of possible solutions

The set of merges depends on the number of linked rows in \mathcal{M} and the number of generated models is given by a *Fubini number* or *Ordered Bell number* [27]: $Fn_{|\mathcal{L}|} = \sum_{k=0}^{|\mathcal{L}|} k! \times \binom{|\mathcal{L}|}{k}$

This number is an upper bound since the number of generated data models can be less depending on references' organization between rows. Our driving example can produce up to 75 models considering just merges $Fn_4 = \sum_{k=0}^4 k! \times \binom{4}{k} = 75$.

Moreover, the number of splits on a row depends on the number of keys in a row $|keys(r)|$ (without primary keys) which is given by the *Bell's number* [28]: $B_{|keys(r)|} = \sum_{k=0}^{|keys(r)|} \binom{|keys(r)|}{k} \times B_{|keys(r)|-1}$

Thus, in our example, *Customer* and *Order* rows contain 3 non-primary keys $B_3 = 5$, for *Warehouse* $B_1 = 1$ and *StoredIn* $B_0 = 1$. And those solutions combined together produce $B_{|keys(C)|} \times B_{|keys(O)|} \times B_{|keys(W)|} \times B_{|keys(S)|} = 25$. To finish with, split and merged rows can be combined to produce all possible solutions in T . Each split row can be merged and adds new nested solutions. Thus, we can formalize the complete denormalization problem with the combination of rows' merges and splits all together as a product. The total number of possible data models gives the following complexity measure:

$$|\mathcal{M}^*| = Fn_{|\mathcal{L}|} \times \prod_{k=1}^{|\mathcal{R}|} B_{|keys(r_k)|}$$

Our example will generate $75 \times 25 = 1,875$ data models. As said previously, since there are several ways to produce a single $\mathcal{M}_i \in \mathcal{M}^{opt}$, it requires to prune the lattice G of solutions \mathcal{M}^* .

2.4.5.2 Application of the refinement rules

Due to the aforementioned problems, we need to provide a heuristic in order to reduce the search space and avoid cycles. The target is a subset of \mathcal{M}^* called \mathcal{M}^{opt} after pruning edges and nodes from T . The heuristic avoids to produce different paths between two data models. In fact, applying splits and merges in different orders will produce the same effects on the resulting data models. Moreover, every merges can be reversed by a split and produce a cycle in the production of data models. Thus, four main rules have to be adopted and are presented in the following.

A row with complex values. Applying a split on a row with complex values (merged rows) gives the following result:

$$s(m(r_i, r_j, ref_{i \rightarrow j}), k_j) = s(r_i\{r_j\}, k_j) = (r_i, r_j, ref_{i \rightarrow j})$$

where r_i and r_j are two rows that are first merged and represented as $r_i\{r_j\}$, $ref_{i \rightarrow j}$ is the reference linking the two rows and used for the merge and k_j is the key on which we apply the split. We notice that this transformation takes the models' generation back in the hierarchy with the initial model

(r_i, r_j) of two rows, hence generates a cycle. Thus, the first rule avoids applying a split on a row with complex values.

A row linked to two or more rows. When a row r_i is linked to two rows r_j and r_k with two references $ref_{i \rightarrow j}$ and $ref_{i \rightarrow k}$, a redundancy will occur, since:

$$m(m(r_j, r_i, ref_{j \rightarrow i}), r_k, ref_{j \rightarrow k}) = m(m(r_j, r_k, ref_{j \rightarrow k}), r_i, ref_{j \rightarrow i}) = r_j\{r_i, r_k\}$$

where $r_i, r_j, r_k \in \mathcal{R}$ and r_j is linked both to r_i ($ref_{j \rightarrow i}$) and r_k ($ref_{j \rightarrow k}$).

Since both transformations are equivalent (the result is the same data model), the heuristic discards one of the two merges.

The use case. By considering the use case, we can reduce the number of joins only to those used in queries that combine rows through references. Also splits should not separate keys if queries of the use case combine them. It avoids solutions which require instance reconstruction with costly joins. Moreover, splits applied considering queries can generate redundancy. If queries $q(K)$ and $q(\bar{K})$ applied on r use complement keys, they will produce the same data model (*i.e.*, $s(r, K) = s(r, \bar{K})$). To avoid this issue, we compute complementary queries to prune redundant splits.

We must notice that Theorem 2.4.1 on the completeness of our approach remains true according to the two first simplification rules. In fact, only duplicate edges in \mathcal{M}^* are removed producing a DFS keeping all the nodes. Thus, it always exists a *single* path, to obtain a data model. However, this last simplification rules removes nodes since we remove edges from \mathcal{M}^* if the refinement rule do not rely on a query from the use case. Commonly said in the literature [40, 48, 6], we assume the fact that those transformations will lead to a *useless* data model as well as all the corresponding subtree. Consequently, the completeness remains valid for data models relying on the use case.

2.4.5.3 Heuristic to Generate Data Models

To formalize the generation of data models with our heuristic, Algorithm 1 gives the recursive function `5FMHeuristic` which generates a list of data models \mathcal{M}^{opt} . It takes a relational model as an entry and produces all possible models (of all families). Starting from $i = 0$, at stage i , it processes a current data model \mathcal{M}_i on which splits and merges will be applied by considering a set of queries \mathcal{Q} from the use case. For simplicity, we consider that a query is a set of involved keys within the data model (for filters, joins, projects, aggregates, etc.).

Each time the `5FMHeuristic` function is called with a new data model from the `5Families` meta-model, it is added to the global output list \mathcal{M}^{opt} (line 2). Then, all keys from the data model ($K \in \mathcal{K}$) are tested except those which are involved in a Primary Key constraint (line 3). To test the keys, the first check (lines 5-7) verifies if the key is used in a query from $\mathcal{Q} \cup \mathcal{Q}'$ (remaining and processed queries). It also checks if the key is the last remaining key from the current row R except the Primary Key (needed for instance reconstruction). In this case, the key is put in a new row by applying the rule `Split` (Definition 12) and then recursively call `5FMHeuristic` on this new data model (\mathcal{M}_{i+1}). If the key K belongs to at least one remaining query q from \mathcal{Q} (line 9), we check the rules from the heuristic. First (lines 10-11), if all keys from q belong to the same row r and the latter can be split (except the Primary Key and non-complementary keys), we split it by taking all the keys from q in a new row, instead of the single key K . The generated data model is then recursively denormalized (line 11) without the query q (moved to \mathcal{Q}').

Algorithm 1 5FMHeuristic

global: A list of data models \mathcal{M}^{opt} from *5FamiliesModel*

input: A data model \mathcal{M}_i from *5FamiliesModel*, a list of input queries \mathcal{Q} (a query is a set of keys from \mathcal{M}_i), a list of used queries \mathcal{Q}'

init: $\mathcal{M}_i = \mathcal{M}_0$, the relational data model, $\mathcal{M}^{opt} = \emptyset$, $\mathcal{Q}' = \emptyset$

Procedure 5FMHeuristic($\mathcal{M}_i, \mathcal{Q}, \mathcal{Q}'$)

```

 $\mathcal{M}^{opt} := \mathcal{M}^{opt} \cup \mathcal{M}_i$ 
foreach key  $K \in \mathcal{K} \wedge !Constraint(K, CType=PK)$  do
     $r := Row(K)$  if  $K \notin \mathcal{Q} \cup \mathcal{Q}'$  then
        if  $\exists k \in r | k \neq K \wedge !Constraint(k, CType=PK)$  then
            | 5FMHeuristic(Split( $\mathcal{M}, K$ ),  $\mathcal{Q}, \mathcal{Q}'$ )
        end
    end
else
    foreach  $q \in \mathcal{Q}$  do
        if  $\forall k \in q | Row(k) = r, \exists k \in R | k \notin q \wedge !Constraint(k, Ctype = PK) \wedge \forall q' \in \mathcal{Q} | \bar{q} \neq q'$  then
            | 5FMHeuristic(Split( $\mathcal{M}_i, q$ ),  $\mathcal{Q} - q, \mathcal{Q}' \cup q$ )
        end
        else if  $\exists k \in q | Row(k) \neq r \wedge Concept(k) = Concept(K)$  then
            | continue
        end
        else if  $\exists k_1, k_2 \in q | ref_{k_1 \rightarrow k_2} \vee ref_{k_2 \rightarrow k_1} \in \mathcal{L}$  then
            | 5FMHeuristic(Merge( $\mathcal{M}_i, q, k_1, k_2$ ),  $\mathcal{Q} - q, \mathcal{Q}' \cup q$ )
            | 5FMHeuristic(Merge( $\mathcal{M}_i, q, k_2, k_1$ ),  $\mathcal{Q} - q, \mathcal{Q}' \cup q$ )
            | 5FMHeuristic(ReferenceToEdge( $\mathcal{M}_i, q, k_1, k_2$ ),  $\mathcal{Q} - q, \mathcal{Q}' \cup q$ )
        end
    end
end
end

```

The second rule concerns the keys from a query q (line 12). When this key belongs to the same concept but in different rows, we avoid applying a merge between those rows and continue to the next query (line 13).

The third rule (line 14) applies the **Merge** rule when a query q uses two keys linked by a *Reference*. In this case, we apply merges in both directions (lines 15-16). Finally, to produce GO data models, references are transformed into *Edges* (line 17) with the **ReferenceToEdge** rule.

Thanks to this heuristic, the application of refinement rules will generate a tree of possibilities whose first nodes try splits and finish with merges (DFS). In addition, the use case reduces drastically the number of nodes for queries using few keys, and drives towards merges for multi-concept queries.

In our example of 4 concepts as input, the number of solutions, initially equal to 1,875, is reduced to 125. This number will be reduced even more by considering multi-concepts queries impacting the *Fubini's* number, and far more with queries involving several keys in a same concept (*Bell's* number).

2.4.5.4 Complexity of the Heuristic

Our problem is abstracted by a graph of solution $G(\mathcal{M}^*, \mathcal{T}^*)$, on which our heuristic has two reduction effects. First, it reduces the graph's size by pruning useless data models since it focuses only on the use case \mathcal{Q} , obtaining \mathcal{M}^{opt} . Second, it applies a DFS approach to avoid redundancies and removes all links in order to produce a tree of solutions called \mathcal{T}^{opt} .

Since the heuristic considers the distinct references used by queries $|refs(Q)|$, the *Fubini* number is impacted. *Bell* number with splits is also modified based on the size of distinct sets of keys from queries for a given row k : $|KeySet(Q_k)|$. Thus, the upper bound of the estimated number of solutions is:

$$|\mathcal{M}^{opt}| = Fn_{|refs(Q)|} \times \prod_{k=1}^{|\mathcal{R}|} |KeySet(Q_k)|$$

And the number of transformations to apply is: $|\mathcal{T}^{opt}| = |\mathcal{M}^{opt}| - 1$.

2.5 Experimental Study

In this section, we present the implementation of our approach shown in Figure 2.1. First, we present the metamodelling and model transformation development. Then, we detail the implementation of the ModelDrivenGuide, and the validation of the heuristic.

2.5.1 Metamodels and Transformation Rules Implementation

The implementation of our approach enlightens the models transformation process by showing the 2 level of transformation: 1) *Conceptual PIM* to *Common Logical PIM* Transformation; 2) *Common Logical PIM* to *MongoDB PSM* Transformation.

2.5.1.1 Experimental Environment

Since our approach is based on MDA, we need an infrastructure suitable for metamodeling, modeling, and for model transformations. For this, we developed our approach using a model transformation environment called *Eclipse Modeling Framework* (EMF⁴). To implement our approach, we have used:

- (1) *Ecore* for the implementation of all the PIM1, PIM2, and PSMx metamodels. Inspired by the object-oriented approach, the Ecore language is based on the notion of package (EPackage), class (EClass), attribute (EAttribute), reference link (EReference), data type (EDatatype), and enumeration (EEnum),
- (2) *XMI*⁵ a format in which instances of metamodels are created,
- (3) *QVTO* (QVT Operational) a Model-To-Model (M2M) transformation tool that implements the QVT language. It is used to formalize both exogenous transformation rules (from conceptual to logical model, and from logical to physical model), and also refinement rules.

2.5.1.2 Conceptual PIM to Logical PIM Transformation

We present here, the *ConceptualMMtoLogicalMM* Transformation, which is the first step of our approach.

Source. The source is the conceptual metamodel (PIM1) which is a simplified UML class diagram metamodel shown in (a) of Figure 2.18 implemented in Ecore. An instance of this metamodel is shown in (a) of Figure 2.19.

Target. The target is the *5Families* metamodel used for model transformation, presented in Section 2.4.2. Our metamodel is formalized in Ecore as shown in (b) of Figure 2.18 in order to verify all transformations and refinements of models. The instance of this metamodel shown in (b) of Figure 2.19 is automatically generated using the *ConceptualToLogicalMM* QVT transformation rules.

ConceptualMMtoLogicalMM transformation. UML classes, associations, etc. from the PIM1 (Conceptual Metamodel) are transformed into concepts, references, etc. of our PIM2 (*5Families* Metamodel) respectively. Rules are expressed in QVT language as shown in Figure 2.20.

⁴EMF: <https://www.eclipse.org/modeling/emf/>

⁵*XML Metadata Interchange*: <https://www.omg.org/spec/XMI/About-XMI/>

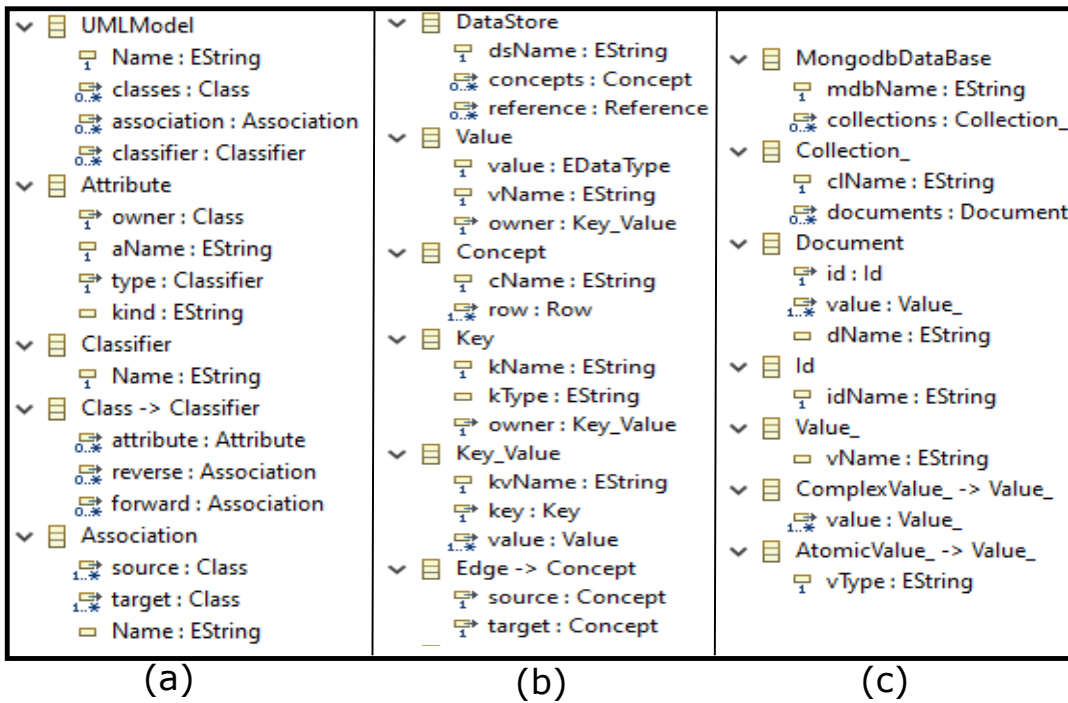


Figure 2.18: Ecore metamodels: conceptual PIM, logical PIM, and MongoDB PSM

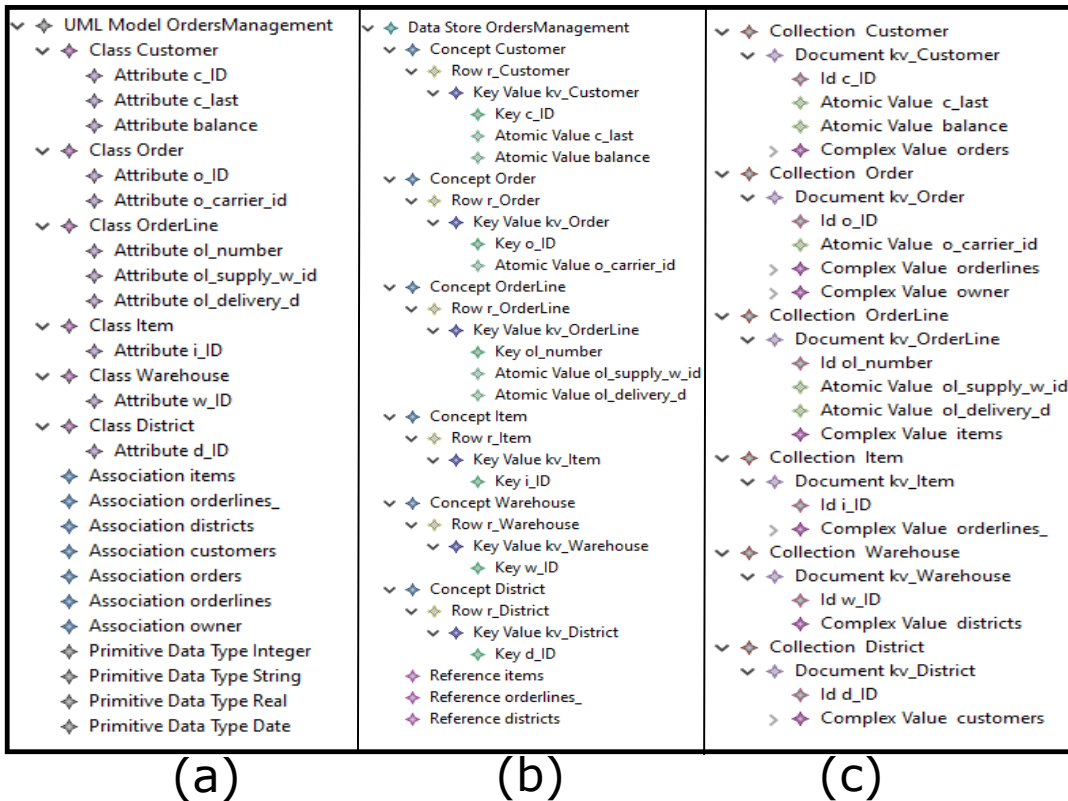


Figure 2.19: Class diagram, 5Families & MongoDB XMI models

2.5.1.3 Logical PIM to MongoDB PSM Transformation

After obtaining the different logical models from the 5Families common metamodel refinement, we transform them into suitable physical models related to the corresponding PSM metamodel. In our case, we transform our relevant data models into MongoDB model. The MongoDB metamodel is implemented in Ecore as stated in (c) of Figure 2.18.

```

modeltype conceptualPIM uses "http://UMLClassDiagram.com";
modeltype logicalPIM uses "http://GenericLogicalModel.com";
transformation TransformationUmlToColumnsOrientedModel(in Source: conceptualPIM, out Target: logicalPIM);
main() {Source.rootObjects()[ClassDiagram] -> map toDataBase();}
mapping ClassDiagram::toDataBase():DataBase{name := self.name ; table:=self.classes -> map toClass();
relationship:=self.links -> toRelationship();}
-- Transforming Class to Table
mapping conceptualPIM ::Class::toClass():logicalPIM::Table{name:=self.name;attributec:=self.attributec -> map
toAttribute();}
-- Transforming Attribute to Column
mapping conceptualPIM ::Attribute::toAttribute():logicalPIM::Attribute{if (self.attType="Oid"){aName:=self.attName;
aType:="Rid";} endif; {aName:=self.attName ; aType:=self.attType;}}
-- Transforming Binary Link to Relationship
mapping conceptualPIM ::Link::toRelationship():logicalPIM::Relationship{name:=self.name ; linkedtable:=self.linkedclass ->
map toLinkedTable() ; typer:=self.typer -> map toRelationType();}
mapping conceptualPIM ::LinkedClass::toLinkedTable():logicalPIM::LinkedTable{name:=self.name ;
cardinality:=self.cardinality -> map toCardinalities();}
mapping conceptualPIM ::TypeL::toRelationType():logicalPIM::TypeR{typer:=self.typer;}
mapping conceptualPIM ::Cardinality::toCardinalities():logicalPIM::Cardinality{nom:=self.nom;}
-- Transforming n-ary link to Table
mapping conceptualPIM ::Class::toClass1():logicalPIM::Tables{tName:=self.cName;key:=self.ident -> map
toId();columns:=self.attributes -> map toRefColumns();}
mapping conceptualPIM ::Attributes::toRefColumns():logicalPIM::Columns{aName:=self.attName;aType:="Rid";}
-- Transforming Association Class to Table
mapping conceptualPIM ::Class::toClass2():logicalPIM::Tables{tName:=self.cName;key:=self.ident -> map
toId();columns:=self.attributes -> map toRefColumns();columns:=self.attributes -> map toColumns();}
mapping conceptualPIM ::Ident::toId():logicalPIM::Key{kname:=self.iName -> map tokName();ktype:=self.itype -> map

```

Figure 2.20: ConceptualMMToLogicalMM QVT transformation

Source. The source is one of the data models (Logical PIM) generated by the refinement script.

Target. For the given logical model, the target is going to be one of the PSM metamodels. For all the PSMs implementation, we refer the reader to A. Ait-Brahim PhD [34]. Here, we present, as example, the transformation to MongoDB platform. The Figure 2.19 (c) shows the generated MongoDB XMI model.

LogicalMMtoPhysicalMM Transformation. In order to achieve this, a QVT rule is defined for each PSM metamodel. For the PSM of MongoDB, we transform Concepts to Collections, Rows to Documents, Complex Values to Nested documents, etc. The QVT transformation rule are shown in Figure 2.21.

```

modeltype LogicalPIM uses "http://GenericLogicalModel.com";
modeltype MongoDBPSM uses "http://MongoDBModel.com";
transformation TransformationGenericModelToCassandraModel
(in Source: LogicalPIM, out Target: MongoDBPSM);
main() {Source.rootObjects()[DataBase] -> map toDataBase();}
-- Transforming DataBase to MongoDB DataBase
mapping DataBase::toDataBase():MongoDataBase{name := self.name;collection:=self.table -> map toCollection();}
-- Transforming Table to Collection
mapping LogicalPIM ::Table::toCollection():MongoDBPSM::Collection {name:=self.name;}
atomicfield:=self.attributec -> map toAtomicField();complexfield:=self.islinkedto -> map toComplexField();}
-- Transforming Attribute to Atomic Field
mapping LogicalPIM ::Attribute::toAtomicField():MongoDBPSM::AtomicField {name:=self.name;}
-- Transforming Composition Relationship using nested data
mapping LogicalPIM ::LinkedToTable::toComplexField():MongoDBPSM::ComplexField {if(self.relationshipType =
"Composition"){name:=self.name;atomicfield:=self.attributeL -> map toField();} endif;}
mapping LogicalPIM ::Attributes::toField():MongoDBPSM::AtomicField {name:=self.name;}

```

Figure 2.21: LogicalMMToMongoDBMM QVT transformation

2.5.2 ModelDrivenGuide Implementation

Our ModelDrivenGuide approach is implemented in *Java*. It starts with an UML model automatically transformed into a relational model (\mathcal{M}_0 in Figure 2.10). Then, refinement rules are applied recursively on data models using the *Naïve* (generate all solutions) and the *5FMHeuristic* strategies.

In order to get the number of generated models and to see the gain obtained by our heuristic, a *signature* file has been implemented. It presents a sorted list of concepts, rows, keys, nesting and references. It ensures their uniqueness and detects identical data models produced by two distinct paths. It will help to cut cycles in the *Naïve* strategy and to count the number of duplicated data models.

Table 2.2: Generated models

Strategy	Nb models	Nb Splits	Nb Merges	Nb unique models
<i>Naïve</i>	2,313	1,646	666	1,445
5FMHeuristic	27	2	24	27

2.5.2.1 TPC-C

To illustrate our approach, we used the TPC-C⁶ benchmark giving a full use case mixing at the same time transactions, joins and aggregations. For this test, we focus on the three concepts (classes in the UML model): *Customer*, *Order* and *OrderLine*. Our driving example in Figure 2.10 is the PIM2 representation of the TPC-C benchmark.

Table 2.2 shows the number of generated data models, splits, merges and unique data models (thanks to signatures). The *Naïve* approach on TPC-C benchmark, produces 2,313 data models by applying 1,646 splits and 666 merges (1,445 distinct data models and thus 868 redundancies).

While the **5FMHeuristic** generates 27 data models with only 2 splits (few splitting queries) and 24 merges. The number of distinct data models is equal to the total number of generated models (no redundancies) which is the goal of our heuristic (DFS approach).

To compare our approach with existing work [49], it is important to remind that they generate a **unique** data model that corresponds to their targeting approach optimizing one criteria. In our case, we produce this data model among 27 others offering a large choice of possible models which contains this solution. After decreasing the number of generated models noticeably, it becomes an easier task for the IS designer to choose the most convenient one by considering the ease of implementation, NoSQL compatibility, security policy, storage or environmental impact, etc. We must notice that among those solutions it is possible to select a data model which can be implemented in several families of DB at the same time called Polystores [11].

2.5.2.2 Data Models Generation

In order to study the behavior of the *Naïve* and the **5FMHeuristic**, we have implemented a generator of data models that simulates the impact of denormalization strategies. This generator produces various data models by varying the number of concepts and keys, and also the topology of linked concepts (lines *vs* stars). We will produce the average number of generated data models per strategy. Moreover, random queries are generated to see the effect of use cases by varying the number of joins and involved keys. Our code is available on Github⁷ with a configuration file to parameter the generator.

To study the impact of the heuristic on the applied rules (Merge, Split and both of them), thanks to the aforementioned signatures, we will count the average number of generated data models from the initial data model (relational) and the generated models before and after applying the heuristic.

Figure 2.22 (a) shows the evolution of the number of generated data models with the *Naïve* approach and our **5FMHeuristic**. We focus only on merges by limiting keys to primary and foreign keys. The *Naïve* approach follows an exponential growth (dashed curve) but slightly lower than the *Fubini* number since simulated data models do not contain cycles (with references) nor star-shaped data models which lead to more combinations and reach this upper-bound. The *Naïve* unique approach avoids cycles and duplicates which reduces significantly the number of solutions. Our heuristic targets required merges only, which can witness a decrease when it reaches bigger data models (here 7 concepts). In fact, since the number of queries/joins are a constant, their impact on the number of merges becomes noticeable. Thus the number of possibilities decrease rapidly.

Figure 2.22 (b) focuses on splits varying number of keys on a single concept. The *Naïve* approach shows the distribution of the *Bell* number with duplicates (dashed curve & *Naïve* unique). The number of data models produced by **5FMHeuristic** is reduced sharply for splits. Here also the number of possible splits is

⁶ http://www.tpc.org/tpc_documents_current_versions/pdf/tpc-c_v5.11.0.pdf

⁷ https://github.com/leonard-de-vinci/5FM_generator/

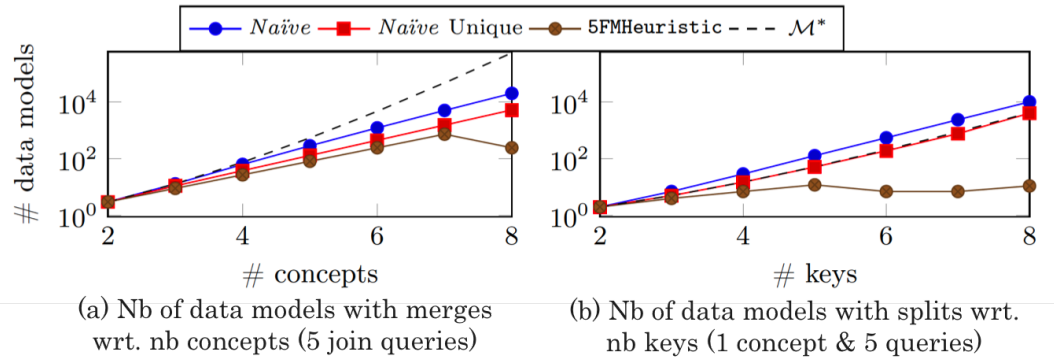


Figure 2.22: Number of data models

bounded by the number of involved queries reaching a threshold based on the size of the KeySet (simplified Bell number to the KeySet).

The impact of the use case on the 5FMHeuristic is shown in Figure 2.23 (a) by varying the number of keys per query. It's applied on 8 concepts of 5 keys, each with different numbers of queries (no joins - no merges). We notice that only single-key queries produce the maximum number of splits since it offers all possibilities, while this number decreases with bigger queries. When the KeySet size is high, it implies that splits cannot be applied since all keys must be put together in a same row. Thus, we can conclude that small queries produce more data models to find more adapted solutions.

Figure 2.23 (b) focuses on join queries by varying the number of joins per query. Contrary to the splits, the bigger the joins are the more data models are generated. This is due to the fact that queries allow merging more concepts with each other allowing to produce data models with a single concept.

Figure 2.24 plots the global impact of the 5FMHeuristic with both merges and splits by varying the number of concepts and the number of keys for each. With 5 different queries (filters and joins), it is interesting to see that the distribution remains similar for all cases. The number of keys per concept has an impact on possible splits and consequently on merges. For 8 concepts, it varies from 726 for 2 keys/concept to 16,767 data models for 4 keys/concept. We plotted the $|\mathcal{M}^*$ complexity (dashed curves) for each key sizes which shows the exponential growth of the produced data models, at most 28,383,420 solutions for 8 concepts of 5 keys. We can see that the 5FMHeuristic drastically reduces the search space.

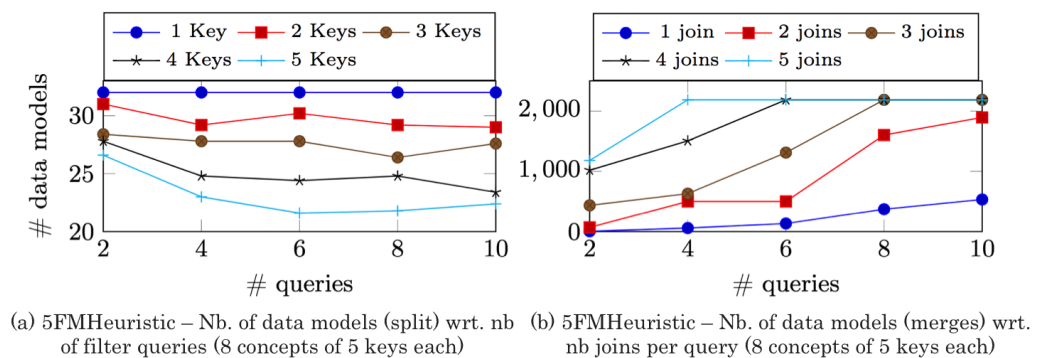


Figure 2.23: 5FMHeuristic - Number of data models

To conclude, queries with more joins and few keys have more impact on the number of generated data models than other ones. Moreover, our heuristic implies that splits impact possible merges due to denormalizations' combination.

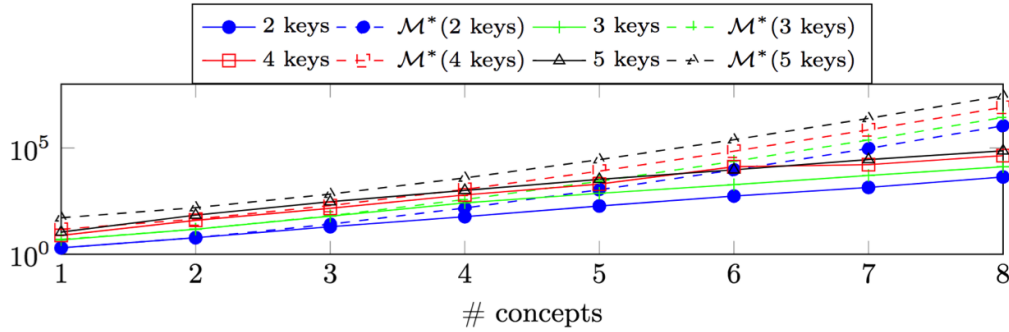


Figure 2.24: 5FMHeuristic - Number of data models wrt. nb of concepts varying keys (5 queries)

2.6 Conclusion

In this chapter we have presented our model driven approach that aims to help, automate, and guide the development of big data IS. The cornerstone of our approach is the common logical metamodel that describes the 5 families of models (relational and the 4 NoSQL families). As part of A. Ait Brahim, PhD thesis, we have presented QVT transformation rules that aims to automate concept-to-concept mapping, from conceptual PIM to the logical PIM and then, to the different target platforms (PSMs). Then, as part of J. Mali PhD thesis which is in progress, we have proposed refinement rules, particularly for splitting and merging concepts' rows, that generates all data models by recursive denormalization to find suitable solutions. Then, a heuristic of generations based on DFS and the use case, allows avoiding the explosion of solutions in terms of the amount of data models and paths to obtain them. Our goal is to offer the IS designer a limited solution space of which he can choose the best model that will be adapted to his constraints (*e.g.*, efficiency, green computing, integration, polystores, security). The IS optimization goes through: a) the choice of denormalization level, b) the choice of the target NoSQL solution, c) the choice of associated indexes and sharding strategies, and d) the combination of the DB transactions. We propose the first full formalization of the problem and issues with refinement rules and data models manipulation.

As this work is in progress, currently, we are working on a global cost model that aims to enrich the choice process and help the designer in his choice by associating each model of the solution space with an implementation cost and then sort them accordingly. This cost model will integrate three dimensions: **time cost**, **environmental cost** and **financial cost**. The time cost model calculates a cost of each query on each data model. The cost of a data model is the sum of costs of the queries on this data model. The environmental cost shows the impact of carbon footprint on the environment, and financial cost is calculated according to the cost of cloud solutions and the number of servers.

This chapter presents a part of the work carried out within the SAFECARE¹ H2020 project. The SAFECARE project aims to provide solutions that improve both physical and cyber security of healthcare organizations in a seamless and cost-effective way. Twenty-one European partners were involved in this project. I collaborated with four permanent members of my team (F. Hamdi, N. Lammari, N. Mimouni, and S. Cherfi), alongside F. Hannou (post-doc), and M. Rihany (research engineer). CNAM's share was 450,000 €, out of a total budget of 7,573,979 €. We were responsible for a fundamental task, which involved developing the impact propagation module, and we were actively engaged in four work packages. Our contributions were published in [16]/10, [17]/32, [18]/34, [19]/35, [68]/7, [69]/1.

Healthcare organizations constitute intricate socio-technical systems that encompass human engagement, business processes, and advanced Cyber-Physical Systems (CPS). These systems amalgamate cyber and physical infrastructure, placing patients, their well-being, and security at the core. Within CPS, the distinction between the cyber and physical domains is becoming increasingly blurred. Indeed, with the recent advances in cloud computing, the Internet of Things (IoT) and other information technologies, the face of healthcare systems is changing. By adopting the usage of Electronic Patient Records, wearable sensors or in-home remote patients monitoring, healthcare organizations are now able to provide more personalized services. This progress induces sharing information about health services, resources availability (beds and medical personnel) or patients' data through open and controlled platform. It also offers new opportunities for new applications such as disease treatment, medical research, care services, etc. Unfortunately, these developments rely on common communication interfaces and standards and thus enhance security breaches exposing hospitals to several threats. Besides, as healthcare organizations deal with human being health and lives, damages are mostly more severe. According to IBM data breach report², healthcare organizations had the highest costs associated with data breaches with \$6.45 million. To increase the efficiency of solutions, it is necessary to examine all the problem facets. The objective of our work is to provide an approach for risk assessment and analysis to be able, further to provide suitable solutions to ensure security of hospital and healthcare organization.

This chapter is organized as follows: Section 3.1 presents an example of a cyber-physical attack scenario that helps to state the problem. Section 3.2 reports the related work. Section 3.3 is dedicated to the ontology-based model. Section 3.4 presents the Safecareonto implementation before concluding in Section 3.5.

3.1 Problem Statement

We consider the "patient data theft" security attack scenario as an ongoing illustration. Patient data holds significant importance, and any unauthorized access to healthcare databases jeopardizes patient privacy, potentially leading to extortion attempts targeting both individuals and medical institutions. This attack can occur through two methods: remote infiltration via an accessible web service that allows entry to the web server, or on-site infiltration by breaching the hospital premises and gaining access to the local network. In both scenarios, the attacker compromises the Picture Archiving and Communication System (PACS) to illicitly access and pilfer health-related data.

- (1) The attacker identifies an ethernet plug in the waiting room of the hospital,
- (2) He/she plug an ethernet cable,
- (3) He/she access the IT network of the hospital,
- (4) He/she scan the network to identify the PACS,

¹<https://www.safecare-project.eu/>

²<https://www.ibm.com/downloads/cas/ZBZLY7KL>

- (5) He/she escalate to administrator privilege,
- (6) He/she connect to the PACS and install executable payload and access data,
- (7) He/she exfiltrate patient data,
- (8) **Result:** Privacy disclosure and patient data theft.

In this scenario, the intrusion into the hospital should trigger both physical and cyber protection mechanisms, considering the nature of the initiated acts. The main challenge facing security architects is to prepare the system to counter this type of complex attacks. This task is particularly complicated since attackers operate in a hybrid cyber-physical fashion, whereas most of the current security systems deal with physical and cyber threats independently.

The purpose of our work is to propose a solution able to:

- Identify the critical assets and their properties,
- Assess the risk to which they are susceptible, considering the characteristics of the assets, their interconnections, and the existing safeguards.
- Provide information to help prevent the propagation of incidents in case of attacks.

Our solution is built upon an ontology-based model. Within this framework, two fundamental challenges come into play: knowledge acquisition and knowledge representation and utilization, specifically tailored to address the targeted security issue. To tackle these challenges, we introduce an incremental approach. The primary goal of this approach is to gather implicit and explicit knowledge concerning the management of structural and behavioral incidents. This process unfolds in three distinct phases. In the initial phase, we elicit both implicit and explicit knowledge through a comprehensive analysis of attack scenarios, conducted in collaboration with security experts from hospitals. This step serves to elucidate and codify the knowledge. The subsequent phase involves structuring this accumulated knowledge into a modular ontology, providing a coherent framework for its organization. In the final phase, we leverage these conceptual elements, refining and reconfiguring them to align with the overarching objective of preventing incidents' propagation.

3.2 Related Work

Ensuring system's security and facing cyber or physical attacks raised major concerns for both practitioners and academics. As commonly known knowledge bases, we mention the Common Vulnerabilities and Exposures (CVE)³, the Common Weakness Enumeration (CWE)⁴, and the Common Vulnerability Scoring System (CVSS)⁵ promoted by MITRE⁶. These bases provide a reference-method for known information-security vulnerabilities and exposures. In [124], the authors present the Unified Cyber security Ontology (UCO) that unifies most commonly used cyber security standards. The NIST institute promotes a more general vulnerabilities ontology [33] as an open industry standard for assessing the severity of computer system security vulnerabilities.

Based on the modeled security breach, we can classify the existing work on cyber and/or physical security modeling into two main categories: risk & threat, and attacks & incident modeling approaches. For each category, a particular attention is given to ontology-based and healthcare dedicated contributions.

3.2.1 Risk & Threat Modeling Framework

Several risk management standard and frameworks have been established to improve the systems' security. Most of the frameworks are based on security and risk standard management process like ISO 14971: 2019⁷. Microsoft promotes the STRIDE threat model as mnemonic that categorizes threats into spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privileges [116]. Each

³<https://cve.mitre.org/>

⁴<https://cwe.mitre.org/>

⁵<https://www.first.org/cvss/>

⁶<https://attack.mitre.org/>

⁷ISO 14971: 2019, Medical devices-Application of risk management to medical devices

of the six threat could exploit the components of information assurance and has an attendant security property that would address the threat. This model is used as part of Microsoft's Security Development Lifecycle to help define the attack surface. The European Commission reported a generic classification of threats in which natural hazards are distinguished from non-malicious man-made hazards and malicious man-made hazards [126]. The Health Information Trust Alliance (HITRUST) [71] provides a framework based on threat taxonomy that distinguish logical, physical, and organizational threats levels. We can also make reference to the CIM standard developed by DMTF (Distributed Management Task Force), an internationally recognized organization, accredited by bodies such as the American National Standards Institute and ISO. Furthermore, numerous security risk analysis methodologies provide asset descriptions, such as EBIOS risk management [52] and MAGERIT 3.0 [12].

Dedicated solutions was provided as in [53], where the authors present the most common threats affecting ICS/SCADA systems. On the other hand, research studies have dealt with the problem of threat management. Regarding assets relationships, we can consider research that give particular attention to the hierarchical links between assets [131, 127, 36]. In [43], the authors present an ontology-based approach that provides classification, relationships, and reasoning about vulnerabilities and threats.

In [114], the authors propose an ontology of risk as a semantic foundation for the representation of risks in enterprise architecture. This ontology integrates three main perspectives on risk: risk as a quantitative notion; risk as a chain of events that impacts an agent's goals, and risk as the relationship of ascribing risk. In [38], the authors present a rich taxonomy of operational cyber risk that attempts to identify and organize the sources of cyber risks. For physical risk assessments, [91] propose a list of threats related to terrorism. We can also find in "Common Criteria" and ANSSI portals security protection profiles for some software and physical equipment of critical infrastructure where threats affecting these components are listed. In [128], the authors present an ontology of hazards and threats that could affect a critical infrastructure. This ontology covers four sectors: energy, transport, water and telecommunication. In the healthcare field, the work presented in [54] provides an overview of the cyber threats that jeopardize smart hospitals. In [9], the authors present taxonomies of threats for healthcare infrastructures.

3.2.2 Attack & Incident Modeling Framework

The MITRE provides the CAPEC⁸ knowledge base that reports attack patterns in cyber security. In [100], the authors propose a taxonomy for classifying security incident that focuses on the cross domain and impact oriented analysis. The work presented in [92], provide a detection model for events occurring in cyber physical systems. In [1], the authors propose a model driven framework based on EBIOS [52] and on attack trees method, in order to identify the critical parts of the systems.

3.2.3 Discussion

The examination of the current body of knowledge reveals variations in the focus of provided standards, knowledge bases, and research contributions. These differences are based on distinct primary objectives: storing common vulnerabilities, modeling and evaluating risks and threats, or depicting incidents and their subsequent consequences. Despite the increasing fusion of interconnected cyber and physical elements, the realms of physical security and cyber security continue to be treated as separate entities. To create a more comprehensive approach, a security mechanism should be formulated to encompass the entire system, rather than exclusively targeting specific segments [14].

Moreover, prevalent healthcare and security ontologies tend to be specialized for particular functions, such as security requisites or mitigation efforts. Besides, most of these ontologies is confined to either the cyber or physical domain, disregarding the intricacies of cyber-physical interactions. It is insufficient to merely juxtapose these distinct aspects. Many healthcare ontologies [60] primarily revolve around medical procedural terminologies, often overlooking security dimensions

Hence, it is crucial to establish an approach that comprehensively addresses the diverse facets of both cyber and physical security. This entails furnishing a semantic depiction of assets, encompassing their

⁸<https://capec.mitre.org/index.html>

vulnerabilities, potential threats, and the range of impacts they may encounter. Furthermore, this approach should encompass incidents and their subsequent cascading consequences.

3.3 An Ontology for Security Management in Healthcare CPS

We propose the **SafecareOnto** that describes healthcare assets, and their cyber and physical security. We target a high-quality ontology that faithfully formalizes experts knowledge, and guarantee reusability. Accordingly, we choose the common main phases across examined methodologies [58], and organized them in a process that considers genericity and usability. The maintained phases that we follow to build the Safecare ontology are: **knowledge elicitation** (Section 3.3.1), **representation** (Section 3.3.2), and **implementation and validation** (Section 3.4).

3.3.1 Knowledge Elicitation

Knowledge acquisition refers to the process of collecting the necessary data to design and populate a knowledge repository, such as an ontology. In the literature, some standards and ontologies are available and are considered as an initial input to understand and formalize the knowledge of the domain. In order to refine this information, an efficient process requires interacting with field experts that hold the domain knowledge, i.e., security in healthcare CPS. This task strives to faithfully capture the system's essential elements that would appear as key concepts in the ontology. It is also important to mirror the way different processes are implemented to represent semantic relationships.

For efficiency and effectiveness purposes, the best practices highly recommend the usage of a data collection methodology, describing the extent of data with the highest added value, and avoiding by the same to omit valuable knowledge. It is necessary to understand the use case constraints to set the right methodology that fits the study requirements correctly.

To ensure a reliable outcome, the following guarantees should be provided:

- **Heterogeneity of terminologies:** the ontology should capture the semantics of security in healthcare CPS, which implies a diversity of used vocabulary. This heterogeneity is to add to the linguistic and technical variations underlying the multiplicity of end users. A unique methodology centralizes similar concepts for a highly coordinated glossary.
- **Hybrid communication channels:** the collection process is designed to alternate two phases. A passive collection where experts autonomously fill methodology support files and active collection phases where ontology designers communicate with experts to check and validate the received data. Even with a fine-grained definition of the methodology, exchange sessions allow verification and validation enabling a higher exploitation rate.
- **Reuse and genericity:** the project involves several end users (hospitals), and many experts from the cyber and the physical security fields. The collection process should cover the specific features of each hospital security system to subserve a fully generic model. The collection has to be reusable as many times as necessary to interface with different partners.

To precisely address the issues raised in the motivation section, the methodology has been designed to handle a single attack scenario at a time. This allows a personalized study comprising three phases:

- (1) Collect the primary list of assets,
- (2) Refine asset lists by exploring their interactions (relationships),
- (3) Describe assets regarding risks and protections.

The figure 3.1 illustrates the entire process.

Phase A: Primary asset identification. Given a security attack scenario, and following a particular operating mode (for example the scenario presented in Section 3.1), the first phase consists in identifying the list of primary assets, i.e., those directly involved in the actions of the attacker. An excel file is provided to the expert, who indicates the list of assets corresponding to each action.

Example 4 (Assets identification): Consider the actions A1 and A4, given a hospital H, the primary assets involved in each action are depicted in Table 3.1.

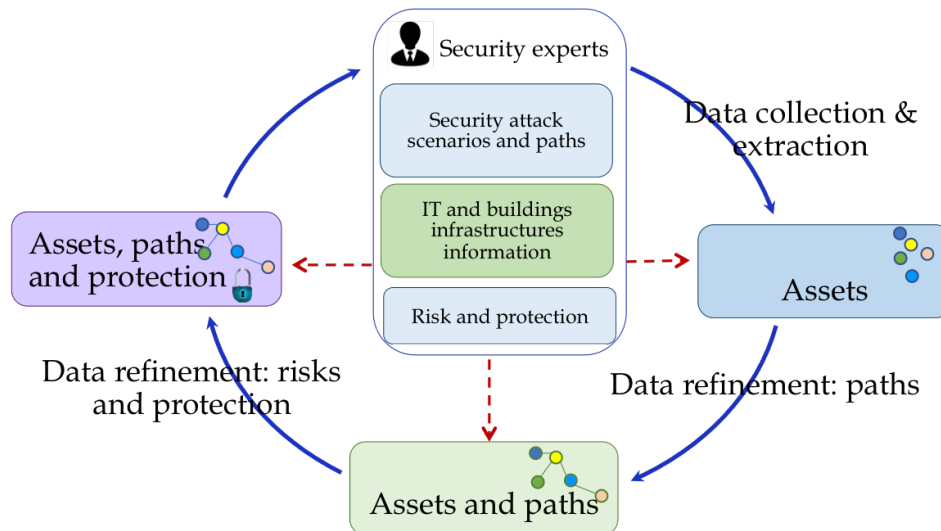


Figure 3.1: Data acquisition methodology phases

Table 3.1: Primary asset identification

Action code	Description	Impacted assets
A1	plug an ethernet cable in the waiting room of the hospital	ethernet plug waiting room
A4	network scan to identify the Picture Archiving and Communication System	core Network PACS

Phase B: Assets relationships. All assets in the hospital are interconnected, either through physical or cyber relationships. These links are potential vectors for incident propagation. The assets identified in phase A as those primarily impacted by attacker actions are linked to other assets, and these links are potential vectors for incident propagation. Phase B is a refinement stage that enables extending the primary asset list to all assets reachable via cyber or physical relationships. To build the relationships graph, two input files should be acquired:

- (1) **Locality architecture:** a schema of hospital rooms localities and functions. It is necessary to enrich this representation with physical devices (medical, computer, cameras, etc.) located in each room to consider physical access links.
- (2) **Cyber architecture:** network infrastructure detailing how different assets communicate.

Example 5 (Cyber and physical infrastructure): The hospital H produces two materials: physical topology of the radiology service, augmented by the list of devices it includes, and the network architecture linking these devices to the network. Based on these files, a graph of dependencies is built to display how primary impacted assets from Example 4 are connected to other cyber/physical assets. Figure 3.2 shows the produced graph.

Phase C: Assets description. Beyond the asset identification phases, the extended list of assets should be enriched with an appropriate description of each single asset following two dimensions:

- (1) **Risks:** assets are exposed to various risks depending on their nature and function. Beyond the attacker’s actions’ scope, it is necessary to gather knowledge about all risks that a particular asset may undergo because some of these risks could rise as cascading effects of the initially intended damages.
- (2) **Protection:** protecting assets are any physical or cyber elements used to stop or reduce damage. These protections could operate before attacks (preventive), during attacks (protection and defense) or to recover a stable state (resilience). They have different efficiency degrees used to determine optimal securing strategies.

Chapter 3. Modeling Security in Healthcare Cyber-Physical Systems

3.3. An Ontology for Security Management in Healthcare CPS

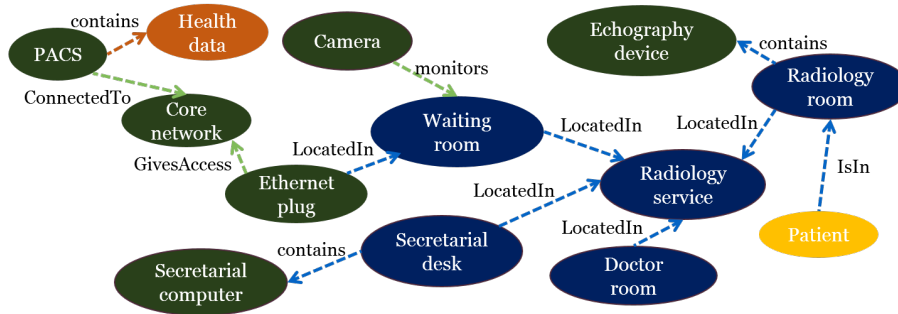


Figure 3.2: Assets relationships extracted from cyber and physical architectures

Example 6 (Assets description): Security experts indicate for all the assets (Examples 4 and 5), what type of risks they are exposed to, and implemented or possible protections. An example of expert knowledge collected, is detailed in Table 3.2.

Table 3.2: Assets description

Asset	Risk	Protection
Health data	theft: cyber	encrypting protection 90%
Waiting room	unauthorized access: physical	door access control protection 100%
Ethernet plug	unauthorized connection: cyber	disconnection prevention 100%

Based on this knowledge acquisition methodology, we adopt a bottom-up fashion for ontology construction. This approach starts with the data instances gathered from experts as input, and operates an incremental generalization task, to identify high-level ontology concepts and axioms.

At the validation phase, alignment tables enable checking the correspondence between ontology concepts and domain knowledge.

3.3.2 SafecareOnto

We propose a modular ontology shown in Figure 3.3 with a central module said Core ontology, and two related and additional modules dedicated to protection and impact propagation.

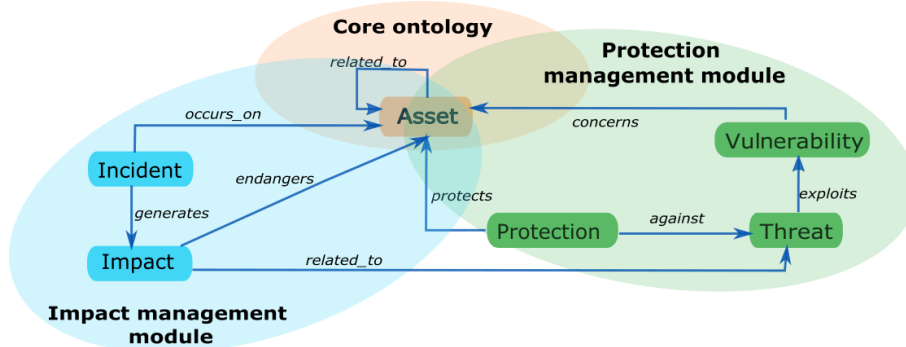


Figure 3.3: The conceptual view of SafecareOnto

The core ontology captures essentially the static knowledge about critical assets. The main purpose of the SafecareOnto is managing security and more precisely incident propagation in order to provide solutions to both prevent and counter the escalation of security incidents. To cope with this objective, the core ontology evolves around the concept of **Asset** and **propagation patterns**.

An **asset** is any "thing" that has value. Assets could be business assets such as "personal data" about "patients" and "personnel" or support assets such as "medical equipment" or "IT devices".

An asset is described by a set of properties that could be common to all assets, such as a unique identifier, essential for identification, localization and a state to track overtime the asset. Assets are related to other assets through several kinds of relationships. Remember that the objective of the ontology is to serve as a basis to reasoning about incidents propagation. To this aim, we have identified 4 structure-based propagation patterns representing 4 propagation schemes.

The reasoning behind our solution is that the spread of incidents is influenced by various factors. These factors encompass the structural arrangement of assets, encompassing their inherent and situational attributes (controlled by the core ontology), their status concerning security breaches (governed by the protection management module), and the characteristics of incidents (handled by the impact management module). The relationships patterns, summarized in Table 3.3 highlight the structural factors.

The protection management module describes protection of assets against attacks. Each asset could have one or several weaknesses said **vulnerabilities** that could be exploited by a **threat** that is a potential of impairment of an asset. A **protection** could be an asset or a policy that protects an asset from **threats**. The information about vulnerabilities needs to be updated consequently to regular maintenance operations or after incident analysis. A **protection** is a solution against a **threat**. For example, a camera is protection against a threat which is unauthorized access.

The impact management module defines the concepts that are essential to the computation of impact propagation and provide indicators to help decide about the suitable countermeasures to face attacks. It relies on **incident** and **impact** concepts.

An **incident** is adverse actions performed by a threat agent on an asset. When an **incident** occurs, there is a risk that it propagates to related assets. An **impact** is the result of such propagation. This propagation needs to be precisely qualified and/or quantified to efficiently help decide about the mitigation plans. In SAFECARE, we handle both physical and cyber incidents. We also have to assess the severity of an incident to better compute its propagation. This assessment considers the known vulnerabilities and protections of the assets. An incident could be the expression of known or unexpected threats. It depends on the probability that a threat will exploit a vulnerability.

3.3.3 Core Ontology Conceptualization

The objective of this phase is to establish a conceptual and structured model for the ontology. The ontological framework encompasses both static knowledge (comprising concepts, relationships, and attributes) and dynamic knowledge, which is formalized through axioms.

3.3.3.1 Concepts Identification and Definition

During the conceptualisation phase, the concepts and their relationships are refined.

Asset concept is a subclass of owl : Thing ($Asset \sqsubseteq \top$) and is further specialised into a set of subclasses that constitute a partition of the concept **Asset** since they have no common instances and that their union completely covers the concept **Asset** as defined for the domain [72]. Given that the ontology's objective is to serve as a basis for reasoning about incidents propagation, we classify the assets according to the nature of the incidents they may suffer from and those they likely propagate. The following set of subclasses constitute a partition of the concept "Asset" since they have no common instances and that their union completely covers the concept "Asset" as defined for the domain [72].

- **Network** ($Network \sqsubseteq Asset$): a computer network denotes a communication and data exchange channel linking at least two devices (nodes). A network can be wired or wireless. There are internal networks in hospitals mainly dedicated to support business processes (medical IS, building management system as surveillance or access control). The hospital has external connections to provide stakeholders with the necessary access to achieve their tasks. Patients have also access to web services for appointment booking, for example. Networks link the different components of the hospital

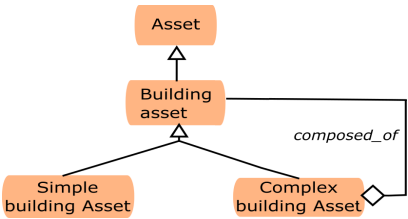
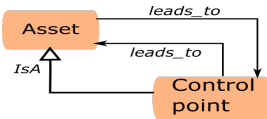
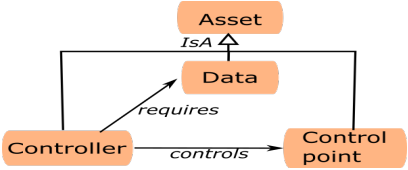
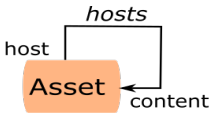
Pattern	Description
 <pre> classDiagram class Asset class Building_asset class Simple_building_Asset class Complex_building_Asset Asset < -- Building_asset Building_asset < -- Simple_building_Asset Building_asset < -- Complex_building_Asset Building_asset o-- Complex_building_Asset : composed_of </pre>	<p>The whole-part pattern assumes that if an incident happens on a whole, then it could impact its parts. Inversely, if parts are attacked, the whole could also suffer from the consequences of the attack. This pattern applies to several assets and essentially to assets representing locations. In SafecareOnto, they are referred to as Building assets. the propagation through these structures are essentially "physical incidents" such as "unauthorized access". For example, an intrusion on one floor of a hospital could potentially affect all the rooms on that floor.</p>
 <pre> classDiagram class Asset class Control_point Asset < -- Control_point : IsA Asset --> Control_point : leads_to Control_point --> Asset : leads_to </pre>	<p>Leads to pattern captures the access and communication possibilities between assets. This access applies for both physical or cyber flows and is materialized through a specific asset referred to as Access point. As an example we could mention a door that allows access from a room to another or a port that is a communication end point in a network. An access point could be one way or bidirectional to represent the possible flow directions explicitly.</p>
 <pre> classDiagram class Asset class Data class Controller class Control_point Asset < -- Data : IsA Controller --> Data : requires Controller --> Control_point : controls Data <--> Control_point </pre>	<p>Controls pattern allows specifying the conditions and mechanisms for granting or revoking access to assets. This pattern is composed of three elements: the Controller applies the access policy, the Control point representing the access point and the Data representing the policy applied by the controller. For example, a smart card based system is composed of: the access rights stored locally or remotely, door readers to check whether data on the card is consistent with the policy and the door.</p>
 <pre> classDiagram class Asset class Asset2 class Asset3 Asset --> Asset2 : hosts Asset --> Asset3 : content </pre> <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <pre> a1 = Asset.content a2 = Asset.host (a1.category=device AND a2.category=softwar OR (a1.category=device AND a2.category=data) .. </pre> </div>	<p>The hosts-content pattern assumes that if an incident happens on an asset named host asset then the content, referred to as content asset could be affected by this incident. The structure of the pattern is enriched by rules to enhance the validity of the relationships description. For example, if the host is a device, IT or medical, a content could be software.</p>

Table 3.3: Structural patterns

cyber-infrastructure.

- **Building** asset ($Building \sqsubseteq Asset$): a building asset is a geographical entity that corresponds to the building in which a hospital (or a part of it) is located. An asset building has a variable granularity going from "room" to a "complex of buildings". The granularity level is stored as an attribute value, together with the use of the location, and its physical level (ground, 1, 2, etc.). The concept Building asset has two subclasses: a **simple building** asset (a non-divisible location in the map, $SimpleBuilding \sqsubseteq Building$) and **complex building** asset that groups multiple simple building asset ($ComplexBuilding \sqsubseteq Building$). The set of instances of the building asset corresponds to the hospital's physical infrastructure.
- **Staff** ($Staff \sqsubseteq Asset$): staff represents any physical person performing regular or occasional tasks within the institution (hospital). In addition to direct employees, the staff includes external stakeholders acting on-site or remotely. This modeling aims to extend assets to the staff who interact with hospital resources and who is exposed to possible threats related to its activity. A staff member has a name, a phone number and a role (administrative, technical or medical). This concept willingly excludes patients since they are covered by another level of formalization (business processes).
- **Device** ($Device \sqsubseteq Asset$): device refers to any tangible equipment, whether associated to a computer software with an automatic action (camera, sensor, server) or not (door, lamp). Devices can be classified according to various criteria. In the healthcare context and given the purpose of managing incident propagation, we consider the following classification: computer device $ComputerDevice \sqsubseteq Device$ (used for software hosting to achieve the hospital's missions), building equipment such as doors, chairs, including those equipped with an automated operating process $BuildingDevice \sqsubseteq Device$ (sensor, camera), medical devices $MedicalDevice \sqsubseteq Device$ (scanner, echography device, or pacemaker). The medical devices are manipulated by specialists and are intended to serve for patients' care. We group devices in the categories above, based on their possible interactions with other classes, which is the main matter in propagation. In general, devices have global features such as the producer, the commissioning year or their operational state.
- **Software** ($Software \sqsubseteq Asset$): softwares are virtual programs (sequences of computer code) with data processing capabilities. They support a determined business process such as medical acts, human resources, finance, or building management. Softwares are the target and a vector for many cyber incidents, and most of the hospital processes depend on their reliability. We record certain properties that enable determining their role in the incident propagation: product name, version, editor, technology, and the platform (web, computer, phone). Note that operating systems are also considered as softwares.
- **Data** ($Data \sqsubseteq Asset$): data play a major role in security management since multiple attacks are carried out using or targeting data. Data are manifold: access control policies, surveillance system, device configurations, patients' health data, research results. We identify two major categories: **patient data**, and **operating data** used to ensure the good working of the hospital (access policies, camera flows, personnel data, metadata).
- **Access point** ($AccessPoint \sqsubseteq \top$): the most crucial part of the incident propagation task is identifying entry points providing access to critical resources. The access points are generally the gateways that enable the use of the resource, and by the same, the occurrence of the incident. That is why security engineers attempt to maximally secure these elements by implementing strong protections. The access point concept is valid for a physical resource (an office's door) as for cyber resources (a network port). Access points are valuable to the hospitals, thanks to their security management role and are considered assets.
- **Controller** ($Controller \sqsubseteq \top$): Controllers are either physical devices or virtual protocols responsible for enforcing access restrictions to assets, as defined by predefined access policies. The access to the surgery rooms requires a door (access point), an entry supervised by a door access controller. Computer is secured by a login page and controlled by a password checker. It is essential to identify

asset access and control mechanisms at the appropriate granularity level, to ensure safe use, and anticipate a possible incident occurrence and propagation.

3.3.3.2 Relations Identification

The relationships depict how assets interact in the healthcare context and what are their properties. We have identified two families of relations:

- The first one corresponds to concepts Attributes (data properties in OWL): a staff **hasRole**, a building **hasLevel**, a software **hasVersion**, Complex buildings are **composed Of** several buildings (complex or simple), which can be spread over multiple vertical levels (floors), etc.
- The second family of relations results from our analysis of propagation channels. This analysis revealed 4 structural patterns that help reasoning on propagation of incidents according to their nature (cyber or physical). We detail these patterns in Table 3.3.

Based on the top-level concepts defined in the previous section, the relationships depict how assets interact in the healthcare context and what are their properties. **Complex buildings** are **composedOf buildings** (complex or simple), which can be spread over multiple vertical levels (floors). The **building** host the **devices** (**HostDevice**), those dedicated to healthcare (**medical device**) and those ensuring other processes support (**operating devices**). Unlike machines, **personnel** are fast moveable assets and have a frequently changing location over time within the hospital. The frequency of the propagation of the incident being often faster, there is no interest in modeling timely the location of persons.

The connection between the physical and the cyber assets is represented by the relations linking devices to other cyber assets: **devices hostSoftware** as operating systems, **hostNetworks** as for wired connections and **hostData** as data configuring **medical devices**.

As explained in the knowledge acquisition step, two architectures enable identifying possible propagation paths: the cyber and the physical architecture. The ontology formalizes these maps thanks to the **leadsto** relations. **leadsToAsset** and **leadToAccessPoint** denote communication channels that provide access to **assets** through **access points**, for example, the waiting room leads to the door, which leads to the doctor's office. As stated earlier, the indication of the door entrance is required to identify where protections should be implemented to reduce the incident occurrence probability.

Controllers effectiveness depends on the definition of access policies that grant or deny access to the asset usage. As access policies are of type **operating data (isA)**; controllers **requireData** operating data. Take the example of a "door access controller" that manages the access point "door", it **requires** on-the-spot checking of the "user's credentials": access codes or biometric data such as fingerprint, voice, facial pattern or iris.

3.3.3.3 Axioms Definition

The use of axioms allows expressing certain capabilities or features of a concept and avoid adding new concepts that would not be reused [123]. For SafecareOnto, a set of formal axioms is defined to specify ontology elements. In the following, we introduce a set of axioms and their description.

- $simpleBuilding \sqsubseteq building$: subclass axiom, all simple buildings are buildings,
- $AccessPoint \sqsubseteq device \sqcup building \sqcup software \sqcup network$: an access point can be either a device, a building, a software or a network,
- $Controller \sqsubseteq \forall controls (building \sqcup device \sqcup network \sqcup software)$: a controller controls a building, a device, a network or a software,
- $PhysicalIncident \sqcap CyberIncidents \sqsubseteq \perp$: CyberIncidents and PhysicalIncidents are totally disjoint concepts,
- $data \sqcap requiredby Controller \sqsubseteq operatingData$: data required by a Controller are exclusively Operating Data,

- $leadsToAsset, leadsToAccessPoint \sqsubseteq leadsTo$: $leadsToAsset$ and $LeadsToAccessPoint$ are sub-relations of $leadsTo$ relation,
- $ComplexBuilding \sqsubseteq \exists composedof Building$: $ComplexBuilding$ is composed of at least one $Building$.

3.4 Implementation

The ontology evaluation strives to validate its effectiveness and efficiency to support security experts in the asset's preventive monitoring and for the mitigation of cyber and physical security incidents' impacts. This section shows how Safecareonto was implemented and illustrates two direct use cases for the ontology assessment.

3.4.1 SafecareOnto Implementation

To implement the ontology, we have used Protégé [104], which is an ontology and knowledge base editor that enables the construction of domain ontologies, and comes with visualization packages. Also, it offers reasoning tools to support editing inference rules for the impact propagation application. Figure 3.4 depicts an extract of the Safecare ontology designed in Protégé. Here, we present concepts that belong mostly, to the core ontology like $Asset$ that could be $Staff$, $Device$, $Data$, or $Building$, etc. with their links, as for instance, a $Device$ hosts $Data$. Also, we show the concept $Threat$, $Vulnerability$, and $Protection$ that belong to the protection management module as well as, $Impact$ and $Incident$ that belong to the impact management module.

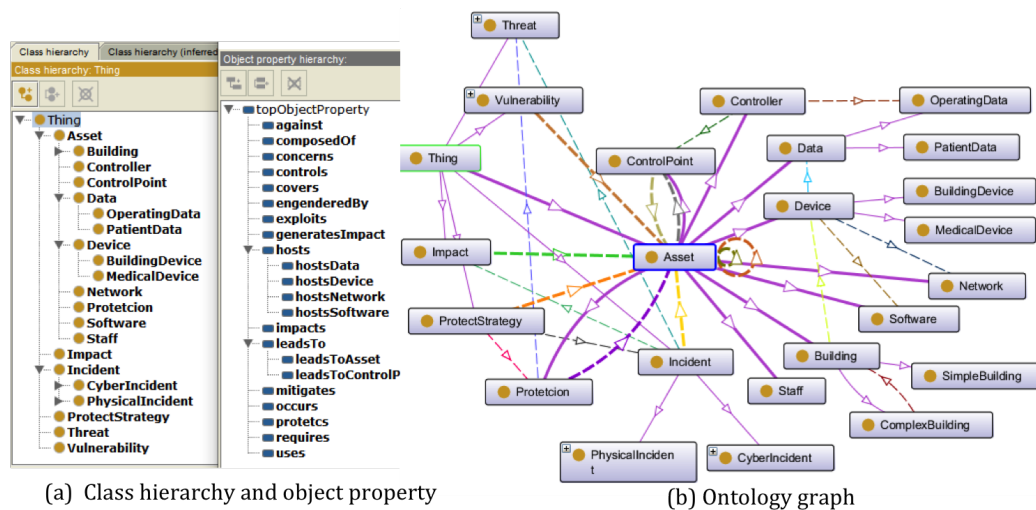


Figure 3.4: SafecareOnto implementation in Protégé

3.4.2 Monitoring Cyber-Physical System

Beyond modeling hybrid cyber-physical interactions, the ontology-based solution's strength is an inference and reasoning mechanism that derives new facts from the implicit knowledge existing in the knowledge base. This tool offers the possibility of reasoning about the incidents, protections, threats, assets connections (asset:leadsTo for example). According to the motivation example, the server's protection is ensured by OS passwords and data encryption systems (cyber protection). Besides, protections assigned to the server room (camera, fire door, the door access controller) complete the protection against physical threats. Using the SWRL (Semantic Web Rule language [73]), we implement inference rules, like the one allowing for the physical protection use transitivity:

$$R_1 : hosts(?a,?d), protects(?p,?a), against(?p,?t), physicalThreat(?t) \longrightarrow protects(?p,?d)$$

The reasoner [118] enables discovering new facts from individuals and object properties and data properties assertions. SPARQL queries enable querying the knowledge base to determine the protection state of assets. The corresponding SPARQL query and its result are depicted in Figure 3.5.

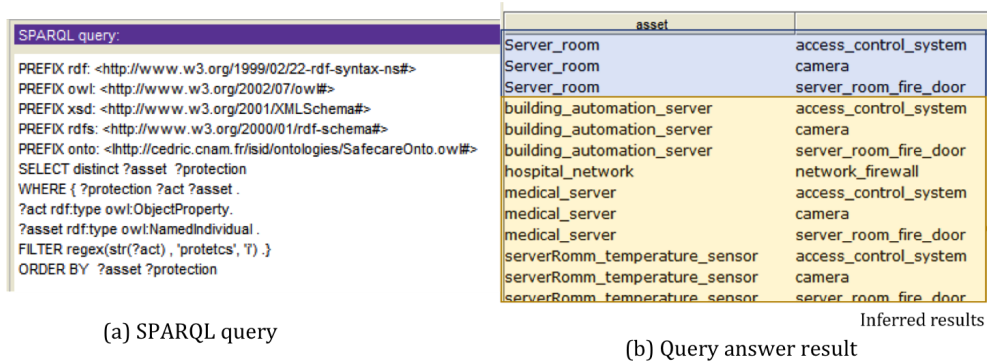


Figure 3.5: Query answer result

3.4.3 Incident Impact Propagation

An incident that occurs on an asset can lead to several impacts on other connected assets, regardless of the cyber or physical nature of incidents or assets. Inference rules determine what assets are reachable from the asset initially suffering from the incident, and can be consequently impacted. The following propagation rules illustrate examples from the attack scenario of physical intrusion:

- (a) An attacker that physically enters an office, can steal the data hosted by any office device.

$$R_2 : occurs(?i, ?a), Intrusion(?i), SimpleBuilding(?a), Device(?d) \\ hosts(?a, ?d), hosts(?d, ?data) \longrightarrow impacts(?impact, ?data), Steal(?impact)$$

- (b) Any code injection occurring on a system connected to the network, potentially generates a network flooding impact.

$$R_3 : occurs(?i, ?a), CodeInjection(?i), Software(?a), Network(?n) \\ leadsTo(?a, ?n) \longrightarrow impacts(?impact, ?n), NetworkFlooding(?impact)$$

- (c) Any code injection into the network generates a service stop incident for any device connected to the network.

$$R_4 : occurs(?i, ?n), CodeInjection(?i), Network(?n) leadsTo(?n, ?d), \\ buildingDevice(?d) \longrightarrow impacts(?impact, ?d), ServiceStop(?impact)$$

Consider that initially an intrusion incident occurs on the technical office door asset. The Figure 3.6 shows the impacts generated by the inference rules R_2, R_3 and R_4 .

Incident propagates through the LeadsTo relationship to the technical office as an intrusion incident. The technical office hosts the technical computer which facilitates the code injection by the attacker. This malicious code spreads through the network and could trigger several undesired impacts such as air cooling system dysfunction, unavailability of care services or sensitive data disclosure because of corrupt medical IS.

3.5 Conclusion

The demand for innovative strategies to safeguard healthcare cyber-physical systems is steadily increasing, necessitating an amalgamation of cyber and physical security aspects within a comprehensive solution. Healthcare systems currently lack a formal repository of knowledge to guide security managers in designing effective security solutions. For this, in this chapter, we proposed an ontology-based model that addresses

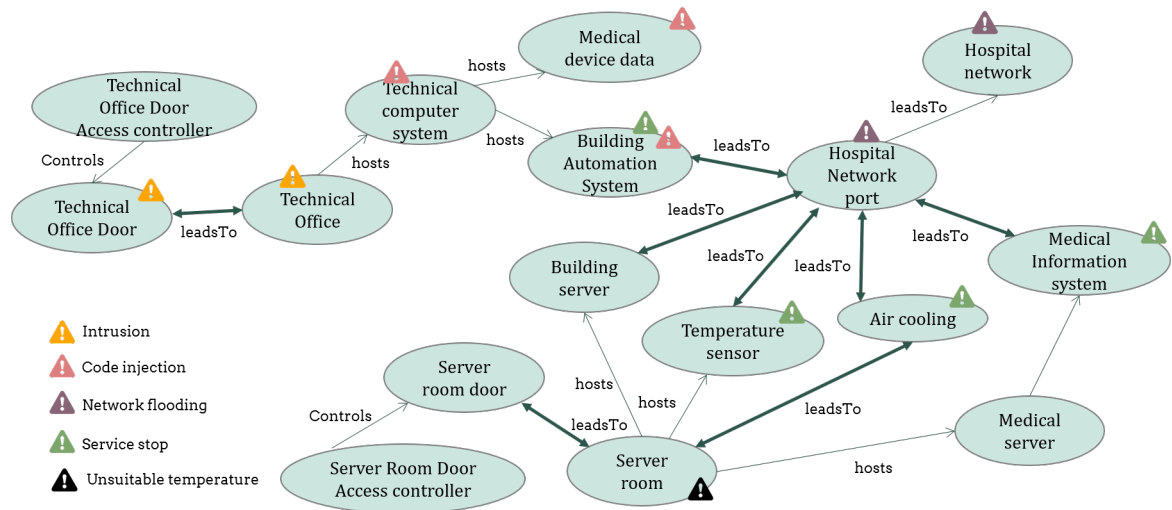


Figure 3.6: Incident's impact propagation on scenario assets

both cyber and physical security concerns in healthcare systems, providing support for incident propagation and mitigation reasoning. Our modular ontology revolves around a foundational core focused on assets and encompasses protective and impact propagation modules.

The process of acquiring knowledge through collaboration with experts has yielded a business domain understanding that facilitates the creation of a decision support system for risk mitigation. This knowledge, predominantly dynamic in nature as it evolves alongside real instances of attacks, represents the subsequent stage in the evolution of our solution. The modular framework of our solution has demonstrated its efficacy, particularly as the acquisition of domain-specific knowledge cannot be accomplished in a single instance due to the diversity and geographical distribution of stakeholders. Our next stride involves the development of the protection and impact management modules [69]. The implementation of the ontology using "Protégé" lays the groundwork for forthcoming applications, particularly the decision support system for risk mitigation.

This chapter showcases the research undertaken within the scope of M. Souibgui's PhD thesis which was successfully defended on December 14th, 2022. This thesis was co-supervised with Pr. S. Ben Yahia from LIPAH, FST, Tunisia, and Pr. S. Cherfi from CEDRIC, CNAM. The result of this work was published in [119]/2, [120]/9, [121]/6, [122]/11.

In this chapter, we contribute to the field of BI&A, intending to exploit schemaless data scattered over several document stores for decision-making. There are three main challenges that we have addressed:

- (1) Schemaless data sources that have to be extracted and transformed to fit decisions needs,
- (2) Data sources that are multiple, scattered, and may not be known in advance by decision-makers,
- (3) The lack of integration among these heterogeneous data sources as they are usually produced by independent producers.

To deal with the first two points, we introduce a new approach that aims to extract, transform, and load NoSQL data sources in an on-demand fashion. We first start with schema extraction from the sources. We focused, particularly, on document stores and addressed the impact of the schemaless nature and heterogeneity within these data. To deal with the lack of integration of data sources, our approach proposes to join sources by discovering the connecting fields. Unlike commercial tools and existing approaches that propose to perform the join on attributes manually using the key fields proposed by the user, our approach is based on an automatic algorithm that we have proposed.

First, we introduce a global BI&A architecture where our contributions are focused on the ETL process starting from document-oriented sources. Then, we propose IRIS-DS (**I**dentifiers and **R**eferences **D**IScovery in **D**ocument **S**tores), a new approach that aims to automatically discover the pairs of join keys (*identifier*, *reference*) by leveraging multiple document stores. Specifically, the distinguishing features of our contributions are as follows:

- To the best of our knowledge, there is currently no comprehensive BI&A approach specifically designed for document oriented sources, and that extracts, transforms, and loads dispersed data across several collections,
- No former work has been dedicated to join keys discovery in the context of document stores. We consider both composite and non-composite join keys:
 - In order to find candidate *identifiers*, we adapt existing features in RDB to the context of document stores and we introduce new ones,
 - To detect the candidate pairs of *identifier* and *reference*, we uptake a graph embedding technique, called *node2vec* [64], which yields significant advantages,
 - Unlike existing works, we use both syntactic and semantic similarity measures for pruning point-less candidates.

This chapter is organized as follows: Section 4.1 shows the related work. Section 4.2 presents our BI&A architecture. Section 4.3 details how to automatically detect identifiers and references in document stores. Section 4.4 presents our experiments before concluding in Section 4.5.

4.1 Related Work

In this section, we survey existing works that paid attention to how to exploit NoSQL data for decision-support and how to prepare these data to fit the analysis requirements. We identify several major streams of approaches. Firstly, in Section 4.1.1, we reviewed the approaches that dealt with NoSQL data integration problem, including ETL process over NoSQL stores, and NoSQL schema extraction. Indeed, the main challenge when dealing with schemaless data is extracting data that is dispersed across multiple NoSQL stores. This challenge leads to the question of how to identify joinable fields from several collections without integrity constraints. Thus, we, secondly, present the approaches that dealt with the problem of

joining dispersed data across NoSQL stores, joinable table discovery in tabular data (cf. Section 4.1.2), and the approaches that have treated the ontology and the schema matching issues (cf. Section 4.1.3). Furthermore, in Section 4.1.4, we review the approaches that tackled the OnLine Analytical Processing (OLAP) analysis over NoSQL stores.

4.1.1 NoSQL Data Integration

In this Section, we, firstly, scrutinize the works regarding the ETL process over NoSQL stores. Since NoSQL stores are schemaless, we, additionally, scrutinize the related works that have proposed approaches to extract the schema from these sources. Finally, we present the works addressing the problem of joining dispersed data across several NoSQL stores.

4.1.1.1 ETL over NoSQL Stores

Most of the existing works have proposed contributions to improve the performance of the ETL process. As the problem of performance is not the core of our work, we present, in this section, four representative works. In [10], the authors conducted a thorough investigation of the existing methodologies for designing, developing, and optimizing ETL workflows in a big data context. They highlight that there is a very limited support for semi-structured and unstructured data, nevertheless, the variety of data formats is expanding rapidly. Designing ETL process for big data is challenging since typical ETL operators are not adequate for handling large amounts of data. For this, the authors proposed a theoretical ETL framework that allows the developer to expand the functionality of an ETL tool with new kinds of cleaning operators, as outlier detection or de-duplication. In [98], the authors proposed a real-time ETL architecture for unstructured data. The main goal is to accelerate stream-disk joins, which almost require frequent disk access, during the transformation phase in the ETL process. Thus, they attempt to overcome the problems of disk overhead and stream data loss. In distributed disk-data, they perform stream-disk join by combining data from stream and disk data based using a common attribute that is defined by the user. In [96], the authors proposed a tool called *BigDimETL*, that extracts, transforms, and load NoSQL stores. This tool aims to minimize ETL time consuming by parallelizing the treatment of *select*, *project*, and *join* operations. The input data extracted from a document store is converted into a column-oriented model in order to apply partitioning techniques. In [79], the authors proposed a hybrid ETL approach that combines eager and lazy ETL. Eager ETL is the traditional ETL process, whereas lazy ETL processes data only when necessary. The approach aims to reduce the execution time for repetitive scientific research queries by putting away the previously integrated and loaded data into an integrated data repository.

In the literature review presented in [110], the authors underlined that some glaring issues regarding NoSQL technologies used in the decision support systems still require more effort. They draw attention to the fact that nowadays, NoSQL technologies have problems, particularly, with join operation and aggregate functions due to the schemaless nature of NoSQL stores. The users have to write their own programs to be able to perform the ETL process on a NoSQL store. Most of the present studies on ETL over NoSQL stores have focused on volume and velocity, and aims to improve the ETL process performance. However, handling variety issues are similarly important and needs more efforts.

4.1.1.2 Schema Extraction

The schema extraction aims to find a list of document fields with their types [8]. Most new data models undertake a schemaless representation which does not imply that these data are stored without a schema. Alternatively, the schema is a soft concept where instances in the same data store referring to the same concept might be stored using distinct schemas to match specific features of each instance [61]. The authors in [81] have proposed a method to extract the global schema of a collection of JSON documents using a graph representation. The nodes represent JSON fields, nested objects, or arrays while the edges reflect the JSON documents' hierarchical structure. Since NoSQL systems do not check any structural constraints, the approach reveals structural data outliers using similarity measures to capture the degree of heterogeneity

of JSON data. In [134], the authors have proposed a schema management framework to extract distinct schemas in a collection. To have one single view of collection data, they offer a new concept called *skeleton* used as a relaxed form of the schema. The approach supports queries by allowing developers to find a suitable collection to persist a new document. In the same direction, the authors in [75] have proposed a tool called *JSONDiscoverer* that aims to represent implicit structures of a given set of JSON documents as a UML class diagram. This step is followed by an advanced discovery that infers the global schema of a set of JSON documents. Baazizi et al. [24] were interested in schema inference of massive JSON datasets. The approach aims to infer the structural features of JSON data that describe JSON objects and arrays, and consider nested values and optional keys. The distinguishing feature of their approach is that it is parametric and allows the user to specify the degree of preciseness and conciseness of the inferred schema. Besides, Gallinucci et al. [61] have extended the schema extraction level of a JSON documents' collection with schema profiling techniques to capture the hidden rules explaining schema variants. These rules are represented using a decision tree as a schema profile. The proposed algorithm combines value-based and schema-based criteria to better capture the rules underlining the use of different schemas within a collection.

4.1.1.3 Joining Dispersed NoSQL Data

In [39], the authors discussed the impact of performing the join operation in document stores. They underline that some operations, which are trivial in RDBMS, may become more complex in NoSQL management systems, particularly the join operation, which is not explicitly available in NoSQL stores. The approach provides an algorithm to perform, at the application layer, an inner-join operation on two collections, which is experimented with MongoDB. However, the algorithm requires to be fueled with join keys, which are generally not indicated in document stores.

Besides, since the join is mandatory for querying tasks, we have also studied the dedicated querying approaches. In [97], the authors proposed *Squerall*, a framework that enables querying heterogeneous data on the fly without prior data transformation. *Squerall* supports MongoDB (document-oriented system) and Cassandra (column-oriented system). In addition, the framework allows the user to declare modifications for altering join keys during query time to make data joinable. Authors in [83] have proposed a data management solution allowing joins over Cassandra DB where the primary keys are considered as partition keys. They offer an implementation of query execution and optimization module to find optimal join algorithms using basic heuristics. The approach proposed in [70] inputs two sets of values from join columns and produces a predicted join relationship using an extensive table corpus. They employ statistical co-occurrence to quantify semantic correlation at the row and column levels. However, when working on a spreadsheet with several tables, it is up to the user to select two join columns from two distinct tables. In fact, they just know the tables but not the precise columns that got to be joined.

4.1.2 Data Discovery

Data discovery is the process of exploring data to automatically detect similar, unionable, or joinable attributes among heterogeneous datasets. We review the related works that have tackled these issues. We distinguish two main streams. The first one is related to primary key and foreign key discovery in RDB. The second stream of approaches has treated the problem of joinable table discovery.

4.1.2.1 Primary and Foreign Keys Discovery

Many efforts were made, particularly, for foreign keys detection in RDB [139, 41, 99, 136]. We present here, three representative works. In [41], the purpose of the proposed approach is to automatically discover a set of foreign keys connecting a given fact and dimension tables. The foreign key relationships are discovered in PowerPivot. The latter is an Excel add-in that can be used to generate pivot tables from different

datasets, perform data analysis, and build data models. The approach is based on a set of pruning rules to filter out candidates, and a scoring function based on a similarity. In [99], the proposed approach aims to profile unary and multicolumn foreign keys from incomplete data. It proposes three different algorithms according to the ways of handling the occurrences of null values: full, simple, and partial semantics that depends on whether the foreign key columns match all values in the referenced tuple. In [136], Wu et al. have proposed a framework to detect foreign keys on web tables. Due to the poor quality of web tables, which can contain noisy data, the authors underline that discovering foreign keys required human intervention. Their approach is based on two phases. They, firstly, find candidate foreign keys using the proposed algorithm. Secondly, the candidates are validated using crowdsourcing. Crowdsourcing is the process of gathering work, information, or views from a large group of people who submit their opinions via social media or mobile applications.

The previous works have proposed approaches to profile foreign keys in RDB and web tables. We note that all of them assume the presence of already known primary keys. On the other hand, quite freshly, Jiang and Naumann [77] have proposed an approach to automatically discover both primary keys and foreign keys in RDB. The approach is based on the functional dependencies that describe the characteristics of a table or relationships between tables, namely unique column combinations and inclusion dependencies. Both dependencies have been used to detect primary keys and foreign keys. A unique column combination is a set of attributes whose projection contains only the column combinations having unique and non-null values. Their work is based on the set of inclusion dependencies given as input. However, this assumption couldn't pertain to the context of document stores. Even if this previous work [77] is the closest one to our problem, we cannot apply it as it is, out of the context of RDB. Thus, it is essential to rethink the problem using alternative methods adapted to document stores' schemaless nature.

4.1.2.2 Joinable Table Discovery

In [32], the authors proposed an approach that aims to detect if attribute values coming from different sources belong to the same domain. Based on this, whether the sources are candidates to be joined or unioned to populate a target is decided. In [141], given a table and one join column, the authors aim to find joinable tables in data lakes by formulating the problem as an overlap set similarity search. Finding a joining table is based on a given join column regardless of whether it is a primary/foreign key. Additionally, their approach is based solely on values, which is unhandy in NoSQL. Indeed, they ignore numeric values since they create casual joins that are not meaningful. However, numeric values are very important to discover identifiers and references in document stores. Similarly, in [57], Fernandez et al. propose an approach that aims to find objects that are semantically related. However, to identify semantic links, their approach requires domain-specific knowledge encoded in an ontology, which is not always available.

4.1.3 Ontology and Schema Matching Approaches

A schema refers to a structure of metadata presenting a blueprint of how the data is stored and accessed by applications [22]. Schema and ontology matching is a crucial task in the schema integration process since it aims to identify semantic correspondences between metadata, thereby reducing manual treatment. In this section, we report our review findings that we broadly classified as follows: document stores matching, ontology alignment, and graph embedding.

4.1.3.1 Document Stores Matching

Schema matching aims to identify correspondences between semantically related elements of heterogeneous schemas [29]. In the literature, approaches are broadly classified into three categories: *(i)* schema-based approaches: consider only schema information; *(ii)* instance-based approaches: rely on data to retrieve relevant insights that can boost the schema matching results; and *(iii)* hybrid approaches: combine several

matching techniques. Most of these approaches rely on manual human intervention. Furthermore, few researchers have considered schema matching for document stores [31]. In [132], the author presents an empirical study on matching JSON files using existing tools. This study aims to survey and evaluate the state of the art to check whether existing data integration approaches and tools can handle the JSON format. The conducted study demonstrates that these tools do not readily bear JSON and that additional effort is needed to understand the underlying issues and develop systems that natively support JSON.

In [82], the author proposed a semi-automatic approach, called *Karma*, for the semantic schema mapping starting from heterogeneous sources, including JSON. To resolve the data format problem, *Karma* starts by converting all the data formats into a nested relational data model using existing methods. Then, the user chooses a column to transform by giving examples of data transformation for particular rows. The system learns the transformation and applies it to the rest of the data. For example, the user inputs the first name and the last name in the correct order: $\{Kerry James, Marshall\}$ instead of $\{Marshall, Kerry James\}$. Then, the system learns the transformation made to apply it to the rest of the data. To cope with various data formats, the user models each dataset as a semantic description using a domain ontology to be then represented by *Karma* in a common schema, which can be in RDF or CSV format. However, the data integration using a semantic mapping method is limited to the cultural heritage domain. Moreover, it requires an expert user to model the input datasets using a domain ontology. Similarly, authors in [31] proposed an approach to match multiple heterogeneous JSON schemas using linguistic, semantic, and instance-based techniques. This is done in the same collection of documents, where the documents' attributes are related to the same concepts.

4.1.3.2 Ontology Alignment

Ontologies encode the concepts and properties of a subject area and the relationships between them. Ontology alignment is the process of finding correspondences between concepts in ontologies [117]. It aims to identify semantic similarities between concepts. Several ontology alignment systems are proposed in the literature. The *Alignment API*, proposed in [47], aims to represent correspondences between two given ontologies. It performs the Cartesian product of possible pairs of entities related to two OWL¹ schemas. In addition, each pair of entities is associated with a similarity measure. Authors in [78] proposed *Kepler*, an ontology alignment system which addresses the key challenges related to heterogeneous ontologies in the semantic web. The system is designed to compute alignments in a multilingual context using a translator. Authors in [51] proposed a matching algorithm called eXtended mapping (*XMap*) dealing with large-scale ontology matching. The application of the *XMap* algorithm requires an RDF² ontology format. The algorithm is based on string, linguistic and structural similarity measures. On the other hand, the alignment procedure proposed in the *AML* [55] tool is essentially based on the lexical and structural aspects. Indeed, it uses four sources of background knowledge, which are limited to the medical field.

4.1.3.3 Graph Embedding

Graph embedding is a technique that converts graph nodes into a low-dimensional vector. The embedding encapsulates the graph topology by preserving the neighborhood similarity between nodes in the embedded space [37]. Recently, various graph embedding techniques have been proposed. Adopting one of these techniques depends on the problem settings and the type of the graph embedding input, i.e., homogeneous graph, heterogeneous graph, or graph with auxiliary information and knowledge.

In [84], the authors proposed a schema matching approach to generate column similarities based on graph embedding in RDB. REMA creates an undirected graph using columns and values as nodes connected with edges that reflect the input table. However, the embedding is based on instances, which are unhandy in NoSQL. In [23], the authors investigated the problem of entities matching across knowledge

¹Web Ontology Language

²Resource Description Framework

graphs. They used graph embedding to benefit from the RDF model's graph nature. Using the graph embedding for learning representations with `RDF2VEC` carried out entity matching with higher accuracy. The authors outline that resolving ambiguous entities can not be worked out by NLP techniques grounded on a text since the source entities and target entities are syntactically the same. Therefore the disambiguation process is performed through graph embedding, which explores the context of entities.

Another advantage of using embedding, which is highlighted by the *node2vec* technique, is that it detects similarities without a requirement for external knowledge, and it can be based on graphs containing heterogeneous nodes. Using a traditional matching technique does not take advantage of all accessible features, e.g., throwing away data types while processing attributes and using the names of the attributes solely.

4.1.4 OLAP Analysis over Document Stores

OLAP is widely used as a structured data analysis approach dedicated to decision-making. With the diffusion of NoSQL stores, a great effort should be devoted to provide solutions for OLAP analysis on NoSQL stores. To the best of our knowledge, few works have tried to find solutions for OLAP analysis on document stores. In [65], the authors proposed a preliminary approach that performs OLAP queries across heterogeneous data models, i.e., relational, document-oriented, and column-oriented, using a dataspace layer on top of the underlying databases. The dataspace is a set of features where each feature is a representation of a set of attributes modeling semantically the same concept. The authors assume that the *SameAs* or foreign key relationships between attributes are already known and that all keys are not composite. They provide an execution plan for a given query launched on different databases. Then, the query defined on the dataspace will be translated into a set of queries to be performed on separate DB.

Chouder et al. [44] have proposed an approach to enable OLAP on document stores in the context of self-service BI. This approach extracts the global schema of one collection of nested JSON documents and generates a multidimensional draft schema. To build the multidimensional schema, the dimensions and measures are first identified among the set of fields contained in the JSON collection. Then, the decision-maker can define his query, which is validated by mining approximate functional dependencies. Once a functional dependency is found, the multidimensional schema is refined to add multidimensional hierarchies. Finally, the validated query is translated into the native language of MongoDB. Similarly, the approach proposed in [61] enables OLAP directly on a JSON collection. But, unlike [44] where the research for functional dependencies is done on-demand, this approach looks for all the approximate functional dependencies. Given a collection, the approach consists in integrating the distinct local schemas extracted from documents to propose a global schema. Then, it provides a multidimensional view of the global schema. The identification of different hierarchies is based on approximate functional dependencies. Most stages require user interaction. These approaches focus on querying a single collection of JSON documents and do not address the problem of integrating several collections.

4.1.5 Discussion

Our goal is to extract, transform, and load data, which is scattered over several document stores, to be queried for decision-making. In our literature study, we reviewed the approaches that proposed solutions for the ETL process over NoSQL stores. The majority of prior research that has been conducted in a big data context has focused on volume and velocity. Nevertheless, addressing variety issues is equally crucial in tackling numerous real-world problems. Moreover, even though many authors have conducted schema extraction, this problem remains under-explored. They extract the set of document fields with their types. The types are identified in a naive way according to the representation of the value. For example, if values are of numeric type, and represented as a string, the real type will not be noticed. This can have major consequences in BI systems because the measures that can be analyzed are often numeric, so an important measure for the decision-maker can be missed. Besides, identifying the hidden types is extremely important in determining the identifier candidates. Hence, the major downside of these approaches is that they do not try to find the hidden type of each field while extracting the schema. It would be of special interest to infer the real types of values since it is very important in data discovery and data integration.

Moreover, most of the above approaches have focused on extracting documents schemata. The transformation phase remains briefly addressed in the literature and requires more effort. Some key questions and notions are still not discussed. In fact, we investigate the focus on the problem of joining scattered data in the context of NoSQL stores. Indeed, even if the problem of the detection of primary keys and foreign keys in RDB is a known problem and dedicated solutions have been proposed in the literature, the issue of automatic detection of identifiers and references in document stores remains a challenge to which no previous contribution has been made.

Additional studies to understand more completely the above-mentioned concerns are required. Therefore, we have also investigated other data discovery methods that seek to identify joinable or unionable tables. These works attempt to find relatedness between tables when explicit relationships are lacking. The relatedness may be for joinability or unionability. In other words, given a set of tables, they look for meaningful datasets to populate a target table. These approaches generate results in the form of similar tables or similar attributes regardless of whether there are primary/foreign keys among these features or not. Additionally, though these works are interesting, they are devoted to tabular data. As a result, it is not appropriate for schema variant data stores as document stores, where we have different levels of object nesting, null and missing values. Furthermore, these approaches are generally based solely on values which is unhandy in a NoSQL context.

On the other hand, we have reviewed works on ontology and schema matching. The formal semantics is more grounded in ontologies than in document stores since ontologies represent the meaning rather than the data. The ontology alignment process is challenging, particularly when the ontologies are retrieved from heterogeneous sources leading to inherent differences. Nearly all of the alignment techniques rely on string-based similarities, which cannot handle the vocabulary mismatch issue. Therefore, finding out the suitable similarity measures and how to viably combine them to be used in alignment solutions is still a challenge [106].

Since a large number of works call for external domain knowledge, which is almost unavailable, current researches opt for embedding techniques to capture semantic similarities without the need for external domain-specific knowledge. Embedding techniques are very interesting to explore, nevertheless, the existent works that used embedding for semantic matching are based on instances, which is impractical in NoSQL.

Finally, existing works that have dealt with the document stores OLAP analysis focus mainly on querying a single collection of JSON documents and do not address the problem of integrating several collections. Besides, as document stores are characterized by their volume and variety, it is important to provide an on-demand ETL that extracts solely the data that meet the decision-makers' requirements. On-demand approaches are worth interest and are not yet used for the ETL in the context of document stores [25, 138]. To the best of our knowledge, no prior works have proposed a BI&A approach based on the on-demand methodology that starts from schema extraction of document stores (with a multitude of varied and dispersed data over more than one collection), performs ETL operations, and reaches OLAP analysis. Besides, no former work have dedicated to join keys discovery in document stores.

4.2 Document Stores Integration Approach

This section provides an overview of our BI&A approach for integrating document stores. First, we present the overall architecture and then we delve into the schema extraction phase. Finally, we briefly touch upon the OLAP analysis part, which is outside the scope of this thesis.

This section outlines our BI&A approach for integrating document stores. Initially, we present the overarching architecture, followed by an in-depth exploration of the schema extraction phase. Lastly, we briefly mention the OLAP analysis component, which falls beyond the scope of this thesis.

4.2.1 Overview of our Approach

We introduce a new approach that aims to extract, transform, and load several document stores in an on-demand fashion and provide dedicated solutions for decision-making. It considers both data sources'

schemaless nature and analytical needs to explore several document stores efficiently. As shown in Figure 4.1, our approach operates in two main stages: (i) on-demand ETL where ETL operations are only performed on the pertinent data that meets the decision-maker requirements; and (ii) OLAP analysis of this data (this stage is out of the scope of our contributions presented in this chapter).

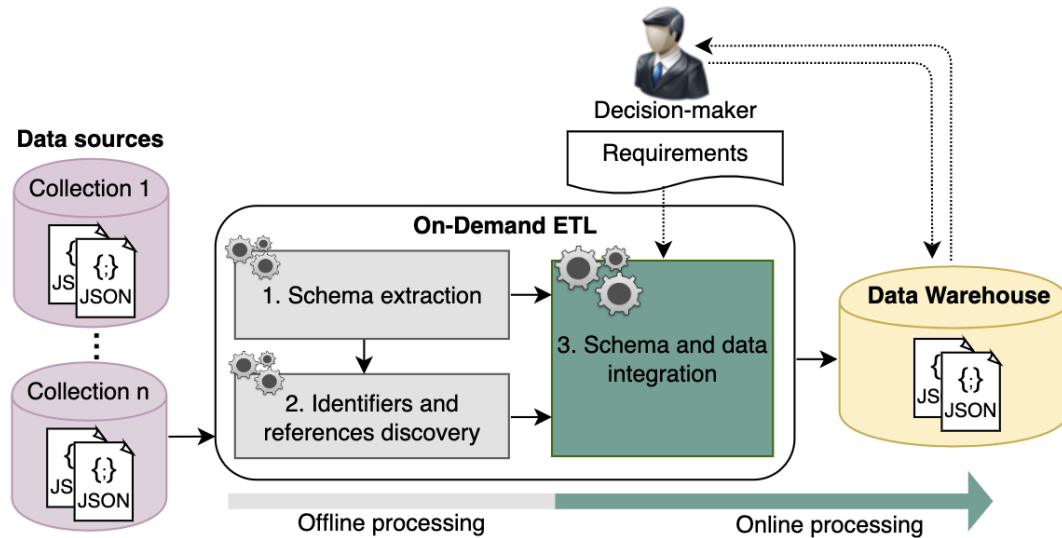


Figure 4.1: Our approach general architecture

In our work, a first objective is to consider a document-oriented data model taking into account schema variety. We consider scattered data over several collections. Since in documents, it is common to nest one object into another object [111], we regard the different cases of nesting objects:

- The document-related objects are represented separately from the original document (i.e., in another collection) and are referenced using identifiers,
- All the document-related objects are nested in the original document with different nesting depths.

Starting from dispersed data across several document stores, our approach is built upon two main stages:

- **Stage 1: On-demand ETL** aims to extract, transform and load the relevant data for each specific analysis. It operates in three phases as follows:
 - **Phase 1: Schema Extraction:** extracts the schema of each collection, in order to deal with the schemaless and variety nature of document stores. This phase is automated and processed when the user is offline.
 - **Phase 2: Identifiers and references discovery:** fetching relevant data often needs to access more than one collection of JSON documents. This multiple access requires finding the "pivot" connecting fields to perform a join operation. The latter is a complex operation to ensure, while no prior definition of join keys. While joining tables in relational data sources is assured by dint of a precise join key, in document stores, collections are unlikely to have an exact join key due to the absence of integrity constraints. So, identifying key fields that are necessary for joining two collections is an important step that we have delved into, and it is detailed in Section 4.3). This phase is automated and processed when the user is offline.
 - **Phase 3: Schema and Data Integration:** starting from several collections, this phase aims to:
 - * perform a mapping between the decision-maker requirements and collections schemas,
 - * perform ETL operations,
 - * create the DW which is a collection of JSON documents.

It is worth mentioning that this phase is processed when the user is online, since it requires interaction

with him, as depicted in Figure 4.1.

- **Stage 2: OLAP Analysis:** the aim of this stage is to ensure an OLAP analysis adapted to document stores. The main difference with respect to other approaches is that OLAP analysis can be performed on several collections which is very important in BI&A applications.

In the remainder, we present the core stage of our approach, which is divided into two phases: (i) schema extraction of document stores; and (ii) schema and data integration.

4.2.2 Schema Extraction

Document stores have a dynamic schema, mainly evident through the presence or absence of specific fields with various types. Although this schemaless nature guarantees some perks, the lack of schema information makes data processing complex and difficult. Hence, it is critical to extract the exact schema of each collection in order to perform the data integration successfully. For this, we propose extracting a flat document schema (cf. Definition 13) and the schema of each collection (cf. Definition 14) focusing on inferring real types.

Definition 13 (Document flat schema): A list of fields with their associated types. Let $S_D = \{(p, t)_i / 1 < i < k\}$ be the schema associated to a JSON document D . It consists of k pairs (p, t) , such that:

p : is the field path from the document root. It is the unique identifier of each field,

t : is the field type. Since a field can have different types from one document to another, we consider the most frequent type.

Example 7: Figure 4.2 depicts a JSON document on the left and shows its associated flat schema on the right, where $\$$ symbol represents the document root.

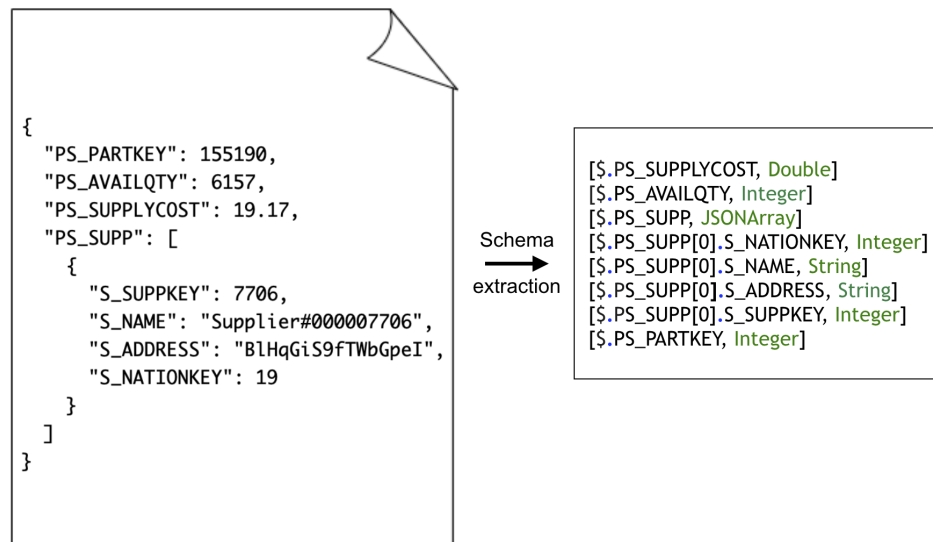


Figure 4.2: Example of document flat schema generation

Definition 14 (Collection schema): the schema of a collection C is $S(C) = \bigcup_{j=1}^l S_{D_j}$, where S_{D_j} is the flat schema associated with a JSON document D_j that belongs to the collection C and l denoting the number of distinct documents schemas that exist in a collection C .

Similarly to data, the quality of the extracted schema influences the data integration process and therefore leads to inaccurate analysis. It is important to overhaul the extracted schemas from document stores.

As mentioned in the related work, we report several methods in the literature to address the schema extraction issue. The proposed solutions generate a schema as a set of document fields with their associated types. Oddly, the types are not thoroughly described. As far as we know, no previous work has provided a

schema that yields the real hidden type for document fields. Since we can hide a real primitive type under another primitive type, we propose a new method that aims to check the type of each field to detect such cases. This method avoids misleading results generated by a wrong data type. For instance, if the values of a given field f_i are of Float type, e.g., 19.17 (cf. Example 4.2), while they are represented between quotes: "19.17", in this case, the real type must be Float instead of String.

Checking the actual type requires access to the values, making it awkward to use in NoSQL stores as the volume of data is important. We thus suggest using the random sampling technique (cf. Definition 15). The latter is an unbiased form to collect a subset of data using randomness [105]. Hence, given a sample of values of a field f_i belonging to a collection C , we check the actual type using a set of regular expressions. For instance, in order to check if the type is really String, Float, or Date, we have used these regular expressions; the above list of regular expressions can be extended and completed gradually.

- check if the type is String: `".*[a-zA-Z].*"`
- check if the type is Float: `"[-+]?[0-9]*\.[0-9]+"`
- check if the type is Date :
 - `"M/dd/yyyy"`,
 - `"dd.M.yyyy"`,
 - `"M/dd/yyyy hh:mm:ss a"`,
 - `"dd.M.yyyy hh:mm:ss a"`,
 - `"dd.MMM.yyyy"`,
 - `"dd - MMM - yyyy"`,
 - `"yyyy - dd - M"`.

Definition 15 (Simple Random Sampling): Simple random sampling is one of the most widely used category of probability sampling techniques in the statistic literature [140]. Each item in the underlying population has an equal probability of being selected in an unbiased fashion. Random numbers are generated in order to select items to constitute the random sample.

Identifying the hidden types is extremely important in determining the identifier candidates, as detailed in Section 4.3.

4.2.3 Schema and Data Integration

This phase aims to ensure the mapping between decision-maker requirements (cf. Definition 16) and collections schemas, to perform ETL operations, and to create the DW which is a collection of JSON documents.

4.2.3.1 Mapping

The goal of this step is to ensure the mapping between collections schemas and the decision-maker requirements so as to unveil the collections of interest, which meet the decision-maker requirements and will be included as input in the rest of the phases.

Definition 16 (Decision-maker requirements): decision-maker requirements are expressed as a multidimensional schema. The latter involves the subject of analysis, which is described by numerical attributes called measures (cf. Definition 18) and the axes of analysis called dimensions (cf. Definition 17) [125]. A multidimensional schema, denoted with MS , is a triple $MS = (D, M, f)$ where:

- $D = \{d_i, 1 < i < l\}$: a finite set of dimensions d_i . Each dimension is associated to a finite set of hierarchy levels $h_j(d_i)$.
- $M = \{(m, o, a)_i, 1 <= i < n\}$: the set of measures to be analysed, where:

m : label of the measure to be analysed.

o :

$$o = \begin{cases} \text{Computation formula,} \\ \emptyset, & \text{otherwise} \end{cases}$$

a : aggregation operator associated with each measure m_i (SUM, AVG, COUNT, etc.).

- f : a function that associates each measure to a finite set of grouping fields, such that $f : M \rightarrow G$ where G is a set of grouping fields.

Definition 17 (Dimension): A dimension is a qualitative value representing an axe of analysis. It is used to categorize and reveal details in data, e.g., region and date.

Definition 18 (Measure): A measure is a quantitative value that can be aggregated according dimensions, e.g., turnover.

Example 8: A decision-maker intends to analyze the order revenue measure according to product type and order year. The multidimensional schema is $MS = (D, M, f)$ where:

- $D = \{\text{product type, date} = \{\text{month, year}\}\}$,
- $M = \{m_1 = (\text{revenue, [price * quantity], SUM})\}$,
- $f(m_1) = \{\text{year, product type}\}$.

Definition 19 (Mapping): given the multidimensional schema and the set of collections schemas, a mapping is defined by the function: $\varphi : \{G, M\} \rightarrow S$
 $x \mapsto \varphi(x)$

where:

- $\{G, M\}$ is the set of multidimensional attributes, i.e., dimensions and measures,
- S is the set of collections' schemas.

This phase is semi-automatic since it requires interaction with the decision-maker. In fact, a multi-dimensional attribute can be found in more than one collection. In this case, the decision-maker has to validate manually the mapping proposed by the system.

4.2.3.2 Performing ETL Operations

ETL is a crucial part of the BI&A chain where most of the data curation is carried out. The latter is made up of a set of operations [21]. Hence, we present an extensible list of ETL operations that we have adapted to document stores. As shown in Table 4.1, we introduce a formalization for each operation and a notation assigned to each one. For each ETL operation, we provide an example of its counterpart in the Talend Big Data (TBD) tool. The latter is a free and open source tool. It offers a development environment dealing with a variety of big data sources and targets. We note that commercial ETL tools, almost provide the same components for different data sources format since they transform the data source format to a flat representation. For instance, Talend offers almost the same ETL components in different products as Talend Data Integration and Talend Big Data.

After pinpointing the collections that meet the decision-maker requirement, the first operation is to join these collections based on the detected identifiers and references. Thus, we obtain a single collection where relevant data are centralized in one document store, which represent the data warehouse after performing the rest of the ETL operations.

4.2.4 OLAP Analysis

Our objective is to establish a comprehensive BI&A architecture that spans from integrating NoSQL data to conducting OLAP analysis. In this stage, the focus is on devising the requisite mechanisms to ensure OLAP analysis of the NoSQL data integrated into the data warehouse. Given the widespread adoption of NoSQL databases for storing semi-structured data, there is an escalating demand for enabling OLAP operations on document stores. This allows even users with limited experience to acquire insights and utilize a NoSQL-based data warehouse. The decision-maker formulates OLAP queries, which are subsequently translated into the native query language of the document store. Notably, while data is stored in and extracted from document stores, the data processing remains transparent to the end user. It's important to note that the **OLAP phase falls beyond the scope of this HDR thesis.**

Chapter 4. Data Driven Approach for Document Integration in BI Systems

4.2. Document Stores Integration Approach

Table 4.1: The main ETL operations for document stores

ETL operations	Description	Notation	Examples of Talend components
Input/Wrapper	Return a collection C_2 containing key-value pairs that are constructed from a collection C_1 .	$C_2 \leftarrow I_{f_1, \dots, f_n}(C_1)$	tMongoDBInput
Project	Pass along the documents of the collection C to acquire only the given fields f_1, \dots, f_n with their values.	$\prod_{f_1, \dots, f_n}(C)$	tExtractJSON-Fields
Filter	Select a subset of documents within a collection C that match a criteria cr .	$\sigma_{cr}(C)$	tFilterRow
Cartesian product	Return a collection containing all possible combinations of the documents belonging to the collections C_1 and C_2 .	$C_1 \times C_2$	tMap*
Join	Combine data from two collections based on one or more fields. Given two collections C_1 and C_2 , we denote: - $d_i^{C_1}, d_j^{C_2}$: documents belonging to the collections C_1 and C_2 , respectively; - $A(d_i^C)$: the list of (field, value) pairs included in the document d_i^C ; - $A_key(C_1, C_2) = (a_1, a_2)$: function applied to search the key fields a_1 and a_2 that link the collections C_1 and C_2 ; - CF : the resulting collection of the C_1 and C_2 join; CF is an array of documents d_{ij} where $d_{ij} = \{A(d_i^{C_1}) + A(d_j^{C_2})\}$ such as $\{a_1.value = a_2.value\}$ where $a_1.value, a_2.value$ are the values respectively associated to a_1 and a_2 .	$C_1 \bowtie_{(a_1, a_2)} C_2$	tMap*
Aggregation	Group fields from multiple documents based on one or more document fields. We denote with: - $G = \{g_j / 1 < j \leq n\}$: a set of fields used for grouping values; - f_i : field that will be aggregated; - F_i : aggregate function.	$Agg(F_i(f_i), G)$	tAggregateRow
Set a default value	Set default value for a document field within a collection C .	$\psi_{f \leftarrow c}(C)$	Edit the schema using built-in settings
Rename	Rename a document field in a collection C .	$\rho_{f_1 \leftarrow f_2}(C)$	Edit the schema using built-in settings
Arithmetic conversion	Convert numerical fields $\{n_1, n_2, \dots\}$ by applying an arithmetic operation $O_t(n_1, n_2, \dots)$, where t is the arithmetic operation type.	$\{n_1, n_2, \dots\} \Rightarrow O_t(n_1, n_2, \dots)$	tMap*
Format conversion	Convert field value format $f_1(v)$ to another given format $f_2(v)$.	$f_1(v) \Rightarrow f_2(v)$	tMap*

* In Talend Big Data, the user has to select manually the join attributes to combine data from two collections.

4.3 IRIS-DS: an Approach for Identifiers and References DIScovery in Document Stores

[IRIS-DS: Identifiers and References DIScovery in Document Stores]

In this section, we detail our approach shown in Figure 4.3 that presents the sequential order of three stages. The primary input is the heterogeneous collections on which the IRIS-DS is processed. Collections are document stores in JSON format. The output of the IRIS-DS is a set of candidate pairs of identifiers and references. The core stages of our approach are: schemas extraction, discovery of candidate *identifiers*, identifying candidate pairs of key fields. Schema extraction stage is presented in Section 4.2.2. In this section, we detail the corner stage of our contribution i.e. Discovery of candidate identifiers, identifying candidate pairs of identifiers, and references, the IRIS-DS algorithm and a case of study that aims to demonstrate the feasibility of our approach.

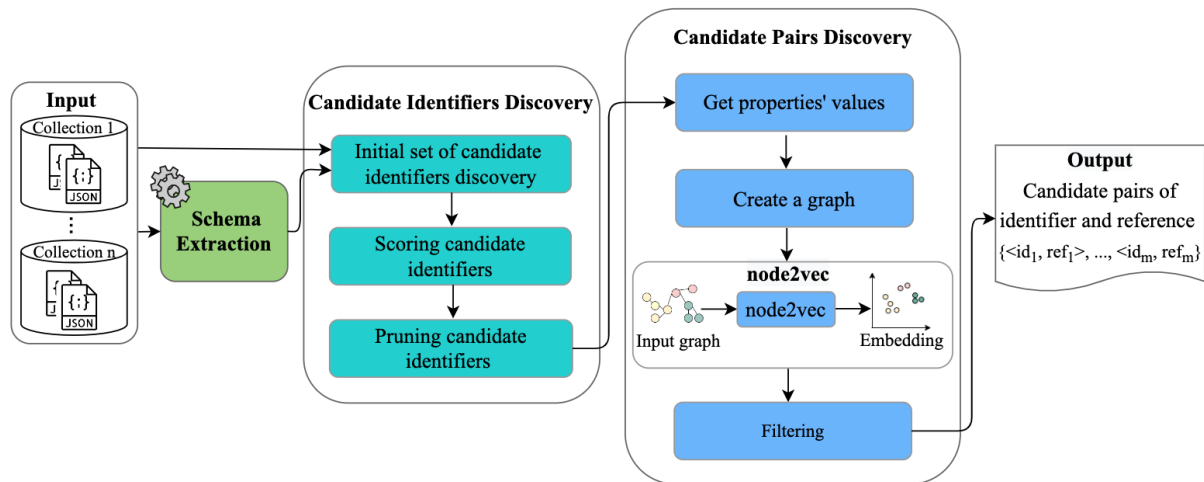


Figure 4.3: IRIS-DS general architecture

4.3.1 Discovery of Candidate Identifiers

In this stage, we restrict our focus on the discovery of candidate *identifiers* on which depends the identification of the pairs (*identifier*, *reference*) afterwards. Hence, this stage aims to start with identifying an initial list of candidate *identifiers* for each collection and come out with a refined list after the scoring and the pruning phases.

4.3.1.1 Identifying the Initial list of Candidate Identifiers

Let us consider a collection C , and its schema $S(C) = T_{cx} \cup T_s$, where T_{cx} is the set of fields with complex types (JSON object or JSON array) and T_s the ones with simple types (primitive types). Since an *identifier* can not, probably, be a *JSON object* nor a *JSON array*, then we limited the search space of candidate *identifiers* to the ones having simple types (T_s). In JSON format, each element in an array can be of three types: objects (set of key-value pairs), arrays, and/or simple values (Integer, String, Boolean, etc.). Since we are looking for candidate *identifiers*, we are interested in JSON elements represented as key-value pairs, whether nested in objects or arrays.

Moreover, due to schema flexibility, documents within the same collection may present some structural variety. Some fields are not present in all documents or may have null values. Thus, we classify fields in T_s as being required (F_r) or optional (F_o) (cf. Definition 20). We limited the search space of candidate *identifiers* to the required fields within F_r . Then, within F_r , we look for single fields and combinations of fields having unique values. We use the acronym *IDc* to refer to a candidate *identifier* (cf. Definition 21)

which can be constituted of one or more fields. We note that we regard only minimal unique fields' combinations. The generation of these combinations is done steadily. Firstly, we look for the *IDc* made up of single fields. Secondly, we generate combinations of two fields from the remaining list of non-unique fields that are frequent and of simple types. We repeat the same step for the triad combinations.

Checking unique values brings us back to evoke the problems related to duplicate detection in case of missing values. To better understand this point, consider for example two instances' values of a composite identifier: ("1", "2") and ("1", "null"). This case calls into question some past assumptions: (i) assume that the two instances' values are identical; or (ii) assume that the two instances' values are different. Multiple strategies have also been proposed to deal with missing values.

By and large, an identifier must be unique and not null [109]. However, owing to schema flexibility, any field in each collection can be missing in some documents or have null values. Thus, we apply the uniqueness checking only to required fields (cf. Definition 20). If the field is required (with a high frequency of appearance with values different from null and missing), we verify the uniqueness constraint. The latter is verified based on non-null values [30]. If so, the field is retained as a candidate *identifier*, and its score is computed in the following stage.

Definition 20: (Required Field) A field f_i is required whenever its frequency is greater than or equal to a threshold ϵ . The frequency is computed as $freq(f_i) = \frac{|\tilde{k}_c|}{|D_c|}$ [44], where $|\tilde{k}_c|$ is the number of documents in which the key in the given field is not missing and has a not null value, and $|D_c|$ stands for the total number of documents within the collection C .

Definition 21: (Candidate Identifier) Given a collection C , a candidate identifier (*IDc*) is one or more fields that are of simple types, required, and form a minimal combination of unique values.

4.3.1.2 Scoring Candidate Identifiers

In the context of relational sources, we explored several primary key features in the literature [77, 107] to distinguish valid primary keys from spurious ones. We reuse some of these features that we have adapted to the context of document stores in our proposal, and we introduce extra features: **depth**, **data type**, and **name prefix**. We describe these features in the following.

- **Cardinality:** in practice, schema designers show a tendency to use fewer fields for the identifier definition: fewer fields enable better understandability and maintainability. The score function is defined as $\frac{1}{|IDc|}$.
- **Name prefix/suffix:** *identifiers* are generally identified by their field name prefix/suffix. We consider the list of possible names' prefixes/suffixes for identifiers as: "id", "key", "nr", "no", "pk", "num", and "code". We define the score function as $\frac{prefixSuffix(IDc)}{|IDc|}$, where $prefixSuffix(IDc)$ counts the number of fields in the *IDc* whose name contains one of the prefixes/suffixes mentioned above.
- **Depth:** *identifiers* often have a shallow depth. In fact, nested fields has a lower chance to be an *identifier* for the entire collection. We define the score function as $\frac{1}{|IDc|} (\sum_{i=1}^{|IDc|} \frac{1}{depth(f_i)+1})$, where $f_i \in IDc$.
- **Data type:** hands-on hints show that a field is likely to be an *identifier* whenever its data type is Integer or String.

We define the data type score as $\frac{1}{|IDc|} (\sum_{i=1}^{|IDc|} type(f_i))$ where $type(f_i)$ is a binary function that returns one if the field f_i has a *String* or an *Integer* type or zero otherwise.

- **Value length:** fields that are used as *identifiers* are supposed to have a short value length, as they are typically non-semantic *identifiers*. The score function is defined as $\frac{1}{\overline{max(1, LengthMax(f_i)-n)}}$, where
 - $\overline{LengthMax(f_i)}$ is the average length of the longest values associated to the *IDc* fields. This function is defined as $\overline{LengthMax(f_i)} = \frac{1}{|IDc|} \sum_{i=1}^{|IDc|} LengthMax(f_i)$.

– n is a parameter used to penalize long values.

Value length is a value-based feature, making it cumbersome to use in a NoSQL context. To take advantage of this critical feature, we use the random sampling technique as reported earlier (cf. Subsection 4.2.2).

We use these features to score each candidate *identifier* related to each collection. We use the overall average of the computed scores for the total score.

4.3.1.3 Pruning Candidate *identifiers*

Expectedly, the set of the initial candidate *identifiers* is very large. Filtering techniques are essential to get rid of irrelevant candidate *identifiers*. For this, for each collection, we score each candidate *identifier* using the above-described features. We use the *Cliff* technique [77] (cf. Definition 22). As described in Example 9, the set of candidate *identifiers* is split into two parts: (i) *Upper*: it contains the candidates before the *Cliff*; and (ii) *Lower*: it contains the remaining candidates. Since the candidates that appear in the *Upper* part do have the highest scores, we prune the candidates belonging to the *Lower* part. We note that in case of multiple instances of *Cliff*, we retain all candidates before the last *Cliff* (cf. Definition 23).

Definition 22: (Cliff [77]) Given $S = \{S_1, S_2, \dots, S_n\}$, the sorted score list of candidate *identifiers* belonging to one collection, and their corresponding score difference list, $SD = \{SD_1, SD_2, \dots, SD_{n-1}\}$, where a score difference is defined as $SD_i = S_i - S_{i+1}$ of each pair of adjacent candidates, the *Cliff* is the pair of adjacent candidates S_i and S_{i+1} having the largest SD score.

Definition 23: (Multiple instances of the *Cliff*) Given $SD = \{SD_1, SD_2, \dots, SD_{n-1}\}$ the set of score differences where SD_i is the largest score difference. $\forall SD_j \in SD$, if $\exists SD_j \mid SD_j = SD_i$ such that $j \neq i$ then we prune the lower part of SD_k , where:

$$k = \begin{cases} i, & S_i < S_j \\ j, & \text{otherwise} \end{cases} \quad (4.1)$$

Example 9: As depicted in Figure 4.4, we suppose having a list S of candidate *identifiers*' scores, which are decreasingly sorted as follows: $S = \{1.0, 0.6, 0.58, 0.39, 0.23, 0.1\}$. We generate the score difference list $SD = \{0.4, 0.02, 0.19, 0.16, 0.13\}$. The *Cliff* is the largest score difference value in SD , i.e., 0.4. The green and the pink squares, shown in Figure 4.4, respectively illustrate, the *Upper* and the *Lower* parts.

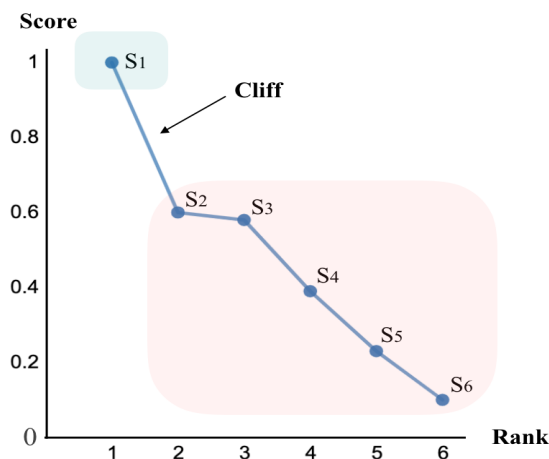


Figure 4.4: Example of the *Cliff* method applied to the ranked scores of candidate *identifiers*

The pruning phase is dedicated to refining the initial list of candidate *identifiers* for each collection. Furthermore, the refined list identifies candidate pairs of key fields as detailed in the remainder.

4.3.2 Identifying Candidate Pairs of Identifiers and References

This stage aims to constitute the pairs of *identifier* and *reference* fields related to every two document stores. Given two collections, we search the candidate pairs in both directions. We first find for each candidate *identifier* $IDc(C_1)$ of the first collection C_1 the most similar candidate *reference*, if it exists, from the set of fields $F(C_2) = f_1, \dots, f_n$ of the second collection C_2 . Then, we find for each $IDc(C_2)$ the most similar candidate *reference*, if it exists, from $F(C_1) = f_1, \dots, f_n$. We filter the obtained candidate pairs afterward.

The degree of similarity depends on the shared properties' values between an *identifier* and its *reference*. In order to inspect these similarities, we model the relation between $IDc(C_i)$ and $F(C_j) = f_1, \dots, f_n$ as a graph. Graph-based techniques are becoming ubiquitous since they are essential to yield new insights into data. However, graphs with their traditional representation do not allow entirely take benefit from the existing machine learning approaches and techniques [45].

In recent years, there has been considerable interest in graph embedding that aims to convert graph nodes into a lower-dimensional space in which the neighborhood similarity between nodes is preserved in the embedded space [37]. To this end, using embedding and vector spaces offers a richer toolset and machine learning approaches. Our objective aligns with the graph embedding goal, particularly the node embedding one.

Hence, we uptake a graph embedding technique, called *node2vec* [64]. The latter learns feature representations for the nodes across a graph for different machine learning tasks. *Node2vec* explores network neighborhood. It designs a flexible neighborhood sampling, called a random walk, by interpolating between Breadth-first and Depth-first sampling strategies.

The graph we defined is simple (do not allow multiple edges), heterogeneous (with nodes of different types), undirected and unweighted.

We denoted it as $G(V, E)$ where:

- $V = \{IDc(C_i), F(C_j), PV\}$: the set of vertices (aka nodes) with three types, where:
 - $IDc(C_i)$: is a candidate *identifier* of the collection C_i ,
 - $F(C_j)$: is the set of fields of the collection C_j ,
 - PV : is the set of properties' values related to both identifiers and references, e.g., String is a value of the data type property.
- E : is the set of edges that link and define the present relationships between nodes. An edge can be established between:
 - a node of an identifier $IDc(C_i)$ and a node of a candidate *reference* from $F(C_j)$ if this pair has a syntactic or semantic similarity greater than a threshold,
 - a field in $\{IDc(C_i), F(C_j)\}$ and a property value.

Example 10: Figure 4.5 shows an example of the input graph where the blue nodes represent the fields belonging to both collections `CountryRegion` and `Orders_customer` and the green nodes represent their properties' values. The candidate *identifier* of the collection `CountryRegion` is `CountryKey`. The colored edge linking the two nodes `CountryKey` and `NationKey` shows the existence of a syntactic or semantic relationship between the two nodes. We note that for simplicity, we use only a simple label for each node, whereas in the schema, each field is identified by its full path from the document root.

We propose a set of rules to identify the properties related to identifiers and references:

- **Rule 1: Compatibility of data type.** The identifier and its reference must have the same type or compatible types. For example, if we have an *identifier* with a *String* type and a *reference* with an *Integer* type, and they are not convertible, this pair will not be considered in this case. Our approach covers all possible combinations of primitive types, e.g., (*String*, *String*), (*String*, *Double*), (*Integer*, *Double*), (*Short*, *Double*). As mentioned earlier in the global schema extraction stage, since an actual primitive type can be hidden under another primitive type, we check the type of each field pair to detect such cases.

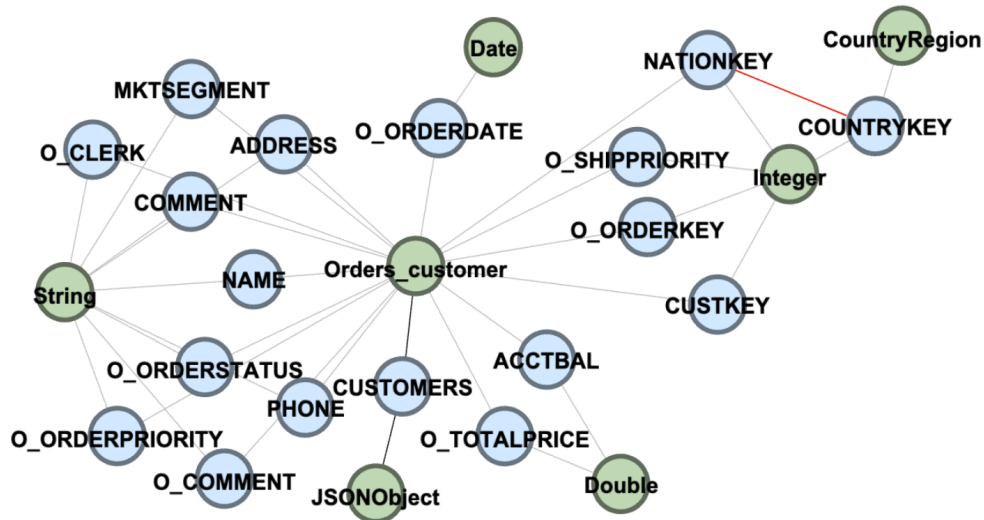


Figure 4.5: Example of the input graph for *node2vec*

- Rule 2: Syntactic similarity-based pruning.** In many instances, fields' names are not randomly assigned for better understanding. Hence, taking into account the similarity between the fields' names of each pair could be a kick-off beacon. To this end, we use a syntactic similarity measure, and we opt for the *Fuzzy-Token* similarity since it is the most suitable for our case [133]. Thus, the similarity combines both token-based similarity and string similarity. To use this similarity function, the input strings s_1 and s_2 are tokenized. We consider both cases for the tokenization: (i) having a delimiter, e.g., "_" and/or uppercase letter; (ii) strings are attached without a delimiter, e.g., "LINESTATUS". The function is defined as $syntac(s_1, s_2) = \frac{|T_1 \tilde{\cap}_\sigma T_2|}{|T_1| + |T_2| - |T_1 \tilde{\cap}_\sigma T_2|}$, where s_1 is the *reference* name and s_2 is either the *identifier* name or the collection name of that identifier. Then, we retain the maximum value obtained between the two similarity measures. We note that s_1 and s_2 are amended compared to [77]³. In addition, T_1 and T_2 are the tokens' sets related to s_1 and s_2 respectively and σ is the edit distance threshold used to penalize lower similarities. To compute this similarity, a weighted bigraph should be constructed using T_1 and T_2 . The weight, i.e., edit distance measure, is assigned to each edge. Then, we keep only the edges with a weight larger than σ . The fuzzy overlap, denoted by $|T_1 \tilde{\cap}_\sigma T_2|$, is used to define the maximum weight matching of the constructed graph. Note that if $|T_1|$ or $|T_2|$ are more significant than one, we filter them from the set of possible suffixes or prefixes such as "key" and "id." For instance, by considering the two fields "CountryKey" and "CustomerKey," the syntactic similarity measure may rise due to the common suffix "Key," which becomes misleading to get the correct result and enhance the probability of getting false-positive results. On the other hand, the case of fields "Id" and "UserId" reveals the necessity to keep T_1 and T_2 without the filtering step.
- Rule 3: Semantic similarity based pruning.** Using only the syntactic similarity between two fields is not sufficient to cover all cases, e.g., `customer` and `client`. It indeed leads to generating some false-positive and false-negative results. To this end, we propose a filtering step based on semantic similarity. To do so, we use the *Wup* semantic similarity measure (cf. Definition 24), which is based on the lexical database WordNet⁴. Like the syntactic measure, we use tokenization to divide the attached words into meaningful separated words. Given two fields f_1 and f_2 , we split each of them into a set of tokens, T_1 and T_2 respectively. We consider f_1 and f_2 semantically similar if exists at least a semantic similarity between elements of a pair (t_1, t_2) , where $t_1 \in \{T_1 \setminus SP\}$ and $t_2 \in \{T_2 \setminus SP\}$. We

³In [77], the authors have concatenated the table name for both primary key and foreign key presented in an inclusion dependency. However, it remains unclear to concatenate table names for both of them because, generally, the foreign key is likely to be similar to the name of the referenced table, but the inverse rarely happens.

⁴<https://wordnet.princeton.edu/>

denote with SP the set of possible suffixes or prefixes such as "key" and "id". Likewise the syntactic similarity, we deal differently with a unary set of tokens (T_1 or T_2) by considering all of the tokens without a filtering step.

Definition 24: (Wup similarity [137, 108]) Wup is a path-based semantic similarity. Given two concepts, it finds the path length to root from the Least Common Subsumer (LCS), the most specific concept they share as an ancestor. The Wup similarity is computed as follows: $sim_{Wup} = \frac{2 \times depth(LCS(C_1, C_2))}{depth(C_1) + depth(C_2)}$ where $depth(C)$ is the depth of the concept in the WordNet hierarchy.

Based on the rules defined above, we define the properties that concern both identifiers and references, namely: collection name, data type, IsSyntacticallySimilar, and IsSemanticallySimilar.

4.3.3 The IRIS-DS Algorithm

Starting from several collections, IRIS-DS algorithm firstly extracts the collections' schemas. Secondly, it discovers the initial set of candidate *identifiers* that will be refined after the scoring and the pruning steps. Then, it performs a graph embedding technique to track down the set of candidate pairs of *identifier* and *reference*. The pseudo-code is sketched in Algorithm 2, which in turn invokes various methods that are detailed separately.

In Algorithm 2, we start with the extraction of collections' global schemas. In line $A_2.L_5$, we search candidate *identifiers* from the set of fields presented in collections' global schemas. This step is explained separately in Algorithm 3.

In line $A_2.L_6$, the list of collections' pairs, denoted as L , is generated using the Cartesian product while keeping only pairs with different elements, i.e., each collection pair (CP) is defined as $CP = (C_i, C_j)$ where $C_i \neq C_j$. The cardinality of L is defined as $|L| = \frac{|C|(|C|-1)}{2}$, where $|C|$ is the number of distinct collections.

Algorithm 2 IRIS-DS

Input: Collections \mathbb{C}

Output: Pairs of candidate *identifier* and *reference* for each collections' pair $IRcand$

$LCP_1 \leftarrow \emptyset, LCP_2 \leftarrow \emptyset$

$SG_{\mathbb{C}} \leftarrow \text{GenerateCollectionsSchemas}(\mathbb{C})$

$\text{SearchCandidateIDs}(\mathbb{C}, SG_{\mathbb{C}})$

$L \leftarrow \text{GetListOfCollectionsPairs}()$

foreach $CP=(C_i, C_j)$ *in* L **do**

▷ $*[f]|L| = \frac{|C|(|C|-1)}{2}$

if $\text{GetID}(C_i) \neq \emptyset$ **then**

$LCP_1 \leftarrow \text{discoverPairs}(C_i, C_j)$

end

if $\text{GetID}(C_j) \neq \emptyset$ **then**

$LCP_2 \leftarrow \text{discoverPairs}(C_j, C_i)$

end

$IRcand \leftarrow \text{filter}(LCP_1, LCP_2)$

$\text{Store}(C_i, C_j, IRcand)$

▷ $*[f]\text{Store}$: store the $IRcand$ for each collection pair

end

return $IRcand$

The idea is to iterate over L to find the set of pairs of candidate join keys (*identifier, reference*), if they exist, between every two collections (lines $A_2.L_{7-16}$). We consider both directions: a field in C_i can refer to another field in C_j or vice versa. The pairs discovery assured in line $A_2.L_{12}$ and line $A_2.L_{15}$ are detailed in Algorithm 4 where the loop from line $A_4.L_3$ to line $A_4.L_{16}$ iterates over the list of *identifiers* of the current collection C_i , which are generated with $\text{GetID}(C_i)$.

In line Algorithm 4, we start by creating the input graph G for the embedding. Building the graph

Algorithm 3 SearchCandidateIDs()

Input: Collections \mathbb{C} , Collections schemas $SG_{\mathbb{C}}$

Output: Initial list of candidate *identifiers* IL

$I \leftarrow \emptyset, R \leftarrow \emptyset, U \leftarrow \emptyset, IDs \leftarrow \emptyset, L \leftarrow \emptyset$

foreach C *in* \mathbb{C} **do**

$I \leftarrow \text{GetFieldsWithSimpleTypes}(SG_{\mathbb{C}})$

$R \leftarrow \text{GetRequiredFields}(I)$

$U \leftarrow \text{GetUnique}(R)$

$S \leftarrow \text{Score}(U)$

$IDs \leftarrow \text{Cliff}(S)$

$L \leftarrow L \cup IDs$

end

return L

has a linear time complexity $O(n)$ as execution time depends on the size of the collection schema. The properties values are among the constituting nodes of the input graph. We detail the extraction of the properties values in Algorithm 5. In lines $A_4.L_{5-6}$, we apply the *node2vec* algorithm, and we generate the model after learning feature representations for the nodes across the graph. The loop in lines $A_4.L_{8-12}$, iterates over the field(s) of an *identifier* and search the most similar field(s) that will be considered as a candidate *reference*.

Algorithm 4 discoverPairs()

Input: Collection C_i , collection C_j

Output: list of candidate pairs where the *identifiers* in C_i and reference in C_j LCP

foreach IDc *in* $\text{GetID}(C_i)$ **do**

$PV \leftarrow \text{GetPV}(IDc) \cup \text{GetPV}(\text{GetFields}(SG(C_j)))$

 ▶ $*[f]PV$: properties values

$G \leftarrow \text{CreateGraphForEmbedding}(IDc, \text{GetFields}(SG(C_j)), PV)$

 ▶ $*[f]G$: input graph for node2vec

$n \leftarrow \text{node2vec}(G)$

$model \leftarrow \text{LearnEmbedding}(n)$

 ▶ $*[f]$ Learning node representations $R \leftarrow \emptyset$

foreach $part$ *in* IDc **do**

if $model.\text{GetSimilar}(part) \neq \emptyset$ **then**

$R \leftarrow R \cup model.\text{GetSimilar}(part)$

 ▶ $*[f]$ list of the most similar fields to IDc

end

end

if $|IDc| = |R|$ **then**

$LCP \leftarrow LCP \cup (IDc, R)$

 ▶ $*[f]LCP$: list of candidate pairs

end

end

return LCP

In Algorithm 5, based on the rules mentioned above, we search the properties' values related to an *identifier* of the 1st collection and the set of fields of the 2nd collection. In line $A_5.L_5$, we find the type of the field f . In line $A_5.L_7$, we check if the current field and the *identifier* have a syntactic similarity measure equal to or greater than a given threshold. If they are syntactically similar, we assign to f the name of the identifier as a property value. In this way, an edge will be established between the *identifier*'s node and the field's node. Similarly, we verify the semantic similarity between the *identifier* and the current field.

Chapter 4. Data Driven Approach for Document Integration in BI Systems

4.3. IRIS-DS: an Approach for Identifiers and References Discovery in Document Stores

Algorithm 5 GetPV()

Input: List of fields L , candidate *identifier* IDc , collections' names $CN(L)$ and $CN(IDc)$

Output: properties values PV

$PV \leftarrow \{CN(L), CN(IDc)\}$

▷ *[f]add collections' names as properties values

foreach f *in* L **do**

$t \leftarrow dataType(f)$

▷ *[f]get the data type of the field f according to Rule 1

$PV_f \leftarrow PV_f \cup t$

if $CheckSyntacticSimilarity() = true$ **then** $PV_f \leftarrow PV_f \cup IDc$ ▷ *[f]check if the field f is syntactically similar to IDc

else $PV_f \leftarrow PV_f \cup null$

if $CheckSemanticSimilarity() = true$ **then** $PV_f \leftarrow PV_f \cup IDc$

else $PV_f \leftarrow PV_f \cup null$

$PV \leftarrow PV \cup PV_f$

end

return PV

4.3.4 Case Study

Figure 4.6 shows two JSON collections, i.e., `Orders_customer` and `CountryRegion`, that are based on the TPC-H⁵ benchmark. This benchmark comprises relational sources that we have transformed into JSON collections⁶. For readability, we only present an excerpt of one document from each collection. Based on this benchmark, our basic scenario is to perform a join operation between `Orders_customer` and `CountryRegion`. In doing so, we should perform *identifiers* and *references* discovery between the two aforementioned collections.

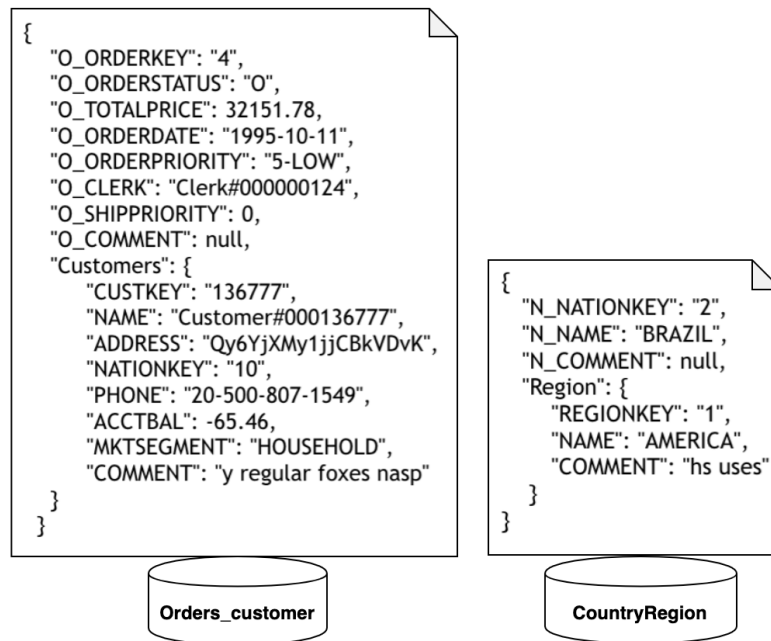


Figure 4.6: Example of JSON documents from the TPC-H benchmark

We start by preparing the graph embedding input. As shown in Figure 4.7, we create three graphs:

- G_1 : presents the relationships between the candidate *identifier* [$$.COUNTRYKEY$] with the different fields of the collection `Orders_customer`.

⁵Decision support benchmark: <http://www.tpc.org/tpch/>

⁶ <https://github.com/souibguimanel/TPCHjson>

- G_2 : presents the relationships between the first part of the composite candidate *identifier* $[\$.O_ORDERKEY, \$.customer.CUSTKEY]$ with the different fields of the collection CountryRegion.
- G_3 : presents the relationships between the second part of the composite candidate *identifier* $[\$.O_ORDERKEY, \$.customer.CUSTKEY]$ with the different fields of the collection CountryRegion.

G_2 and G_3 are related to the same candidate *identifier* $[\$.O_ORDERKEY, \$.customer.CUSTKEY]$, which is composed of two fields. The green nodes denote the candidate *identifier* or a part of the candidate *identifier* of the first collection. The blue nodes indicate the fields of the second collection that can contain the candidate *reference*. Finally, the nodes of properties' values are presented with a pink color. The red edge in G_1 marks the existence of a semantic similarity between the two fields $[\$.customer.NATIONKEY]$ and $[\$.COUNTRYKEY]$.

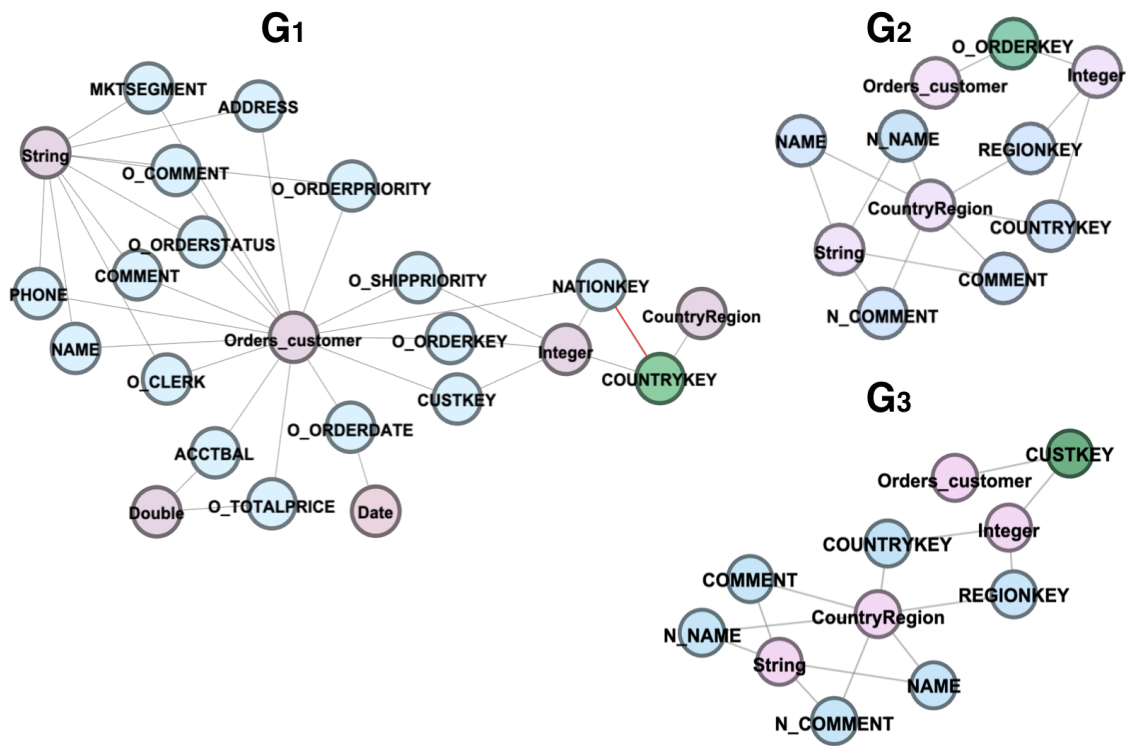


Figure 4.7: Node2vec input

To establish the relationships between nodes in each graph, we identify the values of the properties: collection name, data type, syntactic similarity measure, and semantic similarity measure. Since the two last properties require pairs of fields, we create pairs using the Cartesian product between a candidate *identifier* of the first collection and the group of fields of the second collection. Then, we compute the syntactic and semantic similarity measures for each pair. The pair with a similarity measure greater than a threshold will have an edge between their representing nodes in the graph.

4.4 Experimental Study

This section reports our experimental findings regarding discovering identifiers and references in document stores. As proof of concept of our approach, we have implemented a Java prototype to support the main phases in order to validate and evaluate the relevance of our approach. We, firstly, present the technical architecture of our prototype in Section 4.4.1. Secondly, we describe the data collection and preparation in Section 4.4.2. We base our experimental study on two benchmarks and two real-world datasets. In Section 4.4.3, we describe the evaluation protocol, and finally, we discuss the experimental results in Section 4.4.4.

Table 4.2: Initial list of candidate *identifiers* with their scores

Collection	Candidate <i>identifiers</i>	Score
CountryRegion	[\$.COUNTRYKEY]	1.00
	[\$.N_NAME]	0.63
Orders_customer	[\$.O_ORDERKEY]	1.00
	[\$.Customers.CUSTKEY]	0.90
	[\$.O_CLERK, \$.customer.NATIONKEY]	0.75
	[\$.customer.MKTSEGMENT, \$.customer.NATIONKEY]	0.70
	[\$.O_ORDERSTATUS, \$.customer.NATIONKEY, \$.O_ORDERPRIORITY]	0.70
	[\$.O_ORDERSTATUS, \$.O_CLERK]	0.70
	[\$.O_TOTALPRICE]	0.60
	[\$.O_COMMENT]	0.60
	[\$.Customers.PHONE]	0.52
	[\$.customer.NAME]	0.52
	[\$.O_CLERK, \$.O_ORDERPRIORITY]	0.52
	[\$.customer.ACCTBAL]	0.50
	[\$.O_ORDERDATE]	0.50
	[\$.customer.COMMENT]	0.50
	[\$.customer.ADDRESS]	0.50
[\$.customer.MKTSEGMENT, \$.O_CLERK]	0.49	

4.4.1 Technical Architecture of our Prototype

We have developed a prototype to support the main phases and tested them under macOS High Sierra machine, Processor Intel Core i5, 2.7 GHz, and 8 GB of DDR3 RAM. As shown in Figure 4.8, our prototype is implemented in Java 8 using the integrated development environment Eclipse JEE. We store the used collections of JSON documents on MongoDB as a document-oriented DBMS. To access the document stores, we use the MongoDB Java driver. The latter establishes a connection with the MongoDB database by creating a MongoClient. The rest of the Java code implements the core stages of our approach, namely document stores schema extraction, discovery of candidate identifiers for each collection in the MongoDB database, and creation of the graph that will be used as an input of the graph embedding algorithm. The creation of the graph requires computing syntactic and semantic similarities. Since, we use a token-based syntactic similarity, we use the Python Wordninja library⁷ to split the attached words into tokens. As for the semantic similarity, we use the WS4J API to exploit the large lexical database WordNet in Java. Finally, to discover candidate pairs of *identifiers* and *references*, we use node2vec as a graph embedding algorithm. This algorithm is implemented in Python 3 and available on GitHub. We ran this algorithm on Google Colab⁸, which provides free access to computer resources such as GPUs, making it suitable for machine learning and data analysis.

4.4.2 Data Collection and Preparation

We base our experimental study on two benchmarks and two real-world datasets:

- **TPC-H**: a benchmark for decision support systems. It represents the activity of any industry that manages, sells or distributes a product worldwide.
- **TPC-E**: a benchmark offering a set of flat files that models a brokerage firm with customers who start transactions concerning trades, account inquiries, and market research.
- **Twitter**: these datasets, which comprise users and their related tweets, are scraped from Twitter’s API. We consider two collections: **Tweets** and **Users**. Each document of the **Tweets** collection contains

⁷<https://pypi.org/project/wordninja/>

⁸<https://colab.research.google.com/>

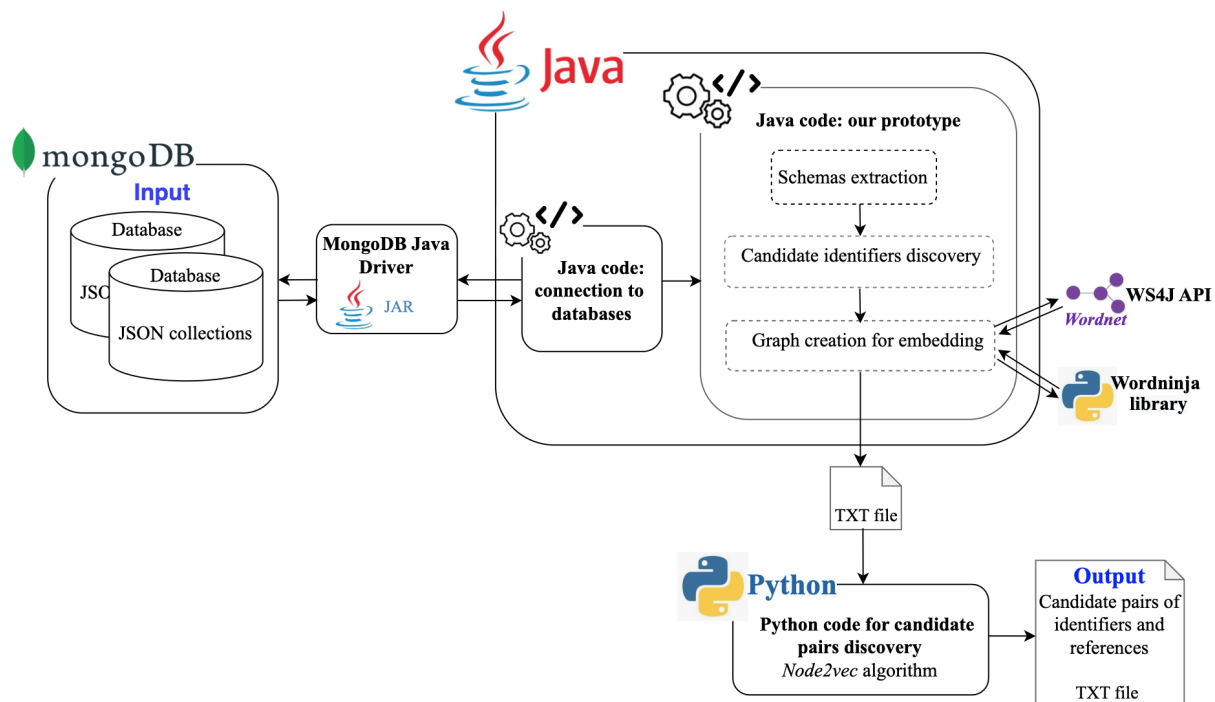


Figure 4.8: Technical architecture of our prototype

Table 4.3: Minimal and maximal depth within the considered datasets

Datasets	Minimal depth	Maximal depth
TPC-H	1	2
TPC-E	1	3
Twitter	1	3
Musicians	1	1

nested objects such as the coordinates object that gives the geographic location. These datasets are initially in JSON format. They are heterogeneous in terms of schema variety, different nesting levels (i.e., field depth), missing values, different types, etc.

- **Musicians:** are datasets extracted from Wikidata⁹, a real-world data source. These datasets are implemented in Valentine [85] where the authors proposed a well-thought-out dataset creation process that is tailored to the scope of matching techniques. The datasets represent data about American singers. It contains around 11k tabular data that we converted into JSON documents. To resemble a real-life scenario, authors in [85] have varied the names of the attributes in the source and the target sources included in Valentine. In addition, they provide with the source and the target dataset a JSON file containing the possible matching that we use as ground truth to check the accuracy of our results.

Collections within considered datasets contain fields having distinct maximal depth values. For example, as shown in Table 4.3, the maximal depth in the Twitter datasets is three, whereas, in the TPC-H datasets, the maximal depth is two.

Since our approach deals with document stores, we carried out a data preparation phase to use the two benchmarks TPC-H and TPC-E, in our experimental process as input to our prototype. We have implemented a transformation phase to convert their generated flat files to JSON ones. We consider each record

⁹ https://www.wikidata.org/wiki/Wikidata:Main_Page

in the flat file as a document in the JSON collection. We perform the data preparation stage regarding the document-oriented model features, e.g., randomly assigning null or missing values. Furthermore, to have different storage models, we have denormalized data in both benchmarks: (i) **TPC - H**, we have denormalized the **Orders** collection by embedding documents from **Customer**. Similarly, we have denormalized the **Nation** collection by embedding documents from **Region**; and (ii) **TPC-E**, we have denormalized the **Trade** collection by embedding documents from **TradeType**. Similarly, we have denormalized **CustomerAccounts** by embedding documents from both **Address** and **Customer** collections. This is done by replacing each foreign key with its full object. We host the generated data in a GitHub repositories^{10,11} to make them openly available.

4.4.3 Evaluation Protocol

The experiments we conduct aim to validate our approach in terms of result relevance. The approach validation comprises both levels: (i) candidate *identifiers* discovery for each collection; and (ii) identification of candidate pairs of key fields (*identifier* and *reference*) for every two collections. To this end, for each level, we use four metrics:

- **precision**: the fraction of the predicted true *identifier*/pairs among the predicted *identifiers*/pairs.
- **recall**: the fraction of the predicted true *identifier*/pairs among *identifiers*/pairs of the gold standard.
- **accuracy**: the number of correct results returned by our algorithm.
- **percentage decrease**: rate the reduction of the number of candidates that will be proposed to the end-user, this metric is computed as $\frac{(OriginalNumber - NewNumber)}{OriginalNumber} * 100$.

We distinguish two cases to compute the percentage decrease:

- Discovery of candidate *identifiers* for each collection:
 - * original number: the schema size of the given collection.
 - * new number: the number of the detected candidate *identifiers*.
- Identification of candidate pairs of key fields (*identifier* and *reference*) for every two collections:
 - * original number: given two collection schemas, the original number is the sum of the cardinality of the Cartesian product between the candidate *identifiers* $IDc(C_1)$ of the first collection and the set of fields of the second collection $F(C_2) = f_1, \dots, f_n$ and the cardinality of the Cartesian product between $IDc(C_2)$ and $F(C_1) = f_1, \dots, f_m$.
 - * new number: the number of the discovered pairs of identifier and reference.

4.4.4 Experimental Results

As shown in Tables 4.4 and 4.6, we compare the output sets of both candidate *identifiers* and candidate pairs (*reference*, *identifier*) with the gold standard of the Twitter collections, Musicians collections, TPC-H benchmark, and TPC-E benchmark, and we report the precision, recall, accuracy, and the percentage decrease. The results show that our approach reaches a high precision and accuracy without diminishing the recall. Furthermore, the percentage decrease metric yields increasingly excellent results by reducing the number of candidates proposed to the end user.

We note that in Table 4.4, our algorithm shows a decrease in the precision to 0.25 when detecting the identifier in the **Financial** collection. This decrease is because of several non-composite unique fields, which generate false-positive results. However, this error accounts for only a tiny portion of the used collections, and fortunately, the percentage decrease is still high.

Although our approach is dedicated to heterogeneous document stores (consisting of diverse data contents), we have also used the two Musicians datasets as part of our experimental study, which encompasses homogeneous data devoted to evaluating matching techniques.

We carry out further tests that corroborated the correlation between the collection size and the values of the evaluation metrics. To gauge this effect, we have varied the **Orders** collection size, as shown in

¹⁰ <https://github.com/souibguimanel/TPCHjson>

¹¹ <https://github.com/souibguimanel/TPCEjson>

Table 4.4: IRIS-DS results for candidate *identifiers* discovery in TPC-H, TPC-E, and Twitter collections

	Collection	# documents	Preci- sion	Re- call	Accu- racy	Percentage de- crease %
TPC-H	Orders	1k	1	1	1	94.11
	NationRegion	24	1	1	1	85.71
	Supplier	1k	1	1	1	94.73
TPC-E	Trade_TT	10k	1	1	1	94.73
	Financial	20k	0.25	1	0.73	71.42
	CustomersAc- counts	10k	1	1	1	97.14
	Holding	10k	1	1	1	83.33
	AccountPermis- sion	10k	1	1	1	80
Twitter	Tweets	1k	1	1	1	96.42
	TwitterUsers	10k	1	1	1	96.66
	Tweets	2M	1	1	1	97.61
	TwitterUsers	2M	1	1	1	97.29
Musicians	MusiciansSource	11k	1	1	1	92.30
	MusiciansTarget	11k	1	1	1	92.30

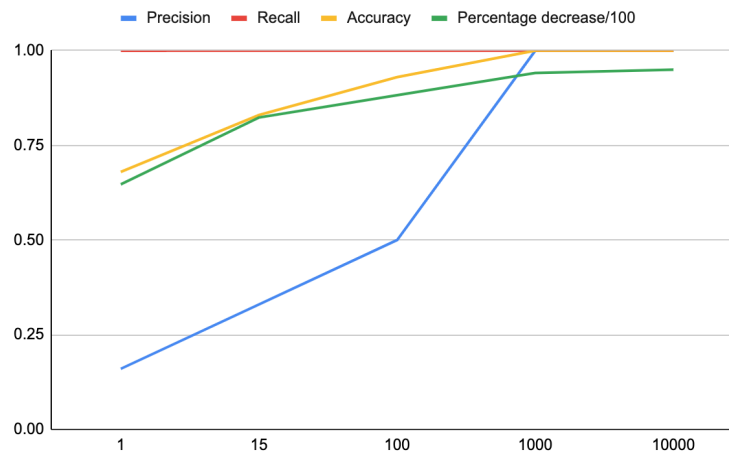


Figure 4.9: Impact of the dataset size: **Orders** collection

Figure 4.9. The result demonstrates that the more we increase the collection size, the better precision, recall, accuracy, and percentage decrease. The number of fields with unique values increases if we reduce the number of documents in the same collection. Although the precision often decreases when using a collection with few documents, the percentage decrease remains significantly high, reflecting the reduction of the final number of candidates proposed to the end user.

On the other hand, since we introduced new features to discover candidate *identifiers*, we perform additional tests to emphasize the importance of these features. For instance, the depth feature is intrinsic to document stores since collections' fields have different depths due to the embedding of objects inside documents. Thus, as shown in Table 4.5, by omitting this feature while computing the score for the **CustomersAccounts** collection, we realize the remarkable decrease in the precision, accuracy, and percentage decrease. Similarly, we remark a significant difference between their absence and presence for the data type and field name prefix features while computing scores and discovering candidate *identifiers*.

Since our approach considers several collections, we apply key pair discovery to every two collections. Indeed, there is at least one pair of collections that are not joinable, so they did not have a relationship (*reference*, *identifier*). Our approach can handle such cases. In fact, as depicted in Table 4.6, the pairs'

Table 4.5: Importance of the depth feature: CustomersAccounts collection

	Precision	Recall	Accuracy	Percentage decrease
Without depth	0.05	1	0.51	48.57
With depth	1	1	1	97.14

Table 4.6: IRIS-DS results for candidate pairs discovery in TPC-H, TPC-E, and Twitter collections

	Collection 1	Collection 2	Precision	Recall	Accuracy	Percentage decrease %
TPC-H	Orders	Nation	1	1	1	95.83
	Supplier	Orders	N/A	N/A	1	100
	CountryRegion	Supplier	1	1	1	92.85
TPC-E	Trade_TT	CustomersAccounts	1	1	1	98.14
	Holding	Trade_TT	1	1	1	96
	Financial	CustomersAccounts	N/A	N/A	1	100
	CustomersAccounts	Holding	1	1	1	97.56
	AccountPermission	CustomersAccounts	1	1	1	97.50
	Holding	AccountPermission	N/A	N/A	1	100
Twitter	Tweets	Users	1	1	1	98.50
Musicians	MusiciansSource	MusiciansTarget	1	1	1	96.15

discovery performed between the collections’ pairs (i) Supplier and Order; (ii) AccountPermission and Holding; and (iii) Financial and CustomersAccounts returns no join key pairs. This implies that the precision and recall are N/A, i.e., Not Applicable because the number of true-positive values is null. For example, this might occur where the gold standard does not contain join key fields, and our algorithm returns no pairs correctly. We note that the parameter settings (*dimensions* and *walk-length*) of the *node2vec* algorithm that we used in our tests gives a greater result when they are set to respectively 10 and 30. The results of our approach evaluation prove the viability of detecting automatically identifiers and references pairs in document stores with dispersed data across many collections.

4.5 Conclusion

In this chapter, we presented the different phases of our BI&A approach that extracts, transforms, and loads the necessary data for decision-making from document stores. We specifically focus on the ETL stage where, in contrast to existing works, we account for data dispersion across several collections. Given that document stores exhibit a variable schema and lack integrity constraints and inclusion dependencies, establishing an exact join key is a challenging task. To address this issue, we have introduced a novel approach for discovering *identifiers* and *references* based on multiple document stores. Our proposed IRIS-DS algorithm identifies candidate *identifiers* for each collection and subsequently pinpoints candidate pairs of *identifiers* and *references* across different collections. We employ scoring features and pruning rules to effectively identify relevant identifier candidates from a pool of initial ones. To establish candidate pairs among various document stores, we employ the *node2vec* graph embedding technique, which leverages syntactic and semantic similarity measures to eliminate irrelevant candidates.

To validate the efficacy of our IRIS-DS approach, we conducted experiments to assess result relevance. The validation encompasses two levels: (i) discovering candidate *identifiers* for individual collections, and (ii) identifying candidate pairs of key fields (*identifiers* and *references*) across pairs of collections. For each level, we employed four metrics: precision, recall, accuracy, and percentage decrease. The results of our approach’s evaluation shed light on the viability of detecting join key fields in document stores containing scattered data across multiple collections. The conducted experiments on the TPC-H and TPC-E benchmarks, as well as the Twitter and Musicians datasets, highlight that our approach effectively maintains the accuracy of the generated results.

5.1 Summary

To store and manage big data, IS rely on NoSQL data stores, as they are highly efficient and able of managing volume, variety, and velocity. The schemaless nature of NoSQL data stores requires careful consideration of data modeling, data quality, and data integration techniques to ensure that meaningful insights can be derived from the stored data. For this, we have focused on the modeling and development challenges of IS leveraging NoSQL data stores. We have also considered the problem of integrating NoSQL data into BI&A systems. Additionally, within the SAFECARE, H2020 project, we have worked on modeling security in healthcare cyber-physical systems. All these contributions aim, in one way or another, to support decision-making based on data and/or models. To deal with the development of IS and managing security in healthcare systems, we have proposed dedicated model-based approach, whereas the integration of NoSQL data into BI&A systems is primarily data driven.

This HDR thesis outlines my contributions in the field of IS, security in healthcare cyber-physical systems, and BI&A. Our main goal is to facilitate and enhance the processes of modeling, developing, and managing security in IS, as well as the integration of big data in BI systems. These contributions form the foundation of my research and aim to advance the fields of IS, security in healthcare systems, and BI, and can be summarized as follows.

5.1.1 Model-Based Approaches

Modeling, and developing big data information systems. We have proposed a model driven approach that formalizes, guides, and automates the development process of big data IS. The cornerstone of our approach is the common logical metamodel that describes the 5 families of models (relational and the 4 NoSQL families). We have presented QVT transformation rules that aim to automate concept-to-concept mapping, from conceptual PIM to the logical PIM and then to the different target platforms (PSM). Besides, we have proposed refinement rules, particularly for splitting and merging concepts' rows that generates all data models by recursive denormalization to find suitable solutions. Then, we have presented a heuristic that allows avoiding the explosion of solutions in terms of the amount of data models and paths to obtain them.

Compared to existing works, our approach has the advantage of considering all four NoSQL families and the relational model. It also leverages the flexibilities provided by NoSQL databases by utilizing model refinements. As a result, it offers a comprehensive and generic framework that automates model transformations while guiding the user in decision-making.

Modeling security in healthcare cyber-physical systems. As part of the H2020 SAFECARE project, and in order to tackle the issue of security in healthcare cyber-physical systems, our proposed solution revolves around an ontology-based model that encompasses both cyber and physical security aspects. This model serves the purpose of supporting reasoning related to incident propagation and mitigation in healthcare systems. Our modular ontology is built around a core ontology focusing on assets, and comprises protection and impact propagation modules. This modular structure allows for a flexible and scalable approach, as it enables the acquisition of domain knowledge in increments. Given the diverse range of stakeholders involved in the project and their geographical dispersion, the ability to incrementally add knowledge proved to be highly valuable. By adopting this ontology-based approach, we aim to enhance the understanding and management of security in healthcare cyber-physical systems. The model facilitates the analysis of incidents, their potential impact, and the corresponding measures required for effective mitigation. This approach contributes to bolstering the security posture of healthcare systems and ensuring the safety of sensitive medical data and critical infrastructure.

5.1.2 Data Driven Approach for Big Data Integration in BI Systems

Our contributions in the field of BI&A intend to exploit schemaless data scattered over several document stores for decision-making. For this, we have introduced a new approach that aims to extract, transform, and load NoSQL data sources in an on-demand fashion. We first start with schema extraction from the sources. We focused, particularly, on document stores and addressed the impact of the schemaless nature and heterogeneity within these data. Notably, unlike existing works, our approach is not limited to one collection, as we consider the dispersion of data across several collections.

To deal with the lack of integration of data sources, our approach proposes to join sources by helping to discover the connecting fields. Unlike commercial tools and existing approaches that propose to perform the join on attributes manually using the key fields proposed by the analyst, our approach is based on an automatic algorithm that we have developed. This algorithm that aims to automatically detect both *identifiers* and *references* on several document stores. The *modus operandi* of our approach underscores three core stages: (i) global schema extraction; (ii) discovery of candidate *identifiers*; and (iii) identifying candidate pairs of *identifier* and *reference* fields.

In addition to ETL process, we claim that our approach for automatically discovering identifier and references might be useful for:

- **Querying tasks:** The join is mandatory for querying tasks. Therefore, it is compulsory to have the join keys beforehand. For instance, MongoDB uses the MQL (MongoDB Query Language) to query stored data in document databases using different operators.
- **Database design:** identifying the interrelationships in legacy databases certainly requires the discovery of join keys. Even though the lack of a rigid schema characterizes NoSQL frameworks, developers would also need to overview the data and take appropriate steps and decisions during the application design process. These decisions can have a significant effect on application efficiency and readability of the code [101].

5.2 Perspectives

Many research questions and perspectives are raised by our proposals. In the following, we present those that we believe are the most promising:

- Our model driven approach that guides and facilitates the development of big data IS generates all possible models by transforming, merging or splitting concepts. This approach produces a set of data models providing choices instead of focusing on a dedicated solution which prevents any trade-off, and reduces the search space using a heuristic by considering the use case (set of queries). To assist users in selecting the most suitable models and solutions for their specific use cases, we are currently developing a cost model. This model aims to compare the generated data models and guide the decision-making process towards the optimal data model(s). The cost model encompasses three dimensions, including time, environmental factors, and financial considerations. It incorporates both query-independent costs (related to the data model itself) and query-dependent costs (involving all queries executed on a data model). Multiple parameters, such as queries, data volume, number of servers, etc. are considered to define the cost associated with each data model.
- As part of our work within the H2020 SAFECARE project that finished by the end of 2021, we have proposed an ontology-based model to manage security in healthcare cyber-physical systems. The ontology was designed to support an impact propagation model. As the project is already finished, the continuity of the work presented in this thesis focused mainly on the impact propagation. The impact propagation approach relies on propagation rules inferring the cascading effects of security incidents occurring in hospitals. An impact score module allows evaluating impacts' severity, considering implemented protection measures. Our work have provided a set of tests and demonstration sessions in partnership with 3 hospitals in Europe. During these tests, we measured effectiveness and efficiency metrics, as well as end users' satisfaction feedback. Experiments performed on real attack scenarios, show high effectiveness rates and a common agreement from end-users about the added value of the

solution to enhance risks analysis practice and increase hospitals mitigation strategies efficiency. The way the ontology and the impact propagation model have been designed makes their extension easy and enables considerably improving performance. For example, we can enrich the rules repository with well documented cyber and physical propagation patterns.

In future research, there is a need to broaden our focus by including other types of threats such as human errors and natural disasters. Additionally, exploring alternative metrics for computing impact scores beyond just protection levels, such as assessing severity based on vulnerabilities, would be valuable. To address the increasing complexity when expanding to larger hospitals with extensive knowledge graphs and rule bases, it is essential to develop technical solutions that can parallelize impact computations for scalability purposes. These solutions will enable efficient processing and analysis of impacts across the expanding datasets and ensure effective scaling of the system.

- Regarding our approach that aims to integrate scattered document-oriented sources in BI&A systems, we have the following main futur directions:
 - For the identifiers and references discovery, we aim to utilize the **DBRefs** proposed by MongoDB. When searching for references, **DBRefs** allow us to establish connections between documents by utilizing the value of the `_id` field in the first document, along with the collection name and, if necessary, the database name. This feature was introduced in the latest versions of MongoDB and proves highly beneficial in simplifying the search process for references when properly implemented within documents.
 - We aim to study the impact of the data incremental refresh [112] and how to schedule the process of identifiers and references discovery in document stores accordingly. In our approach, the discovery of identifiers and references is processed when the user is offline. If new changes are produced within one or more collections that have already been processed, the process is relaunched again in order to identify the new identifiers and references, which can remain the same. As the processing is done offline, the laboriousness of this process does not impact the user; nevertheless, we consider the incremental processing one of our future goals.
 - Our contribution in the BI&A is mainly data driven as the most significant contribution focused on identifiers and references discovery starting from data sources. However, as fetching relevant data that meet the decision-maker requirement often needs to access more than one data source, this implies facing mainly two major difficulties. The first is related to **independence between production context and usage context** of data. The consequence is an absence of complete match between the decision makers' needs and the data used. To avoid loading unnecessary data for the envisaged analyses, our approach proposes to reduce the loaded data by **mapping** the multidimensional attributes reflecting the decision makers' needs and the collections schemas. The second difficulty is related to the evolution of data sources. As it is crucial for analysis to be up-to-date regarding the data sources, it is necessary to offer an approach allowing the revision of the aligned schema **on-demand**. For this, we propose to integrate the decision-makers' requirements by incorporating a multidimensional modeling phase based on several document stores. There is an increasing need for supporting OLAP on document stores, to help non-expert users in the decision-making process. Current works focus on extracting a multidimensional schema from a single collection [44, 62], whereas in BI&A fetching relevant data, often needs to access more than one collection of JSON documents.
 - We intend to extend our approach to support additional NoSQL data models. Our aim is to provide a generic BI&A approach that considers the 4 NoSQL families and the relational model.

5.3 Future Research Projects

My research work focuses on IS and decision support. In this context, I have a particular interest in model driven development, as well as data quality management and governance. In the short and medium term, I am continuing my work on these issues, not only by providing solutions in various domains, such

Chapter 5. Conclusion and Research Perspectives

5.3. Future Research Projects

as online learning environments and sports events, but also by adopting new approaches and techniques such as machine learning algorithms and the semantic web.

Data lake and metadata repositories to develop city sustainability indicators using open data applied to sports practices. A sports event is a spatio-temporal phenomenon that structurally, economically, and socially impacts a territory (the location hosting the event), thereby generating a legacy. Studying the impact of sports events on territories and sports practices relies on the exploitation and integration of available data (open data). This can be particularly challenging when dealing with data known by their volume, variety, and velocity that surpass the capacities of traditional data storage and processing systems. In the literature, existing works aim to improve the integration of heterogeneous data and establish more open IS, such as semantic data lakes or metadata models to ensure the governance of open data.

Challenges and research questions. In the absence of a unifying framework to leverage data with semantic heterogeneities, it is essential to enhance cross-data usability and improve access to metadata necessary for adopting a critical perspective on the results. Our goal is to prepare and structure data and metadata to enable critical and comparative analyses regarding the impact of events and mega-events on urban sports practices (Sustainable Development Goals - SDG11). This topic is of great interest to local authorities, digital project stakeholders focusing on sports practices, and sponsors of major events such as the 2024 Olympics Games and other national and international sporting events (Roland Garros, La Parisienne, etc.). Therefore, our goal is to reconcile this heterogeneity and facilitate the manipulation and analysis of data with strong connectivity.

In collaboration with my colleagues from the IGN and DVRC research centers, starting from these open big data, our objective is to leverage the knowledge from previous events, such as Olympic Games, and study the various impacts. For example, we aim to analyze the impact of the event on the host city's tourism image, the development of sustainable infrastructure, gentrification, and more.

To do so, we propose to (i) study and cross-reference the available data¹, addressing issues of quality, reliability, and consistency; (ii) develop a conceptual modeling of sports heritage; (iii) align the targeted SDGs with the extracted concepts; (iv) propose a domain ontology related to the objects of sports event legacies to capitalize on the knowledge in this field. This work began in January 2023 with a doctoral thesis funded by IGN and DVRC.

Governance of massive data from educational systems. The platformization of learning presents a significant change management challenge for educational institutions due to the behavioral changes resulting from new practices and innovative approaches to communication and marketing. Intelligent technologies are being utilized through online platforms, commonly known as Learning Management Systems (LMS) like Moodle, Blackboard, or Canvas. These systems connect tutors and learners, allowing instructors to create and offer online courses, as well as provide various services such as synchronous and asynchronous communication, content creation, assessment, and tracking of learner progress. They offer cost savings and flexibility to learners, enabling them to engage in learning at their own pace, anytime and from anywhere.

Challenges and research questions. In the higher education system, platformization serves to address the many challenges universities face today, such as the massive influx of learners with diverse profiles and varied learning approaches, student dropout rates, universities' stagnant position in international rankings, and more. It provides personalized learning paths and intelligent recommendations of resources and content. In order to offer tailored services, relevant recommendations, and personalized support to learners (based on their profiles, skills, and learning histories), the governance of data in educational systems represents a major challenge. The study of the literature has highlighted the absence of an architecture specifying the components, processes, and policies for distributed management, semantic integration, and accessibility of data, as well as the definition of roles and rights of different stakeholders.

Therefore, we propose a new approach for data governance that focuses on the collection, integration, quality assessment, and management of massive data. This approach aims to support collaborative decision-making and improve the effectiveness and efficiency of the learning process, aligned with Goal 4 of the SDG for 2030. This research work have started in January 2023, as a collaborative thesis between the CNAM

¹<https://www.data.gouv.fr/fr/datasets/recensement-des-equipements-sportifs-espaces-et-sites-de-pratiques/>

and ENSI - Tunis. Additionally, in collaboration with my colleagues from ENSI Tunis, and Télécom Sud Paris, we have submitted a PHC-Utique 2024 project.

Big data schema inference. NoSQL databases offer dynamic and flexible data structures which can vary from one record to another. This flexibility enables scalability and agility in handling diverse data formats, unstructured or semi-structured data, and rapidly changing data requirements. It also supports horizontal scaling by distributing the data across multiple nodes in a cluster, allowing for high-performance processing of big data workloads. However, the absence of a fixed schema in NoSQL systems can introduce complexities in data integration and analysis. Integrating data from various sources into a unified format may require additional efforts for mapping and transforming the data. Furthermore, the lack of a predefined schema poses challenges in ensuring data quality, consistency, and interoperability across different data sources.

Challenges and research questions. Without a predefined schema, the data integration process initially requires schema extraction, which involves identifying the schema that describes the data. Subsequently, comes the schema alignment step, whose main objective is to identify correlations between schemas, which will enable data integration. Indeed, in NoSQL sources, schemas can exhibit flexibility, with the number of attributes varying across instances, or they can be completely absent, meaning that data sources contain data without explicitly specifying the attributes. In the latter scenario, inferring the schema from the data becomes crucial. However, aligning with an existing schema or schema fragments presents challenges related to vocabulary or semantic heterogeneity, which often necessitates human intervention. This task becomes particularly daunting due to the large volume, high velocity, and variety of the data.

To address these challenges, we propose leveraging the data and/or (parts of) schemas to automate the process of (i) schema detection and (ii) data integration, using machine learning algorithms. This work presents a new doctoral thesis topic that builds upon our previous research conducted within M. Souibgui's PhD thesis, which was defended in December 2022. Our aim is to generalize the approach of discovering identifiers and references and to consider the 4 NoSQL families.

- [1] Rouwaida Abdallah, Anas Motii, Nataliya Yakymets, and Agnes Lanusse. Using model driven engineering to support multi-paradigms security analysis. In *Model-Driven Engineering and Software Development - Third International Conference, MODELSWARD 2015, Angers, France, February 9-11, 2015, Revised Selected Papers*, volume 580, pages 278–292. Springer, 2015.
- [2] Fatma Abdelhedi, Amal Ait Brahim, Faten Atigui, and Gilles Zurfluh. Towards Automatic Generation of NoSQL Document-Oriented Models. In *24th International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2018)*, pages 47–53, Las Vegas, Nevada, United States, 2018.
- [3] Fatma Abdelhédi, Amal Ait Brahim, Faten Atigui, and Gilles Zurfluh. Big data and knowledge management: How to implement conceptual models in nosql systems? In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 3: KMIS, Porto - Portugal, November 9 - 11, 2016*, pages 235–240. SciTePress, 2016.
- [4] Fatma Abdelhédi, Amal Ait Brahim, Faten Atigui, and Gilles Zurfluh. Processus de transformation MDA d’un schéma conceptuel de données en un schéma logique nosql. In *Actes du XXXIVème Congrès INFORSID, Grenoble, France, May 31 - June 3, 2016*, pages 15–30, 2016.
- [5] Fatma Abdelhédi, Amal Ait Brahim, Faten Atigui, and Gilles Zurfluh. Logical unified modeling for nosql databases. In *ICEIS 2017 - Proceedings of the 19th International Conference on Enterprise Information Systems, Volume 1, Porto, Portugal, April 26-29, 2017*, pages 249–256. SciTePress, 2017.
- [6] Fatma Abdelhédi, Amal Ait Brahim, Faten Atigui, and Gilles Zurfluh. Mda-based approach for nosql databases modelling. In *Big Data Analytics and Knowledge Discovery - 19th International Conference, DaWaK 2017, Lyon, France, Proceedings*, pages 88–102, 2017.
- [7] Fatma Abdelhédi, Amal Ait Brahim, Faten Atigui, and Gilles Zurfluh. Umltonosql: Automatic transformation of conceptual schema to nosql databases. In *14th IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2017, Hammamet, Tunisia, October 30 - Nov. 3, 2017*, pages 272–279. IEEE Computer Society, 2017.
- [8] Fatma Abdelhédi, Amal Ait Brahim, Hela Rajhi, Rabah Tighilt Ferhat, and Gilles Zurfluh. Automatic extraction of a document-oriented nosql schema. In *Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS 2021, Online Streaming, April 26-28, 2021, Volume 1*, pages 192–199. SCITEPRESS, 2021.
- [9] Ioannis Agrafiotis, Jason R. C. Nurse, Michael Goldsmith, Sadie Creese, and David Upton. A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate. *J. Cybersecur.*, 4(1):tyy006, 2018.
- [10] Syed Muhammad Fawad Ali. Next-generation ETL framework to address the challenges posed by big data. In *Proceedings of the 20th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data co-located with 10th EDBT/ICDT Joint Conference (EDBT/ICDT 2018), Vienna, Austria, March 26-29, 2018*, volume 2062 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018.
- [11] Rana Alotaibi, Bogdan Cautis, Alin Deutsch, Moustafa Latrache, Ioana Manolescu, and Yifei Yang. ESTOCADA: towards scalable polystore systems. *Proc. VLDB Endow.*, 13(12):2949–2952, 2020.
- [12] MA Amutio, J Candau, and J Mañas. Magerit-version 3, methodology for information systems risk analysis and management, book i-the method. *Ministerio de administraciones públicas*, 2014.
- [13] Rupali Arora and Rinkle Rani Aggarwal. Modeling and querying data in mongodb. *International Journal of Scientific and Engineering Research*, 4(7):141–144, 2013.

- [14] Yosef Ashibani and Qusay H. Mahmoud. Cyber physical systems security: Analysis, challenges and solutions. *Comput. Secur.*, 68:81–97, 2017.
- [15] Faten Atigui. *Approche dirigée par les modèles pour l’implantation et la réduction d’entrepôts de données*. Thèse de doctorat, Université Toulouse 1 Capitole, décembre 2013. (Soutenance le 05/12/2013). URL: http://www.irit.fr/publis/SIG/2013_These_FatenAtigui.pdf.
- [16] Faten Atigui, Fayçal Hamdi, Fatma-Zohra Hannou, Nadira Lammari, Nada Mimouni, and Samira Si-Said. Managing cyber-physical incidents propagation in health services. In *Research Challenges in Information Science: RCIS 2020*, number 385. Springer, 2020.
- [17] Faten Atigui, Fayçal Hamdi, Nadira Lammari, and Samira Si-said Cherfi. Cyber-Physical Threat Intelligence for Critical Infrastructures Security. In *Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures*. Now publisher, 2020.
- [18] Faten Atigui, Fayçal Hamdi, Fatma-Zohra Hannou, Nadira Lammari, and Samira Si-Said Cherfi. Specification of the impact propagation and decision support models. In *Deliverable D6.6, SAFECARE*, 2019.
- [19] Faten Atigui, Fayçal Hamdi, Fatma-Zohra Hannou, Nadira Lammari, Mohamed Rihany, and Samira Si-Said Cherfi. Impact propagation and decision support model. In *Deliverable D6.7, SAFECARE*, 2020.
- [20] Faten Atigui, Asma Mokrani, and Nicolas Travers. Dataguide : une approche pour l’implantation de schémas nosql. In *Extraction et Gestion des Connaissances, EGC 2020, Brussels, Belgium, January 27-31, 2020*, volume E-36, pages 407–408. Éditions RNTI, 2020.
- [21] Faten Atigui, Franck Ravat, Olivier Teste, and Gilles Zurfluh. Using OCL for automatically producing multidimensional models and ETL processes. In *Data Warehousing and Knowledge Discovery - 14th International Conference, DaWaK 2012, Vienna, Austria, Proceedings*, pages 42–53, 2012.
- [22] David Aumueller, Hong Hai Do, Sabine Massmann, and Erhard Rahm. Schema and ontology matching with COMA++. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA, June 14-16, 2005*, pages 906–908. ACM, 2005.
- [23] Michael Azmy, Peng Shi, Jimmy Lin, and Ihab F. Ilyas. Matching entities across different knowledge graphs with graph embeddings. *CoRR*, abs/1903.06607, 2019.
- [24] Mohamed Amine Baazizi, Dario Colazzo, Giorgio Ghelli, and Carlo Sartiani. Parametric schema inference for massive JSON datasets. *VLDB J.*, 28(4):497–521, 2019.
- [25] Lorenzo Baldacci, Matteo Golfarelli, Simone Graziani, and Stefano Rizzi. QETL: an approach to on-demand ETL from non-owned data sources. *Data Knowl. Eng.*, 112:17–37, 2017.
- [26] Shalini Batra et al. MongoDB versus sql: a case study on electricity data. In *Emerging research in computing, information, communication and applications*, pages 297–308. Springer, 2016.
- [27] Hacène Belbachir, Yahia Djemmada, and László Németh. The deranged bell numbers. *Mathematica Slovaca*, 73(4):849–860, 2023.
- [28] Eric Temple Bell. Exponential polynomials. *Annals of Mathematics*, pages 258–277, 1934.
- [29] Zohra Bellahsene, Angela Bonifati, Fabien Duchateau, and Yannis Velegarakis. On evaluating schema matching and mapping. In *Schema Matching and Mapping*, pages 253–291. Springer, 2011.
- [30] Laure Berti-Équille, Hazar Harmouch, Felix Naumann, Noël Novelli, and Saravanan Thirumuranathan. Discovery of genuine functional dependencies from relational data with missing values. In *Actes du XXXVIIème Congrès INFORSID, Paris, France, June 11-14, 2019*, pages 287–288, 2019.
- [31] Vitor Marini Blaselbauer and Joao Marcelo Borovina Josko. Jsonglue: a hybrid matcher for json schema matching. In *Companion Proceedings*, page 77, 2020.

- [32] Alex Bogatu, Alvaro A. A. Fernandes, Norman W. Paton, and Nikolaos Konstantinou. Dataset discovery in data lakes. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*, pages 709–720, 2020.
- [33] Harold Booth and Christopher Turner. Vulnerability description ontology (vdo): a framework for characterizing vulnerabilities. Technical report, National Institute of Standards and Technology, 2016.
- [34] Amal Ait Brahim. *Approche dirigée par les modèles pour l’implantation de bases de données massives sur des SGBD NoSQL*. PhD thesis, University of Toulouse 1 Capitole, France, 2018.
- [35] Amal Ait Brahim, Fatma Abdelhédi, Gilles Zurfluh, and Faten Atigui. Modeling framework for nosql systems. In *Proceedings of the XX Iberoamerican Conference on Software Engineering, Buenos Aires, Argentina, May 22-23, 2017*, pages 57–70. Curran Associates, 2017.
- [36] Jakub Breier and Frank Schindler. Assets dependencies model in information security risk management. In *Information and Communication Technology - Second IFIP TC5/8 International Conference, ICT-EurAsia 2014, Bali, Indonesia, April 14-17, 2014. Proceedings*, volume 8407 of *Lecture Notes in Computer Science*, pages 405–412. Springer, 2014.
- [37] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. *IEEE Trans. Knowl. Data Eng.*, 30(9):1616–1637, 2018.
- [38] James L Cebula and Lisa R Young. A taxonomy of operational cyber security risks. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 2010.
- [39] Antonio Celesti, Maria Fazio, and Massimo Villari. A study on join operations in mongodb preserving collections data models for future internet applications. *Future Internet*, 11(4):83, 2019.
- [40] Artem Chebotko, Andrey Kashlev, and Shiyong Lu. A Big Data Modeling Methodology for Apache Cassandra. In *2015 IEEE ICB D*, pages 238–245. IEEE, 2015.
- [41] Zhimin Chen, Vivek R. Narasayya, and Surajit Chaudhuri. Fast foreign-key detection in microsoft SQL server powerpivot for excel. *Proc. VLDB Endow.*, 7(13):1417–1428, 2014.
- [42] Max Chevalier, Mohammed El Malki, Arlind Kopliku, Olivier Teste, and Ronan Tournier. Implementing multidimensional data warehouses into nosql. In *ICEIS 2015 - Proceedings of the 17th International Conference on Enterprise Information Systems, Volume 1, Barcelona, Spain, 27-30 April, 2015*, pages 172–183. SciTePress, 2015.
- [43] Michal Choras, Adam Flizikowski, Rafal Kozik, and Witold Holubowicz. Decision aid tool and ontology-based reasoning for critical infrastructure vulnerabilities and threats analysis. In *Critical Information Infrastructures Security, 4th International Workshop, CRITIS 2009, Bonn, Germany, September 30 - October 2, 2009. Revised Papers*, volume 6027, pages 98–110. Springer, 2009.
- [44] Mohamed Lamine Chouder, Stefano Rizzi, and Rachid Chalal. Exodus: Exploratory OLAP over document stores. *Inf. Syst.*, 79:44–57, 2019.
- [45] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. A survey on network embedding. *IEEE Trans. Knowl. Data Eng.*, 31(5):833–852, 2019.
- [46] Gwendal Daniel, Gerson Sunyé, and Jordi Cabot. UMLtoGraphDB: mapping conceptual schemas to graph databases. In *ER’16*, pages 430–444. Springer, 2016.
- [47] Jérôme David, Jérôme Euzenat, François Scharffe, and Cássia Trojahn dos Santos. The alignment API 4.0. *Semantic Web*, 2(1):3–10, 2011.
- [48] Myller Claudino de Freitas, Damires Yluska Souza, and Ana Carolina Salgado. Conceptual mappings to convert relational into nosql databases. In *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems, Volume 1, Rome, Italy, April 25-28, 2016*, pages 174–181. SciTePress, 2016.

- [49] Alfonso de la Vega, Diego García-Saiz, Carlos Blanco, Marta Zorrilla, and Pablo Sánchez. Mortadelo: Automatic generation of nosql stores from platform-independent data models. *Future Generation Computer Systems*, 105:455–474, 2020.
- [50] Khaled Dehdouh, Fadila Bentayeb, Omar Boussaid, and Nadia Kabachi. Using the column oriented nosql model for implementing big data warehouses. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, page 469. The Steering Committee of The World Congress in Computer Science, Computer ..., 2015.
- [51] Warith Eddine Djeddi, Sadok Ben Yahia, and Mohamed Tarek Khadir. Xmap: results for OAEI 2018. In *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.*, pages 210–215, 2018.
- [52] EBIOS. Ebios risk manager – the method. https://www.ssi.gouv.fr/uploads/2019/11/anssi-guide-ebios_risk_manager-en-v1.0.pdf, 2019.
- [53] ENISA. Communication network dependencies for ICS/SCADA Systems. <https://www.enisa.europa.eu/publications/ics-scada-dependencies>, 2016.
- [54] ENISA. Cyber security and resilience for Smart Hospitals. <https://www.enisa.europa.eu/publications/cyber-security-and-resilience-for-smart-hospitals>, 2016.
- [55] Daniel Faria, Catia Pesquita, Booma Sowkarthiga Balasubramani, Teemu Tervo, David Carriço, Rodrigo Garrilha, Francisco M. Couto, and Isabel F. Cruz. Results of AML participation in OAEI 2018. In *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, 2018*, pages 125–131, 2018.
- [56] Wenduo Feng, Ping Gu, Chao Zhang, and Kai Zhou. Transforming uml class diagram into cassandra data model with annotations. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 798–805. IEEE, 2015.
- [57] Raul Castro Fernandez, Essam Mansour, Abdulhakim Ali Qahtan, Ahmed K. Elmagarmid, Ihab F. Ilyas, Samuel Madden, Mourad Ouzzani, Michael Stonebraker, and Nan Tang. Seeping semantics: Linking datasets using word embeddings for data discovery. In *34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018*, pages 989–1000. IEEE Computer Society, 2018.
- [58] Mariano Fernández-López, Asunción Gómez-Pérez, and Natalia Juristo. Methontology: from ontological art towards ontological engineering. 1997.
- [59] A. Fox and E. A. Brewer. Harvest, Yield, and Scalable Tolerant Systems. In *Workshop on Hot Topics in Operating Systems*, pages 174–178. IEEE, 1999.
- [60] Fred Freitas, Stefan Schulz, and Eduardo Moraes. Survey of current terminologies and ontologies in biology and medicine. *RECIIS-Electronic Journal in Communication, Information and Innovation in Health*, 3(1):7–18, 2009.
- [61] Enrico Gallinucci, Matteo Golfarelli, and Stefano Rizzi. Schema profiling of document-oriented databases. *Inf. Syst.*, 75:13–25, 2018.
- [62] Enrico Gallinucci, Matteo Golfarelli, and Stefano Rizzi. Approximate OLAP of document-oriented databases: A variety-aware approach. *Inf. Syst.*, 85:114–130, 2019.
- [63] Felix Gessert, Wolfram Wingerath, Steffen Friedrich, and Norbert Ritter. NoSQL Database Systems: a Survey and Decision Guidance. *CSR D*, 32(3-4):353–365, 2017.
- [64] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864. ACM, 2016.

- [65] Hamdi Ben Hamadou, Enrico Gallinucci, and Matteo Golfarelli. Answering GPSJ queries in a polystore: A dataspace-based approach. In *Conceptual Modeling - 38th International Conference, ER 2019, Salvador, Brazil, November 4-7, 2019, Proceedings*, volume 11788, pages 189–203. Springer, 2019.
- [66] Shady Hamouda and Zurinahni Zainol. Document-oriented data schema for relational database migration to NoSQL. In *Innovate-Data'17*, pages 43–50. IEEE, 2017.
- [67] Mohamed Hanine, Abdesadik Bendarag, and Omar Boutkhom. Data Migration Methodology from Relational to NoSQL Databases. *Int. J. IJECE'12*, 9(12):2369–2373, 2016.
- [68] Fatma-Zohra Hannou, Faten Atigui, Nadira Lammari, and Samira Si-Said Cherfi. An ontology-based model for cyber-physicalsecurity management in healthcare context. In *Advanced Information Systems Engineering Workshops - CAiSE 2021 International Workshops*, Lecture Notes in Business Information Processing. Springer, 2021.
- [69] Fatma-Zohra Hannou, Mohamad Rihany, Nadira Lammari, Fayçal Hamdi, Nada Mimouni, Faten Atigui, Samira Si-Said Cherfi, and Philippe Tourron. Semantic-based approach for cyber-physical cascading effects within healthcare infrastructures. *IEEE Access*, 10:53398–53417, 2022.
- [70] Yeye He, Kris Ganjam, and Xu Chu. SEMA-JOIN: joining semantically-related tables using big table corpora. *PVLDB*, 8(12):1358–1369, 2015.
- [71] HITRUST. The HITRUST Threat Catalogue. <https://hitrustalliance.net/threat-catalogue/>, 2019.
- [72] Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. A practical guide to building owl ontologies using the protégé-owl plugin and co-ode tools edition 1.0. *University of Manchester*, 2004.
- [73] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosf, Mike Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21(79):1–31, 2004.
- [74] W. H. Inmon. *Building the Data Warehouse*. QED Information Sciences, Inc., Wellesley, MA, USA, 1992.
- [75] Javier Luis Cánovas Izquierdo and Jordi Cabot. Jsondiscoverer: Visualizing the schema lurking behind JSON documents. *Knowl.-Based Syst.*, 103:52–55, 2016.
- [76] Nishtha Jatana, Sahil Puri, Mehak Ahuja, Ishita Kathuria, and Dishant Gosain. A survey and comparison of relational and non-relational database. *Int. J. IJERT*, 1(6):1–5, 2012.
- [77] Lan Jiang and Felix Naumann. Holistic primary key and foreign key detection. *J. Intell. Inf. Syst.*, 54(3):439–461, 2020.
- [78] Marouen Kachroudi, Gayo Diallo, and Sadok Ben Yahia. KEPLER at OAEI 2018. In *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018*, volume 2288, pages 173–178. CEUR-WS.org, 2018.
- [79] Pradeeban Kathiravelu, Ashish Sharma, Helena Galhardas, Peter Van Roy, and Luís Veiga. On-demand big data integration: A hybrid ETL approach for reproducible scientific research. *CoRR*, abs/1804.08985, 2018.
- [80] Anneke G. Kleppe, Jos Warmer, and Wim Bast. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Longman Publishing Co., USA, 2003.
- [81] Meike Klettke, Uta Störl, and Stefanie Scherzinger. Schema extraction and structural outlier detection for json-based nosql data stores. In *Datenbanksysteme für Business, Technologie und Web (BTW), 16. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), 4.-6.3.2015 in Hamburg, Germany. Proceedings*, volume P-241, pages 425–444. GI, 2015.

- [82] Craig A. Knoblock and Pedro A. Szekely. Exploiting semantics for big data integration. *AI Mag.*, 36(1):25–38, 2015.
- [83] Haridimos Kondylakis, Antonis Fountouris, Apostolos Planas, Georgia Troullinou, and Dimitris Plexousakis. Enabling joins over cassandra nosql databases. In *Big Data Innovations and Applications - 5th International Conference, Innovate-Data 2019, Istanbul, Turkey, August 26-28, 2019, Proceedings*, volume 1054, pages 3–17. Springer, 2019.
- [84] Christos Koutras, Marios Fragkoulis, Asterios Katsifodimos, and Christoph Lofi. REMA: graph embeddings-based relational schema matching. In *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference, Copenhagen, Denmark, March 30, 2020*, volume 2578. CEUR-WS.org, 2020.
- [85] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. Valentine: Evaluating matching techniques for dataset discovery. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*, pages 468–479, 2021.
- [86] Rakesh Kumar, Shilpi Charu, and Somya Bansal. Effective way to handling big data problems using nosql database (mongodb). *J. Adv. Database Manag. Syst.*, 2(2):42–48, 2015.
- [87] Chao-Hsien Lee and Yu-Lin Zheng. Automatic sql-to-nosql schema transformation over the mysql and hbase databases. In *2015 IEEE International Conference on Consumer Electronics-Taiwan*, pages 426–427. IEEE, 2015.
- [88] Mark Lewis, Gary Kochenberger, and Bahram Alidaee. A new modeling and solution approach for the set-partitioning problem. *COR*, 35(3):807–813, 2008.
- [89] Chongxin Li. Transforming relational database into HBase: A case study. In *ICSESS'10*, pages 683–687. IEEE, 2010.
- [90] Yan Li, Ping Gu, and Chao Zhang. Transforming UML class diagrams into HBase based on meta-model. In *ISEEE'14*, volume 2, pages 720–724. IEEE, 2014.
- [91] Chunlin Liu, Chong-Kuan Tan, Yea-Saen Fang, and Tat-Seng Lok. The security risk assessment methodology. *Procedia Engineering*, 43:600–609, 2012.
- [92] Meng Ma, Ling Liu, Yangxin Lin, Disheng Pan, and Ping Wang. Event description and detection in cyber-physical systems: An ontology-based language and approach. In *2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1–8. IEEE Computer Society, 2017.
- [93] Jihane Mali, Shohreh Ahvar, Faten Atigui, Ahmed Azough, and Nicolas Travers. A global model-driven denormalization approach for schema migration. In *Research Challenges in Information Science - 16th International Conference, RCIS 2022, Barcelona, Spain, May 17-20, 2022, Proceedings*, volume 446, pages 529–545. Springer, 2022.
- [94] Jihane Mali, Faten Atigui, Ahmed Azough, and Nicolas Travers. How to Implement NoSQL Schemas with ModelDrivenGuide? In *Conférence sur la Gestion de Données – Principes, Technologies et Applications (BDA'20)*, number 4771, page <https://easychair.org/publications/preprint/mVv7>, Paris, France, 2020. URL: <https://hal.archives-ouvertes.fr/hal-03020532>.
- [95] Jihane Mali, Faten Atigui, Ahmed Azough, and Nicolas Travers. ModelDrivenGuide: An Approach for Implementing NoSQL Schemas. In *Database and Expert Systems Applications - 31st International Conference, DEXA, Bratislava, Slovakia, Proceedings, Part I*, pages 141–151, 2020.
- [96] Hana Mallek, Faiza Ghozzi, and Faïez Gargouri. Towards extract-transform-load operations in a big data context. *Int. J. Sociotechnology Knowl. Dev.*, 12(2):77–95, 2020.
- [97] Mohamed Nadjib Mami, Damien Graux, Simon Scerri, Hajira Jabeen, Sören Auer, and Jens Lehmann. Squerall: Virtual ontology-based access to heterogeneous and large data sources. In *The Semantic Web - ISWC 2019 - 18th International Semantic Web Conference, Auckland, New Zealand, October 26-30, 2019, Proceedings, Part II*, volume 11779, pages 229–245. Springer, 2019.

- [98] Erum Mehmood and Tayyaba Anees. Distributed real-time etl architecture for unstructured big data. *Knowledge and Information Systems*, pages 1–27, 2022.
- [99] Mozhgan Memari, Sebastian Link, and Gillian Dobbie. SQL data profiling of foreign keys. In *Conceptual Modeling - 34th International Conference, ER 2015, Stockholm, Sweden, October 19-22, 2015, Proceedings*, volume 9381, pages 229–243. Springer, 2015.
- [100] William B Miller. *Classifying and cataloging cyber-security incidents within cyber-physical systems*. Brigham Young University-Provo, 2014.
- [101] Michael J. Mior and Kenneth Salem. Renormalization of nosql database schemas. In *Conceptual Modeling - 37th International Conference, ER 2018, Xi'an, China, October 22-25, 2018, Proceedings*, pages 479–487, 2018.
- [102] ABM Moniruzzaman and Syed Akhter Hossain. NoSQL Database: New era of databases for Big Data analytics-classification, characteristics and comparison. *arXiv preprint arXiv:1307.0191*, 2013.
- [103] Steve Ataky Tsham Mpinda, Luís Gustavo Maschietto, and Patrick Andjasubu Bungama. From relational database to columnoriented nosql database: Migration process. *International Journal of Engineering Research & Technology (IJERT)*, 4:399–403, 2015.
- [104] Mark A. Musen. The protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12, 2015.
- [105] Trong Duc Nguyen, Ming-Hung Shih, Sai Sree Parvathaneni, Bojian Xu, Divesh Srivastava, and Srikanta Tirthapura. Random sampling for group-by queries. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020*, pages 541–552. IEEE, 2020.
- [106] Ikechukwu Nkisi-Orji, Nirmalie Wiratunga, Stewart Massie, Kit-Ying Hui, and Rachel Heaven. Ontology alignment based on word embedding and random forest classification. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I*, volume 11051, pages 557–572. Springer, 2018.
- [107] Thorsten Papenbrock and Felix Naumann. Data-driven schema normalization. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21-24, 2017*, pages 342–353. OpenProceedings.org, 2017.
- [108] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. Wordnet: : Similarity - measuring the relatedness of concepts. In *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, pages 1024–1025. AAAI Press / The MIT Press, 2004.
- [109] David Pejcoch. Critical evaluation of validation rules automated extraction from data. *Journal of Systems Integration*, 5:32–46, 2014.
- [110] Lucija Petricoli, Luka Humski, and Boris Vrdoljak. The challenges of nosql data warehousing. In *International conference on E-business technologies (EBT)*, 2021.
- [111] Jaroslav Pokorný. JSON functionally. In *Advances in Databases and Information Systems - 24th European Conference, ADBIS 2020, Lyon, France, August 25-27, 2020, Proceedings*, volume 12245, pages 139–153. Springer, 2020.
- [112] Weiping Qu and Stefan Deßloch. Incremental ETL pipeline scheduling for near real-time data warehouses. In *Datenbanksysteme für Business, Technologie und Web (BTW 2017), 17. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS), 6.-10. März 2017, Stuttgart, Germany, Proceedings*, volume P-265, pages 299–308. GI, 2017.
- [113] Leonardo Rocha, Fernando Vale, Elder Cirilo, Dárlinton Barbosa, and Fernando Mourão. A framework for migrating relational datasets to nosql. *Procedia Computer Science*, 51:2593–2602, 2015.
- [114] Tiago Prince Sales, Fernanda Baião, Giancarlo Guizzardi, João Paulo A Almeida, Nicola Guarino, and John Mylopoulos. The common ontology of value and risk. In *International Conference on Conceptual Modeling*, pages 121–135. Springer, 2018.

- [115] Lucas C. Scabora, Jaqueline Joice Brito, Ricardo Rodrigues Ciferri, and Cristina Dutra de Aguiar Ciferri. Physical data warehouse design on nosql databases - OLAP query processing over hbase. In *ICEIS 2016 - Proceedings of the 18th International Conference on Enterprise Information Systems, Volume 1, Rome, Italy, April 25-28, 2016*, pages 111–118. SciTePress, 2016.
- [116] Adam Shostack. *Threat modeling: Designing for security*. John Wiley & Sons, 2014.
- [117] Pavel Shvaiko and Jérôme Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176, 2013.
- [118] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2):51–53, 2007.
- [119] Manel Souibgui, Faten Atigui, Sadok Ben Yahia, and Samira Si-Said Cherfi. An embedding driven approach to automatically detect identifiers and references in document stores. *Data Knowledge Engineering*, 139:102003, 2022.
- [120] Manel Souibgui, Faten Atigui, Sadok Ben Yahia, and Samira Si-Said Cherfi. Business intelligence and analytics: On-demand ETL over document stores. In *Research Challenges in Information Science - 14th International Conference, RCIS 2020, Limassol, Cyprus, September 23-25, 2020, Proceedings*, volume 385 of *Lecture Notes in Business Information Processing*, pages 556–561. Springer, 2020.
- [121] Manel Souibgui, Faten Atigui, Sadok Ben Yahia, and Samira Si-Said Cherfi. IRIS-DS: A new approach for identifiers and references discovery in document stores. In *54th Hawaii International Conference on System Sciences, HICSS 2021, Kauai, Hawaii, USA, January 5, 2021*, pages 1–10, 2021.
- [122] Manel Souibgui, Faten Atigui, Saloua Zammali, Samira Si-Said Cherfi, and Sadok Ben Yahia. Data quality in ETL process: A preliminary study. In *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23rd International Conference KES-2019, Budapest, Hungary, 4-6 September 2019*, volume 159, pages 676–687. Elsevier, 2019.
- [123] Steffen Staab and Rudi Studer. *Handbook on ontologies*. Springer Science & Business Media, 2010.
- [124] Zareen Syed, Ankur Padia, Tim Finin, M. Lisa Mathews, and Anupam Joshi. UCO: A unified cybersecurity ontology. In *Artificial Intelligence for Cyber Security, Papers from the 2016 AAAI Workshop, Phoenix, Arizona, USA, February 12, 2016*, volume WS-16-03. AAAI Press, 2016.
- [125] M Thenmozhi and K Vivekanandan. A framework to derive multidimensional schema for data warehouse using ontology. In *Proceedings of National Conference on Internet and WebService Computing, NCIWSC*, 2012.
- [126] Marianthi Theocharidou and Georgios Giannopoulos. Risk assessment methodologies for ci protection. part ii: A new approach. *tech. report EUR27332 EN*, 2015.
- [127] Xin Tong and Xiaofang Ban. A hierarchical information system risk evaluation method based on asset dependence chain. *International Journal of Security and Its Applications*, 8(6):81–88, 2014.
- [128] Paolo Trucco, Boris Petrenj, Sara Bouchon, and Carmelo Di Mauro. Ontology-based approach to disruption scenario generation for critical infrastructure systems. *International Journal of Critical Infrastructures*, 12(3):248–272, 2016.
- [129] Tamás Vajk, Péter Fehér, Krisztián Fekete, and Hassan Charaf. Denormalizing data into schema-free databases. In *CogInfoCom'13*, pages 747–752. IEEE, 2013.
- [130] Panos Vassiliadis. A survey of extract-transform-load technology. *International Journal of Data Warehousing and Mining IJDWM*, 5(3):1–27, 2009.
- [131] Jan vom Brocke, Alessio Maria Braccini, Christian Sonnenberg, and Paolo Spagnoletti. Living it infrastructures—an ontology-based approach to aligning it infrastructure capacity and business needs. *International Journal of Accounting Information Systems*, 15(3):246–274, 2014.
- [132] Kunal Waghray. JSON schema matching: Empirical observations. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, pages 2887–2889. ACM, 2020.

Bibliography

Bibliography

- [133] Jiannan Wang, Guoliang Li, and Jianhua Feng. Fast-join: An efficient method for fuzzy token matching based string similarity join. In *Proceedings of the 27th International Conference on Data Engineering, ICDE, Hannover, Germany*, pages 458–469, 2011.
- [134] Lanjun Wang, Oktie Hassanzadeh, Shuo Zhang, Juwei Shi, Limei Jiao, Jia Zou, and Chen Wang. Schema management for document stores. *PVLDB*, 8(9):922–933, 2015.
- [135] Hugh J. Watson. Business intelligence: Past, present and future. In *Proceedings of the 15th Americas Conference on Information Systems AMCIS*, page 153, San Francisco, California, USA, 2009.
- [136] Xiaoyu Wu, Ning Wang, and Huaxi Liu. Discovering foreign keys on web tables with the crowd. *Comput. Informatics*, 38(3):621–646, 2019.
- [137] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, ACL '94*, page 133–138, USA, 1994. Association for Computational Linguistics.
- [138] Ying, Niccolò Meneghetti, Ronny Fehling, Zhen Hua Liu, and Oliver Kennedy. Lenses: An on-demand approach to ETL. *PVLDB*, 8(12):1578–1589, 2015.
- [139] Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M. Procopiuc, and Divesh Srivastava. On multi-column foreign key discovery. *Proc. VLDB Endow.*, 3(1):805–814, 2010.
- [140] Zhuoyue Zhao, Robert Christensen, Feifei Li, Xiao Hu, and Ke Yi. Random sampling over joins revisited. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*, pages 1525–1539. ACM, 2018.
- [141] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. JOSIE: overlap set similarity search for finding joinable tables in data lakes. In *Proceedings of the 2019 International Conference on Management of Data, SIGMOD Conference 2019, Amsterdam, The Netherlands, June 30 - July 5, 2019*, pages 847–864. ACM, 2019.

1 Curriculum Vitae

Faten ATIGUI

Associate Professor, CNAM-Paris

CEDRIC Lab- ISID Team

2 Rue Conté, 75003, Paris

Email: faten.atigui@lecnam.net

2021 - French National Award of Scientific Excellence (PEDR)

1.1 Education

2013 PhD in Computer Science, University of Toulouse 1 Capitole, France

Title: Model driven methodology for data warehouses development

Supervisor: Gilles ZURFLUH

Co-supervisor: Franck RAVAT

Lab: Toulouse Computer Science Research Institute (IRIT)

Thesis defense: December 5th, 2013

2009 Master Degree in Computer Science, University of Paul Sabatier, Toulouse, France

Master of Information Retrieval and Databases

Rank: 1st

University of Paul Sabatier, Toulouse France

2008 Bachelor in Computer Science and Multimedia, University of Gabès, Tunisia

Rank: 2nd

1.2 Professional Experience

Since September, 2014

Associate Professor

Conservatoire National des Arts et Métiers (CNAM), Paris

ISID Team - CEDRIC Lab

September 2012 - August 2014

Temporary Lecturer and Research Assistant (ATER)

University of Toulouse 1 - Capitole

Toulouse Computer Science Research Institute (IRIT)

September 2009 - August 2012

PhD student - Lecturer (DCCE)

University of Toulouse 1 - Capitole

Toulouse Computer Science Research Institute (IRIT)

February 2009 - June 2009

Master Internship

Title: Customizing OLAP queries based on contextual preferences

Toulouse Computer Science Research Institute (IRIT)

2 Teaching

I started my teaching activities in 2009 according to my PhD agreement for 3 years at the University of Toulouse 1 Capitole, Then, I was a lecturer for two years at the same university. Since September 2014, I have been an Associate Professor at the CNAM PARIS. During those 13 years, I oversaw both training level (Master and Bachelor) in diverse teaching units. Otherwise, students had diverse backgrounds: Economics, Management, Computer science, Law, etc. A wide diversity of audience requires a pedagogical challenge. I should be flexible based on students' needs. To lead the way of pedagogical constraints, I prepared courses materials as well as tutorials. These courses were given on two ways: the Face-to-Face and online. For the online courses I am using the Moodle platform. Table 1 summarizes my teaching activities at the CNAM. For each course I describe the training level, number of hours, and the year.

- CNAM Paris – Associate Professor since September 2014
 - Multidimensional Databases and Data Warehousing (Master Degree)
 - Data Warehousing and Data Mining (Master Degree)
 - Database design and Administration (Master’s & B.Sc. Degree)
 - Modeling Information Systems (B.Sc. Degree)
 - Business intelligence and knowledge management (3rd year Engineer)
 - Business intelligence and analytics (Master Degree)
 - Organization and information systems (Master Degree)
 - Databases (B.Sc. Degree)
 - Information systems methodology (B.Sc. Degree)

Table 1. Summary of teaching hours per year (CNAM)

Year	Teaching hours (HED)
2014-2015	289
2015-2016	334
2016-2017	405
2017-2018	432
2018-2019	438
2019-2020	422
2020-2021	385
2021-2022	377
2022-2023	~250
Total	3332

- University of Toulouse 1 Capitole – Temporary Assistant Professor September 2012 – August 2014 (288 hours) & Instructor September 2009 – August 2012(64 hours per year)
 - Multidimensional Databases (Master Degree)
 - Data warehouse (Master Degree)
 - Decision support systems (Master Degree)
 - Information systems (Master Degree)
 - E-Management (Master Degree)
 - Databases (B.Sc. degree)
 - Reporting tools (Master Degree)

3 Responsibilities

3.1 *Research Master Degree in Information Systems and Business Intelligence (SIBI)*

From 2016 to 2019, I was the responsible for the Research Master's Degree in Information Systems and Business Intelligence. I managed the entire process, which included tasks such as student selection, organizing and overseeing master thesis defenses, facilitating jury deliberations, and recruiting instructors.

In collaboration with my colleagues, I was involved in preparing the **HCERES** evaluation and the **accreditation** renewal application for the Master's program. It was a challenging task, but the evaluations were highly positive, and as a result, the accreditation was successfully renewed until 2025.

I actively promoted the SIBI Master's degree to various French institutions, including the ESILV¹ engineering school, which led to the establishment of a double degree agreement between the two institutions. Additionally, I successfully promoted the SIBI Master's degree to public and private institutions in Algeria (USTHB), Morocco (University of Fes), and Tunisia (Faculty of Sciences of Tunis, Central University, and University of Jendouba).

Since September 2022, I have been co-responsible for the Master's in Information Systems and Business Intelligence (SIBI), which opened in continuing education (HTT).

3.2 *Other Responsibilities*

- In 2022: member of the working group on distance training at the CNAM
- Since 2021: member of the development committee for the Professional Bachelor - Web, Mobile and Business Intelligence of the CNAM
- Since 2020: member of the development committee Bachelor in Computer Science
- Since 2015: responsible for the module Advanced Information Systems (NFE103)
- Since 2017: responsible for the module Methods for computerization - complements (NFA013)
- Since 2016: responsible for the module Business Intelligence and Knowledge management (Engineer)
- 2016 – 2022: responsible for the bloc Business Intelligence and Analytics, in Specialized Master in Decision and Geo-located Information Systems (DeSIgeo).
- 2016-2019: responsible for the module Multidimensional Databases in Master Degree in Information System and Business Intelligence

4 Supervision, Tutoring, and Jury

As Associate Professor at the CNAM, I have been involved in several side activities such as supervising, tutoring and jury for several programs. Each year, for the engineering degree program, I have been supervised on average one student per year in case of continuing education and 6 in apprenticeship education. The supervisions had spread out to various levels: the first, the second and the third grade Engineer. Also, I have supervised on average 2 students per year for the specialized DESIGEO Master (Decision and Geo-located Information System). Finally, Table 3 presents the list of research master information system and business intelligence's students supervised since 2014.

Besides, I have been a tutor of bachelor's in computer science - Web, Mobile and Business Intelligence's students at both university courses: continuing and apprenticeship. The number of students is 4 per year.

Furthermore, I have regularly participated in several juries of defense for various certificates (master, research master and Computer science bachelor) for university courses: continuing and apprenticeship as tutor or chairman.

¹ <https://www.esilv.fr/>

I also have been a part of several validation panels: VAE (Validation of Acquired Experience), DPE (Engineer graduated by the State) (on average about ten per year) for the engineering diploma in Information Systems and Business Intelligence, the general license in computer science from the CNAM (on average 5 per year) and the ENG221 jury (on average 2 to 4 juries per year).

5 Research Activity

My research activity focuses on the modeling, the development, and the security of information systems, and business intelligence systems.

Model Driven Development of Business Intelligence Systems

As part of my PhD thesis at the University of Toulouse 1 Capitole, we have dealt with the problem of formalizing and automating the development process of a Business Intelligence system. We have proposed a model driven approach (MDA²) that aims to formalize and automate the process of building a BI system from its conceptual modeling to its physical implementation. Our approach integrates both multidimensional data modeling and Extract Transform and Load (ETL) processes. Also, we have proposed an approach that reduces the historical data of a warehouse, over time. Our contributions have been published in national and international journals and conferences (BDA'10, ICEIS'11, EDA'11, DaWaK'12, JDS'12, INFORSID'12, EDA'12, DaWaK'14, EDA'14).

Model Driven Development of Big Data Information Systems

For many decades, the storage and the use of data have mainly relied on Relational Databases (RDB). With the advent of Big Data, the volume of data has exploded; the variety has increased causing several issues related to digital transformation, whether in terms of storage, data exploitation, cost or performance. For this, new data management systems, highly efficient, and easy to use called NoSQL systems have appeared. In these systems, the data is organized according to 4 different families of structures, namely, *key-value*, *document-oriented*, *column-oriented*, and *graph-oriented*. Today, in addition to the existing relational solutions, there are more than 225 different NoSQL solutions. It is difficult to determine the most suitable solution that meets both functional and technical needs. Besides, transferring the database from one solution to another is a heavy and costly process. Inadequate choices can lead to scalability, data consistency, and cost issues. Following an approach that supports the user with a well-founded decision to carry out the IS project would be very helpful.

- **Contributions**

- A global approach that facilitates and automates the process of transforming a conceptual model into physical models related to the 4 NoSQL families, but also to the relational model. Moreover, this approach guides the choice of models and platforms most suited to business and functional needs. We adopt the Model Driven Architecture (MDA), which provides a formal framework for modeling and transforming models.
- A cost model that aims to guide the choice at the logical level (which family of models, normalization, partial or total nesting), and at the physical level considering technical criteria like costs, performances, financial and environmental cost.

- **Publications:** INFORSID'16, KMIS'16, DaWaK'17, AICSSA'17, ICEIS'17, CIBSE'17, PDPTA'18, EGC'20, DEXA'20, BDA'20, RCIS'22

- **Co-supervising:** 1 defended PhD (Amal AIT BRAHIM, 2015-2018), 1 PhD in progress (Jihane MALI, 2020-2023), 1 master thesis with publication (Asma MOKRANI, 2017)

Modeling Cyber-Physical Security in Healthcare Systems

Hospitals and health organizations are among the most critical and vulnerable cyber-physical infrastructures. By relying on last technological advances such as wearable sensors or in-home remote patients monitoring,

² <https://www.omg.org/mda/>

Appendix A. Curriculum Vitae of Faten Atigui

healthcare organizations are now able to provide more personalized services and open their information systems to inform patients and partners about health services, resource availability (beds and medical personnel) or patients' data through open and controlled platforms. Unfortunately, these new technologies that rely on common communication interfaces and standards, enhance security breaches, and open the door to hackers exposing hospitals to several threats.

- **Contributions**

We provide domain ontology for modeling cyber-physical security in healthcare systems. This ontology was designed to support an impact propagation model application and highlights cyber-physical interactions among hospital assets and the consequence of these hybrid relationships on incidents they may encounter.

- **Publications:** RCIS'20, CAISE'21, IEEE Access'22, 2 deliverables, 1 book chapter
- **Co-supervising:** In this project, I mainly collaborated with F. HANNOU (post-doc)
- **Project:** SAFECARE (<https://www.safecare-project.eu/>)

Data Driven Approach for Big Data Integration in Business Intelligence (BI) Systems

NoSQL systems are commonly used to store and manage big data through flexible data models, and distributed systems. However, in the rush to solve big data challenges, NoSQL systems have abandoned some of the core features of relational databases, basically, schema and integrity constraints. Although these systems are widely used today, BI remains associated with RDBs. Exploiting NoSQL data for BI requires reviewing the entire BI architecture to deal with the heterogeneity of big data. In fact, fetching relevant data that meets the decision-maker requirements, often needs to access more than one data store, thereby needs to use the join operation. While joining tables in relational data sources is straightforwardly owed to the availability of a precise join key, in NoSQL sources, like, document stores, collections are the furthest from having an exact join key due to the absence of integrity constraints.

- **Contributions**

We propose a new approach that aims to extract, transform, and load on demand document-oriented data sources. We provide a multi-source approach that enables automatic detection of join attributes between multiple collections despite the lack of integrity constraints.

- **Publications:** KES'19, RCIS'20, HICSS'21, DKE'22
- **Co-supervising:** 1 PhD (Manel SOUIBGUI, 2018-2022) – PhD defense: December 14th, 2022

5.1 Publications

The Table below summarizes the list of my publications.

Table 2. Summary of my publications

Publication type	Number of publications
International journals	3 + 1 preface
International conferences	17 (1A + 1 forum A, 9B, 5C +1 conf. ranked in SJR)
National journals	1
National or European conferences	9
Book chapter, PhD thesis, deliverables	1+1+2
Total	35

a. International Journals

1. Fatma-Zohra Hannou, Mohamad Rihany, Nadira Lammari, Fayçal Hamdi, Nada Mimouni, **Faten Atigui**, Samira Si-Said Cherfi, Philippe Tourron: Semantic-Based Approach for Cyber-Physical Cascading Effects Within Healthcare Infrastructures. *IEEE Access* 10: 53398-53417 (2022). **(Impact factor: 3,476)**
2. Manel Souibgui, **Faten Atigui**, Sadok Ben Yahia, Samira Si-Said Cherfi: An embedding driven approach to automatically detect identifiers and references in document stores. *Data Knowl. Eng.* 139: 102003 (2022). **(Impact factor: 1,99)**
3. Max Silberztein, Elisabeth Metais, Farid Meziane, Elena Kornyshova, **Faten Atigui**: Preface of the Data & Knowledge Engineering special issue following NLDB'18. In *Data and Knowledge Engineering (DKE)* (127): 101775, 2020. **(Preface, Impact factor: 1,99)**
4. **Faten Atigui**, Franck Ravat, Jiefu Song, Olivier Teste, Gilles Zurfluh: Facilitate Effective Decision-Making by Warehousing Reduced Data: Is It Feasible? *Int. J. Decis. Support Syst. Technol.* 7(3): 36-64 2015. **(Impact factor: 0.78)**

b. International Conferences³

5. Jihane Mali, Shohreh Ahvar, **Faten Atigui**, Ahmed Azough, Nicolas Travers: A Global Model-Driven Denormalization Approach for Schema Migration. *RCIS 2022*: 529-545. **(Regular paper, Conf. B)**
6. Manel Souibgui, **Faten Atigui**, Sadok Ben Yahia, Samira Si-Said Cherfi : IRIS-DS: A New Approach for Identifiers and References Discovery in Document Stores. *HICSS 2021*. **(Regular paper, Conf. A)**
7. Fatma-Zohra Hannou, **Faten Atigui**, Nadira Lammari, Samira Si-Said Cherfi : An Ontology-based Model for Cyber-Physical Security Management in Healthcare Context. *CAISE'21 Forum*. **(Forum paper, Conf. A)**
8. Jihane Mali, **Faten Atigui**, Ahmed Azough, Nicolas Travers: ModelDrivenGuide: An Approach for Implementing NoSQL Schemas. *DEXA (1) 2020*: 141-151. **(Short paper, Conf. B)**
9. Manel Souibgui, **Faten Atigui**, Sadok Ben Yahia, Samira Si-Said Cherfi: Business Intelligence and Analytics: On-demand ETL over Document Stores. *RCIS 2020*: 556-561. **(Short paper, Conf. B)**
10. **Faten Atigui**, Fayçal Hamdi, Fatma-Zohra Hannou, Nadira Lammari, Nada Mimouni, Samira Si-Said Cherfi: Managing Cyber-physical Incidents Propagation in Health Services. In *Research Projects, RCIS 2020*. **(Poster, Conf. B)**
11. Manel Souibgui, **Faten Atigui**, Saloua Zammali, Samira Si-Said Cherfi, Sadok Ben Yahia: Data quality in ETL process: A preliminary study. *KES 2019*: 676-687. **(Regular paper, Conf. B)**
12. Fatma Abdelhedi, Amal Ait Brahim, **Faten Atigui**, Gilles Zurfluh Towards Automatic Generation of NoSQL Document-Oriented Models, *PDPTA 2018*, p. 47-54, 2018. **(Regular paper, Conf. B)**
13. Fatma Abdelhédi, Amal Ait Brahim, **Faten Atigui**, Gilles Zurfluh: MDA-Based Approach for NoSQL Databases Modelling. *DaWaK 2017*: 88-102. **(Regular paper, Conf. B)**
14. Fatma Abdelhédi, Amal Ait Brahim, **Faten Atigui**, Gilles Zurfluh: UMLtoNoSQL: Automatic Transformation of Conceptual Schema to NoSQL Databases. *AICCSA 2017*: 272-279. **(Regular paper, Conf. C)**

³ Le classement des conférences est celui proposé par Core (<http://portal.core.edu.au/conf-ranks/>)

15. Fatma Abdelhédi, Amal Ait Brahim, **Faten Atigui**, Gilles Zurfluh: Logical Unified Modeling for NoSQL Databases. ICEIS (1) 2017: 249-256. **(Regular paper, Conf. C)**
16. Amal Ait Brahim, Fatma Abdelhédi, Gilles Zurfluh, **Faten Atigui**: Modeling Framework for NoSQL Systems. ClbSE 2017: 57-70. **(Regular paper, SJR : 0.121)**
17. Rahma Djiroun, Kamel Boukhalfa, Zaia Alimazighi, **Faten Atigui**, Sandro Bimonte: A data cube design and construction methodology based on OLAP queries. AICCSA 2016: 1-8, **(Regular paper, Conf. C)**
18. Fatma Abdelhédi, Amal Ait Brahim, **Faten Atigui**, Gilles Zurfluh: Big Data and Knowledge Management: How to Implement Conceptual Models in NoSQL Systems?. KMIS 2016: 235-240 **(Regular paper, Conf. C)**
19. **Faten Atigui**, Franck Ravat, Jiefu Song, Gilles Zurfluh: Reducing Multidimensional Data. DaWaK 2014: 208-220 **(Regular paper, Conf. B)**
20. **Faten Atigui**, Franck Ravat, Olivier Teste, and Gilles Zurfluh. Using OCL for automatically producing multidimensional models and ETL processes. DaWaK 2012, pages 42-53, **(Regular paper, Conf. B)**
21. **Faten Atigui**, Franck Ravat, Ronan Tournier, and Gilles Zurfluh. A unified model driven methodology for data warehouses and ETL design. ICEIS 2011, pages 247-252, **(Short paper, C)**

c. National Journals

22. **Faten Atigui**, Franck Ravat, Olivier Teste, Gilles Zurfluh. Modélisation conjointe des données et des processus pour l'implantation de schémas d'entrepôts. Journal des Systèmes Décisionnels (Journal of Decision System), JDS 21(1) : 27-49, 2012.

d. National and European Conferences

23. **Faten Atigui**, Asma Mokrani, Nicolas Travers: DataGuide : une approche pour l'implantation de schémas NoSQL. EGC 2020: 407-408. **(Poster, European conference C)**
24. Jihane Mali, **Faten Atigui**, Ahmed Azough, and Nicolas Travers : How to Implement NoSQL Schemas with ModelDrivenGuide?. In Conférence sur la Gestion de Données -- Principes, Technologies et Applications BDA 2020. **(National conference)**
25. Fatma Abdelhédi, Amal Ait Brahim, **Faten Atigui**, Gilles Zurfluh: Processus de transformation MDA d'un schéma conceptuel de données en un schéma logique NoSQL. INFORSID 2016: 15-30. **(Regular Paper, national conference)**
26. **Faten Atigui**, Franck Ravat, Jiefu Song, Gilles Zurfluh: Entrepôts de données multidimensionnelles réduites : principes et expérimentations. EDA 2014: 103-118. **(Regular Paper, national conference)**
27. **Faten Atigui**, Franck Ravat, Olivier Teste, Gilles Zurfluh. Modèle d'archivage d'entrepôt de données multidimensionnelles. INFORSID 2012: 473-488. **(Regular Paper, national conference)**
28. **Faten Atigui**, Franck Ravat, Olivier Teste, Gilles Zurfluh. Archivage d'entrepôt de données multidimensionnelles. EDA 2012: 129-138. **(Regular Paper, national conference)**
29. **Faten Atigui**, Franck Ravat, Olivier Teste, Gilles Zurfluh. Modèle unifié pour la transformation des schémas en constellation. EDA 2011, pp. 5–22 **(Regular Paper, national conference)**
30. **Faten Atigui**, Franck Ravat, Olivier Teste, Gilles Zurfluh. Démarche dirigée par les modèles pour la conception d'entrepôts de données multidimensionnelles. BDA 2010 **(Regular Paper, national conference)**

e. Book Chapter

31. **Faten Atigui**, Fayçal Hamdi, Nadira Lammari, Samira Si-Said Cherfi: Cyber-Physical Threat Intelligence for Critical Infrastructures Security. In Cyber-Physical Threat Intelligence for Critical Infrastructures Security: A Guide to Integrated Cyber-Physical Protection of Modern Critical Infrastructures, Now Publishers, 2020

f. Conference Proceeding

32. Max Silberztein, **Faten Atigui**, Elena Kornyshova, Elisabeth Metais, Farid Meziane: Natural Language Processing and Information Systems. 23rd International Conference on Applications of Natural Language to Information Systems, NLDB 2018, Paris, France, June 13-15, 2018, Proceedings. Springer, Lecture Notes in Computer Science 10859, 2018.

g. Deliverables

33. **Faten Atigui**, Fayçal Hamdi, Fatma-Zohra Hannou, Nadira Lammari, Samira Si-Said Cherfi: Specification of the impact propagation and decision Support models, Dec. 2019, Deliverable D6.6 (SAFECARE)
34. **Faten Atigui**, Fayçal Hamdi, Fatma-Zohra Hannou, Nadira Lammari, Nada Mimouni, Mohamed Rihany, Samira Si-Said Cherfi: Impact propagation and decision support model, Dec. 2020, Deliverable D6.7 (SAFECARE)

h. PhD Thesis

35. **Faten Atigui**, Model driven approach for the development and reduction of data warehouses. PhD Thesis, University of Toulouse 1 Capitole, December 2013.

5.2 PhD and Master Thesis co-Supervising

5.2.1 PhD co-Supervising

- **Amal Ait Brahim (supervision rate: 40%)**
 - Title: Model driven approach for big data modeling and implementing over NoSQL
 - Co-supervision: Gilles Zurfluh (University of Toulouse 1 Capitole), and Fatma Abdelhédi (Trimane)
 - Period: 2015-2018 – **Phd defense: October 31st, 2018**
 - Funding: grant from the University of Toulouse 1 Capitole
- **Manel Souibgui (supervision rate: 50%)**
 - Title: Towards On-demand ETL over Document Stores
 - Co-supervision: Sadok Ben Yahya (University of Tunis), and Samira Si-Said Cherfi (CEDRIC-CNAM)
 - Period: 2018-2022 - **Phd defense: December 14th, 2022**
 - Funding: Joint PhD with University of Tunis el Manar (mobility grant)
- **Jihane Mali (supervision rate: 33%)**
 - Title: Model driven development of big data information systems
 - Co-supervision: Shoreh Ahvar (ISEP), Nicolas Travers (DVRC), Ahmed Azough (DVRC)
 - Period: 2020-2024
 - Funding: co-funding from DVRC and ISEP
- **Wissal Ben Jira (supervision rate: 25%)**
 - Title: Data Lake and metadata repositories to develop city sustainability indicators using open Big Data Applied to sports practices
 - Co-supervision: Bénédicte Bucher (IGN), Malika Grim (IGN), Nicolas Travers (DVRC)

Appendix A. Curriculum Vitae of Faten Atigui

- Period: 2023-2026
- Funding: co-funding from DVRC and IGN
- **Aya Hidri (supervision rate: 40%)**
 - Title: Governance of big data retrieved from education systems
 - Co-supervision: Manel Ben Sassi (RIADI), Henda Ben Ghezala (RIADI), Samira Cherfi (CEDRIC-CNAM)
 - Period: 2023-2026
 - Funding: Joint PhD with the National School of Computer Science Tunis – RIADI Lab (mobility grant)

5.2.2 Master Thesis co-Supervising

The table below shows the students that I have supervised at the Master 2 Research in Information System and Business Intelligence.

Table 3. Research Master thesis co-supervising

Student	Thesis defense date	Title	Training place	Co-supervision rate
Amina KASRAOUI	Sept. 15, 2015	Data warehouse reduction approach	CEDRIC, CNAM	75%
Soufiane MIR	Sept. 15, 2015	Business Intelligence on the cloud	AEROW Decision	100%
Amina BOURHRARA	Sept. 12, 2016	Business Intelligence and NoSql systems	TRIMANE	100%
Rabah TIGHILT	Sept. 18, 2017	Wireless sensor networks Big Data integration and analysis	CEDRIC, CNAM	80%
Abdelaali RAJLI	Sept. 18, 2017	Studying the protection of Web applications against cyber attacks	CBI ² -TRIMANE	45%
Kaba KERFALA	Sept. 18, 2017	Quality management in BI systems	CEDRIC, CNAM	50%
Ahmed JEBALI	Sept. 18, 2017	Use of unstructured data for decision support	Bolloré - Transport and logistics	100%
Maria LEE	Sept. 18, 2017	Optimizing ETL process	Sixense Digital	100%
Ali Nabil BEN MIRA	Sept. 18, 2017	Automatic ETL to generate automatic and configurable Export files	2B-Consulting	100%
Sarah GEBRATI	Sept. 18, 2017	Study and implementation of dynamic extraction process of document-oriented schema	CBI ² -TRIMANE	45%
Fernandez ROBERTO	Sept. 18, 2017	Automatic process for the generation of a multidimensional schema	SGS Group	100%
Asma MOKRANI	Sept. 17, 2018	Denormalizing NoSQL schema	CEDRIC, CNAM	50%
Amel ZRELLI	Sept. 17, 2018	Improvement of energy	Gisgo Sas	100%

Appendix A. Curriculum Vitae of Faten Atigui

		approximation algorithm		
Karim KIDISS	Sept. 18, 2018	Identification of tourist behavior: analyzing social networks	Léonard de Vinci-ESILV	20%
Salsabil AMARA	Sept. 18, 2018	User trace analysis - Learning analytics	University of Paris Sorbonne	20%
Corentin LEMAITRE	Sept. 24, 2019	Methodology for chaining legal decisions and enriching data in a decision-making database	CBI ² .TRIMANE	50%
Salaheddine BALHOUSSE	Sept. 12, 2019	Implementation of a CRM solution	BMUST BI Consulting	100%
Ismail BALHOUSSE	Sept. 12, 2019	Business Intelligence solution: ETL and data warehouse	BMUST BI Consulting	100%

5.3 *Scientific Patency*

5.3.1 Jury Member of PhD Thesis Defense

- Candidate: Rabah TIGHILT
 - PhD thesis defended on November 16th, 2021, at the University of Toulouse 1 Capitole
 - Title: Model-driven approach to extract models from document-oriented NoSQL database
 - Supervisor: Gilles ZURFLUH, Professor, University of Toulouse 1 Capitole

5.3.2 Invitation and Expertise

- **Erasmus program 2022**
The exchange program between the CNAM and the University of Carthage in which we benefit from 3 scholarships as follows:
 - From March 27 to March 31st, 2022: stay at the National Institute of Applied Sciences of Tunis (INSAT): I gave a course on BI for the 2nd year engineer students
 - In 2023: a one-week mission to welcome my colleague Sonia GABBOUJ, Professor at the University of Carthage, to provide a course on industry 4.0 to our students at the CNAM
 - Scientific internship in 2023: a doctoral student will be invited at the CNAM for 4 months: we will work on the integration of big data into BI systems
- **Expertise CIFRE, October 2021**
Evaluation of a proposal for a doctoral thesis subject to obtain CIFRE funding
- **International expertise 2020**
Associate Professor promotion at **Zayed University, Dubai, Emirats Arabes Unis**
- **Guest speaker** at the Big Data days, organized in Tunis (Beit el Hikma), October 28, 2016
- **Guest speaker** at the workshop on active pedagogy from November 1st to November 3rd, 2018, Hammamet-Tunisia
- **In February 2017, invited Professor**, University of Jendouba, Tunisia:
 - Teaching: 12 hours of training on Business Intelligence and Analytics, Master Degree
 - Research: 3 meeting to prepare a project proposal on learning analytics and business intelligence (cf. submitted project in section Research Project)

- **In December 2016, invited Professor**, Central University, Tunis, Tunisia:
 - Teaching: 15 hours of courses on information systems, Bachelor Degree
 - Engineer thesis defense: invited member of the jury in 4 engineering defenses in information systems and business intelligence
- **CNAM-Tunisia action plan**
 - Welcome of the delegation University of Carthage and University of Sfax at CNAM from April 24 to April 25, 2018
 - Promotion and presentation of the Business Intelligence and Big Data advances and of the Information System and Business Intelligence Master program.
- **Committee Selection (COS) member for Associate Professor or equivalent**
 - Associate Professor position n°4200, profile: Information Systems Security, **vice president of the COS, 2019**
 - Associate Professor position n°4459, profile: Information Systems Engineering, University of Paris 1 Panthéon-Sorbonne, **2018**
 - Associate Professor position, Télécom Sud Paris, profile: Management of Information Systems, **2018**
 - Associate Professor position n°4139, profile: modeling, verification, and evaluation of distributed service-oriented systems (IMO-VESPA team), **2016**
 - Assistant Professor position section 27 (MCF 1272), profile: software engineering, University of Auvergne, **2015**

5.3.3 Chair, Organizing, and Program Committee of Conferences

- **Workshop co-chair**: 1st International Workshop on Business Intelligence over Big Data (BIBD), September 8, 2019 Bled, Slovenia, 8 September 2019 collocated with ADBIS 2019
- **Workshop and Tutorial co-chair**⁴: 14th International Conference on Computer Systems and Applications AICCSA'2017
- **Organizing committee**⁵: 23rd International Conference on Natural Language & Information Systems, NLDB'2018
- **Co-organization of the « Focus group and Management Meeting »** January 23 and 24 2020, Paris – SAFECARE project
- **Member of the organizing committee** 26th Advanced Database Days, Toulouse, 2010
- **Regular participation in program committees and reviewer**
 - **International conferences**: ACS/IEEE International Conference on Computer Systems and Applications (**AICCSA**) 2017, 2018, 2019, 2020; International Conference on Database and Expert Systems Applications (**DEXA**) 2017, 2018, 2019, 2020, 2021, 2022; International Conference on Natural Language & Information Systems (**NLDB**) 2018, 2019, 2020, 2021, 2022; International Workshop Quality of Models and Models of Quality (**QMMQ**) 2017, 2018, 2019, 2020; International Conference on Information and Communications Security (**ICICS**) 2017, 2018; Hawaii International Conference on System Sciences – (**HICSS**) 2021
 - **National conferences**: **INFORSID** 2016, 2017, 2018; **EDA** 2017, 2018, 2019, 2020, 2021
 - **International journal**: **ACM Computing surveys** 2018, 2019, 2020.

⁴ <http://www.aiccsa.net/AICCSA2017/conference-committees>

⁵ <http://nldb2018.cnam.fr/organizing-team.php>

Appendix A. Curriculum Vitae of Faten Atigui

- **National journal:** Revue des Nouvelles Technologies de l'Information (TSI) 2017, 2018, 2019.

5.4 *Research Project and Scientific Responsibilities*

5.4.1 Research Project

The table below summarizes my participation in national and international projects.

Table 1. Research project

Project	Period	My role	Budget	Collaboration & Partners
SAFECARE H2020: SAFEguard of Critical health infrastructure	2018-2021	Participant	454K€ (Total project budget: 7 573 979)	21 European partners
DAPHNE	2018-2021	Participant	464K€	CEDRIC-Cnam, LAMOP UMR 8589, LARHRA Rhône-Alpes, TECHNE, UM-LIRMM Montpellier
QUALHIS	2017-2018	Participant	~25K€	LAMOP Univ. Paris 1, CEDRIC-CNAM, TECHNE - Univ. Poitiers

- **SAFECARE (2018-2021) - SAFEguard of Critical health infrastructure, H2020 project**

- Objective: Security management of health services in a cyber-physical environment
- Consortium: 21 European industrial, academic, and administrative partners
- CNAM task manager: N. Lammari and S. Cherfi
- Budget for the Cnam € 454,000 (Total project budget: € 7,573,979)
- Coordinator: APHM (Hospitals of Marseille), P. Tourron

The objective of SAFECARE is to provide solutions to strengthen the cyber and physical security of healthcare organizations. As partner of SAFECARE Consortium, we were responsible for a primary task, which is the development of the impact propagation module, and we were engaged in 4 work packages.

In this context, I collaborated with my colleagues and partners in acquiring knowledge from business experts in hospitals and collectively modeling the domain ontology dedicated to the management of security. This ontology served as a foundation for the impact propagation module.

I have also participate in meetings organized with the 21 European partners (3 conference calls per fortnight), discussion groups, etc. I am also involved in the drafting of the different deliverables.

- **DAPHNE (2017-2022) – ANR project**

- Objective: Discovery in the historical prosopographical basics of knowledge
- Consortium: CNAM-CEDRIC, LAMOP (Paris), LARHRA (Rhône-Alpes), TECHNE Digital technologies for education, UM-LIRMM Laboratory of Computer Science, Robotics and Microelectronics of Montpellier,
- Coordinator: C. du MOUZA (CNAM-CEDRIC)

- **QualHIS (2017-2018) - CNRS Mastodons project**

- Objective: A quality approach for developing and querying large historical prosopographical databases.
- Consortium: LAMOP Univ. Paris 1, ISID CNAM Paris, TECHNE - Univ. Poitiers.
- Coordinator: LAMOP, S. Lamassé

- **PHC UTIQUE 2024 (project submitted as project manager)**
 - Objective: Educational data governance for intelligent assistance of distance learning
 - Consortium: CEDRIC-CNAM, Research Laboratory in Software Engineering & Applied, Distributed and Intelligent Computing (RIADI) - National School of Computer Science, Experimentation and Innovation Laboratory (IS Lab) - Institut Mines Télécom business school
 - **Coordinators**
 - French partner: **F. Atigui** (CNAM-CEDRIC)
 - Tunisian partner: H. Ben Ghezala (Professeur, RIADI, ENSI – Tunis)

5.4.2 Scientific Responsibilities

- In 2022, member of the working group on distance training and “Qualiopi” certification at the CNAM
- Since 2022: member of the improvement committee for the master's degree in information systems and business Intelligence (SIBI)
- Since 2021: member of the improvement committee for the professional Bachelor - Web, Mobile and Business Intelligence of the CNAM
- Since 2020: member of the improvement committee for the Bachelor's in computer science
- Since 2019: elected member of the scientific committee of the CEDRIC laboratory, for a 4-year term
- In 2019: member of the CEDRIC and SMI PhD committee
- In 2019: member of the IEEE association

