Semantic-enhanced Programmable Knowledge Graph (SPG)

White paper（v1.0）

——The new generation industrial-grade knowledge semantic framework and engine

Utilize SPG's domain type constraints and the fusion representation of facts and logic to automatically complete and increase the semantic relations between knowledge elements, promoting the explicit densification of sparse relationships between knowledge elements. Using the SPG framework can accelerate the knowledge integration of massive enterprise data, and seamlessly connect AI applications through its knowledge symbolic representation and programmability capabilities.

The main Idea of SPG

Ant Group × OpenKG co-produced

August 2023

Copyright Notice

Writing instructions

Lead Writing Unit: Ant Technology Group Co., Ltd.

Participating Writing Units: Tongji University, Tianjin University, Hundsun Electronics Co., Ltd., Zhejiang Chuanglin Technology Co., Ltd., Daguan Data Co., Ltd., Haiyizhi Information Technology (Nanjing) Co., Ltd., Zhejiang University, Zhijiang Laboratory, Institute of Computing Technology, Chinese Academy of Sciences.

Writing team member

Ant Technology Group Co., Ltd.: Lei Liang, Zhiqiang Zhang, Jin Peng, Peilong Zhao, Zhihui Guo, Yuxiao He, Lin Yuan

Tongji University: Haofen Wang

Tianjin University: Xin Wang, Xiang Wang

Hundsun Electronics Co., Ltd.: Shuo Bai, Jiao Chen

Zhejiang Chuanglin Technology Co., Ltd.: Yan Zhou, Chen Zhang

Daguan Data Co., Ltd.: Wenguang Wang, Mengjie He

Haiyizhi Information Technology (Nanjing) Co., Ltd.: Fanghuai Hu, Jun Ding

Zhejiang University: Huajun Chen, Wen Zhang

Zhijiang Laboratory: Heng Zhang

Institute of Computing Technology, Chinese Academy of Sciences: Long Bai

Recommendations

The knowledge graph is an extension of early expert systems and semantic web technology. Since Google applied it to the search recommendation field in 2012, knowledge graph technology has been widely adopted in various domains. However, the semantic representation and technical framework of knowledge graphs have not made significant progress for a long time, leading to increased costs and complexities in constructing knowledge graphs across different fields. I am pleased to learn about the collaboration between Ant Group and OpenKG, which leverages Ant Group's extensive industrial experience in knowledge graphs to propose a knowledge semantic framework called SPG. SPG is compatible with big data systems and AI technology systems, and it offers programmability, framework characteristics, and strong cross-scenario migration capabilities. This accelerates the industrialization of knowledge graphs and represents a breakthrough in the knowledge graph technology framework. Since the end of 2022, large language models (LLMs) such as ChatGPT and GPT4 have triggered a new wave of artificial intelligence. However, current LLMs still face challenges such as knowledge illusion, complex reasoning fallacies, and high computational costs. As a complement to LLMs, the technical system of symbolic knowledge graphs enables controlled content understanding and generation. It provides support for accurate domain knowledge and complex reasoning capabilities, facilitating the implementation of LLMs across different industries. We anticipate that SPG will become an important technology in the field of knowledge graphs. Through Ant Group's continuous refinement across diverse scenarios and its collaboration with the OpenKG community, it will drive industry development in the field of knowledge graphs, promote knowledge interconnection across different domains, and enable the controlled and low-cost implementation of LLMs and knowledge graph technology.

——Juanzi Li, Director and professor of the Knowledge Intelligence Research Center of the Institute of Artificial Intelligence of Tsinghua University

As a symbolic knowledge representation system, the knowledge graph possesses capabilities such as high-order semantics, rigorous structure, and complex reasoning. In the era of rapid development of large language models (LLMs), there is a rich interactive relationship between knowledge graphs and LLMs. On one hand, LLMs provide a powerful tool for constructing large-scale knowledge graphs at a low cost. Whether leveraging LLMs to build a world knowledge graph beyond the existing scale by 1-2 orders of magnitude has become an intriguing research question. On the other hand, the knowledge graph, with its high-quality and interpretable knowledge representation and reasoning capabilities, offers a potential exploration direction for addressing the idealistic challenges of LLMs.

Traditional knowledge semantic frameworks like RDF/OWL and LPG have significant limitations in knowledge management and struggle to support the construction and application of knowledge graphs in the

Limitations of Traditional Frameworks

era of LLMs. SPG, derived from the extensive business practices of the Ant Knowledge Graph team, effectively addresses the deficiencies of RDF/OWL and LPG in knowledge management. It represents a new generation of knowledge semantic framework that builds upon the engine architecture by leveraging SPG's semantic specifications and programmable paradigms. SPG facilitates efficient graph construction in various domains and enables semantic alignment of knowledge across different fields.

SPG's Semantic Specifications and Programmable Paradigms

The future development of the knowledge graph relies heavily on an active community. Ant Group will continue collaborating with the OpenKG community in areas such as SPG, the construction and evolution of the world knowledge graph, to accelerate its technological maturity and industrial implementation. We also welcome colleagues from industry and academia to actively participate in co-creation, jointly promoting the maturity and progress of knowledge graph technology, facilitating knowledge exchange and circulation between different fields, and building a new generation of AI technology systems driven by the controllable implementation of knowledge graph + LLMs.

—— Wenguang Chen, President of Ant Group Technology Research Institute

Ant Group possesses diversified business scenarios and massive amount of domain data. The SPG framework has been developed based on the extensive practical experience of Ant Group in knowledge graphs. The characteristics of Ant Group's business data, such as multi-source heterogeneity, temporal dynamics, and complex correlations, provide an excellent environment for constructing large-scale knowledge graphs. The SPG framework, by abstractly addressing multi-business and multi-scenario challenges, defines a new generation enterprise-level knowledge management paradigm with strong adaptability for enterprise-level applications. Through data intellectualization, the SPG framework transforms massive data into knowledge and solves high-dimensional business problems through methods like complex pattern calculation and graph learning reasoning. The SPG framework presents innovative possibilities for efficient domain knowledge graph construction and cross-domain knowledge graph semantic alignment. Furthermore, in the era of large language models (LLMs), the SPG framework, along with the domain knowledge graph built upon it, enables controlled implementation of LLMs in various business fields such as security risk control, micro credit, and digital finance. Through collaboration with OpenKG, we aim to accelerate the enhancement of the SPG framework by harnessing the power of the community and industry, promote the maturity of knowledge graph technology, and advance industry development. Throughout this journey, we welcome the active participation of all colleagues in co-creation, jointly driving the development and innovation of knowledge graph technology, and realizing controllable AI driven by both LLMs and knowledge graphs, ultimately expediting industry implementation.

—— Jun Zhou, Head of Machine Intelligence Department and Researcher at Ant Group.

Preface

As a method of modeling and managing data, knowledge graph has played a crucial role in the digitalization of enterprises. However, with the increasing demand for knowledge graph, traditional knowledge graph technology is encountering several challenges. Through extensive research and practical experience, Ant Group has identified limitations in traditional knowledge graph technology when dealing with complex business scenarios and large-scale data. For instance, the construction of knowledge graphs requires a unified industrial-level knowledge modeling framework that can adapt to diverse fields. The reasoning capabilities of knowledge graphs need to be more efficient and interpretable. Furthermore, the construction and reasoning processes of knowledge graph require enhanced programmability and cross-scenario transferability.

Lei Liang, as the Head of Ant Group's Knowledge Engine, led the team in developing an industrial-level knowledge graph semantic framework called SPG (Semantic-enhanced Programmable Graph). During his initial introduction of the idea and SPG to me, I was pleasantly surprised to find that we were solving similar challenges at the same time. What was initially planned as a one-hour meeting gradually evolved into a morning of in-depth discussions. Subsequently, I felt increasingly compelled to integrate our efforts in expanding SPG to address new opportunities and needs in the era of large language models (LLMs), while also open-sourcing this comprehensive and innovative knowledge graph platform to the entire community. When I shared this idea with Lei Liang, both he and Ant Group provided strong support. We actively promoted collaboration between the various R&D teams of OpenKG and the Ant Knowledge Graph team, ultimately forming a virtual team to facilitate bi-weekly communication, design planning, and research and development work.

Structure of SPG

The SPG framework is based on the property graph, combining the semantic nature of RDF/OWL and the structural nature of LPG. It offers the advantages of semantic simplicity and compatibility with big data. Through the SPG framework, we can achieve automatic layering of knowledge from dynamic to static, ensure the uniqueness of knowledge within domains, and define dependencies between knowledge. Furthermore, the SPG framework provides a programmable paradigm, supporting the rapid construction of new domain knowledge graph and cross-scenario migration. It has wide-ranging applications in solving typical problems and scenarios. In the context of risk mining knowledge graph and enterprise causal knowledge graph, the SPG framework can assist enterprises in identifying and addressing illicit activities, enhancing risk prevention and control capabilities. In terms of knowledge reasoning and intelligent question answering, the SPG framework can provide more accurate and interpretable inference results, improving user experience and decision-making effectiveness.

Knowledge Layering of SPG

Programmability of SPG and new domain knowledge graph

Accurate and interpretable inference results of SPG

In this whitepaper, we will provide a detailed introduction to the design principles, technical modules, and application cases of the SPG framework. We hope that through this whitepaper, readers will have a comprehensive understanding of the SPG framework and be inspired to engage in further discussion and collaboration. We believe that the SPG framework will provide stronger and more flexible support for

enterprise digitalization, driving the development and application of knowledge graph technology. Lastly, we would like to express our gratitude for your attention and support of this whitepaper. If you have any questions or suggestions regarding the SPG framework or knowledge graph technology, please feel free to contact us. Let's work together to create a future for the next generation of industrial-level knowledge graph!

Thank you!

——Haofen Wang, Lei Liang, and the SPG team

Contents

Chapter 1 From Data-Driven to Knowledge-Driven: Enterprises Deepen Competitive Advantages with Evolving Knowledge Graph Technology . . . 1
1.1 The Expectations of Knowledge Graph as the Next-generation Enterprise Knowledge Management Paradigm . . . 1
1.2 Transition from Binary Static to Multidimensional Dynamic: Shift in Knowledge Management Paradigms . . . 2
1.3 Integrating Domain Knowledge Provides New Approaches for AI Implementation. 5
1.4 The Development of Knowledge Graph Technology System Needs to Keep Pace with the Times. . . . 7
1.5 Industrial Knowledge Graph Engine Based on SPG . . . 7
Chapter 2 Challenges of Knowledge Management base on Labeled Property Graph . . . 10
2.1 Typical Case 1: Risk Mining Knowledge Graph . . . 10
2.2 Challenges in Applying LPG to the Risk Mining Knowledge Graph . . . 13
2.3 Typical Case 2: Enterprise Causal Knowledge Graph . . . 14
2.4 Challenges in Applying LPG to Enterprise Causal Knowledge Graph . . . 18
2.5 Complexity and heterogeneity caused by the coupling of structural definition and semantic representation in knowledge modeling . . . 19
2.6 Insufficient expressive power for representing diverse and heterogeneous domain knowledge. . . . 22
2.7 Consistency and propagative reasoning issues caused by logical dependencies between knowledge. . . . 25
2.8 Graph Construction and Evolution Problems for Incomplete Data Sets . . . 27
2.9 Summary of Problems with Semantic-less, Non-programmable Labeled Property Graph. . . . 29
Chapter 3 Semantic Enhancement Programmable Framework (SPG) . . . 30
3.1 The semantic model of SPG . . . 30
3.2 SPG Layered Architecture . . . 32
3.3 The Objectives of SPG . . . 33
Chapter 4 SPG-Schema Layer . . . 35
4.1 Overall Architecture of the SPG-Schema. . . . 35
4.2 Semantic Enhancement of Nodes and Edges . . . 40
4.3 Semantic Enhancement of the Predicates and Constraints . . . 44
4.4 Semantic Enhancement through Rule Definitions . . . 51
4.5 The Relationship between SPG-Schemas and PG-Schemas . . . 53
4.6 Summary of SPG-Schema . . . 54
Chapter 5 SPG-Engine Layer . . . 55
5.1 The Architecture of SPG-Engine . . . 55
5.2 SPG2LPG Translator . . . 56
5.3 SPG2LPG Builder. . . . 59
5.4 SPG2LPG Executor . . . 60
5.5 Basic Requirements for SPG-Engine on the Property Graph Systems. . . . 64
5.6 Advanced Requirements for SPG-Engine on the Property Graph Systems . . . 65

5.7 Summary . . . 67
Chapter 6 SPG-Controller Layer . . . 68
6.1 The Architecture and Workflow of SPG-Controller . . . 68
6.2 Parsing, Compilation, and Task Planning . . . 69
6.3 Task Distribution and Invocation . . . 69
6.4 Knowledge Graph Construction . . . 70
6.5 Knowledge Query . . . 70
6.6 Knowledge Graph Reasoning . . . 71
6.7 Full-Text Search and Vector Search . . . 71
6.8 Deployment of the Services and Tasks . . . 71
6.9 Summary . . . 72
Chapter 7 SPG-Programming Layer . . . 73
7.1 SPG Semantic Programmable Architecture . . . 73
7.2 The Construction and Transformation from Data to Knowledge . . . 74
7.3 Logical Rule Programming . . . 76
7.4 Knowledge Graph Representation Learning . . . 77
7.5 Summary . . . 78
Chapter 8 SPG-LLM Layer . . . 79
8.1 SPG-LLM Natural Language Interaction Architecture . . . 79
8.2 Automatic Extraction and Automated Construction of Knowledge Graphs . . . 79
8.3 Domain Knowledge Completion with LLMs . . . 82
8.4 Natural Language Knowledge Querying and Intelligent Question Answering . . . 82
8.5 Summary . . . 83
Chapter 9 New Generation Cognitive Application Cases Driven by SPG . . . 84
9.1 Enterprise Causal Knowledge Graph Driven by SPG . . . 84
9.2 Comparison between SPG and LPG in the context of Enterprise Causal Knowledge Graph . . . 89
9.3 SPG-Driven Risk Mining Knowledge Graph . . . 90
9.4 Comparison between SPG and LPG in the Risk Mining Knowledge Graph . . . 96
9.5 Summary . . . 96
Chapter 10 SPG Embracing the New Era of Cognitive Intelligence . . . 97
10.1 SWOT Analysis of SPG Compared to Property Graphs . . . 97
10.2 Problem Resolution and Outstanding Issues from Chapter 2 . . . 99
Chapter 11 Outlook on the Future of SPG . . . 100
References . . . 103

Chapter 1 From Data-Driven to Knowledge-Driven: Enterprises Deepen Competitive Advantages with Evolving Knowledge Graph Technology

In the process of digitalization, enterprises have accumulated massive amounts of data. This includes both unstructured and semi-structured data such as text, images, videos, and audio, as well as structured data such as user behavior, product orders, services, and merchant profiles. Additionally, there are professional knowledge bases and industry data obtained from external channels to support business development. Faced with this vast amount of data, enterprises need to continuously create value for users while ensuring efficient management and risk control. This places high demands on the digital infrastructure of enterprises and provides diverse scenarios for AI technologies such as Knowledge Graphs (KGs) and Large Language Models (LLMs). It also brings new opportunities and challenges. AI technologies can help enterprises quickly discover patterns, analyze trends, and predict the future from massive amounts of data. This enables enterprises to better understand customer needs, optimize product design, and improve production efficiency. AI can also assist in intelligent risk management and anti-fraud detection. However, enterprises often face challenges such as data silos, data consistency conflicts, and data duplication due to business development and departmental differences. To improve data utilization efficiency, it is necessary to strengthen data management and application, and increase the utilization and value of data. Enterprises need to establish user-friendly management paradigms, define data structures based on business models, clarify semantics, eliminate ambiguities, and identify errors. They also strive to establish mechanisms for connecting data silos, enabling cross-system and cross-department data sharing and collaborative utilization. In addition, enterprises need to establish standardized data and service agreements to achieve efficient data collaboration, expert experience collaboration, and human-machine collaboration. Efficient data management mechanisms, standardized data modeling, ambiguity elimination to enhance consistency, and data silo connection are key issues faced by enterprises in their digitalization journey. More efficient organization and management of enterprise data and the utilization of AI technologies to fully explore data value have become the core driving forces for future enterprise growth. KG Description & Usage

1.1 The Expectations of Knowledge Graph as the Next-generation Enterprise Knowledge Management Paradigm

As an important branch of AI technology, Knowledge Graph has gained increasing popularity due to its ability to help enterprises organize and manage knowledge data more effectively. By semantically modeling and constructing a knowledge graph, enterprises can gain a better understanding of the relationships between data, uncover hidden values, and make informed decisions. In fact, Gartner predicted in 2021 that Data Fabric, based on Knowledge Graph technology, would become the next-generation data architecture. Neo4j and Cambridge Semantic have also released whitepapers introducing a new generation

of knowledge management paradigms based on Knowledge Graph. Neo4j considers a Knowledge Graph as a semantically enhanced graph, leveraging certain paradigms to semantically enhance the graph and discover more implicit clues from multidimensional relations. Cambridge Semantic believes that Knowledge Graph is a killer application for Data Fabric. It models entities, facts, concepts, and their relations in the real world, providing consistent modeling capabilities for different roles. It enables more accurate representation of organizational data and effectively connects data sources, graph storage, and downstream AI/BI tasks, breaking down data silos and enabling on-demand integration, loading, and seamless connection. Since 2018, Knowledge Graph applications in enterprise digitalization have been widely adopted in various vertical domains such as finance, healthcare, public security, and energy [1, 2, 3]. According to a report [4], the market size of Knowledge Graph in China is expected to reach 29 billion RMB by 2026, with finance and public security being the main driving forces. In the context of enterprise digitalization, the application of Knowledge Graph, such as merchant knowledge graph for merchant risk control, requires a deeper understanding of knowledge, particularly the need for in-depth context (i.e., Deep Context) perception for profiling and risk insights on thin data customer groups such as small and medium-sized businesses, new users, and dormant users [1]. Enterprise-level knowledge management is undergoing a transition from binary static models to dynamic multidimensional models.

1.2 Transition from Binary Static to Multidimensional Dynamic: Shift in Knowledge Management Paradigms

What is Knowledge Graph?

Knowledge Graph is a method of modeling and managing data that utilizes graph structure, knowledge semantics, and logical dependencies to provide capabilities for storing, reasoning, and querying factual knowledge. In its early applications, Knowledge Graph mainly involved extracting <s, p, o> triplets from public corpora to construct static knowledge graphs, aiming to improve search and recommendation efficiency and user experience. As Knowledge Graph applications have shifted from consumer-oriented applications like search and recommendation to enterprise-level applications in risk control and business management, as mentioned earlier, there is a growing demand for profiling and risk insights on long-tail sparse customer groups. This necessitates domain-specific graphs that possess comprehensiveness, accuracy, and interpretability. Moreover, the data sources for knowledge graphs have expanded beyond textual corpora to include diverse and heterogeneous enterprise data. These data sources include unstructured or semi-structured User-Generated Content (UGC) or Professionally-Generated Content (PGC), structured profiles derived from business operations, transactional data, log records, as well as domain-specific business expert knowledge. To support growth management and risk control, it is crucial to build comprehensive profiles of customers, materials, channels, and other dimensions. Taking merchants as an example, Figure 1 illustrates the process of constructing such multidimensional profiles.

Subject , Predicate , Object - enabling structured data storage and efficient querying.

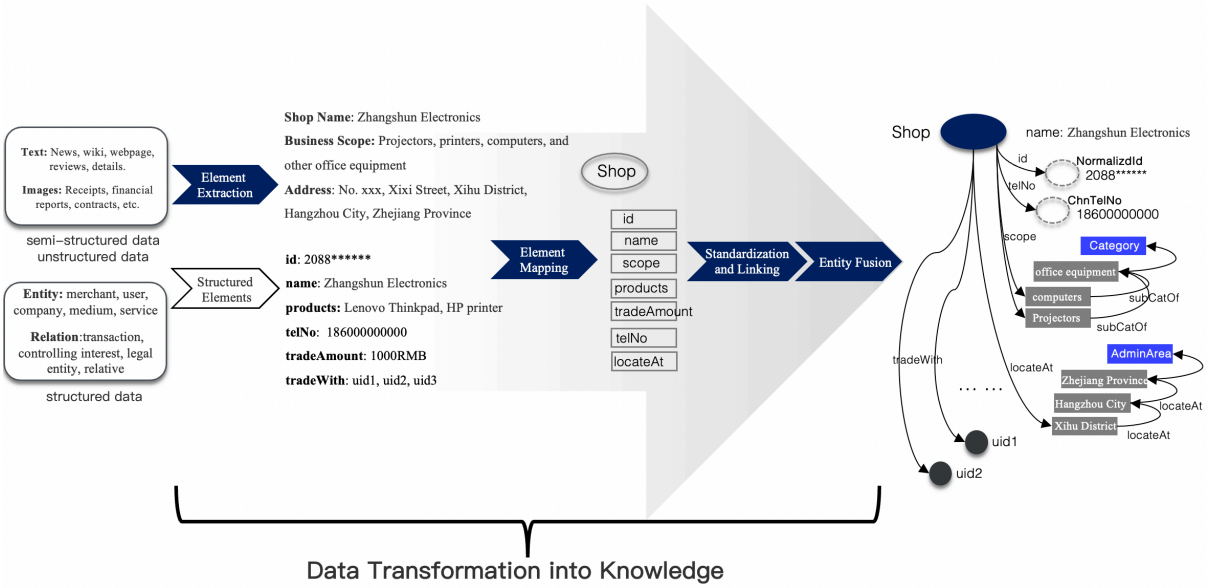

Figure 1: Constructing Merchant Entities

Merchants have surpassed the limitations of static physical stores, as anyone can become a merchant through payment codes. However, this also increases the difficulty of risk control. It is meaningless to rely solely on textual concept tags for risk control, and adding actual factual relations such as transactions and social connections is far from sufficient. As shown in Figure 2, deep collaborative information from multiple aspects of entities is needed to discover more effective associations. The requirements for Knowledge Graph construction have shifted from static common knowledge to dynamic Deep Context in temporal and spatial dimensions. This requires relation propagation based on media (such as Wi-Fi, phone, email) and boundary-based aggregation associations in continuous spatial dimensions [5,6]. It also involves tracking the multidimensional propagation context of events at different levels (micro, macro, and meso), achieving dense representation of sparse semantic relations between entities that are interpretable.

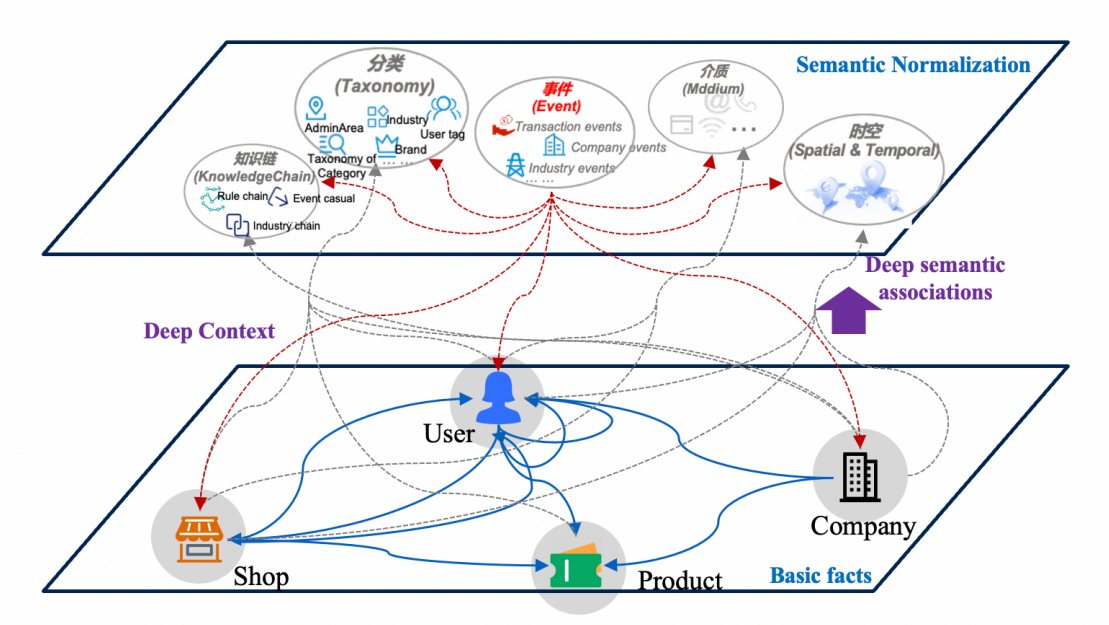

Figure 2: Deep Context Semantic Expansion of Foundational Fact Knowledge Graph

In terms of business applications, Knowledge Graphs can be used to construct knowledge reasoning tasks, such as: 1) Product recommendations: By leveraging semantic connections like category, intent, and temporal information, the semantic associations between people-products, people-merchants, and products-channels can be established, enabling semantic recall of products and representation transfer. 2) eKYB (Electronic Know Your Business): By leveraging media associations, behavioral events, and temporal aggregation, identification of shared merchants or individuals can be achieved, enabling effective profile completion and risk insights. In addition, Knowledge Graphs can also facilitate structured-aware text generation [7], such as: 1) Anti-money laundering intelligent adjudication and qualitative report generation: By combining Deep Context to predict risk behavior and detect criminal networks, the structure of networks and anomalies can be reconstructed through financial chains, temporal aggregation, and device associations, and then transformed into interpretable reports through knowledge graph to text conversion. 2) AI phone call victim alert: Suspicious devices, phishing domains/AppIDs, and criminal networks can be associated with transactional users, generating scripted conversations to alert users and intercept risks. These applications aim to achieve more intelligent and precise risk control and business inference, enhancing the efficiency and value of business operations.

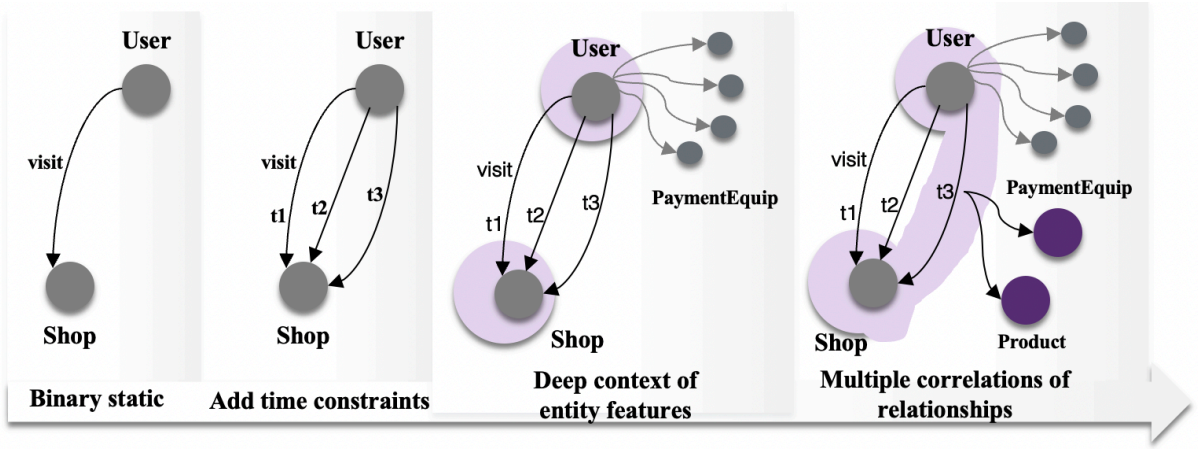

Figure 3: Evolution of Knowledge Representation from Binary to Multivariate

In the case of merchant management and risk control, knowledge management requires strong contextual awareness. Common knowledge graphs, which lack the ability to perceive contextual information and temporal associations, often suffer a significant decrease in effectiveness when applied to scenarios with diversified or intertwined argument elements, as they are unable to perceive individual differences and rely solely on concept-level induction for reasoning [8]. Similar challenges arise in fields such as anti-fraud in public security, insurance claims, medical consultations, and corporate credit assessment. As a result, there has been a significant shift in the expectations of vertical industries towards knowledge graph. Knowledge representation has evolved from the binary static structure depicted in Figure 3 to multidimensional dynamic associations in temporal and spatial dimensions, better aligning with the requirements of real-world applications.

1.3 Integrating Domain Knowledge Provides New Approaches for AI Implementation

According to Yunhe Pan, a member of the Chinese Academy of Engineering, data and knowledge are the two most important elements in the development of AI 2.0. Dealing with big data and multiple knowledge domains forms the core technologies for AI development, as knowledge can effectively assist in AI cognition, decision-making, and learning. Combining KG & LLM

During the process of digitalization, a large amount of domain-specific knowledge, such as factual knowledge, expert experience, and operational procedures, can be accumulated through the extraction of massive data or business practices. This knowledge, existing in various industries and difficult to obtain publicly, holds immense value. By effectively integrating industry expert knowledge with AI, issues related to controllability, safety, and interpretability in AI applications can be addressed. By the end of 2022, ChatGPT had gained global popularity, followed by a surge of similar models in the domestic market. However, as Large Language Models (LLMs) are black-box probabilistic models [9], they struggle to capture and acquire factual knowledge, resulting in illusions and logical errors [10]. Meanwhile, Knowledge Graph (KG) provide factual accuracy, timeliness, and logical rigor, making them an excellent complement to LLMs. The application paradigm of LLM+KG, where Knowledge Graph serve as constraints and a source of complex reasoning capabilities, has attracted widespread attention and sparked numerous application explorations and research studies [9,10,11].

Table 1: Applications of LLMs and KGs in different digital enterprise scenarios.

Scenarios and applications		LLM only	KG enhanced LLM	LLM augmented KG	KG only
Business Growth	Interactive applications	Chat, write poems and songs	Knowledge Q&A, service retrieval, report analysis, etc.	-	Marketing recommendation, event context, marketing decision-making, etc.
	Marketing Recommendation	-	Data report query, crowd label selection, intelligent copywriting, etc.	-	Event analysis, materials analysis, crowd analysis, etc.
Risk Control	Risk forecasting and control	-	Explanatory message generation, waking up the robot, etc.	-	Clues tracking, events transmission, rule based claims, corporate credit, ultimate beneficiaries, equity penetration, etc.
Knowledge Construction	Knowledge extraction	-	-	Document element extraction, event extraction, entity linking, etc.	Knowledge construction based on structured business data
	Knowledge completion	-	-	Obtain the entity LLM embedding representation, extract and supplement the missing knowledge in the knowledge graph from the LLM	Relationships mining, properties prediction, groups mining, rules mining, etc.

In various application scenarios, taking merchant management and risk control as an example, the algorithm tasks can be categorized into the following five aspects: (1) Interactive Applications: including displaying products/services on the consumer end (C) and onboarding services/merchants on the business end (B). (2) Business Management: necessary business analysis and material management for enterprise and merchant operations. (3) Risk Control: combating illicit activities is an ongoing challenge for businesses, requiring enhanced awareness of thin data customer groups and rapid identification of new risk patterns. (4) Knowledge Construction: transforming external unstructured/semi-structured and structured data into

domain knowledge. (5) Knowledge Mining: continuous improvement of coverage for key elements and cross-entity relations to facilitate business growth and risk control. Table 1 lists potential applications of LLMs, KG and the mutual enhancement of LLMs and KG across different categories. These applications can help enterprises achieve better results and outcomes in the field of merchant management and risk control.

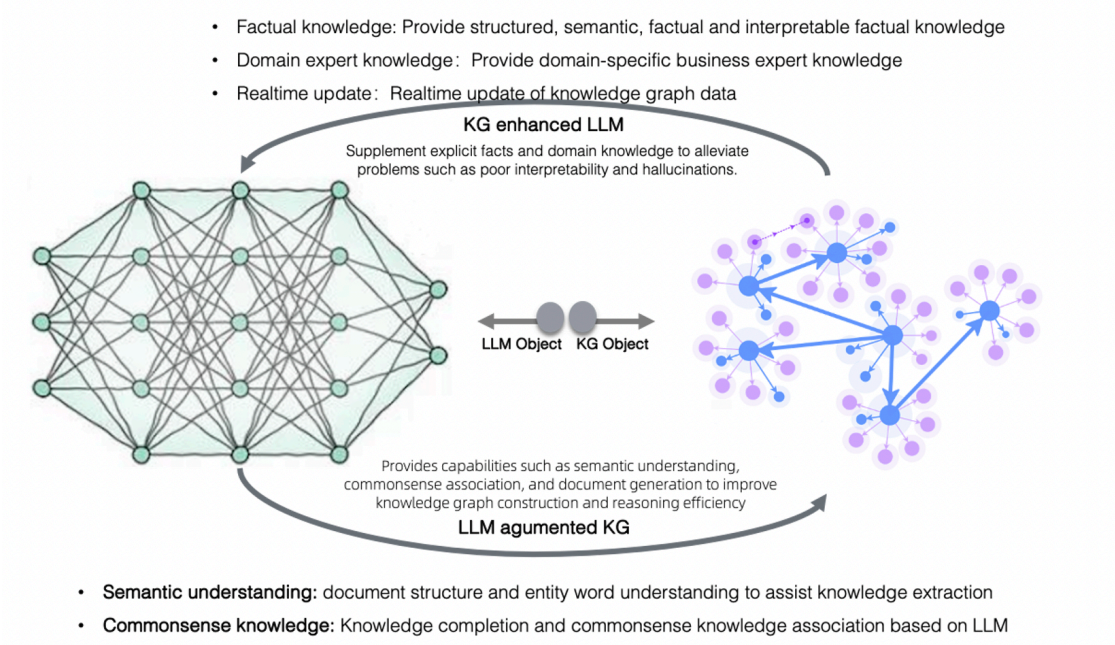

Figure 4: Mutual Drive between Large Models and Knowledge Graph

In general, taking the scenario of merchant management and risk control as an example, the algorithm tasks for LLM and KG applications can be categorized into three types: (1) LLM only: Due to the requirements for domain expertise and factual accuracy, there are currently no clearly applicable scenarios for LLM in the field of merchant management and risk control. (2) LLM + KG dual drive: This is mainly reflected in user interaction scenarios such as knowledge question-answering and report generation, as mentioned earlier, such as AI-powered phone call victim awakening and anti-money laundering intelligent trial report generation. Additionally, there are knowledge element extraction, entity linking, and other knowledge construction scenarios. The detailed description of the dual-drive of LLM and KG is presented in reference [10], including KG-enhanced LLM, LLM-enhanced KG, and the collaborative LLM+KG framework, as shown in Figure 4. (3) KG only: In decision-making, analysis, querying and knowledge mining scenarios that do not require complex language interaction and intent understanding, knowledge graph-based structured knowledge can be directly used for graph representation learning, rule reasoning, knowledge querying, and other tasks. By implementing the collaborative framework of LLM and KG, support is provided for cross-modal knowledge alignment, logic-guided knowledge reasoning, natural language knowledge querying, and more. This presents higher demands for unified representation of KG knowledge semantics and cross-scenario transferability of engine frameworks.

1.4 The Development of Knowledge Graph Technology System Needs to Keep Pace with the Times

The development of the knowledge graph technology does not completely match the expectations of its application in the new paradigm of managing new knowledge data and the dual drive of Large Language Models. The development of knowledge graph technology also needs to keep pace with the times. Specifically, the following issues exist:

Lack of Traditional KG Modeling Systems

Firstly, there is a lack of an industrial-level unified knowledge modeling framework. Despite the development of semantic-rich and loosely structured technologies such as Resource Description Framework (RDF) and Web Ontology Language (OWL) for many years, successful enterprise-level/commercial applications have not emerged. Instead, property graph with strong structure and weak semantics, such as Labeled Property Graph (LPG), has become the preferred choice for enterprise applications.

Secondly, there is a lack of a unified technical framework [2], resulting in poor cross-domain transferability. Due to the variety of tools and complex links, knowledge construction in each domain needs to start from scratch. In addition, there are also significant technical challenges in other aspects, as listed in Table 2.

Table 2: Technical Challenges Faced by Knowledge Graph in the New Paradigm.

Classification	Challenges	Description
1. Overall Framework	1.1 Industrial-Usable Knowledge Semantic Framework	Connecting big data with AI technology system, supporting the knowledge semantic framework that integrates factual and logical representations.
	1.2 Transferable Knowledge Graph Engine across Scenarios	The knowledge graph engine has transferability across scenarios, supporting the rapid incubation of new domain knowledge graphs.
2. Knowledge Graph Construction	2.1 Unified Knowledge Extraction Framework	Unified knowledge extraction based on unstructured/semi-structured data, ensuring throughput and performance under large-scale data.
	2.2 Unified Entity Linking Framework	Unified entity linking/standardization framework for knowledge elements, ensuring throughput, performance, and consistency under large-scale data.
3. Knowledge Graph Reasoning	3.1 Expert Rule Knowledge Representation	Constructing layered representation of logical dependencies between decision rules in the end-to-end business system.
	3.2 Rule-Guided Explainable Reasoning	Implementing effective rule injection and rule constraints, generating explainable inference results [7].

The goal of knowledge graph is to construct a machine-understandable and machine-reasonable digital world, achieving unified representation of knowledge semantics and hierarchical capability. This enables rapid construction of domain-specific knowledge graph and cross-scenario transferability, which is a fundamental core issue that must be addressed in the accelerated industrialization of knowledge graph.

Effective Construction of Digital Worlds

1.5 Industrial Knowledge Graph Engine Based on SPG

The Knowledge Graph Platform of Ant Group, supported by years of experience in the financial industry, has developed a semantic framework based on property graph called Semantic-enhanced Programmable Graph (SPG). It creatively integrates the structural nature of Labeled Property Graph (LPG) with the semantic nature of RDF, overcoming the challenges of industrial implementation faced by

RDF/OWL's semantic complexity while inheriting the advantages of the simplicity of LPG's structure and compatibility with big data systems.

SPG defines digital knowledge clearly

Firstly, SPG provides a clear definition of knowledge in the digital world. Knowledge is the accumulation of human exploration in the material and spiritual world, but how should machines perceive knowledge in the digital world? SPG defines the concept of knowledge in the digital world through formal description and objective facts. In conjunction with Figure 5, SPG provides a formal definition from three dimensions:

(1) Domain Type Structure Constraint: In the objective world, every entity (Thing) belongs to at least one type (Class), and the digital world follows the same principle. Based on SPG, the Domain Model Constrained (SPG DC) provides a constraint on the domain structure type, enabling automatic classification of knowledge subjects and hierarchical organization from dynamic spatiotemporal to static common knowledge.

Automatic classification and hierarchical organization

(2) Uniqueness of Instances within a Domain: In the objective world, there are no two identical entities, and the digital world should be the same. However, due to issues such as data duplication in the digital world, caused by multiple sources and data copying, data redundancy and repetition are common. SPG Evolving utilizes the capabilities of entity linking, concept standardization, and entity resolution provided by the SPG Programming (Knowledge Construction SDK) framework. It combines natural language processing (NLP) and deep learning algorithms to enhance the uniqueness level of different instances within a single type (Class), supporting continuous iteration and evolution of the domain knowledge graph.

Ensuring uniqueness in digital entities using SPG

(3) Logical Dependencies between Knowledge: In the objective world, everything is connected to other things, and there are no isolated entities, which holds true in the digital world as well. SPG Reasoning utilizes predicate semantics and logical rules to define dependencies and transitivity between knowledge, providing a programmable symbolic representation to facilitate machine understanding.

Main focus on the full lifecycle of knowledge management, knowledge construction, and knowledge reasoning

Connecting big data and AI technology systems to help machines better understand the world

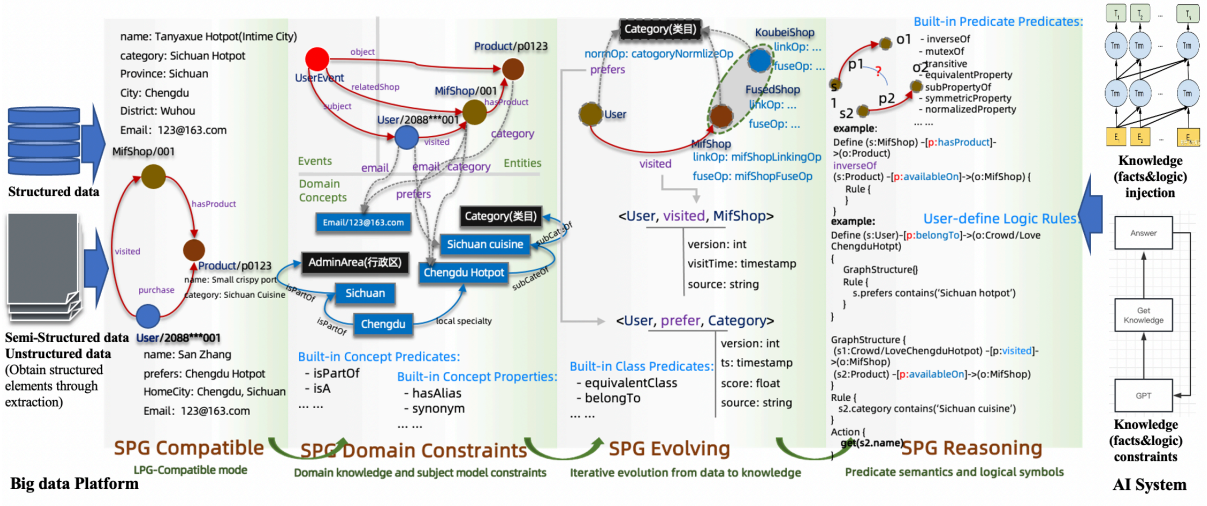

Figure 5: SPG Knowledge Semantic Framework

Main focus on the full lifecycle of knowledge management, knowledge construction, and knowledge reasoning

Knowledge engine connection with big data architecture

SPG fully integrates the advantages of LPG and is compatible with big data systems. The knowledge engine built on SPG seamlessly connects with the big data architecture during the knowledge construction phase, providing a framework of knowledge construction operator to facilitate the transformation from data to knowledge. During the storage phase, it can adapt to property graph to fully leverage their storage and computational capabilities. During the reasoning phase, it is formalized as KGDSL (Knowledge Graph Domain Specific Language), which provides a machine-understandable symbolic representation to support downstream rule reasoning, neural/symbolic fusion learning, KG Prompt collaborates with LLM for knowledge extraction/reasoning, and more. Additionally, through a layered architecture, the construction of a new domain knowledge graph only requires defining the schema, preparing the data, and developing construction/reasoning operators.

The knowledge graph technology is still in a period of rapid development and at a critical turning point in terms of technology. A unified technical framework can significantly lower the application threshold and promote the prosperity of the ecosystem. Therefore, this whitepaper focuses on the fundamental issue of enterprise-level knowledge management and deduces the full lifecycle of knowledge management, knowledge construction, and knowledge reasoning. The goal is to achieve an industrial-level, portable knowledge representation and engine framework. As mentioned earlier, Labeled Property Graph (LPG) has become the preferred choice for most enterprise knowledge modeling due to their unique compatibility with big data architectures. This whitepaper also derives the semantic capabilities required for enterprise-level knowledge management from practical business issues.

Main focus on the full lifecycle of knowledge management, knowledge construction, and knowledge reasoning

Chapter 2 Challenges of Knowledge Management base on Labeled Property Graph

LPG efficiency

In enterprise-level knowledge graph applications, as discussed in Chapter 1, Labeled Property Graph (LPG) is preferred for domain knowledge graph modeling due to its efficiency and compatibility with large data systems. it enables the rapid realization of business value. While the LPG-based knowledge construction has lower initial costs, as business rapidly grows and the volume of knowledge increases, the shortcomings of LPG become increasingly evident due to the lack of knowledge semantics and management capabilities. Firstly, the evolution of knowledge models becomes increasingly challenging, with schemas becoming more complex. Secondly, the flexibility of the node/edge model leads to redundant type creations and repetitive data preparation, making it difficult to maintain consistency and rationality among different relations / properties. Thirdly, the naive property/relation model is insufficient to depict the intrinsic semantics of entities and the semantic dependencies between them, resulting in significant obstacles to the continuous iteration and upgrading of knowledge graph projects. When the scale becomes unmanageable, new projects have to be created to rebuild schemas and graph data. Additionally, a large amount of hard-coding is required during the business application phase to achieve semantic parsing and alignment. This chapter combines two business cases, namely, risk mining knowledge graph and enterprise causal knowledge graph, to introduce the background and main pain points of business application and summarize the related issues. Next, in Chapters 3/4/5/6/7, we will attempt to propose solutions. Finally, in Chapter 9, we will provide two complete SPG-based solutions, aiming to leverage the advantages of LPG while avoiding its drawbacks, and providing efficient semantic modeling and knowledge management tools for enterprise-level knowledge graph applications.

"First challenge - continius integration and evolving data structures

"2nd Challenge - redudant type creation and consistency

3rd Challenge - The naive property-relation model fails to capture semantics and dependencies, hindering knowledge graph evolution

Too much coding work

2.1 Typical Case 1: Risk Mining Knowledge Graph

To achieve the primary business objectives of the risk mining knowledge graph, we aim to construct user-related risk profiles and associated networks for devices, media, transactions, and other relevant factors. Based on explicit or implicit associations, we identify individuals involved in the risky activities and implement risk control measures. Taking the app network risk prevention and control as an example, a particular app is found to be involved in risky activities such as gambling, pornography, and fraud. The following two objectives are expected to be achieved through the associated network of this risky app: (1) Identify the individuals behind the risks and apply corresponding risk control strategies based on the mining clues. (2) Discover other undetected risky apps and prevent the spread of the risks.

However, in practice, individuals involved in risky activities often disguise or hide their behaviors. They may use a large number of virtual devices, virtual IPs, or virtual identities, which are concealed among normal users. Therefore, this chapter will provide examples using a portion of the data listed in Tables 3, 4, 5, and 6 to illustrate the problems encountered in current LPG-based knowledge management. In the given example, the app (denoted as “** Entertainment”) should be identified as a gambling app developed by

"Wang Wu". "Li Si" should be recognized as the boss of the gambling company B, and "Zhang San" and "Li Si" are the same individual.

Table 3: Basic Information of User Entity

User Id	User Name	Phone Num	List of MAC addresses for owned devices	Owned certificate	Entity type
1	Wang Wu	154xxxx3456	06:8A:5F:2E:AB:85 06:8A:5F:2E:AB:86	Certificate 1	Person
2	Li Si	135xxxx5532	06:8A:5F:2E:A1:85		Person
3	Zhang San	135xxxx5532	06:8A:5F:2E:A1:85		Person
4	Company B	131xxxx3456		Certificate 2	Company
...	...	...	...	...	...

Table 4: Basic Information of the Shareholding Relation

Shareholder's Name	Company Being Held	Shareholding Percentage
Zhang San	Company A	100%
Company A	Company B	100%
...	...	...

Table 5: Basic Information of the Application Entity

App Id	App Name	List of MAC addresses of installed devices	Owned certificate	Is it a gambling application (based on user complaint labeling)
1	**Entertainment	06:8A:5F:2F:AB:85, 06:8A:5F:2E:AB:86	Certificate 1	yes
2	Fishing Master	06:8A:5F:2E:AB:85	Certificate 2	unknown
...	...	...	...	

Table 6: Transfer Relation

Transferring User	Receiving User	Transfer Amount
Li Si	Wang Wu	10000
...	...	....

There is a significant gap between the expression of data and the business expectations. This is manifested in the following ways:

- Difficulty in representing deep-level associations between different entities: It is challenging to directly derive the relations between applications and users, as well as the associations between different applications, from the data structure.
- Alignment of different characterizations of the same entity: Natural persons and users cannot be directly equated. For example, in this case, the users "Zhang San" and the one labeled as a gambling boss may refer to the same individual.

In business practice, although there may not be direct associations between applications and users, or between different applications, indirect associations can often be discovered through intermediaries such as devices or certificates. Similarly, relations between users can be explored through methods like shared

phones or devices. To cope with such complex network relations, the knowledge graph typically evolves as follows:

Step 1: Transforming tabular data into property graph representation.

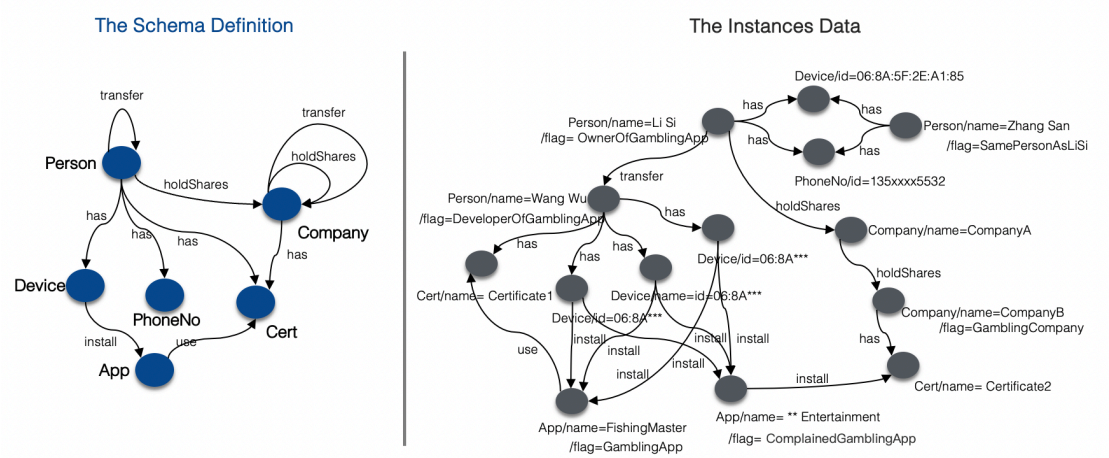

Figure 6: Building the Knowledge Graph by Directly Mapping Tabular Data to Property Graph

Figure 6 illustrates the mapping of tabular data to the data structure of the knowledge graph. At this stage, it is possible to derive the relations between risky applications and risky individuals based on multi-hop relations. However, further analysis and judgment by business experts are still required to directly achieve the business objectives, as depicted in Figure 7. The textual structure of entity instances in both Figure 6 and Figure 7 is as follows: Type / PropertyName = PropertyValue.

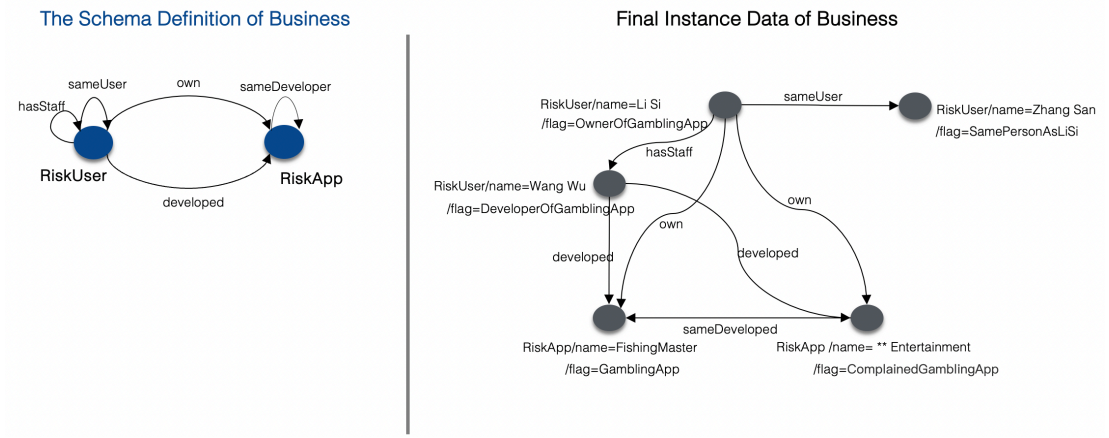

Figure 7: Knowledge Graph Structure Derived through Implicit Inference to Meet Business Expectations

The data structure of the knowledge graph required by the business is typically different from the original graph data. The original graph data represents objective basic data, while the data required by the business is based on the mined associations derived from the objective data. These associations need to be re-integrated into the original data. To uncover these implicit relations, business experts formulate a series of rules, such as rules for determining the same user, rules for user-app ownership, and rules for app developer relations. For example, if two users use the same phone number or device, they are considered to have a “same phone” or “same device” relation. If a user has a controlling relation with a legal entity, the app released by that legal entity is considered to be owned by that user. If a user has multiple devices that

have the same app installed, the user is considered to be the developer of that app. By applying these rules and performing rule calculations using external big data systems, the required data structure for the knowledge graph is obtained, including the addition of new types and relations. The original basic information definitions are also retained to support better decision-making and risk control in the business domain.

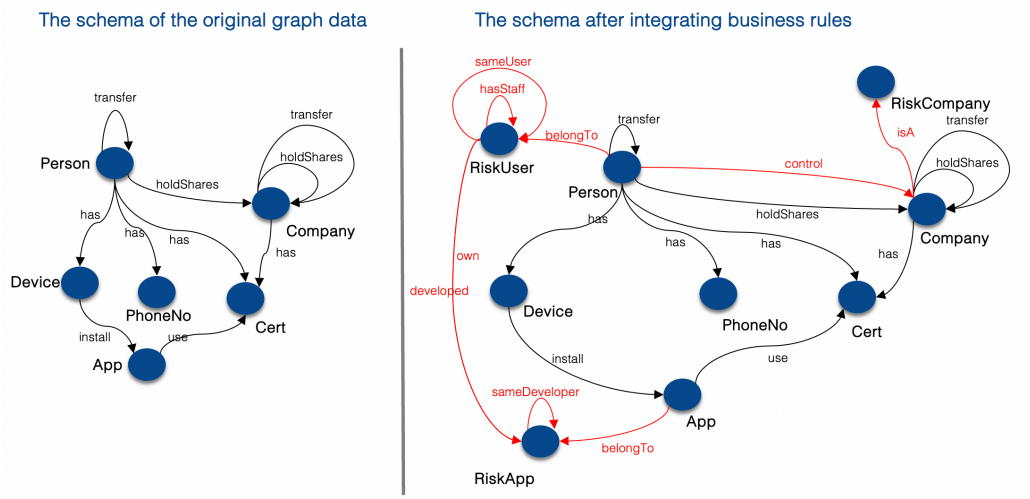

Figure 8: Schema Differences after Incorporating Business Rules

The above example demonstrates a portion of the redundant creations that occur during the business decision-making process. In knowledge graph management, extracting complex implicit associations from basic facts is a fundamental requirement. However, we need to address how to avoid continuous schema expansion caused by the refinement of business objectives and ensure the logical consistency between rule calculations and the underlying facts. These are fundamental issues that knowledge management must address.

Avoid schema expansion and ensure rule-fact consistency.

Main challenges list

2.2 Challenges in Applying LPG to the Risk Mining Knowledge Graph

High construction cost

- **The independent data preparation for nodes and edges significantly increases the construction cost of the knowledge graph.** To construct the required entities and relations for the risk mining knowledge graph, data preparation for nodes and edges is required, which is much larger than the original four tables.

Redundant data preparation

- **The difficulty in directly reusing different knowledge graphs leads to redundant data preparation.** In this business case, it is necessary to construct knowledge graph data for fund transfers and equity structures. Typically, these data already exist as foundational data in other knowledge graphs.

Inconsistencies caused by logical dependencies between entities and elements

- **Inconsistencies caused by logical dependencies between entities and elements.** In the business modeling, the new types and relations in Figures 7 and 8 are derived from the existing data in Figure 6. When the underlying data changes, such derived data must be synchronized, or else inconsistencies in the knowledge graph data will occur.

Continuous expansion of the knowledge graph structure

- **Continuous expansion of the knowledge graph structure due to the migration and changes of business objectives.** In this case, implicit associations through intermediaries are used to identify risky users behind the applications. However, as risky activities evolve rapidly, there will be frequent updates and creation of different entity and relation types. The size of the knowledge graph schema and instances will continue to expand, making it difficult to manage.

Difficult to manage

Therefore, when constructing a knowledge graph, these challenges need to be considered, and appropriate measures should be taken to optimize the data transformation process, improve the reusability of knowledge graph, and design the schema to support the expression of logical relations between knowledge, thereby enhancing the efficiency of business semantic migration. This helps us build a more efficient, reliable, and maintainable knowledge graph system.

2.3 Typical Case 2: Enterprise Causal Knowledge Graph

The knowledge management of an enterprise causal knowledge graph focuses more on depicting the logical relations of causality, conditionality, hierarchy, and sequentiality between events. Therefore, the foundation of an enterprise causal knowledge graph is events. In practical applications, it generally evolves from the application of events and graphs to the causal knowledge graph, this evolves capturing production and operational events related to the enterprises, extracting key elements of events, establishing the linkage between event elements and internal enterprise/industry chain knowledge graph, and constructing causal logical chains between risk events and enterprise/industry chain knowledge graph. This allows for quick linkage to internal warnings or risk management when external risk events are identified. When a financial event occurs, we need to infer the event based on basic facts in order to try to obtain answers to the following questions:

- The nature and impact of the event.
- The entities involved and the impact on related entities.
- Whether the associated entities will further generate other impacts and how they will be affected.

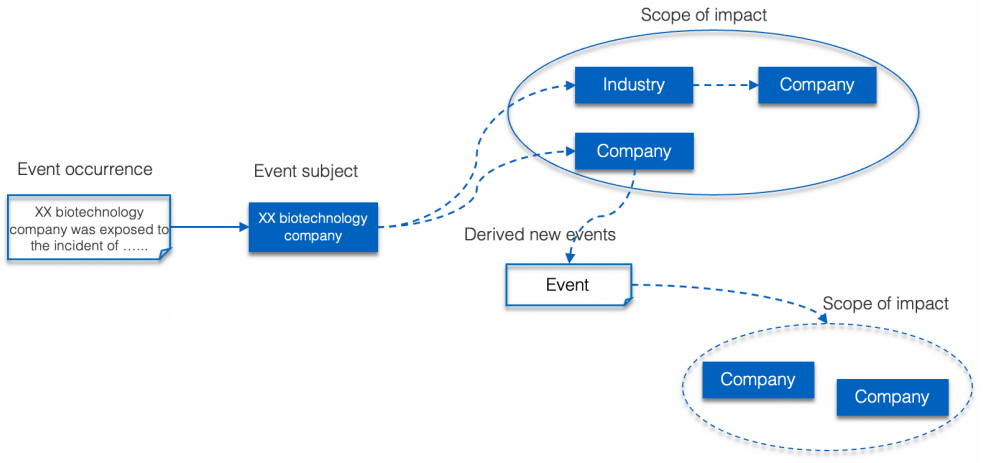

Figure 9: Illustration of Event Impact Propagatio

For example, let's consider an event where a biotech company is exposed for producing fertilizers with severe heavy metal contamination. The extent of the event's significance needs to be further analyzed. When analyzing a specific event, analysts need to repeatedly query and gain insights from basic factual knowledge, based on their understanding of the event and combining it with common knowledge to draw conclusions about the event's impacts. However, various reasoning logic and data are often fragmented and dispersed, requiring effective integration and connection. Therefore, there are many unresolved issues in the application of knowledge graphs in understanding causality. In the case of this event, it is necessary to analyze the impacts on which entities in the enterprise network, the paths and degrees of impact, and whether the impact on other entities will give rise to new events, thus further expanding the scope of influence.

Problem 1: Complex and Diverse Event Classification, Predefined Event Taxonomies Cannot Fully Cover Real-World Application Scenarios.

The traditional approach is to define multi-level event types and construct an event type tree through the expertise of business professionals. This involves defining, describing, and classifying events based on the understanding of equity markets, fixed income markets, and macroeconomic changes. Different events are delineated by their boundaries. Additionally, events can be defined as “changes” in the financial market, typically associated with a series of financial indicators. A set of predefined labels in the form of an “event tree” is created by business experts, and different financial event propagation networks are built based on this tree and historical data. However, such an approach often fails to meet the needs of real-world financial markets, primarily due to the following reasons:

1. Different interpretations of event types. Due to different backgrounds, business experts may have diverse understandings of event trees, leading to inconsistencies and variations in their definitions of the same events. The boundaries between different event types and events may not be clear.
2. Static event trees cannot accommodate the dynamic development of the financial market and the emergence of new financial event types. Especially after the 2008 financial crisis, the global economy entered a new normal, and the domestic economy has shown new features in recent years. For example, the COVID-19 pandemic has had a significant impact on the global economy and various industries. However, in the existing event trees, it is generally classified under the category of “major health security” events, and many business analyses often compare it to the SARS outbreak in 2003 to predict future impacts. However, although both are “major health security” events, they differ significantly in terms of impact time, scope, and other aspects.

In conclusion, due to the complexity of financial events, relying solely on a group of business experts to predefine events cannot fully cover real-world application scenarios. We need a system to dynamically generate derived financial event taxonomies.

Problem 2: Multiple Interrelationships Exist Between Events, Such as Causality and Sequence, which often Require Dynamic Connection through Entity Networks, Demanding Strong Descriptive Capability.

Due to the complexity of the financial event network, the impact to other events after the occurrence of a particular event may vary. This often depends on the differences in entities and relations behind the different events, which determine the different directions of impact for each event.

For example, if Company A's stock price rises due to its expansion of production capacity, and the capital market has a positive outlook on its future development, whether the stock price of its competitor, Company B, will rise or fall will often depend on various factors, such as market demand, the scale of capacity expansion, and the relative market share between the companies and their competitors. Suppose Company A is a semiconductor manufacturer and decides to expand its production capacity. For its competitor, Company B, this may be good news. If there is strong global demand for semiconductors and a tight supply, A's capacity expansion may help alleviate this supply-demand imbalance and stabilize the entire market. In this case, as the market environment improves, the competitor B may also benefit from it. The logic in this case is: if the demand for the entire industry exceeds supply, any action that increases supply may have a positive impact on the entire industry as it helps maintain market stability and prevent price surges or other factors that may lead to market instability. On the other hand, if Company A is an automobile manufacturer and decides to expand its production capacity, it may have a negative impact on its competitor, Company B. In this case, if market demand does not grow, A's capacity expansion may lead to market oversupply, triggering price competition. Therefore, for competitor B, this may result in lower sales volume and profits, making it a negative news. The logic in this case is: if the supply growth in an industry exceeds demand, it will lead to oversupply, potentially triggering price competition, which in turn affects the profit levels of all firms.

In conclusion, due to the complexity of the financial event network, when describing the transmission relations between different events, it is necessary to dynamically link them with the relevant entity network and build a strong descriptive capability based on it.

Problem 3: How to better describe and analyze the propagation of event impacts?

Propagation in entity networks and event networks

Due to the complexity of financial event inference, it is necessary to analyze the propagation effects of events from two perspectives: the propagation in entity networks and the propagation in event networks. Taking the example of “Company A announces bankruptcy/bond default”, we can analyze the event's propagation effects from these two angles:

1. Propagation in entity networks: Company A's bankruptcy will directly impact its shareholders, especially major shareholders, whose financial conditions may be affected, thus further influencing their investments in other companies. Additionally, Company A's competitors may benefit from its bankruptcy, potentially gaining market share. Similarly, suppliers and creditors of

Company A may suffer economic losses due to the bankruptcy. These impacts will propagate in the entity network, affecting other relevant entities.

2. Propagation in event networks: Company A's bankruptcy may serve as a cautionary example for other companies, preventing similar occurrences. For example, it may enhance risk awareness in related industries or markets, prompting companies with issues in financial management and risk control to learn from it and make necessary improvements. The impact of this event will propagate in the event network, forming new events and affecting other entities.

These two propagation processes are not isolated but intertwined. For example, Company A's bankruptcy may draw the attention of its competitors and influence their decision-making, thereby triggering new events in the entity network. Simultaneously, this new event may also become a new node in the event network, further influencing the behavior of other companies.

Problem 4: The process of financial event inference is not sufficient solely based on relation network, it often requires the use of extensive external data for analysis.

In 2019, a dam collapse incident occurred at the Vale of Brazil, resulting in an increase in iron ore prices, which in turn led to a rise in steel production costs. Within the entire chain of event impacts, some companies involved in industry competition benefited from this incident, as their profits rising. However, it also had a negative impact on the downstream of the industry chain, as rising costs led to a decrease in profits.

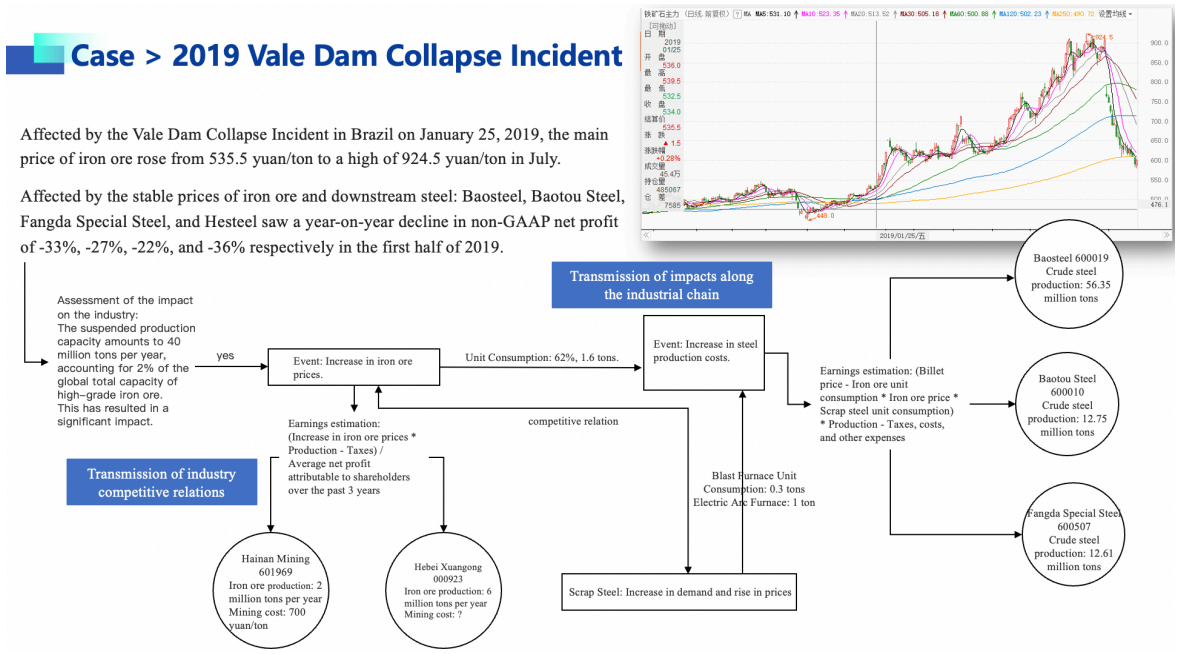

Figure 10: Transmission Diagram of the Impact Chain of the Vale Dam Collapse Incident

The entire iron ore industry chain starts with iron ore extraction, and Vale S.A. is an important participant in the global mining industry, with its operations significantly impacting global iron ore supply and prices. It is precisely because of the company's importance in the global iron ore industry chain that the dam collapse incident led to a global increase in iron ore prices.

China, as a major infrastructure country and the largest consumer of iron ore globally, heavily relies on the global iron ore market. Therefore, a major incident in a Brazilian company can lead to an increase in raw material prices in the iron ore industry chain and successfully transmit the event to the domestic capital market. On the other hand, after the import of iron ore, it goes through the process of smelting, refining in a converter, and casting to produce pig iron. Pig iron is then further processed into various steel products, such as long products (rebars, wire rods) and flat products (hot-rolled coils, cold-rolled coils), which are used in industries such as automobiles, appliances, and shipbuilding. Additionally, there are also pipe products (seamless steel pipes, welded steel pipes), special steel, high-strength steel, and other different products. Various Chinese listed companies are involved in these upstream and downstream segments of the industry chain, such as Baosteel, Baotou Steel, and Fangda Special Steel. The specific transmission logic and impact need to be analyzed in conjunction with the details of the arguments, including the following aspects:

1. Whether the company engages in hedging in the derivatives market, and the value of hedging transactions.
2. The market share of the company's products and the competition landscape in the segmented industry. Generally, the competition landscape for special steel is considered better than ordinary steel.
3. Whether the company has the ability to transfer upstream production pressures to downstream, and whether there are upstream alternatives domestically.

Only by dissecting the above arguments into finer granularity and introducing relevant external data can a complete transmission network be constructed.

2.4 Challenges in Applying LPG to the Enterprise Causal Knowledge Graph

Reasoning with LPG

In general, the analysis of event impacts is based on the analyst's understanding of the event, repeatedly querying and gaining insights from basic factual knowledge, and combining it with common-sense knowledge to draw conclusions about the event's impact. It can be seen that the entire process of event inference is outside of the knowledge graph, as basic factual knowledge lacks common-sense and reasoning logic, and a pure event knowledge graph cannot express the context of the event. In practical applications, in order to complete event inference, various logics have to be scattered in various places outside of the knowledge graph, and reasoning is performed through various external plugins. Such methods inevitably bring many application issues to enterprise causal knowledge graph:

Events to complex. Schema-constrained property graph

- **Contradiction between the static nature of predefined schemas and the dynamic nature of events.** Events are often complex and diverse, and if a strongly schema-constrained property graph is used, it is generally impossible to predefine events. It can only be tailored to specific scenarios. If a schema-free property graph is used, the overly lenient mode will result in increasing data management and usage costs.

Missing context! No expert rules for event analysis, no entire context of event propagation

- **Inability to express the entire event transmission context.** Since the knowledge graph only contains basic facts without the definition and transmission relations of the events, it is impossible to establish expert rules for event analysis, let alone represent the entire context of event propagation. To express the event context, it is necessary not only to illustrate the evolutionary process of events over time but also to combine abstract entities to express the relevance of events within the event domain through abstract levels.

Validation issue and reasoning problem when Separation of the knowledge graph and reasoning logic

- **Separation of the knowledge graph and reasoning logic makes it difficult to evaluate the correctness of the reasoning logic and hinders the reuse of reasoning logic.** Due to the separation of schema and reasoning logic, when maintaining basic factual data, it is impossible to assess the impact on the correctness of external reasoning logic. For example, changes in data such as property names or deleted relations may cause the failure of reasoning logic that exists outside of the knowledge graph. Such problems are unavoidable in traditional event knowledge graph. Furthermore, reasoning logic may consist of a combination of query statements and scripts, and these contents may be managed in analysts' local storage, making it difficult to reuse reasoning logic that is highly generic.

black box reasoning - Poor interpretability of the conclusions derived from event propagation reasoning

- **Poor interpretability of the conclusions derived from event propagation reasoning.** Since the external reasoning logic may be a combination of multiple query statements and scripts, it is not possible to visually observe the deductive process from the cause to the result when the entities affected by the event are calculated. In this case, interpretability becomes a black box, and understanding the query statements and scripts is necessary to comprehend the reasoning logic.

2.5 Complexity and heterogeneity caused by the coupling of structural definition and semantic representation in knowledge modeling

RDF/OWL aproach

RDF/OWL is a syntax-level representation framework, leading to a higher learning cost. In traditional ontology modeling in knowledge engineering, a classification system needs to be defined through description logic syntax.

LPG Syntax and Structure

The labeled property graph (LPG), has simple syntax elements but only represents the data structure. None of the above methods address the problem of “design patterns” themselves. In the process of practical business implementation, the coupling of data structure definition and knowledge semantic ontology design in the modeling process leads to difficulties in decision-making. The schema design of domain knowledge graph is subjective, where entities of the same type are defined differently due to naming and granularity differences. Heterogeneity issues caused by different schema definitions are prevalent, hindering the dissemination and reuse of knowledge, and further exacerbating knowledge inconsistency.

Schema Design and Heterogeneity

2.5.1 Repetitive construction issue caused by differences in entity type granularity due to different business goals

In the application of the risk mining knowledge graph, there is a need to classify the "Person" entity and determine whether they are involved in risky activities. The risky personnel can be further divided into categories such as gambling individuals, bookmakers, money launderers, and so on, as shown in Figure 11.

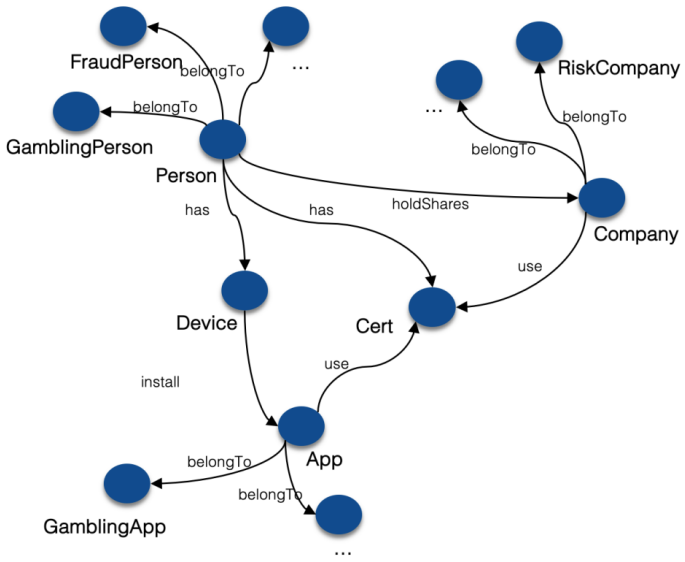

Figure 11: Entity granularity expansion in the modeling process of the risk mining knowledge graph

Even within the same knowledge graph project, different internal demands may lead to the creation of new entity types. The requirement for an entity to have multiple types is often resolved by creating redundant new types. This results in increasingly complex schemas. At a certain stage of business evolution, there may be a need to start from scratch and redesign the knowledge graph. Taking the risk mining knowledge graph as an example:

- **Demand for analyzing different apps:** Black industry groups produce a large number of apps through batch repackaging, similar to an app factory. It is necessary to classify and refine the types of apps to address different risk control strategies.
- **Demand for on-demand refinement of entity types:** When investigating black industry groups, some individuals may be associated with bookmakers, fraud activities, and other specific roles. This leads to the creation of additional types such as "GamblingPerson" and "FraudPerson". As the business evolves, the classification of entities continues to become more refined.

These issues are often strongly correlated with specific business scenarios, and change as the business evolves, and different scenarios arise. From the data management perspective, these apps or persons may use the same or similar data structures. However, from the business logic perspective, there is a need for semantic-level type differentiation. The mixture of schema/ontology modeling from different perspectives

leads to continuously increasing costs for user understanding and maintenance. The redundant construction of entity types also increases the preparation and maintenance costs of data tables.

2.5.2 Different knowledge graph defining the same entity differently

Taking fund flow as an example (cross knowledge graph), as shown in Figure 12.

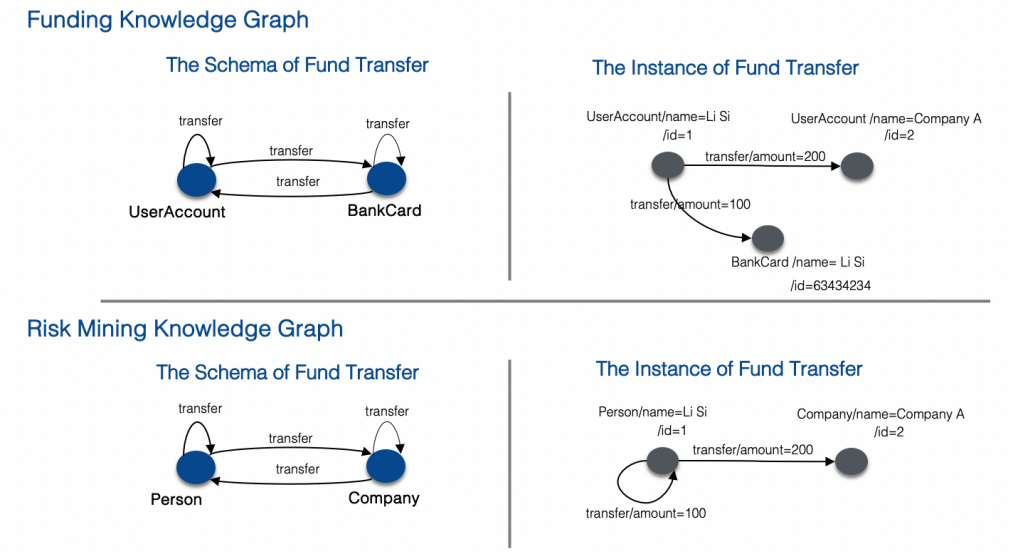

Figure 12: Illustration of cross knowledge graph fusion

In the risk mining knowledge graph, we focus on the transactional relations between users and companies to identify the masterminds behind illegal activities. In the fund knowledge graph, our focus is on analyzing the flow of funds. Therefore, we deploy tracking and control strategies for the involved financial products and treat them as more granular entity types. In both of these scenarios, we deal with transactional relations to meet our respective business requirements. However, there are two problems:

- Different businesses handling the same data in a similar way result in the inability to consolidate common requirements and share accumulated business experiences. Each newly business scenario needs to start from scratch to prepare the data, increasing the threshold for business usage.
- Knowledge sharing across knowledge graphs. For example, the “BankCard” entity exists in the fund knowledge graph. However, it cannot be securely used for business needs such as anti-money laundering or anti-fraud.

2.5.3 Difficulties of making the choice between defining as properties or relations due to construction costs

In the labeled property graph model, each entity and relation type require independent data preparation. With M types of entities and N types of relations, due to differences in property quantities, M + N message structures or data tables need to be prepared to complete the knowledge graph construction. Leading to high data preparation costs. As the demand for knowledge graph-based analysis increases, this cost continues to

escalate. All entities and relations defined in the labeled property graph require separate data preparation, forcing businesses to balance between current simple applications and future scalability. When properties are directly constructed as relations, it increases the complexity of simple application usage, such as the lack of property filtering.

Consider a simple question: connecting devices that use the same Wi-Fi. As shown in Figure 13, our usual approach is as follows: (1) Construct entity types for Device and Wi-Fi, and a relation type "Device -[useWifi]-> Wi-Fi". (2) Prepare data separately for the entities and relations mentioned above, and generate unique entity IDs for Wi-Fi.

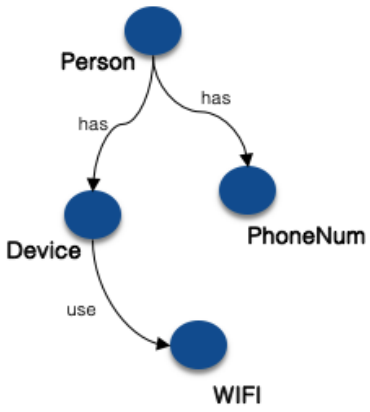

Figure 13: Connecting devices using the same Wi-Fi

This significantly increases the complexity of data preparation, as each entity and relation type require separate data preparation. In extreme cases, if each type requires its own data table, the number of tables increases from 2 to 6. This results in a larger workload for data cleansing. Assuming there are "m" entity tables, with an average of "n" property columns per table that need to be transformed into relation, we would need to generate a total of "m*(2*n+1)" tables. This raises the threshold for user usage.

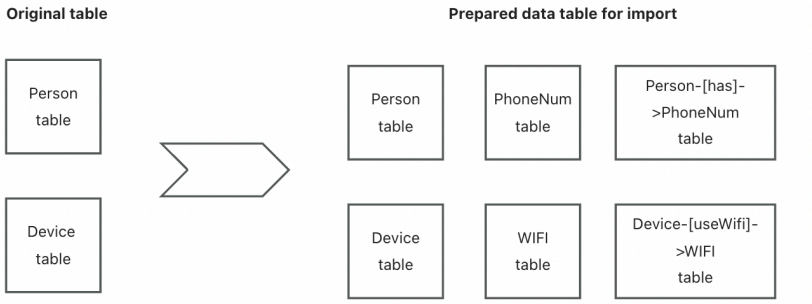

Figure 14: Cost escalation due to entity/relation data preparation

2.6 Insufficient expressive power for representing diverse and heterogeneous domain knowledge

During the implementation of knowledge graph in the financial domain, there is a need for heterogeneous representations of temporal and spatial aspects, such as user behavior, industry events, macro

events, and so on. For example, the enterprise causal knowledge graph needs to express the temporal and spatial relations of individual events as well as model simple or complex logical relations such as causality, succession, co-occurrence, and structure. It is difficult to achieve lossless expression using RDF/OWL, and although the introduction of the HyperGraph [12] can alleviate some of these issues, it does not integrate well with the RDF/OWL system, thereby increasing the cost of user application and understanding.

2.6.1 Representation issues of temporal and spatial structures in events

Representing the multi-element structure of events is also a problem of lossless representation, similar to the HyperGraph. It expresses the temporal relations of various elements in a multi-element structure, where events are temporary associations formed by these elements due to certain behaviors. Once the behavior ends, the association disappears, as shown in Figure 15.

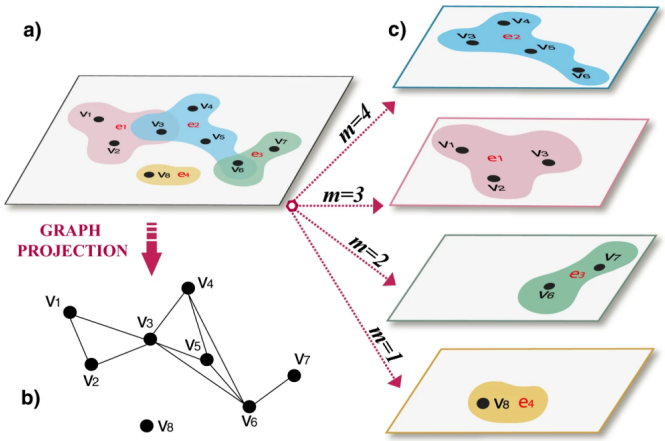

Figure 15: Representation of the HyperGraph [13]

The representation method of RDF-Star [14] extends the modeling capabilities of RDF for such scenarios, and in 2022, the W3C established the RDF-Star Working Group to further enhance RDF. Taking the application of the enterprise causal knowledge graph as an example, the simple structure of a safety production event in a company is represented as shown in Figure 16.

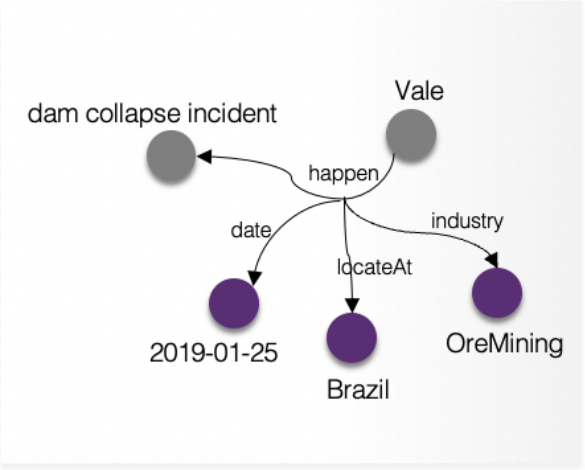

Figure 16: Extension of multi-element relations based on RDF-Star triples

In the representation format of "<s, p, o>" triples, the first step is to extend it with a time element to "<s, p, t, o>" in order to further represent temporal constraints, as shown in the example "<Company, Occurrence, OccurrenceTime, SafetyRiskEvent>". However, event associations are often complex combinations of multiple elements. Breaking down the different aspects of an event into independent elements is necessary, as shown in Figure 16. In the construction of domain knowledge graph based on labeled property graph, which has been developed for many years, there is no solution for how RDF-Star can be applied. We need the representation capabilities of a spatiotemporal event hypergraph based on labeled property graph in order to build the event representation and reasoning capabilities required for enterprise causal knowledge graph.

2.6.2 Problems with causal succession, composition, structure, and logical dependencies

The enterprise causal knowledge graph have an ontology layer, which means that there are not only horizontal associations between events and entities, but also vertical associations from specific to general or from general to specific. Horizontal associations involve roaming, association, and analogy, while vertical associations involve induction, deduction, and evolution. Therefore, the corresponding architecture should carefully consider these situations when making decisions. In terms of the definition and instantiation layer, there are four components: abstract entities, concrete entities, abstract events, and concrete events. They are physically connected as a single graph, but logically can be divided into entity domain and event domain horizontally, and ontology domain and instance domain vertically, forming a so-called four-quadrant architecture [15], as shown in Figure 17.

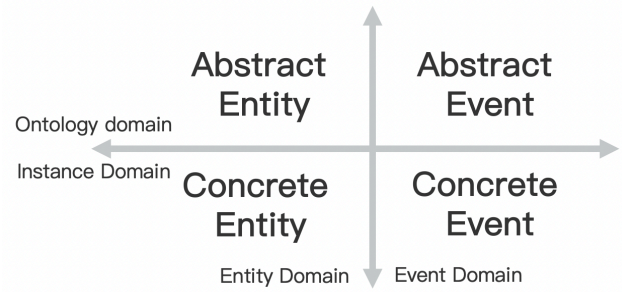

Figure 17: Four-quadrant architecture for enterprise causal knowledge graph

The main challenge is the coexistence of event models and causal models. Common event graphs only represent the relational connections between bare events without their arguments. However, in enterprise-level applications, event instances contain richer information. For example, in an enterprise risk event, it may include information about the entities involved, the industry involved, and whether the production is halted. These additional details can complement the bare events, and both aspects are mutually beneficial. We need the coexistence of event models and causal models. The event model represents a spatiotemporal multi-element hypergraph structure, while the causal layer involves reasoning about causality, succession, and logical combinations. For example, when land prices increase, it leads to an increase in fiscal revenue. The combination of "Land prices in Province A increase" is a binary relation between an administrative entity and an abstract event, and should also lead to the deduction of "Fiscal revenue in Province A increases".

The increase in fiscal revenue then cascades down the impact chain. Similarly, there are expressions of hierarchy between arguments, such as “Interest rate increase” and “Yen interest rate increase”. Ultimately, this can form a specific path of “Event → Abstract entity (superior) → Abstract entity (inferior) / Specific entity → Event”.

2.7 Consistency and propagative reasoning issues caused by logical dependencies between knowledge

In the domain knowledge graph, there are implicit logical dependencies between different properties and relations. Applications in financial risk control, for example, require the establishment of logical dependencies between property elements to construct the automatic propagation capability of risks. In the LPG model, it is necessary to prepare all relations and properties. However, inconsistencies may arise due to factors such as computational timeliness and logical correctness. These issues become more apparent when dealing with logical dependencies between multiple elements, increasing the complexity of pre-computation/construction.

2.7.1 Inconsistency and redundancy construction issues caused by logical dependencies in data

Figure 18 provides a simple example of the issue of properties errors caused by implicit logical linkage across entities in the risk mining knowledge graph.

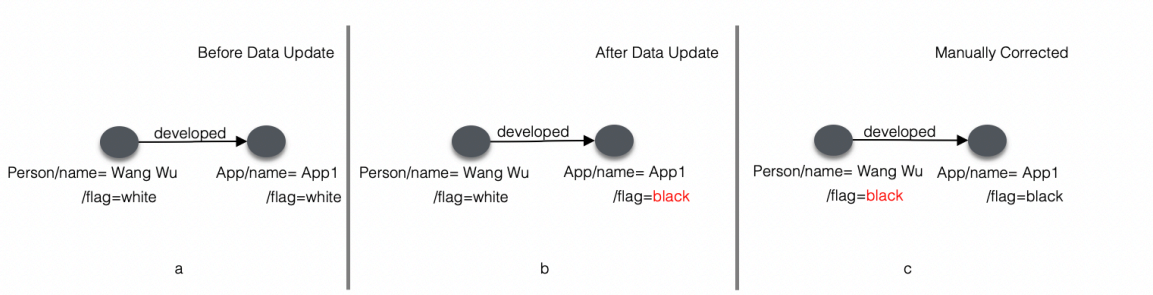

Figure 18: Implicit logical association

Looking from the perspective of discovering risky activities, a rule is defined: “When an app released by Company A is flagged as black, Company A should also be flagged as black”, as shown in Figure 18b. Both the company and the app have a “flag” property. However, when App A is reported and identified as black, Company A is still marked as white. In this case, data inconsistency occurs. It requires waiting for the completion of external system calculations before updating, as shown in Figure 18c, or manual intervention to address the issue. Incorrect data or delayed updates can result in incorrect conclusions and the knowledge graph being unavailable during the correction period.

2.7.2 Problem of obstructed risk propagation/transmission due to logical dependency transfer

In Section 2.3, the dam collapse incident at Vale in 2019 resulted in a rise in raw material prices, subsequently causing an increase in production costs for downstream companies, ultimately leading to a decline in their profits. From a causal perspective, this event originated from a production accident at a company and propagated through the industry chain, triggering financial risks for downstream companies. During the propagation process, it is not a simple diffusion of relations but rather a causal transmission with logical dependencies. Moreover, each instance in the propagation chain still retains the key elements of the initial event. These complexities are challenging to capture in an event knowledge graph based on foundational facts, as the presence of logical dependencies can hinder the propagation of events. To construct the propagation of event risks, it is necessary to consider the triggering mechanisms, the transmission of event impacts, and the rules of transmission.

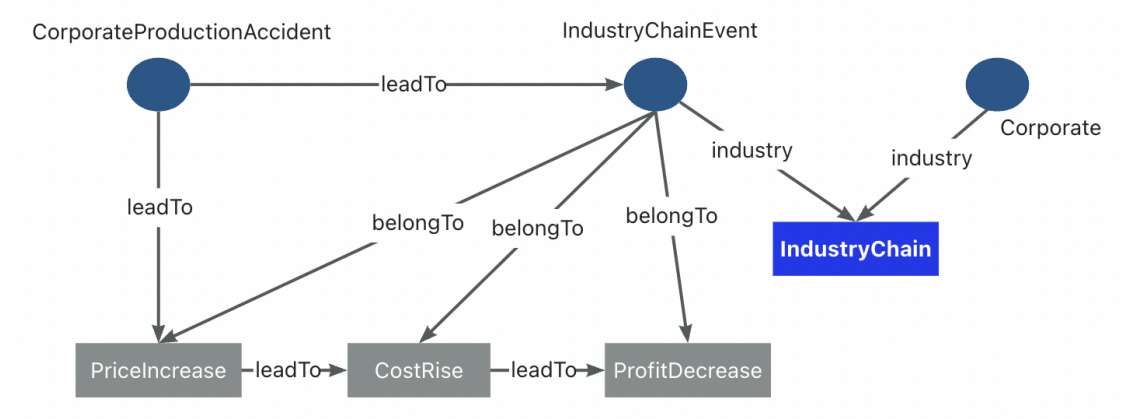

Figure 19: Impact propagation of events between instances

- **Event triggering mechanism:** Structured event elements are obtained based on the extraction of external relevant information or monitoring of the key data changes, resulting in event instances. Based on specific event instances, corresponding event propagation rules are triggered.
- **Event impact propagation:** Event impacts are propagated directly along relations. For example, in the enterprise causal knowledge graph, the impact of a company's safety production accident is propagated to the industry which it belongs to, based on the industrial characteristics of the occurrence subject in the event instance. The relation transmission of events can express which conditions allow an event to be transmitted to another target event. These conditions can utilize various properties of entities and relations obtained through query of associated subgraphs in the transmission path, such as determining whether the occurrence subject is a listed company, the industry of the company, and downstream industries.

During the propagation process, the logical judgment can reference the relevant properties of all preceding entities/relations in the current judgment condition's position. As shown in Figure 20, the “Price increase” event needs to reference the industry property of the subject in the “Vale dam collapse incident”,

the “Cost increase” event needs to reference the downstream industry property of the “Price increase” event, and the “Profit decrease” event needs to reference the industry property of the “Cost increase” event.

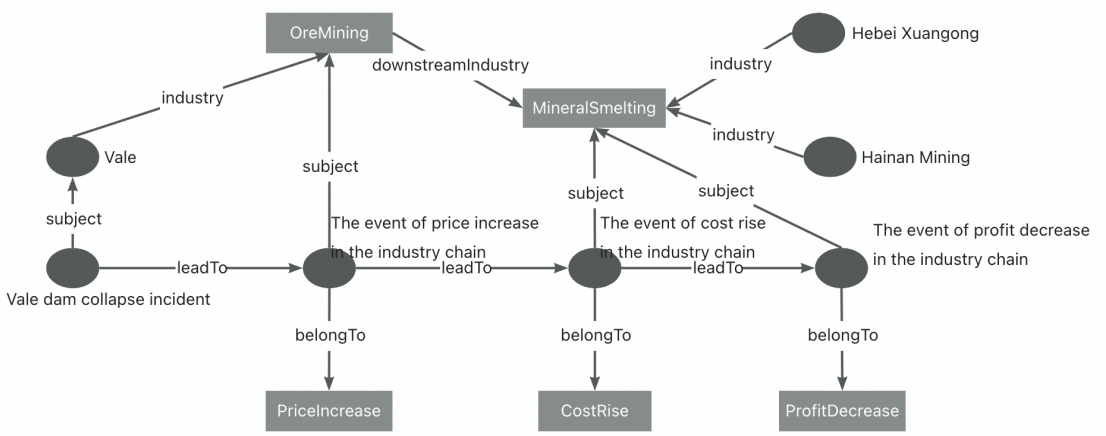

Figure 20: Concept induction and impact propagation based on rules

2.8 Graph Construction and Evolution Problems for Incomplete Data Sets

The construction of enterprise-level knowledge graph is often based on incomplete data sets, with constantly changing sources and construction strategies. Continuous iterations are needed to improve coverage, accuracy, and reduce conflicts and errors. This incompleteness typically includes two aspects: the heterogeneity of data sources and the heterogeneity across multiple knowledge graphs. The heterogeneity of data sources manifests as different instances and properties of the same entity type coming from different data sources. It requires addressing disambiguation, alignment, and fusion of different data sources, as well as evaluating and selecting data sources based on different confidence strategies to achieve traceability and quantifiability. The heterogeneity across multiple knowledge graphs arises from the presence of duplicate definitions of the same entity type in different domain knowledge graphs, which need to be merged and linked across knowledge graph based on business domain requirements and data differences.

2.8.1 Reliable Fusion and Trustable Traceability of Heterogeneous Data from Multiple Sources in Graph Construction

In enterprise knowledge graph applications, different properties and relations of the same entity type may come from different data sources. The common practice to construct entities based on heterogeneous data sources is entity linking and entity resolution. Entity linking involves finding an accurate and unique entity ID for each data update, while entity resolution merges the updated properties and achieves the consolidation of the properties and relations. As shown in Figure 21, in the enterprise causal knowledge graph, the construction process of the enterprise entities, involves the merging of various data sources, such as company announcement extraction, basic business information, and court announcements.

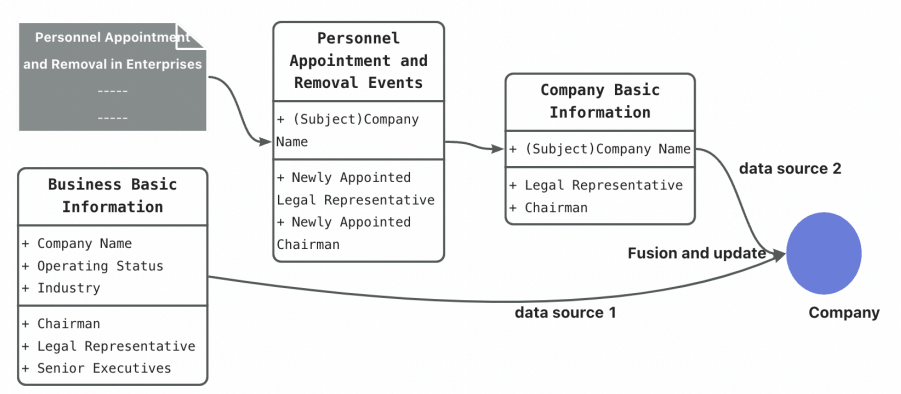

Figure 21: Entity updates based on heterogeneous data sources

The definition of an enterprise entity is generally represented as "<Company, legalPerson, String>". In practical applications, when conflicts occur in property values, decisions on how to retain or update them need to be made based on dimensions such as source type (sourceType) and algorithm prediction scores (score). For example, the confidence level of basic business information is the highest, so it needs to be unconditionally overridden. However, the timeliness of updates for business information and company announcements may not be consistent. There may be cases where company announcements have been captured but the business information has not been synchronized. In such cases, secondary descriptive information needs to be preserved on property elements. This can be formally represented as: "<Company, legalPerson, String>" as p, with the addition of "p.sourceType", "p.score", and the recording of the coverage rules for p in the schema. For example, p.fuseRule = "sourceType == 'business information'; score > p.score".

2.8.2 Entity Alignment, Real-time Updates, and Fusion/Traceability Problems in Cross-Graph Fusion

The problem of cross-graph fusion is similar to 2.5.2. When merging user entities from the risk mining knowledge graph and the fund knowledge graph, it is necessary to determine how to preserve the properties and relations in the new "FuseEntityType".

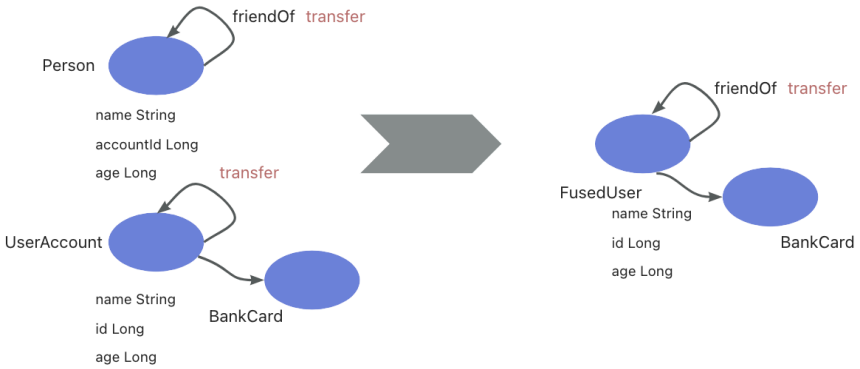

Figure 22: Stable fusion and traceable update problems in cross-graph fusion

In order to ensure that updates to properties/relations of "Person" and "UserAccount" can trigger timely updates to "FusedUser", while forwardly ensuring the stability of the results and backwardly supporting interpretability and traceability of the results, we need to extend properties and relations, and record supplementary properties for fusion and update strategies. These strategies are then executed during the entity update stage. For the iterative evolution of knowledge graph based on incomplete data sets, the knowledge modeling framework needs to address the following problems:

- The properties/relations can carry supplementary properties: These supplementary properties are used to describe the source, confidence, relevance, author, and other relevant asset information of the properties/relations.
- The properties/relations can define update strategies: Support for executable rule expressions is needed to define selection and prioritization strategies for the properties/relations. This ensures the stability of results even when data from the different sources arrive randomly.
- The entity types can be bound to entity linking operators: In industrial-level applications, many data sources do not provide standardized IDs. Therefore, we need to use entity linking strategies such as text matching and spatiotemporal clustering to find the target entity ID. Support for binding entity linking operators to target entity types is needed to ensure the execution of the same entity linking operator when different source data updates occur, thus ensuring the result is stability.

2.9 Summary of Problems with Semantic-less, Non-programmable Labeled Property Graph

Iterative evolution & avoiding duplication

Firstly, knowledge management is associated with the entire lifecycle of the business, requiring the ability to evolve iteratively and support continuous business iteration while effectively avoiding combinatorial explosion and duplicate construction. Secondly, knowledge management faces the complex problem of modeling knowledge from incomplete data sets, heterogeneous data sources, and multiple business expert perspectives. This requires the ability to implement differentiated perspectives and lightweight alignment of heterogeneous data sources through programmable paradigms, thereby reducing system complexity. Lastly, knowledge management needs to establish necessary knowledge hierarchies and classification systems to achieve effective linkage, induction, and deduction between different levels. This enables automatic extraction of static common knowledge to support efficient cross-business reuse and effective accumulation of core assets. The following chapter 3 and 4 will provide detailed explanations of the semantics-enhanced programmable framework (SPG).

Multi-perspectives & lightweight alignment , reduce complexity via programmable paradigms

Knowledge hierarchies & efficient reuse