Engineering Technology Trends Vol 3 Issue 4 2025

Original Research Article

Open Access

Omniscient Pte.Ltd.

Discussion on Strategies for Information System Integration and Data Integration

Zhi-Qiang Gan*

Sichuan Hongxin Software Co., Ltd., Mianyang, Sichuan, 621000, China

***Correspondence to:** Zhi-Qiang Gan, Sichuan Hongxin Software Co., Ltd., Mianyang, Sichuan, 621000, China, E-mail: gzq85159814@163.com

Abstract: With the rapid advancement of digitalization across industries, enterprises are facing an increasingly urgent demand for the efficient utilization of information systems and data. Information system integration focuses on breaking down system barriers to achieve seamless business process connectivity, while data integration aims to address issues such as data fragmentation and heterogeneous formats, ensuring the full realization of data value. This paper concentrates on integration strategies for information systems and data, exploring them from perspectives including technological integration, standard unification, and security protection. The study aims to provide enterprises with scientifically sound integration solutions, thereby improving operational efficiency and decision-making accuracy.

Keywords: Information system integration; data integration; strategy

Introduction

In the current digital era, the widespread application of information technology has led enterprises to rely heavily on various information systems for daily operations. However, the coexistence of multiple systems and the dispersion of data have become significant bottlenecks restricting enterprise development. Information system integration enables the organic combination of systems with different functions and architectures, enhancing the coordination of business processes. Data integration, on the other hand, eliminates data silos and promotes efficient data circulation and sharing. The two approaches are complementary and play a crucial role in optimizing resource allocation and improving the scientific basis of decision-making. An in-depth exploration of integration strategies can help enterprises better cope with complex and dynamic market environments and gain a competitive advantage in an increasingly fierce market.

1. Theoretical Foundations and Literature Review

1.1 Definition of Core Concepts

(1) Information System Integration

Information system integration refers to the process of integrating multiple independent information systems into a collaboratively operating whole. According to architectural models, it can be classified into Enterprise Application Integration (EAI), Service-Oriented Architecture (SOA), and Microservices Architecture. EAI focuses on interface connectivity between systems to address data silo issues. SOA is

service-centric and enables cross-platform invocation through standardized protocols. Microservices architecture decomposes systems into lightweight and independent services, thereby enhancing flexibility and scalability.

(2) Data Integration

Data integration aims to achieve unified access to and utilization of data from different sources. From a layered perspective, it can be divided into the data storage layer (such as data warehouses and data lakes, which enable centralized data storage), the data transmission layer (such as ETL tools and APIs, which ensure efficient data flow), and the data application layer (such as reporting and decision support systems, which mine data value) [1].

1.2 Theoretical Foundations

(1) System Interoperability Theory

The core objective of system interoperability theory is to enable data exchange and functional collaboration among heterogeneous systems. It encompasses three levels—syntactic, semantic, and pragmatic interoperability—providing a theoretical basis for compatibility in information system integration.

(2) Data Governance Theory

Taking the DAMA framework as an example, data governance theory regulates the full data lifecycle from dimensions such as data strategy, data architecture, and data quality, ensuring the accuracy and security of data integration.

(3) Distributed System Architecture Theory

For instance, the CAP theorem states that a distributed system cannot simultaneously satisfy consistency, availability, and partition tolerance. This principle offers a theoretical basis for trade-off decisions in the architectural design of information system integration.

1.3 Research Status at Home and Abroad

(1) Academic Research Hotspots and Gaps

Current research hotspots focus on integration technologies under microservices architectures and data governance in big data environments. Research gaps include standardization for cross-industry system integration and studies on the integration of edge computing with integration technologies.

(2) Industry Practice Cases and Lessons Learned

Representative cases include SOA-based system integration in financial institutions and data lake construction in manufacturing enterprises. Lessons learned highlight that neglecting data quality can lead to integration failure, and excessive pursuit of advanced technologies without alignment with business requirements can undermine integration effectiveness.

2. Analysis of the Relationship between Information System Integration and Data Integration

2.1 Logical Interrelationship

(1) System integration as the foundational support for data integration

Information system integration establishes cross-system data circulation channels by integrating dispersed hardware, software, and network resources. For example, enterprises must first achieve interoperability among business systems through EAI or microservice architectures to provide a stable "transmission carrier" for data integration. Without unified interfaces and functional coordination at the system level, data integration would face the dilemma of "no data to transmit and no channels available," making multi-source data aggregation impossible.

(2) Data integration as the value extension of system integration

System integration focuses on "physical connectivity," addressing whether systems can interoperate. Data integration, however, goes further by cleaning, consolidating, and analyzing the integrated data to realize "data value mining," upgrading the significance of system integration from mere connectivity to business empowerment. For instance, retail enterprises first connect in-store systems with e-commerce platforms through system integration, and then apply data integration to analyze consumer behavior, thereby guiding inventory allocation and marketing strategy formulation.

2.2 Synergistic Challenges

(1) Technological heterogeneity

Different systems may adopt diverse programming languages (e.g., Java, Python), API standards (e.g., REST, SOAP), and database types (e.g., MySQL, Oracle). This heterogeneity increases the difficulty of interface adaptation during system integration and often leads to data format incompatibility during transmission, thus hindering the efficient advancement of data integration.

(2) Data semantic conflicts

Due to differences in system development timelines and business objectives, issues such as inconsistent master data (e.g., different customer IDs in sales and financial systems) and missing metadata (e.g., unclear definitions of data fields) frequently arise. These problems prevent the formation of a unified understanding after data integration, thereby reducing data usability.

(3) Security and compliance risks

During system and data integration, cross-system and cross-regional data transmission must comply with regulations such as the GDPR and the Data Security Law. Inadequate encryption mechanisms and access control may lead to data leakage risks. Moreover, data sovereignty requirements in some countries (e.g., data localization policies) further restrict the scope of cross-regional system and data integration[2] .

2.3 Analysis of Typical Application Scenarios

(1) System integration after enterprise mergers and acquisitions

Post-merger enterprises often face significant differences in system architectures. System integration is first implemented through SOA or microservices, followed by data integration of core domains such as finance and human resources, to resolve issues of parallel systems and isolated data, ensuring a smooth business transition after the merger.

(2) Cross-departmental business process collaboration

In scenarios such as government "one-stop online services," business systems of departments including market regulation, taxation, and social security must first be integrated. Subsequently, data integration enables "single submission with multi-department sharing," reducing repeated material submission by enterprises and improving administrative efficiency.

(3) Multi-source data fusion in Internet of Things (IoT) environments

In IoT environments, structured data (e.g., temperature readings) and unstructured data (e.g., video streams) generated by devices such as sensors and smart terminals must first be integrated at the system level through edge computing nodes. Data integration technologies are then applied for format transformation and correlation analysis, supporting decision-making in smart factories, smart cities, and related applications.

3. Design of Information System Integration Strategies

3.1 Architectural Design Strategies

(1) Applicable scenarios of centralized versus distributed architectures

Centralized architectures are suitable for scenarios with relatively simple business processes, small data volumes, and high requirements for system consistency, such as the integration of internal financial systems in small enterprises. Their advantages lie in simplified deployment and maintenance as well as centralized data control; however, they suffer from limited scalability. In contrast, distributed architectures are more appropriate for complex business environments with large data volumes and high availability requirements, such as multi-system integration in large e-commerce platforms. By decomposing systems into multiple nodes, distributed architectures enhance scalability and fault tolerance, but they place higher demands on technical operation and maintenance capabilities and require careful trade-offs between consistency and availability as defined by the CAP theorem.

(2) Layered architecture based on the "middle platform" (platform-based) concept

Integration can be achieved through the construction of a business middle platform and a data middle platform. The business middle platform consolidates common business capabilities (e.g., user management and order processing) and provides standardized service interfaces for front-end systems, thereby reducing redundant development. The data middle platform aggregates multi-source data for cleansing, modeling, and analysis, supporting both the decision-making needs of the business middle platform and the establishment of unified data assets for data integration. A typical example is Internet enterprises adopting a dual middle-platform architecture combining "business + data," which enables efficient collaboration among marketing, supply chain, and other systems[3].

3.2 Technology Selection Strategies

(1) Interface standardization

RESTful APIs are preferred for lightweight system interactions. Based on the HTTP protocol, they are compatible with most application scenarios and well suited for high-frequency, low-complexity data transmission. For scenarios requiring flexible queries

across multiple associated datasets—such as querying user relationship data on social networking platforms—GraphQL can be adopted. GraphQL allows clients to customize response formats, thereby reducing the number of API calls and improving data retrieval efficiency.

(2) Message middleware selection

In high-concurrency scenarios, such as flash sales on e-commerce platforms, Apache Kafka is recommended due to its high throughput and persistent storage capabilities, which ensure stable data transmission. For scenarios requiring strong transactional support, such as financial transfer systems, RabbitMQ is more suitable. Its rich routing mechanisms and message acknowledgment features help reduce the risk of data loss and enhance message reliability.

(3) Service orchestration tools

Docker can be employed to containerize services, ensuring deployment consistency across different environments. When combined with Kubernetes for service orchestration, it enables automatic scaling, fault self-healing, and efficient lifecycle management of services. This approach is particularly suitable for managing multi-service integration in microservices architectures, significantly improving system operation and maintenance efficiency.

3.3 Implementation Path Strategies

(1) Incremental iteration versus disruptive reconstruction

Incremental iteration is appropriate for scenarios where core business operations cannot be interrupted, such as the integration of banking core systems. By adopting a "pilot implementation–effect verification–gradual rollout" approach, non-core systems (e.g., customer service systems) are integrated first, followed by a gradual transition to core systems, thereby reducing implementation risks. Disruptive reconstruction, on the other hand, is suitable for legacy systems facing urgent business transformation needs, such as digital transformation in traditional manufacturing enterprises. Although it involves complete replacement of existing systems, comprehensive contingency plans for business continuity and data migration backups must be established.

(2) Application of agile development methodologies

Integration projects can be divided into short iteration cycles of one to two weeks, with each cycle focusing on core functionalities, such as the development of specific system interfaces or data cleansing modules. Daily stand-up meetings facilitate progress synchronization and rapid response to requirement changes. By introducing user stories and iteration reviews, integration outcomes can be closely aligned with business needs, avoiding disconnection between technology and operations. For example, a retail enterprise successfully completed a pilot integration of its in-store systems and online marketplace within three months through agile development practices[4].

4. Optimization Strategies for Data Integration

4.1 Data Governance Strategies

(1) Construction of a Master Data Management (MDM) system

Master Data Management focuses on core enterprise master data—such as customer, product, and supplier data—by establishing unified data standards (e.g., coding rules and field definitions) and management processes to achieve centralized control. For example, retail enterprises can build an MDM system to unify customer IDs and product classifications across physical stores and e-commerce platforms, thereby avoiding data integration inconsistencies caused by fragmented master data and laying a solid foundation for cross-channel business collaboration.

(2) Data quality management mechanisms

A full-process data quality management framework comprising "pre-event prevention, in-process monitoring, and post-event remediation" should be established. Prior to data integration, validation rules—such as format checks and logical consistency checks—are defined. During integration, automated tools continuously monitor data quality and trigger alerts when anomalies such as missing values or duplicate records are detected. Afterward, data cleansing techniques, including deduplication, data completion, and format transformation, are applied to correct problematic data, ensuring the accuracy and usability of integrated data.

4.2 Technical Implementation Strategies

(1) ETL tool selection

For complex enterprise-level scenarios—such as large-scale data integration in the financial sector—Informatica is recommended due to its powerful data

processing capabilities and extensive set of built-in connectors, which effectively support heterogeneous data sources. For small- and medium-sized scenarios or budget-constrained projects, open-source tools such as Talend can be adopted. Talend supports custom development at a lower cost while satisfying basic requirements for data extraction, transformation, and loading.

(2) Real-time data integration solutions

Change Data Capture (CDC) technologies can be used to capture real-time database changes, including data insertions and updates. When combined with stream processing frameworks such as Apache Flink or Spark Streaming, real-time data processing can be achieved. This approach is particularly suitable for scenarios with stringent timeliness requirements, such as real-time recommendation systems and financial risk control, effectively overcoming the latency limitations of traditional batch-based ETL processes.

(3) Application of blockchain in data lineage and traceability

By leveraging the immutability and traceability of blockchain technology, the source, processing nodes, and modification records throughout the data integration process can be securely recorded. This approach is especially applicable to cross-border data integration or sensitive data scenarios, such as healthcare data integration, ensuring full lifecycle auditability and significantly enhancing data trustworthiness[5].

4.3 Security and Privacy Strategies

(1) Encrypted data transmission and storage

During data transmission, encryption protocols such as TLS/SSL should be employed to secure data channels and prevent interception or eavesdropping. In the storage phase, sensitive information—such as identification numbers and bank card details—should be protected using symmetric encryption algorithms (e.g., AES) or asymmetric encryption algorithms (e.g., RSA), thereby ensuring data security at rest.

(2) Dynamic access control and audit tracking

A Role-Based Access Control (RBAC) model should be adopted to allocate data access permissions, dynamically adjusting authorization scopes according to user roles to prevent unauthorized access. Meanwhile, an audit logging system should be established to record user data operations, including querying, modification, and deletion activities. In the event of a security incident, such logs enable rapid traceability of responsible entities, ensuring compliance with regulations such as the GDPR and the *Personal Information Protection Law of China*.

Conclusion

Information system integration and data integration constitute critical components of enterprise digital transformation. Their coordinated implementation can significantly enhance operational efficiency and strengthen competitive advantages. Through scientifically sound and well-planned integration strategies, enterprises can effectively address challenges such as system heterogeneity and data silos, thereby achieving deep integration between business processes and data assets. Looking ahead, as technologies continue to evolve, integration strategies must also be continuously refined and upgraded. Enterprises should keep pace with technological advancements and actively explore integration models suited to their own development needs, leveraging data-driven approaches to foster business innovation and achieve sustainable growth in the digital era.

References

[1] Wang Tianyu. Strategies for Information System Integration and Data Integration [J]. Science and Technology Wind, 2020(25): 98–99.

[2] Zhou Yongli. Strategies for Information System Integration and Data Integration [J]. Modern Industrial Economy and Informationization, 2021, 9(6): 67–68.

[3] Li Wei. Research on Information System Integration and Data Integration Strategies [J]. Electronic World, 2020(11): 86–87.

[4] Wang Liangjun. Information System Integration and Data Integration Strategies [J]. Dike World, 2021(8): 15–16.

[5] Duan Chenggang. A Discussion on Strategies for Information System Integration and Data Integration [J]. Encyclopedia Forum (Electronic Journal), 2021(19): 71–72.