World Journal of Innovation and Modern Technology, Vol. 9, Issue 7 (Jul) 2026
ISSN 2682-5910
DOI: 10.66069/ojspub.26820706

Design and Implementation of an Adaptive RBAC Model with Dynamic Permission Allocation for Industrial Control Systems in Smart Power Plants

Zhiling Yuan

Xi'an Thermal Power Research Institute Co., Ltd., Xi'an 710054, Shaanxi, China

Abstract: ***Amidst escalating global energy demand and the accelerating deployment of renewable energy technologies, smart power plants have emerged as indispensable pillars of modern power systems, integrating distributed energy resources, advanced metering infrastructure, and real-time supervisory control and data acquisition (SCADA) networks. However, this digital transformation concurrently exposes these critical infrastructures to unprecedented cybersecurity vulnerabilities—including data tampering, unauthorized access, advanced persistent threats (APTs), and insider data breaches—which, if exploited, could trigger cascading failures or inflict substantial economic losses. Consequently, information security and permission management have become paramount concerns for plant operators and regulators alike. The conventional Role-Based Access Control (RBAC) model, while widely adopted for its stability and interpretability, exhibits inherent limitations in the dynamically evolving operational environment of smart power plants. Static role-to-permission mappings cannot adapt to fluctuating operational contexts, evolving user behaviors, or real-time risk postures—a shortcoming that grows increasingly critical as plants incorporate remote operations, multi-stakeholder coordination, and dynamic grid conditions. Recognizing this gap, researchers have pursued enhanced RBAC methodologies that decouple data logic control from business code, enabling the dynamic creation and modification of data logic during system runtime without service interruption. This approach, as demonstrated in practical implementations, effectively accommodates the fluid permission requirements inherent in smart power plant environments. Empirical validation of such dynamic frameworks is emerging across multiple domains: a 2024 study introduced a distributed smart power plant access control mechanism based on blockchain and ciphertext updatable functional encryption, leveraging decentralized, tamper-proof features to ensure data integrity and transparency while enabling fine-grained, dynamic authorization; experimental results confirmed that blockchain transaction latency remained under 5 milliseconds for typical operations, underscoring the practicality of such approaches for time-sensitive industrial control. Similarly, a context-aware framework integrating large language models with retrieval-augmented generation on an RBAC backbone has demonstrated the ability to infer implicit behavioral semantics from unstructured access logs, thereby refining authorization decisions beyond static policies. In substation automation, the combination of RBAC with mandatory access control has enabled hierarchical, sub-authorized access to IEC 61850 communications, substantially improving the controllability of remote equipment operations. Furthermore, a zero-trust security model for power plants and substations—comprising identity authentication, dynamic access control, and trust evaluation modules—has been proposed to address the complexity of modern communication networks, while attribute-based access control models integrated with variational autoencoders have achieved anomaly detection accuracies as high as 97.2% in power grid IoT terminal access management. The efficacy and feasibility of these enhanced, dynamically configurable permission management approaches are thus validated through their successful application in real-world smart power plant scenarios, offering a viable pathway toward securing critical energy infrastructure against evolving cyber threats while maintaining operational agility and regulatory compliance.***

Keywords: RBAC, Smart power plant, Dynamic permission configuration.

1. INTRODUCTION

With the continuous growth of global energy demand and the rapid development of renewable energy technologies, the power industry is undergoing a profound digital and intelligent transformation [1]. As an important part of the future power system, smart power plants [2] have gradually become a new trend in industry development due to their significant advantages in energy management, equipment monitoring, and operation optimization. In the process of building smart power plants, information security and permission management issues are particularly important [3]. To ensure the safe, reliable, and efficient operation of power plant systems, a flexible and efficient permission control method is urgently needed.

Role-Based Access Control (RBAC) [4], as a widely used permission management method, simplifies the complexity of permission management by associating users with their roles and assigning permissions based on roles. However, the traditional RBAC model is inadequate in coping with the increasingly complex and dynamically changing environment in smart power plants [5]. For example, in the actual operation of power plants, personnel shift information and users' specific task requirements may change frequently. These dynamic factors require the permission management system to quickly adjust user permissions to adapt to real-time business needs [6]. Therefore, an enhanced RBAC model is needed to meet the dynamic permission control requirements in smart power plants to ensure the security and flexibility of the system.

This research aims to propose an enhanced RBAC method to adapt to dynamic permission management in smart power plants. Specifically, this research will cover the following aspects:

(1) Analyze the limitations of the traditional RBAC model: Explore the shortcomings of the traditional RBAC model in smart power plant applications, especially its limitations in handling dynamic changes.

(2) Design an enhanced RBAC model: Integrate dynamic expression control into the traditional RBAC model to build a more flexible and adaptive permission management framework.

(3) Verification and application: Construct a set of dynamic permission control systems.

The structure of this paper is arranged as follows: Chapter 2 introduces the basic concepts of the traditional RBAC model, presents an enhanced RBAC dynamic permission control method, and describes the core expression parser. Chapter 3 will demonstrate the specific design of the enhanced RBAC model in smart power plants and discuss its practical effects and application prospects.

2. TECHNICAL FRAMEWORK

2.1 RBAC Model

In permission design, the RBAC permission model uses roles to associate users with permission points, and permission control is achieved by associating permission codes with corresponding business logic. In this model, permissions are merely markers, and specific data validation logic needs to be implemented in relevant business code, resulting in data logic-based permission validation being mixed with business code. For example, when querying data for the same business function, isolation can be based on organizational fields or time ranges according to requirements. The traditional implementation treats each specific data isolation method as a separate permission, and data isolation is implemented through logical code or database SQL. When new data isolation requirements arise or existing control logic needs to be modified, in addition to adding new permissions, business code must be modified, compiled, and re-released, significantly increasing system development workload and Post operation and maintenance costs

2.2 An Enhanced RBAC Dynamic Permission Control Method

This research innovates on the basis of the traditional RBAC model and provides a A brand new method for verifying permissions, including detailed definition and management mechanisms for operation permissions and data permissions. It decouples the data logic control involved in permission control from business code, enabling dynamic creation and modification of data isolation logic without affecting program operation.

Operation permissions refer to the specific operations a user can perform in the system. Each operation permission is associated with a business function through a permission code, ensuring that users can only access specific functional modules if they have the corresponding permissions.

Data permissions are used to control the range of data a user can access and operate when performing operations. They are defined through data expressions, which can include common logical operators and custom functions to achieve precise control over data. Data expressions can contain logical operators such as ">", "<", "==", "!=", as well as custom functions. Expressions are dynamically interpreted and executed by a parser during system runtime. To facilitate permission management and flexibility, data permissions also support grouping data according to specific rules. Users can define data groups through a configuration interface and reference these groups in data permission expressions.

When a user has multiple data permissions simultaneously, these permissions may have inclusion relationships. A priority mechanism is supported, where higher-priority data permissions will override or simplify lower-priority ones, ensuring the efficiency and accuracy of permission validation.

The permission model associates users with permissions through roles. There is a many-to-many relationship between users and roles, as well as between roles and permissions. When setting permissions in the system, first assign operation permissions to roles, then further set expressions based on the permissions, and dynamically parse the expressions through the expression parser to complete the process of dynamic permission allocation.

2.3 Expression Parser

This section implements a highly flexible and feature-rich expression parser, specifically designed to handle complex logic and data operations, especially suitable for scenarios where dynamic parsing and execution of conditional expressions, mathematical operations, and even SQL fragment generation are required in Java applications. The core capability of this parser lies in its ability to parse composite expressions containing variable references, arithmetic operations, logical comparisons, custom function calls, and specific SQL operators, and accurately compute results or construct SQL instructions based on input parameters. As described in Figure 1, which is the flow chart of the expression parser, after the user logs in, the system queries the user's permission codes and corresponding expressions, parses the expressions through lexical and syntactic analysis, and obtains the final permission isolation result through calculation logic and SQL generation to achieve the purpose of permission control.

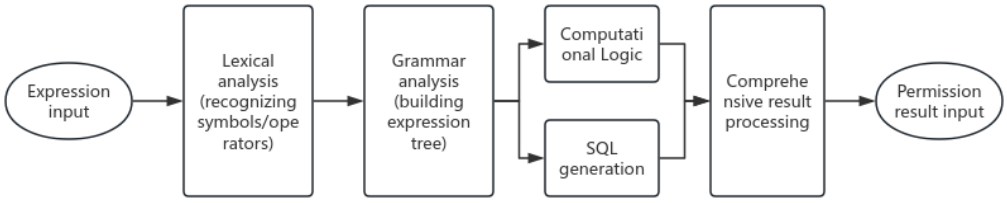

Figure 1: Flow Chart of Expression Parser

Expressions are used to define data access rules when a user performs an operation. For example, a user can only view data entered by themselves, and the expression can be defined as. The system supports using common logical operators and extended functions to build complex permission logic.

During the system initialization phase, a set of predefined operators are registered through the install0perators() method, covering basic arithmetic, comparison, logical operators, and parentheses, allowing users to extend more custom operators as needed. This provides the parser with powerful expression processing capabilities.

The main interfaces of the parser, eval() and getBooleanResult(), are used to calculate the value of the expression and directly obtain the boolean result, respectively. The initSQL() method converts the expression into an SQL fragment for dynamically building query statements and supports parameterized queries.

The parsing process of the expression parser is mainly divided into two parts: parse() for direct calculation logic and parse2Sql() for SQL generation. Both use a stack structure to assist parsing, separate operands and operators by identifying character sequences, and traverse and build an expression tree in a recursive descent manner to handle operator precedence and bracket matching issues. For basic types such as strings, numbers, and booleans, the parser can accurately identify and convert them, and also supports looking up variable values from external mappings, enhancing the dynamics of expressions. The parser executes user-defined function logic through the getFunctionResult() method, extending the flexibility and practicality of expressions.

In summary, the expression parser is a comprehensive solution that enables efficient parsing and computation of complex expressions, adapts to specific requirements such as dynamic SQL generation, and has application value in fields such as data processing and dynamic configuration of business logic.

3. SYSTEM DESIGN

The enhanced RBAC dynamic control permission method for smart power plants described in this paper is technically implemented based on TPRI-DMP, a back-end low-code platform with independent intellectual property rights of Xi'an Thermal Power Research Institute, verifying the feasibility of this method. This section describes the specific design path of this method based on the TPRI-DMP platform.

To implement this set of dynamic control permission methods, it is necessary to build user modules, role modules, permission modules, etc.

The user maintenance module is mainly responsible for managing user information, including functions for adding, modifying, deleting, and querying users. In addition, the user maintenance module also provides operations for assigning and revoking user roles to ensure that each user is correctly assigned the corresponding roles and permissions.

The role maintenance module is responsible for managing role information in the system. Administrators can define new roles, modify the permission combinations of existing roles, and delete roles that are no longer needed through this module. At the same time, the module also provides a query function that can display all user information included in a specific role, facilitating administrators to manage and adjust roles.

The permission maintenance module is the core part of the entire system, responsible for defining and managing operational permissions and data permissions in the system. Operational permissions describe specific operations that users can perform in the system, such as querying, modifying, deleting, and reviewing; data permissions describe the scope of data that users can access or operate when performing these operations. When defining permissions, the permissions corresponding to this permission code are dynamically configured according to expressions, and the back-end re-parses and precompiles them through an expression engine, thereby realizing dynamic modification and expansion of data logic control. Through flexible configuration and management, the permission maintenance module decouples permission definition from business logic, making the system highly scalable and flexible. As shown in Figure 2, when defining permission codes, multiple expressions can be flexibly defined or modified. Roles and expressions are associated in a many-to-many relationship through permission codes, and after parsing by the expression parser, corresponding permissions are assigned to roles, thereby enabling corresponding personnel to have corresponding permissions.

Figure 2: Permission Configuration Interface

4. CONCLUSION

The enhanced RBAC method proposed in this paper effectively solves the problem of dynamic permission control in smart power plants. By analyzing the limitations of the traditional RBAC model, a new permission model is designed, which includes detailed definitions of operational permissions and data permissions. It associates users with permissions through roles and realizes dynamic and precise control of permissions using an expression parser. The database is designed to support the dynamic permission requirements of power plant systems, and multi-factor authentication is adopted for user login permission design to enhance security. This method can improve the

efficiency of permission management and the ability to respond to dynamic changes, ensuring the security and flexibility of the system.

REFERENCES

[1] Wang Shudong. Safeguarding National Energy Security, Promoting Green and Low-Carbon Transformation, and High-Quality Assistance in Building Chinese-Style Modernization [N]. China Electric Power News, 2024-06-27 (001).
[2] Wang Tao, Shan Zhengtao, Du Jingang, et al. Research and Development of an Intelligent Power Plant Production Management Development Platform [J]. Thermal Power Generation, 2019, 48(09):115-119. DOI: 10.19666/j.rlfd.201906129.
[3] Zhao Jinsong, Zhang Chaoyang, Gu Weifeng, et al. Platform Architecture of Smart Power Plant Based on Industrial Internet [J]. Thermal Power Generation, 2019, 48(09): 101-107. DOI: 10.19666/j.rlfd.201906114.
[4] Yu Yangkui. Research on Role-Based Access Control Model (RBAC) [J]. Computer Technology and Development, 2019, 29(01): 198-201.
[5] Sun Hengyi. Design and Implementation of an Extended RBAC Permission Model for Power Trading Systems [J]. Network Security Technology and Application, 2018, (03): 42-43.
[6] Hu Wenyu, Chen Jinbo. Research on Dynamic Access Control Model Oriented to Access Process [J]. Computer Technology and Development, 2022, 32(04): 92-96+108.

Author Profile

Zhiling Yuan (1999-), male, Xianyang, Shaanxi, master's student, assistant engineer, energy and power informatization.