An ATM Knowledge Graph-Based Method of Dynamic Information Fusion and Correlation Analysis

Qiucheng Xu(✉), Yongjie Yan, Yungang Tian, Weiyu Jiang, and Jinglei Huang

State Key Laboratory of Air Traffic Management System and Technology,
The 28th Research Institute of China Electronics Technology Group Corporation,
Nanjing 210007, China
xuqiucheng1987@163.com

Abstract. In order to fuse information and analyze correlation more efficiently and flexibly, an air traffic management (ATM) knowledge graph-based method is proposed to reorganize the information flexibly and manage the fusion process dynamically. After that, a breadth-first and depth-first search-based correlation analysis method is designed to find deeper correlations and improve the searching efficiency.

Keywords: Air traffic management · Knowledge graph · Information fusion · Correlation analysis

1 Introduction

With the rapid development of air transport in China, the number of flights is increasing rapidly at an average annual rate of over 10% in recent years [1]. It is predicted that civil aviation transportation in China will maintain a growth rate of about 12%. By 2030, there will be more than 450 civil transport airports, more than 95% of county-level administrative regions and population in China will receive air services, and the volume of passenger traffic will reach 1.8 billion [2]. However, the rapid development has also brought more and more challenges and pressures to the safe and efficient operation of air transport, for example, flight delays have become more common than ever [3], and the aviation safety accidents demonstrate a steadily rising tendency [4,5].

Air traffic management (ATM) has taken many measures to meet the challenges brought by rapid development of air traffic, such as building more surveillance and sensing equipment; developing a variety of information systems for auxiliary control aiming at different stages and scenarios of air traffic control [6]. In addition, the air traffic control department has strengthened cooperation with diverse departments, including airports, airlines, etc., to access more and more kinds of information from different sources [7]. Above solutions can solve the specific problems of air traffic controllers, while on the other hand, they could also bring a series of new problems, such as the explosion of information scale, difficulty of integration, and adding new workload to the controllers.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 487–496, 2020.
https://doi.org/10.1007/978-981-15-0187-6_58

Many researchers in the field of ATM focus on information management and efficient use methods in the operation of air traffic control.

Some researchers [8,9] introduce the concept of System Wide Information Management (SWIM). As an information switch platform based on network technology, it has been shown that relevant data can be safely, effectively, and timely shared and exchanged between different units and information systems.

Medina et al. [10] describe a usability analysis tool which computes estimate of trials-to-mastery and the probability of failure-to-complete for each task. The information required to complete a task on the automation under development is entered into the Web-based tool via a form. Yang et al. [11] look ahead the concept of intellectualized air traffic management technology, represented by deep learning, emphasizes that judgment and decision-making based on a large number of prior knowledges, which is consistent with the decision-making process of ATM.

However, most of the proposed methods put much emphasize on the breadth while not depth of information that system can handle, and thus leading to the failure to solve the depth-related problems. For example, is there relationship between flight CCA4228 and CES2471? If yes, what is the relationship? Is Shanghai Pudong Airport related to Beijing Capital Airport? If so, how is it related?

Inspired by the research idea of applying knowledge graph in social network, which can represent the breadth of a person's social relationship and analyze the depth of the relationship between any two people, we propose an information fusion and correlation analysis method based on ATM knowledge graph [12]. The main contributions of this paper are outlined as follows.

(1) Using ATM knowledge graph to reorganize the information flexibly and manage the fusion process dynamically.
(2) Based on the ATM knowledge graph, a breadth-first and depth-first search-based correlation analysis method is proposed to get all related feature instances on breadth and get relative path on depth.

Experimental results show that, compared with the traditional fusion and search method, the proposed method can find more deeper correlation and improve the searching efficiency.

The structure of the paper is organized as follows. Section 2 describes the problem definition. Section 3 shows the ATM knowledge graph-based information fusion and correlation analysis method. Section 4 shows the experimental results, and at last conclusions are made at Sect. 5.

2 Problem Definition

The ATM feature types commonly used in ATM information system include airports, runways, routes, airspace, flights, airlines, route points, control units, etc. Most of these commonly used information uses object-oriented design method, that is, to design a type template for each kind of feature, and mainly uses tables in entity-relationship databases for storage, which finally convert each data entity into a row of records.

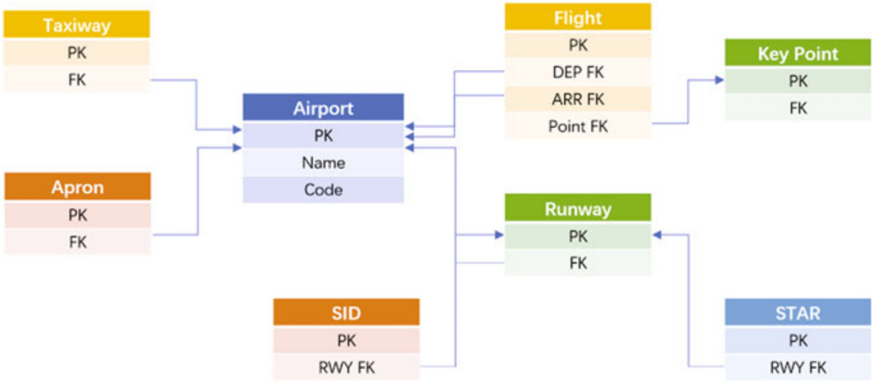

Fig. 1. Different types of ATM features are constructed as data table and connected with each other by PK and FK.

Nowadays, the commonly used method for ATM information system to construct data association and fusion is searching-related feature from multiple data tables through the form of primary key (PK) and foreign key (FK). Figure 1 shows an example of how different feature types are connected by PK and FK.

When the breadth-first search is needed to traverse all relevant data, such as "Pudong Airport in Shanghai as the center, search for all relevant information features," the following questions are asked:

(1) If FK is known, thus FK → PK, the correlation can be obtained. For example, Runway [FK] → Airport [PK], Taxiway [FK] → Airport [PK], Apron [FK] → Airport [PK].
(2) Conversely, if PK is known, then PK → ?, forward associations cannot be performed to know the FK tables who are associated directly unless all data tables are traversed.

When the depth-first search is needed to traverse the association path of two known elements, such as "Query the relationship between flight CCA4228 and Shanghai Pudong Airport," the following questions are asked:

(1) Existing processing capacity, if the take-off and landing airport of flight CCA4228 and the standby airport are not Shanghai Pudong Airport, there is no relationship between them.
(2) The real situation may have the following correlation: Shanghai Pudong Airport → Runway → SID → Route Point → Flight CCA4228, which can be interpreted as that the waypoint of flight CCA4228 at a certain time is the relevant waypoint of departure procedure at 17R_35L runway of Shanghai Pudong Airport.

Through the analysis of the two above-mentioned cases, it can be found that there existing the following shortcomings in nowadays' information fusion association methods:

(1) The traversal mode of breadth-first search is fixed and cannot be exhausted effectively;

(2) The structure of entity relationship is difficult to change dynamically;
(3) The depth-first search traversal ability is insufficient to carry out association reasoning.

3 ATM Knowledge Graph-Based Fusion and Correlation Analysis Method

In order to get all related feature objects on breadth and get relative path on depth, an ATM knowledge graph-based fusion method is proposed to reorganize the information, and after that, a breadth-first and depth-first search-based correlation analysis method is designed to achieve the required goal.

3.1 ATM Knowledge Graph-Based Fusion Method

In this work, we first build the ATM knowledge graph to reorganize all the important ATM features, such as Airport, Route, Airspace, Flight Plan, Airline, Land Mark, Control Unit, and so on. The graph data structure is chosen to represent and save the relationship between all the ATM feature instances, which is different from the E-R divided tables storage method.

The ATM knowledge graph G consists of a collection of nodes N and edges E, which is $G = (N, E)$. The nodes represent all the ATM feature instances and the edges represent the relationship between different feature instances. Figure 2 gives a partial fragment of how the ATM knowledge graph looks like.

In the graph, all the ATM features are organized together and all the relationships between each feature instance are saved as edges. With the graph, we

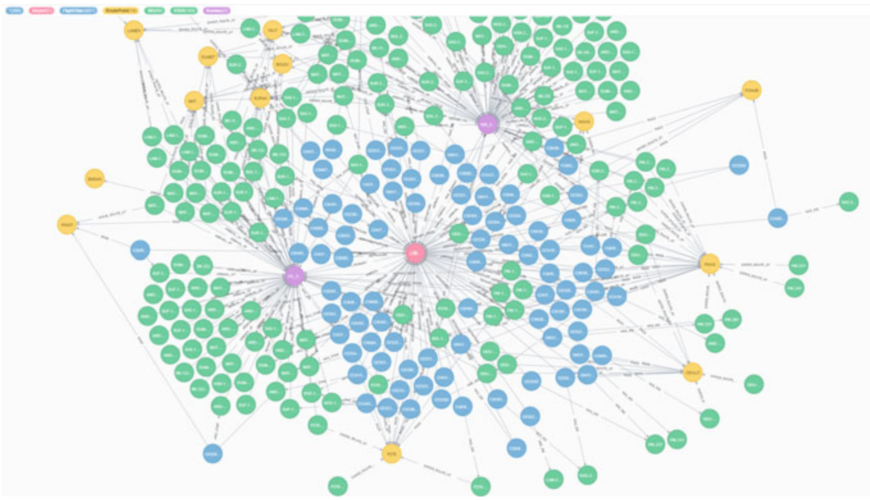

Fig. 2. A partial fragment of the ATM knowledge graph. Different colors of the nodes represent different feature type and the edge between each two nodes has its semantic meaning.

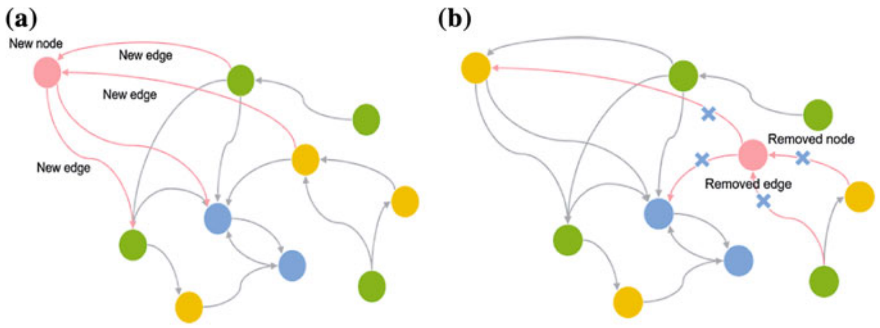

Fig. 3. The graph can be changed to fuse various kinds of data dynamically according to the need. **a** Add new node and edges. **b** Remove old node and edges.

can use degree distribution, average path length, and clustering coefficient index to describe the characteristic of the graph in the future.

Besides, we consider the graph as a dynamic change graph, as shown in Fig. 3, new nodes and edges can be added into the graph at any time. Also, old nodes can be removed from the graph along with the related edges.

3.2 Breadth-First and Depth-First Search-Based Correlation Analysis Method

Based on the ATM knowledge graph, we try to do some correlation analysis with breadth-first and depth-first search in the graph.

For the breadth-first search, we can get the relevant features of the searched feature. Algorithm 1 shows the overall process of the breadth-first search.

Algorithm 1 Breadth-first search method

```
Function BFS (center_node, distance):
Initialize a queue q and a list result_list;
q.push_back(center_node);
center_node.distance = 0;
while q is not empty do
    first_node = q[0];
    if first_node.distance > distance then
        break;
    end if
    first_node.visited_flag = true;
    result_list.push_back(first_node)
    q.pop_front();
    Get all the adjacent nodes of first_node and put them in a temporary list adjacent_list;
    for each adj_node in adjacent_list do
        if adj_node.visited_flag is not true then
            adj_node.distance = first_node.distance + 1;
            q.push_back(adj_node);
        end if
    end for
end while
return result_list;
```

The BFS function needs two parameters, one is the searched feature node and another one is the distance to constrain the search range. Each circulation between line 5 and line 20, we take out the first node of the queue and push back its adjacent nodes at the end of the queue unless the adjacent node has not been visited. In line 7, we check the nodes distance to decide whether the search should be end. At last, we get all the features whose distance from the searched feature node is less than the given distance parameter.

For the depth-first search, we can get all the relative paths of two features. Algorithm 2 shows the overall process of the depth-first search. The search starts from the start node. The recursion function between line 4 and line 19 compares the current node with the end node recursively to find the path. At last, we get all the relative paths of the start and end features by the DFS method.

Algorithm 2 Depth-first search method

```
1: With the input parameters start_node, end_node;
2: Initialize a list result_list;
3: DFS(start_node);
4:  Function DFS (current_node):
5: current_node.visited_flag = true;
6: if current_node == end_node then
7:     result_list.push_back(current_node);
8:     return true;
9: end if
10: Get all the adjacent nodes of current_node and put them in a temporary list adja-
    cent_list;
11: for each adj_node in adjacent_list do
12:     if adj_node.visited_flag is not true then
13:         path_is_right = DFS(adj_node);
14:         if path_is_right == true then
15:             result_list.push_back(adj_node);
16:         end if
17:     end if
18: end for
19: return false;
```

4 Experimental Results

The proposed method has been implemented in Java language on a Windows 64-bit workstation (Intel 2.2 GHz, 64 GB RAM).

In this work, we build an ATM knowledge graph which has about 20,000 feature nodes and 100,000 relationship edges.

Table 1. The breadth-first search results of different feature type.

Searched feature instance	Feature type	Distance from searched feature instance					
		1 (direct connect)		2		3	
		Feature numbers	Elapsed time (ms)	Feature numbers	Elapsed time (ms)	Feature numbers	Elapsed time (ms)
Shanghai Pudong	Airport	146	58	905	113	3873	258
CSN3173	Flight	8	33	1020	126	2959	283
A593	Route	27	4	55	6	296	36
DOGAR	Route point	84	8	413	39	3150	198

Table 2. The depth-first search results of relative path.

First feature instance	Second feature instance	Relative path numbers	Shortest length	Longest length	Search elapsed time (ms)
Shanghai Pudong airport	CSN3173	85	3	3	702
CCA4228	CES2471	15	2	6	453
Shanghai Pudong airport	Beijing Capital airport	124	2	8	1560

Table 1 shows the case of BFS results of different feature type. We test the distance (1, 2, 3) from searched feature instance to verify the efficiency of the BFS method. Distance 1 represents those feature nodes connect with searched feature node directly, distance 2 represents those nodes connect through two edges to the searched feature node, and so on. The result shows that the proposed method can find the same amount of direct-connected feature nodes more efficiently and flexibly, besides can find more deeper relative feature nodes which is hard to achieve by the traditional method.

Figure 4 shows an example of a breadth-first search result of route A593 with the distance (1, 2, 3).

Table 2 shows the case of depth-first search results of relative path. We choose any two feature instances to get their relative paths which cannot be achieved by the traditional method. The result shows that there may be more than one relative path between two feature instances, some of the paths are useful but some others are not. Further research work is needed to describe the path from semantic view.

Figure 5 shows the example of a depth-first search result of relative path between Shanghai Pudong international airport and flight CSN3173. We can

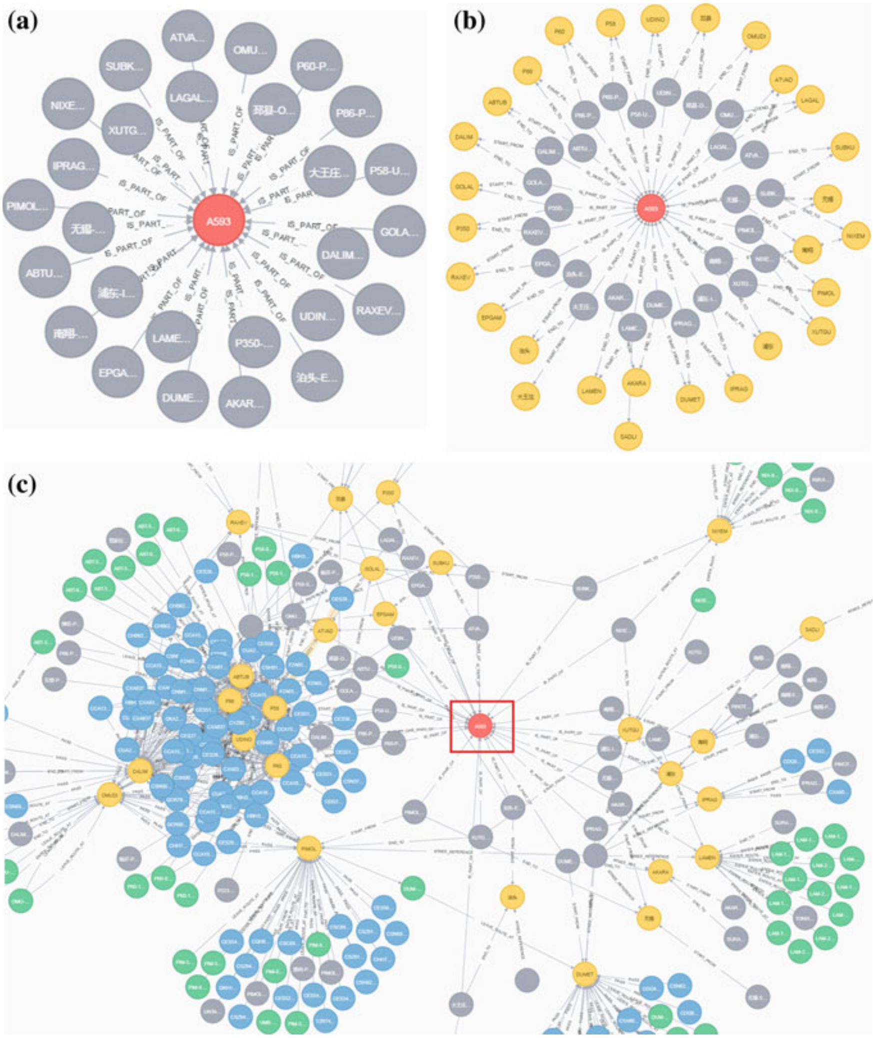

Fig. 4. Example of breadth-first search result of all features with **a** all features distance 1; **b** all features distance 2; **c** all features distance 3.

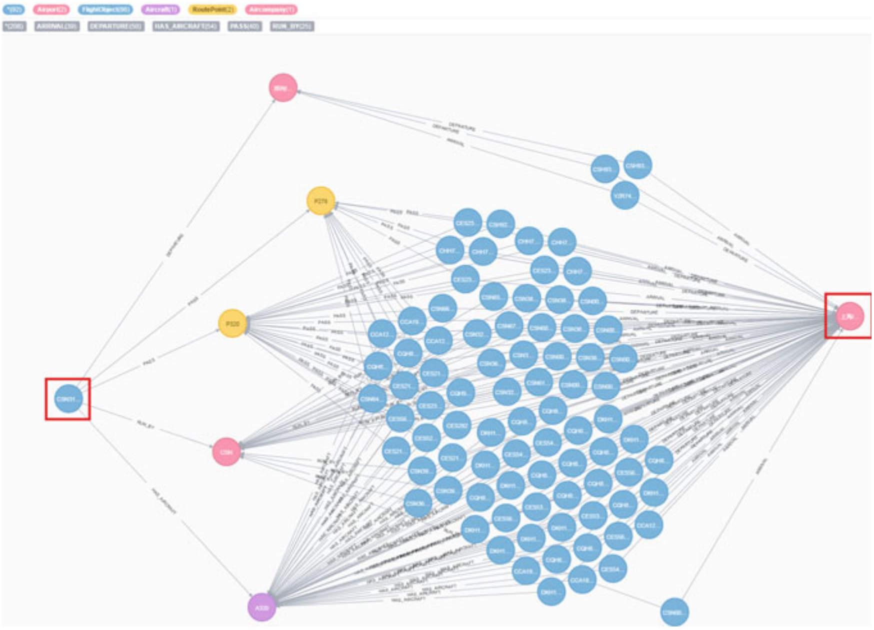

Fig. 5. A depth-first search result of relative path between Shanghai Pudong international airport and flight CSN3173.

find many different paths between them and know how they are related which is difficult to figure out by traditional methods.

5 Conclusion

This paper proposes an ATM knowledge graph-based method to reorganize the information flexibly and manage the fusion process dynamically. After that, a BFS- and DFS-based correlation analysis method is designed to find deeper correlation and improve the searching efficiency. Experimental results show that the proposed method can get all related feature instances on breadth and get relative path on depth with high efficiency and flexibility.

References

1. Zhao W (2017) The opportunities, challenges and obligations in internationalization of China civil aviation. Civil Aviat Manag 09:6–11
2. Yan Y, Cao G (2018) Operational concepts and key technologies of next generation air traffic management system. Command Inf Syst Technol 9(3):8–17
3. Dong N (2013) Current situation, cause of flight delays and governance path of China's civil aviation. J Beijing Univ Aeronaut Astronaut (Soc Sci Ed) 26(06):25–32

4. Netjasov F, Crnogorac D, Pavlovic G (2019) Potential safety occurrences as indicators of air traffic management safety performance: a network-based simulation model. Transp Res Part C: Emerg Technol 102:490–508
5. Yuan L (2108) Correlation among air traffic controllers' fatigue, workload and error. J Civil Aviat Univ China 36(06):31–33
6. Lin J (2014) Construction and development of air traffic control automation system for civil aviation. Electron Technol Softw Eng 08:255
7. Menon PK, Park SG (2016) Dynamics and control technologies in air traffic management. Annu Rev Control 42:271–284
8. Ai B (2012) New-generation ATM information system: SWIM. Chinas Civil Aviat 144(12):20–21
9. Li Q, Huang H, Chen Q (2010) SWIM—the new technology of air traffic management. Digit Commun World 05:50–53
10. Medina M, Sherry L, Feary M (2010) Automation for task analysis of next generation air traffic management systems. Transp Res Part C: Emerg Technol 18(6):921–929
11. Yang H, Yang B, Wu X et al (2018) Research and prospect of intellectualized air traffic management technology. Adv Eng Sci 04:12–21
12. Qi G, Gao H, Wu T (2017) The research advances of knowledge graph. Technol Intell Eng 3(1):4–25