Ontology of Airworthiness Rules based on Knowledge Graph

Honggang Liu[1, a], Yong Cai [1,2, b] and Weidong Yang[1, c]

[1]School of computer science, Fudan University, Shanghai 200000, China

[2] Shanghai Aircraft Airworthiness Certification Center of CAAC, Shanghai 201203, China

[a]929557497@qq.com, [b]13110240031@fudan.edu.cn, [c]yweidong@gmail.com

Keywords: Airworthiness Rules; Knowledge Graph; Aviation field.

Abstract: Since the airworthiness rules are the most basic status in the aviation field, the airworthiness rules must be fully matched in the initial stage of aircraft designing, later modification would lead to huge risks. At present, the rules for aircraft design are reviewed by domain experts. The shortage of domestic talents and complicated audit process have become the bottleneck of aircraft design. This paper will use the judicial aids of the knowledge graph to ontology the ontology of the airworthiness rules. Once the user has created the original design ontology for the aircraft, the design ontology can be compared to the airworthiness rule ontology to determine if the design is vulnerable.

1. Introduction

Knowledge graph is a knowledge based technology Google uses to enhance its search engine capabilities. In essence, knowledge graph is a semantic network that reveals the relationships between entities, and their interrelationships. The current knowledge graph has been used to refer to a variety of large-scale knowledge bases.[1]

A triple is a general representation of a knowledge graph, ie G=(E,R,S), where E={e1,e2,⋯,e|E|} is a collection of entities in the knowledge base, including | E| kinds of different entities; R={r1,r2,⋯,r|E|} is a set of relations in the knowledge base, including a total of |R| kinds of different relationships; S⊆E1×R×E2 represents triple collection in the knowledge base. The basic forms of triples mainly include entity1, relationship, entity2 and concepts, attributes, attribute values, etc. Entities are the most basic elements in the knowledge graph, and different entities have different relationships. Concepts mainly refer to collections, categories, objects, types of object, such as characters, geography, etc.; attributes mainly refer to features, characteristics and parameters that an entity may have, such as nationality, birthday, etc.; attribute values mainly refer to object-specific attributes. Values such as China, 1988-09-08, etc. Each entity can be identified by a globally unique ID, each attribute-value pair (AVP) can be used to characterize the intrinsic properties of the entity, and the relationship can be used to connect two entities. Describe the connection between them.

As we all know, the main purpose of civil aircraft is to carry passengers and cargo, the accident rate of aircraft accidents is extremely low. Once an accident occurs, it means heavy casualties and huge property losses. Through a series of operations that guarantee airworthiness, the probability of a machine crash is reduced to a very low level acceptable to the government and the public. It is worth emphasizing that the design side plays a decisive role in this, that is, the establishment process of airworthiness, and all subsequent work will be based on this. These need to be achieved through the government through legislation, law enforcement, technical appraisal, continuous supervision, etc., through various means to ensure the safety of the aircraft.

The current standard of civil aviation aircraft commonly used around the world is Federal regulation part25- Airworthiness standards: transport category airplanes. This paper will use knowledge graph technology build a semantic ontology of this regulation. Users could use their own airworthiness design craft compare with the semantic ontology; it would be much faster than manually compare two documents.

2. Method

The structure of the triad is {entity1, relationship, entity2}. In order to completely cover the regulations, this article makes specific provisions on the Concepts (the father class of entity) and Properties (relationships and entity's properties), Federal regulation D branch ontology will be made as an example. Since the data set is very large, in order to make the ontology could be scalable in the future, and it could cover all contents in Federal regulation, some relationships were replaced by one entity and two properties.

The original triad should be like figure1. EntityA has RelationC to EntityB. But if it is in a regulation, there should be a term number, like termE, so the knowledge graph we design is like figure2, termE(entity) point to EntityA, RelationC(entity) and EntityB. The relation between EntityA and EntityB has two path, one is through RelationC's father class RelationD(if RelationC has no fater class, RelationD is RelationC itself), the other one is through an Entity(RelationC).

Fig.1. Original knowledge graph

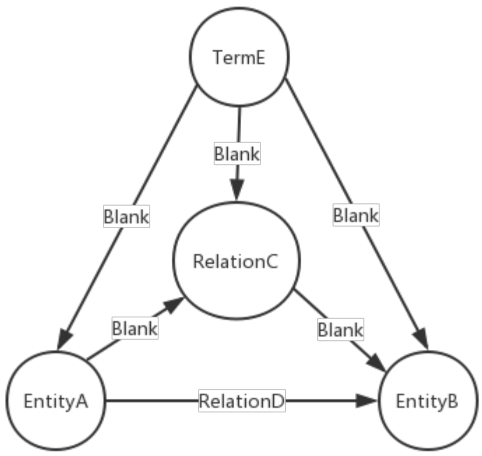

Fig.2. Graph for regulations

The advantage is that this knowledge graph could use term number as index, and user could search for limited content as {EntityA, RelationD, EntityB} in Fig2, the search speed would be improved.

This article used Federal regulation part25 sectionD as example, Part3.1 would detail the Concepts of sectionD and Part3.2 would detail the Properties of sectionD. Node that RelationC in Fig2 would be a child entity of Concept list as RelationC's node type is Entity, RelationD in Fig2 would be a type in Properties list as it is a true property. The Black Property is also a property type it means word order. For example: *Control surfaces require limit load tests*. EntityA is *Control surface*, RelationC/RelationD is *require*, EntityB is *limit load test*. If other part of the regulation is applied to this knowgraph, the Concepts and Properties should add contents.

3. Federal Regulation Part25 SectionD

3.1 Concepts

The first layer classes are listed in Table1. This is the father classes of all entities appears in the knowledge. Node that C00002 Position, C00005 Action, C00010 Compare are the RelationC entity in Fig.2, They should be use with specific Properties.

Table 1. Concepts in Federal regulation part25 sectionD

No.	Name	Example
C00001	Group	G00000001
C00002	Postion	up, inside, with
C00003	Entity_name	Control surface
C00004	Other_name	Limit load test
C00005	Action	require
C00006	Number	300
C00007	Unit	kg
C00008	Term	$25.651
C00009	Condition	A sentence after ‘which’
C00010	Compare	max
C00011	Adjective	enough

Class Group is a group of Entities, for example, the Group G0000001 means all the content in Federal regulation No.25.651. So, the entity $25.651 should point to G0000001 and G0000001 should point to all the entities related to the Federal regulation No.25.651.

Class Position is the position related words, include direction information like *up, left, front*; fuzzy positional information like *inside, nearby, far away*; and contact information like *contact, touch, clutch*.

Class Entity_name is a specific airplane part. Like *control surface, control system, hinges.*

Class Other_name is the father class of all the noun appear in Federal regulation but not a specific airplane part. Like *limit load test, strength, figidity*.

Class Action is a verb or adverb. In normal knowledge graph, it should be a properties, based on the method in Fig.2, this type of properties would be ontologyed as RelationC and RelationD. If RelationC is Class Action, Relation D should be Properti Action. The Class Actions include *require, attach, show*, etc.

Class Number is the numbers, like 3.0, 300, 100

Class Unit is the unit of numbers, like cm, kg, Pa.

Class Term is the number of a term, like $25.651.

Class Condition is a requirement of entity, it usually be a sentence after ‘when’, ‘which’. For example, *there is no interference between any surfaces when one is held in its extreme position and the others are operated.* The condition is *one is held in its extreme position and the others are operated.*

Class Compare is the selection criteria in a group, or in an entity. Like *max, min, better.*

Class Adjective is some emphasized adjectives like *adjustable, enough, nonstandard.*

3.2 Properties

The properties are list in the table 2. This is all the properties appears in the knowledge.

Property Blank could be understood as has/with/be/about/. For example, term $25.651 has term a and term b, the relation between $25.651 and term a is Blank. Another example is ‘adjustable stabilizer’, to describe Entity ‘stabilizer’ is ‘adjustable’, Entity ‘stabilizer’ should point to ‘adjustable’ with Blank property. In the federal regulation, there are so many similar words like has, include, is. These words would all be in Property Blank.

Table 2. Properties in Federal regulation part25 sectionD

No.	Name	Remark
P00001	Blank	Means has/is/with/about
P00002	Attribute	Between Entity_name and Other_name
P00003	Contain	Point to Group
P00004	Then	Point to Group
P00005	Function	Also means ‘so that’
P00006	Or	
P00007	And	
P00008	Not	Also means ‘prevent’
P00009	Satisfy	Point to Group or Term
P00010	Compare	Use as RelationD
P00011	Position	Use as RelationD
P00012	Action	Use as RelationD

Property Attribute describe the belong relationship between two Entities. For example, Limit load tests of control surfaces, in this sentience, Entity ‘limit load test’ belongs to entity ‘control surface’.

Property Contain usually used with ‘include/contain’ words, it emphasizes the group relation.

Property Then describe orderly actions, it usually in a group. For example, *move the joystick forware and then drop the wing windshield*. ‘move …’ and ‘drop …’ two movement is in a time order, the relation between them is Property Then.

Property Funciton describes an Entity or Group as a result of another Entity or Group. For example, *if the joystick is moved forward, the airplane should descend.* In this sentence, Group ‘joystick move forward’ point to Group ‘airplane descend’.

Property Or/And/Not describe the logic of Group or Entity.

Property Satisfy usually point to a Term name of a group of terms. For example, *$25.651 proof of strength compliance with the special factor’s requirements of $25.619 through 25.625 and 25.657.* Here, Proof of strength should point to a group include %25.619 through 25.625 and 25.657.

Property Compare, Position and Action are used as RelationD in Fig.2, they need to be use with Entity Compare, Position and Action’s child classes.

3.3 Example

As an example, an ontology of $25.651 is used. The content is:

(a) Limit load tests of control surfaces are required. These tests must include the horn or fitting to which the control system is attached.

(b) Compliance with the special factors requirements of §§25.619 through 25.625 and 25.657 for control surface hinges must be shown by analysis or individual load tests.

Based on the definition in 3.1 and 3.2, the yellow background words are Entities, the red words are Properties, some of them need to use the method in Chapter 2 to create RelationC Entities, the blue words need to be manually changed to other words.

The software to build this semantic ontology is Protégé, the server running SPARQL Api is Apache Jena. The machine running this server is ThinkPad X1 Carbon, with Win10 operating system. The ontology is show in Fig.3

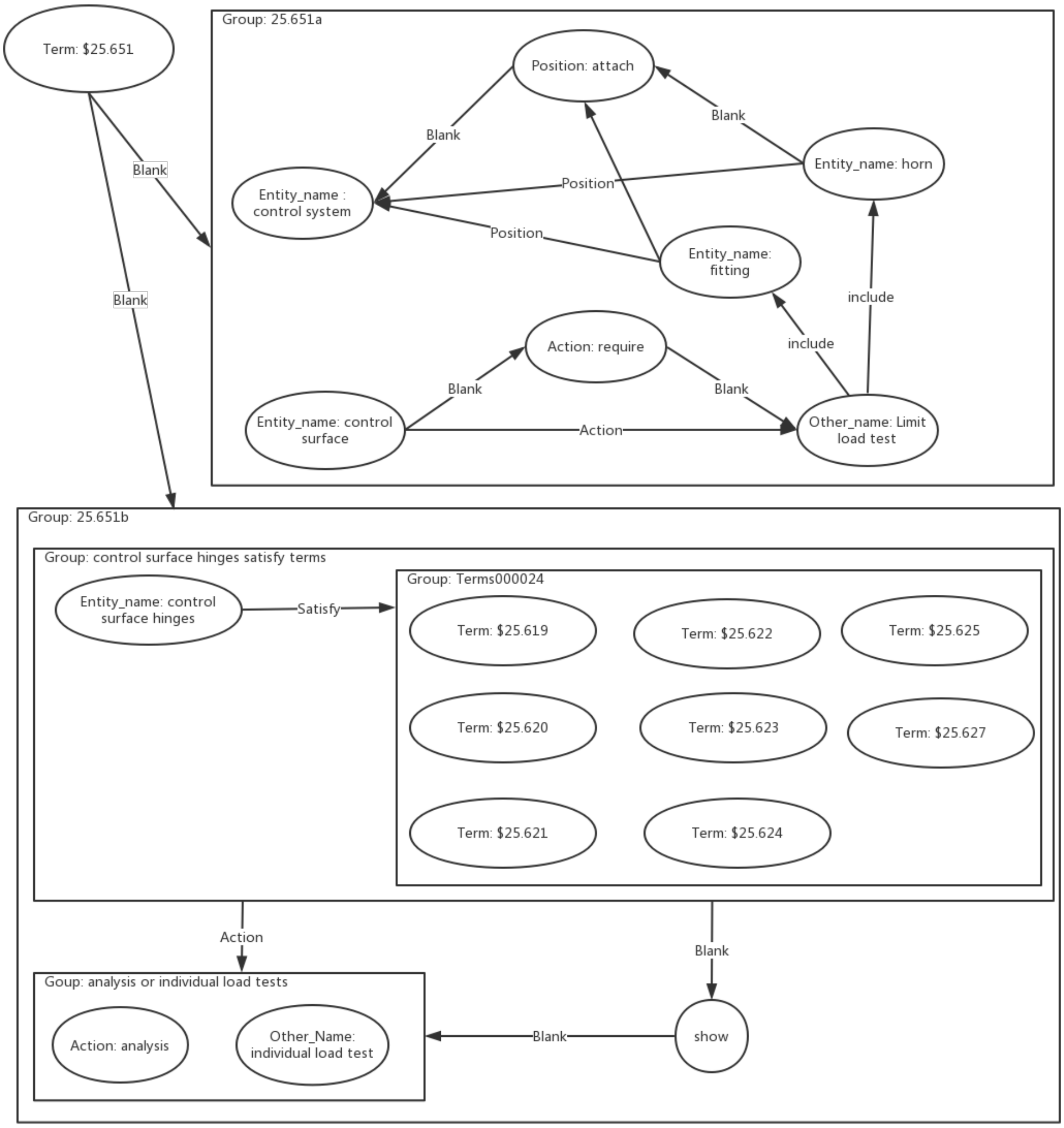

Fig.3 Ontology in Federal regulation part25 sectionD 25.261

If we want to query all the information related to control surface, after import the ontology into Jena database, in the default query webpage, type in SPARAL query string:

```
Select ?relate where{
        prefix:control_surface ?* ?relate.
}
```

And all the information about control surface would come out.

4. Summary

From the result in 3.3, it is obviously that the content of $25.651 has been fully expressed by the ontology.

The method used in this paper, is the first time to apply the knowledge graph to the field of airworthiness rules, it is an innovation. The ontology of knowledge graph is only built from serval regulations, not all of the regulations of Part 25, but if the whole part is on processed, the ontology will be more and more suitable for the field of airworthiness regulations through the continuous iterative update in the later stage. The ultimate goal is to build a system which could replace the work of experts and become an efficient tool to verify whether the design documents are reasonable.

The method of this paper is a landing attempt of the knowledge graph. At present, the research of knowledge graph is mainly distributed in algorithm improvement work of knowledge alignment, automatically extract ontology from information. There are few projects that are actually applied in practical scenario. A detailed introduction to the deployment process in Chapter 3 provides a reference for knowledge graph researchers in the airworthiness field and regulation field.

This paper is supported by Project: Construction of a typical industry industrial Internet enterprise-level centralized security monitoring platform (electronic industry) and Project: General aircraft big data.

References

[1] Zenglin Xu, Yongpan Sheng, Lirong He, Yafang Wang. Knowledge Graph Review. University of Electronic Science and Technology of China, 2016.

[2] Qiao Liu, Yang Li, Hong duan, Yao Liu, Zhiguang Qin. Overview of Knowledge Graph Construction Technology. Computer Research and development, 2016.

[3] Min Xiao. Research on the Construction Method of Domain Ontology. Information Magazine, 2006.

[4] Guiyang Jin, Fuzai Lv, Zhanqin Xiang. Enterprise Information Integration Method Based on Knowledge Graph and Semantic Web Technology.

[5] Guilin Qi, Heng Gao, Tianxing Wu. Progress in knowledge graph. Information Project, 2017.