

基于本体的飞机舵面结构故障诊断方法

袁侃, 胡寿松

(南京航空航天大学 自动化学院, 南京 210016)

摘要 从系统工程的角度分析了飞机系统的复杂性, 将飞机族的概念引入到飞机的本体建模中, 并以舵面故障诊断过程为研究对象, 首先用 Protégé 建立了飞机本体的领域知识模型, 然后将单故障和组合故障的诊断知识列为本体中的 SWRL 规则, 最后利用 JESS 推理出新知识得出诊断结果, 实现了用本体来选择修复方案的过程. 该方法能够实现复杂系统的建模及故障诊断方案的准确选择, 并可通过增加新的诊断知识来完善故障诊断知识库.

关键词 系统工程; 飞机; 故障诊断; 本体; 自修复

Fault diagnosis of aircraft control surface based on ontology

YUAN Kan, HU Shou-song

(College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract The complexity of aircraft system was analyzed from the engineering point of view and the concept of aircraft family was introduced to the ontology modeling of aircraft. At the same time, the fault diagnosis of aircraft control surface was also studied. First, the domain knowledge model of aircraft ontology was built by Protégé. Then the knowledge of single fault and combined faults diagnosis were listed as SWRL rules. Finally, the diagnosis result could be concluded through JESS reasoning to select the right plan of self-repairing. This method can be used to realize the modeling of complex systems and accurate selection of fault diagnosis plan. And the knowledge base of fault diagnosis can also be improved by adding new rules.

Keywords system engineering; aircraft; fault diagnosis; ontology; self-repairing

1 引言

本体为研究者定义了一种通用的术语表用以在某一领域共享信息. 它包括该领域中能被机器解释的基础概念定义以及它们之间的关系, 具备类、属性、关系、函数、公理和实例 6 种基本元素^[1]. 建立本体能够分析领域知识, 并能够在人或软件智能体之间共享关于信息结构的通用理解, 从而促进领域知识的再应用^[2-3]. 本体研究中的一项重要内容是利用语义 Web 规则语言 SWRL 来进行产生式规则的制定. SWRL 是由 OWL 子语言 OWL DL 与 OWL Lite, 以及 Unary/Binary Datalog RuleML 为基础的规则描述语言, 其目的是为了驱使 Horn-like 规则可与 OWL 知识库产生结合, 并利用高度抽象的语法表达 OWL 本体论所描述的知识^[4-6]. JESS 是由美国 Sandlia 国家实验室用 Java 开发的的专家系统推理引擎, 它采用产生式规则作为基本的知识表达模式, 其核心由事实库 (working memory)、规则库 (rule base)、推理机 (inference engine) 三大部分组成. 利用 JESS 可对 SWRL 进行推理, 并将推理所产生的新知识更新到 OWL 知识库中^[7].

飞机是非常庞大的系统, 任何一个子系统的故障都可能带来灾难性的后果. 在众多飞行事故中, 由舵面的结构性故障引起的事故占很大的比例. 因此在考虑飞机的安全性时, 对舵面故障的检测与诊断非常重要. 在某舵面出现结构性故障时, 可以用剩余完好舵面进行重构控制使飞机重新稳定是一种自修复的思想, 美国 NASA 已经分别在 F-15 和 X-36 实验机上验证了这种方法的有效性^[8], 但这种方法在我国尚处于理论研究

收稿日期: 2010-04-17

资助项目: 国家自然科学基金重点项目 (60234010); 航空科学基金 (05E52031)

作者简介: 袁侃 (1979-), 女, 江苏徐州人, 博士, 研究方向: 故障诊断与自修复控制; 胡寿松 (1937-), 男, 江苏南京人, 教授, 博士生导师, 研究方向: 故障诊断、自适应及自修复控制.

阶段. 目前关于飞机舵面故障的诊断方法主要有模糊观测器法^[9], 最小二乘估计方法^[10]等, 但这些方法只考虑了如何进行故障诊断, 并未考虑出现故障后如何进行自修复的决策问题, 另外它们大多假设同一时刻只出现一种故障, 即只考虑了单故障的情况, 而在现实情况中组合故障是完全有可能发生的. 因此, 研究如何对单故障和组合故障进行分组诊断, 并在故障诊断完成后进行修复方案的选择很有意义.

本文从系统工程的角度分析飞机系统的组成, 用飞机族的思想构建飞机系统模型, 并将研究的重点放在舵面故障的诊断上, 首先利用本体论的六大要素建立了飞机系统的本体模型, 然后将舵面的单故障及组合故障的判别方法写成 SWRL 产生式规则, 最后在设置初始条件的情况下, 通过 JESS 推理得出故障诊断的结果, 为后期进行修复方案的制订提供了依据.

2 飞机系统的复杂性分析

系统工程是组织系统的规划、研究、设计制造、试验和使用的科学方法. 它强调科学性、客观性及有目标、有准备、有秩序的工作, 以达到系统整体最优的效果. 根据国内外新机研制情况分析, 飞机的设计研究由于技术知识密集且多学科综合, 具有耗资大、周期长、技术要求高、协作面广等特点, 因此运用系统工程来组织实施其研制工作是十分必要的.

飞机研制的系统工程是制定设计目标、选定设计方案并不断进行验证的过程, 它分为如下几个阶段: 确定任务、探索概念、论证批准、具体设计、飞机试制与鉴定批准成批生产等. 其中, 确定任务与探索概念阶段对新机研制工作的成败至关重要^[11]. 为了满足项目运行成本最低的要求, 需要在设计之初就考虑飞机的建模及构型等因素. 因此, 非常有必要引入飞机族的概念.

飞机族的概念起源于模块化和通用性的思想^[12], 它被定义为一组共享通用部件或子系统的性能或使用要求不同的相关飞机产品系列, 比如波音 737 的 600/700/800/900 型 4 种机型间具有 98% 的机械零部件通用性和 100% 的发动机通用性, 美国联合攻击机 JSF 也是一种在设计之初就考虑不同军种使用要求的战斗机^[13]. 这是未来飞机生产的发展方向^[14], 也为用类和属性的概念建立飞机的本体模型提供了可能^[15]. 因此, 引入飞机族的概念可以有效降低成本, 使资源达到最优化.

3 飞机系统的本体设计

3.1 飞机系统的组成

飞机的组成非常复杂, 需要从系统工程的角度对其进行分析. 按照不同功能系统的地位和重要性可将组成飞机的模块分类为:

- 1) 通用模块: 具有反复使用的基础功能;
- 2) 局部通用模块: 对基本功能具有完善作用;
- 3) 专用模块: 具有某一类型特别需要的功能;
- 4) 接口模块: 扩展其他用途的功能.

以战斗机为例, 其通用模块包括结构模块、飞保模块、动力模块、航电模块和副油箱模块, 专用模块包括卫星通信模块、救生通信模块、参数测量和记录模块等, 接口模块主要负责各模块之间的信息传递. 而将每个模块继续分类, 则可分为更多的下级子模块, 如结构模块可分为机翼模块、机身模块、垂尾模块、平尾模块、起落架模块等, 飞保模块可分为操纵控制模块、液压模块、供电模块、环控模块、救生模块等^[16].

3.2 飞机飞行问题分析

飞机在飞行过程中要考虑到周围环境的影响, 其中环境包括地面和机身周围的气流, 有时还需包括同编队的其它飞机, 因此要考虑这个大系统的整体运行, 必须要同时对环境进行建模. 另外, 不但是飞机自身的子系统间会存在信息交流, 而且飞机与环境间也存在着信息交换, 这些影响因素的繁多导致飞行问题非常复杂. 在飞机的飞行过程中, 最重要的是要保证飞机在按照预定航线飞行的同时完成预定任务, 并且在出现故障时通过故障诊断和紧急修复手段使飞机尽快回复到稳定状态以安全降落.

飞机的舵面故障通常发生在空中飞行时, 因此可暂不考虑地面的影响. 由于舵面故障会引起机身周围的气流参数变化, 从而影响控制律的参数变化, 如果不对控制律进行调整, 很有可能会造成机毁人亡的后果. 本文继承了对故障飞机进行直接自修复的思想, 即在修复过程中, 可以先不进行故障辨识, 重构在发生故障的第一时间内就产生作用, 从而使飞机在最短时间内恢复至稳定状态^[17]. 为了选择合适的故障方案, 故障诊断

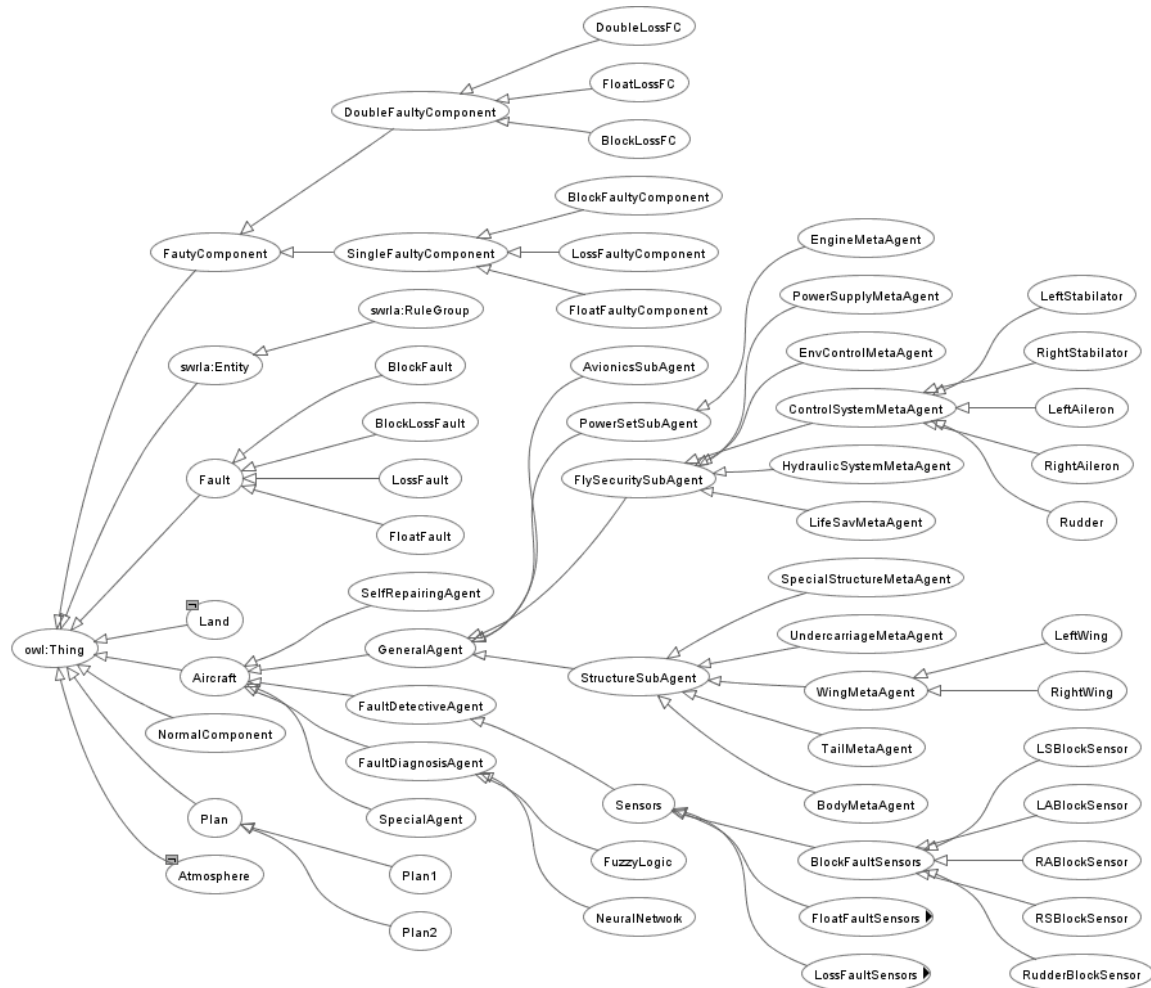


图 1 某型飞机本体关系图

系统必须首先确定故障发生的位置和类型, 因此需为不同故障设置相应的传感器, 这就为本体的建立提出了一定的约束条件.

3.3 飞机实例本体的建立

本文主要考虑飞机舵面的三种主要故障, 即卡死、浮松和缺损, 当舵面发生不同的故障时, 将利用剩余完好舵面实施不同的修复方案. 假设飞机舵面的故障传感器得到的信息准确无误, 其值为 true 时代表相应的故障发生, 值为 false 时代表该舵面正常无故障. 本体的建立过程如下 (本体关系图如图 1 所示):

1) 首先建立飞机本体的类描述, 将飞机系统各层级的每一个功能模块都定义为一个类, 并为每个类建立相应的实体, 以具有 5 个操纵面 (左副翼、右副翼、左平尾、右平尾、方向舵) 的典型飞机为研究对象.

2) 建立飞机的类属性和数据属性, 比如类属性 is_part_of 代表着某些子模块只能从属于特定的父模块, 则传感器类 Sensors 只能是故障检测类 FaultDetectiveAgent 的子类; 数据类型以机翼类 Wing 为例, 可分为左翼 LeftWing 和右翼 RightWing 两个子类, 则对于 F16 来说, 其机翼具有数据属性: 翼展 WingSpan=9.144, 参考翼面积 WingArea=27.87, 平均动力弦 cbar=3.45 等; 传感器具有 hasTest 的数据属性, 当 hasTest=true 时代表该传感器检测到故障, 当 hasTest=false 时代表该传感器正常.

3) 建立关系和公理, 如故障类中分为单故障和组合故障两类, 这两类应为互斥的关系 (disjoints), 即单故障和组合故障不属于同类故障, 另外, 传感器只能检测相应舵面的故障值, 如左副翼的卡死传感器 LABlockSensor 只能检测左副翼的故障值, 即在约束中规定 “LABlockSensor sense only LeftAileron”.

✓ Single-LABlockFault	→ LABlockSensor(?x) ∧ LeftAileron(?y) ∧ hasTest(?x, true) ∧ LAFloatSensor(?z) ∧ hasTest(?z, false) ∧ LALossSensor(?a) ∧ hasTest(?a, false) ∧ RightAileron(?b) ∧ LeftStabilator(?c) ∧ RightStabilator
✓ Single-LAFloatFault	→ LAFloatSensor(?x) ∧ LeftAileron(?y) ∧ hasTest(?x, true) ∧ LABlockSensor(?z) ∧ hasTest(?z, false) ∧ LALossSensor(?a) ∧ hasTest(?a, false) ∧ RightAileron(?b) ∧ LeftStabilator(?c) ∧ RightStabilator
✓ Single-LALossFault	→ LALossSensor(?x) ∧ LeftAileron(?y) ∧ hasTest(?x, true) ∧ LABlockSensor(?z) ∧ hasTest(?z, false) ∧ LAFloatSensor(?a) ∧ hasTest(?a, false) ∧ RightAileron(?b) ∧ LeftStabilator(?c) ∧ RightStabilator
✓ Single-LSBlockFault	→ LSBlockSensor(?x) ∧ LeftStabilator(?y) ∧ hasTest(?x, true) ∧ LSFloatSensor(?z) ∧ hasTest(?z, false) ∧ LSLossSensor(?a) ∧ hasTest(?a, false) ∧ RightStabilator(?b) ∧ LeftAileron(?c) ∧ RightAileron
✓ Single-LSFloatFault	→ LSFloatSensor(?x) ∧ LeftStabilator(?y) ∧ hasTest(?x, true) ∧ LSBlockSensor(?z) ∧ hasTest(?z, false) ∧ LSBlockSensor(?a) ∧ hasTest(?a, false) ∧ RightStabilator(?b) ∧ LeftAileron(?c) ∧ RightAileron
✓ Single-LSLossFault	→ LSLossSensor(?x) ∧ LeftStabilator(?y) ∧ hasTest(?x, true) ∧ LSFloatSensor(?z) ∧ hasTest(?z, false) ∧ LSBlockSensor(?a) ∧ hasTest(?a, false) ∧ RightStabilator(?b) ∧ LeftAileron(?c) ∧ RightAileron
✓ Single-RABlockFault	→ RABlockSensor(?x) ∧ RightAileron(?y) ∧ hasTest(?x, true) ∧ RALossSensor(?z) ∧ hasTest(?z, false) ∧ RAFloatSensor(?a) ∧ hasTest(?a, false) ∧ LeftAileron(?b) ∧ LeftStabilator(?c) ∧ RightStabilator
✓ Single-RAFloatFault	→ RAFloatSensor(?x) ∧ RightAileron(?y) ∧ hasTest(?x, true) ∧ RALossSensor(?z) ∧ hasTest(?z, false) ∧ RABlockSensor(?a) ∧ hasTest(?a, false) ∧ LeftAileron(?b) ∧ LeftStabilator(?c) ∧ RightStabilator
✓ Single-RALossFault	→ RALossSensor(?x) ∧ RightAileron(?y) ∧ hasTest(?x, true) ∧ RAFloatSensor(?z) ∧ hasTest(?z, false) ∧ RABlockSensor(?a) ∧ hasTest(?a, false) ∧ LeftAileron(?b) ∧ LeftStabilator(?c) ∧ RightStabilator
✓ Single-RSBlockFault	→ RSBlockSensor(?x) ∧ RightStabilator(?y) ∧ hasTest(?x, true) ∧ RSLossSensor(?z) ∧ hasTest(?z, false) ∧ RSFloatSensor(?a) ∧ hasTest(?a, false) ∧ LeftStabilator(?b) ∧ LeftAileron(?c) ∧ RightAileron
✓ Single-RSFloatFault	→ RSFloatSensor(?x) ∧ RightStabilator(?y) ∧ hasTest(?x, true) ∧ RSLossSensor(?z) ∧ hasTest(?z, false) ∧ RSBlockSensor(?a) ∧ hasTest(?a, false) ∧ LeftStabilator(?b) ∧ LeftAileron(?c) ∧ RightAileron
✓ Single-RSLossFault	→ RSLossSensor(?x) ∧ RightStabilator(?y) ∧ hasTest(?x, true) ∧ RSFloatSensor(?z) ∧ hasTest(?z, false) ∧ RSBlockSensor(?a) ∧ hasTest(?a, false) ∧ LeftStabilator(?b) ∧ LeftAileron(?c) ∧ RightAileron
✓ Single-RudderBlockFault	→ RudderBlockSensor(?x) ∧ Rudder(?y) ∧ hasTest(?x, true) ∧ RudderLossSensor(?z) ∧ hasTest(?z, false) ∧ RudderFloatSensor(?a) ∧ hasTest(?a, false) ∧ LeftStabilator(?b) ∧ RightStabilator(?c) ∧ LeftAileron
✓ Single-RudderFloatFault	→ RudderFloatSensor(?x) ∧ Rudder(?y) ∧ hasTest(?x, true) ∧ RudderLossSensor(?z) ∧ hasTest(?z, false) ∧ RudderBlockSensor(?a) ∧ hasTest(?a, false) ∧ LeftStabilator(?b) ∧ RightStabilator(?c) ∧ LeftAileron
✓ Single-RudderLossFault	→ RudderLossSensor(?x) ∧ Rudder(?y) ∧ hasTest(?x, true) ∧ RudderFloatSensor(?z) ∧ hasTest(?z, false) ∧ RudderBlockSensor(?a) ∧ hasTest(?a, false) ∧ LeftStabilator(?b) ∧ RightStabilator(?c) ∧ LeftAileron

图 2 单故障判别的 SWRL 规则

3.4 建立 SWRL 规则

根据故障诊断的过程建立 SWRL 规则:

1) 建立单故障的判别规则:

如图 2 所示, 以诊断左副翼卡死故障为例, 该规则 Single-LABlockFault 可解释为, 当左副翼卡死故障的传感器检测到故障, 而其它舵面均正常时, 则判定为左副翼卡死故障, 即:

$$\begin{aligned} & \text{LABlockSensor}(?x) \wedge \text{LeftAileron}(?y) \wedge \text{hasTest}(?x, \text{true}) \wedge \text{LAFloatSensor}(?z) \wedge \text{hasTest}(?z, \text{false}) \\ & \wedge \text{LALossSensor}(?a) \wedge \text{hasTest}(?a, \text{false}) \wedge \text{RightAileron}(?b) \wedge \text{LeftStabilator}(?c) \wedge \text{RightStabilator}(?d) \\ & \wedge \text{Rudder}(?e) \wedge \text{NormalComponent}(?b) \wedge \text{NormalComponent}(?c) \wedge \text{NormalComponent}(?d) \\ & \wedge \text{NormalComponent}(?e) \rightarrow \text{BlockFaultyComponent}(?y), \end{aligned}$$

其中判定某舵面为正常的规则可解释为, 当该舵面的卡死、松浮及缺损故障传感器均未检测到故障时, 可判断该舵面处于正常状态, 如判断右副翼正常的规则是:

$$\begin{aligned} & \text{RightAileron}(?x) \wedge \text{RABlockSensor}(?y) \wedge \text{RAFloatSensor}(?z) \wedge \text{RALossSensor}(?a) \wedge \\ & \text{hasTest}(?y, \text{false}) \wedge \text{hasTest}(?z, \text{false}) \wedge \text{hasTest}(?a, \text{false}) \rightarrow \text{NormalComponent}(?x). \end{aligned}$$

2) 建立组合故障的判别规则:

组合故障可分为可修复故障和不可修复故障, 将可修复的组合故障列为诊断规则, 以左平尾和右平尾同时出现缺损故障为例, 判断出现组合的故障的规则 Double-LRSLossFault 为:

$$\begin{aligned} & \text{LeftStabilator}(?x) \wedge \text{RightStabilator}(?y) \wedge \text{LSLossSensor}(?z) \wedge \text{hasTest}(?z, \text{true}) \\ & \wedge \text{RSLossSensor}(?a) \wedge \text{hasTest}(?a, \text{true}) \rightarrow \text{DoubleLossFC}(?x) \wedge \text{DoubleLossFC}(?y). \end{aligned}$$

3.5 利用 JESS 进行推理

本文采用 Protégé 3.4.4 版本建立本体, 并用 JESS7.0p2 根据所有建立的 SWRL 规则进行推理, 当设置左副翼卡死传感器检测到故障时, 结果如图 3 所示。

Inferred Axioms
http://www.owl-ontologies.com/Ontology1269702681.owl#NormalComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16RightAileron)
http://www.owl-ontologies.com/Ontology1269702681.owl#NormalComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16LeftStabilator)
http://www.owl-ontologies.com/Ontology1269702681.owl#NormalComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16Rudder)
http://www.owl-ontologies.com/Ontology1269702681.owl#BlockFaultyComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16LeftAileron)
http://www.owl-ontologies.com/Ontology1269702681.owl#NormalComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16RightStabilator)

图 3 左副翼出现卡死故障, 而其它舵面均正常

当设置左平尾、右平尾缺损传感器检测到故障值时, 采用第 1 号修复方案 Plan1, 结果如图 4 所示。

如在实际诊断过程中出现了不可修复的组合故障, 则 JESS 无法推理出新规则, 可对飞行员提出报警, 告知其该组合故障不可修复, 建议实施其它紧急方案。

Inferred Axioms
http://www.owl-ontologies.com/Ontology1269702681.owl#DoubleLossFC(http://www.owl-ontologies.com/Ontology1269702681.owl#F16RightStabilator)
http://www.owl-ontologies.com/Ontology1269702681.owl#NormalComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16Rudder)
http://www.owl-ontologies.com/Ontology1269702681.owl#DoubleLossFC(http://www.owl-ontologies.com/Ontology1269702681.owl#F16LeftStabilator)
http://www.owl-ontologies.com/Ontology1269702681.owl#Plan1(http://www.owl-ontologies.com/Ontology1269702681.owl#F16LeftStabilator)
http://www.owl-ontologies.com/Ontology1269702681.owl#NormalComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16RightAileron)
http://www.owl-ontologies.com/Ontology1269702681.owl#NormalComponent(http://www.owl-ontologies.com/Ontology1269702681.owl#F16LeftAileron)

图 4 左、右平尾缺损组合故障诊断结果

4 结论

本文用本体的领域知识来建立飞机模型,使飞机各功能模块及子模块之间的关系更加清晰,并利用 JESS 对 SWRL 规则进行推理所得出的新知识进行舵面故障诊断,为故障修复的决策制定提供了依据,这种方法可以通过增加新的 SWRL 规则来完善故障诊断知识库,从而提高故障诊断的准确性。

参考文献

- [1] Noy N, McGuinness D. Ontology development 101: A guide to creating your first ontology[R]. Technical Report, Citeseer, 2001.
- [2] 杜小勇, 李曼, 王珊. 本体学习研究综述 [J]. 软件学报, 2006, 17(9): 1837-1847.
Du X Y, Li M, Wang S. A survey on ontology learning research[J]. Journal of Software, 2006, 17(9): 1837-1847.
- [3] Uschold M, Gruninger M. Ontologies: Principles, methods and applications[J]. The Knowledge Engineering Review, 2009, 11(2): 93-136.
- [4] McGuinness D, Van Harmelen F. OWL web ontology language overview[R]. Technical Report, W3C Recommendation, 10 February 2004.
- [5] Horrocks I, Patel-Schneider P, et al. SWRL: A semantic web rule language combining OWL and RuleML[R]. Technical Report, W3C Member Submission, 21 May 2004.
- [6] O'Connor M, Shankar R, et al. Efficiently querying relational databases using OWL and SWRL[J]. Web Reasoning and Rule Systems, 2007: 361-363.
- [7] Laboratories S N. JESS, the Rule Engine for the Java Platform[EB/OL]. <http://Herzberg.ca.sandia.gov/jess>, 2009.
- [8] Eslinger R, Chandler P. Self-repairing flight control system program overview[C]// Dayton: Proceedings of the IEEE National Aerospace and Electronics Conference, 1988: 504-511.
- [9] 周川, 胡维礼, 陈庆伟, 等. 飞机舵面结构故障检测与重构的模糊观测器方法 [J]. 信息与控制, 2001(S1): 703-709.
Zhou C, Hu W L, Chen Q W, et al. Fault detection and reconfigurable control for aircraft based on fuzzy observers[J]. Information and Control, 2001(S1): 703-709.
- [10] 苏浩秦, 宋述杰, 邓建华. 基于限制最小二乘估计的飞机舵面故障诊断方法 [J]. 机械科学与技术, 2005, 24: 1033-1035.
Su H Q, Song S J, Deng J H. Detection and diagnosis method of impairment effectors based on the constraint least square method[J]. Mechanical Science and Technology, 2005, 24: 1033-1035.
- [11] 何永钧. 通过实践探索适用于飞机研制的系统工程 [J]. 航空系统工程, 1995(3): 20-25.
He Y J. Exploration of applicable system engineering for aircraft research through practice[J]. Aeronautics System Engineering, 1995(3): 20-25.
- [12] Willcox K, Wakayama S. Simultaneous optimization of a multiple-aircraft family[J]. Journal of Aircraft, 2003, 40(4): 616-622.
- [13] 雍明培, 余雄庆. 基于模块化产品平台的飞机族设计技术探讨 [J]. 飞机设计, 2006(4): 30-37.
Yong M P, Yu X Q. Aircraft family design using modular product platform methodology — An exploratory study[J]. Aircraft Design, 2006(4): 30-37.
- [14] Fuller J, Aeronautics L, Texas F. The role of manned aircraft in the future[C]// Dayton: AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Years, 2003.
- [15] Allison J, Roth B, et al. Aircraft family design using decomposition-based methods[R]. Technical Report, Citeseer, 2006.
- [16] 党育辉. 通用型舰载支援飞机的模块化布局设计概念 [J]. 飞机工程, 2006(2): 10-15.
Dang Y H. Modular design concept of carrier based common support aircraft[J]. Aircraft Engineering, 2006(2): 10-15.
- [17] 葛铁. 多故障的直接自修复控制 [D]. 南京: 南京航空航天大学自动化学院, 2002.
Ge T. Direct self-repairing control of multiple faults[D]. Nanjing: College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, 2002.