

面向多航天器协同的自主任务规划方法研究

陈丹¹, 程伟¹, 高延超¹, 王怡健²

(1. 北京空间技术研制试验中心, 北京 100094; 2. 北京卫星制造厂有限公司, 北京 100094)

摘要: 针对多航天器的协同任务规划, 设计了一种基于多 Agent 技术的体系结构, 将航天器任务体系分为多航天器协同层、系统层和分系统层的多层结构, 分析了同构式多航天器以及异构式多航天器的任务差异, 给出了多航天器协同任务模型; 在此基础上, 提出了多航天器的招投标任务协商策略和评估发布协商策略, 以及基于 SGPLAN 的任务协同算法; 以多航天器进行推进补加问题为例通过仿真进行了验证, 表明了模型的合理性以及算法的有效性, 为多航天器协同任务实际应用提供了一种可行的方法。

关键词: 多航天器; 协同; 任务规划

Research on Self-organization Mission Planning for Multi-spacecraft Coordination

Chen Dan¹, Cheng Wei¹, Gao Yanchao¹, Wang Yijie²

(1. Institute of Manned Space System Engineering, China Academy of Space Technology, Beijing 100094, China;

2. Beijing manufacturer of Space Crafts, China Academy of Space Technology, Beijing 100094, China)

Abstract: A novel spacecraft autonomy architecture based on Multi-agent is put forward to fulfill the self-organization mission planning demands of multi-spacecraft-coordination. The architecture is divided into Multi-spacecraft-coordination Layer, Spacecraft System Layer and Spacecraft subsystem Layer. The difference between isomorphic spacecraft and heterogeneous spacecraft is analysed, and the model of multi-spacecraft coordination has been taken. On this basis, bidding mechanism and estimate-publishment mechanism is designed for the two kind of multi-spacecraft mission planning, and the algorithm based of SG Plan. has been taken. At last, multi-spacecraft propellant refueling problem was taken as an example, and satisfactory results were obtained through simulation, which shows that the model was reasonable and the algorithm was efficient. The self-organization mission planning in this article can be referred to real multi-spacecraft-coordination application.

Keywords: multi-spacecraft; coordination; mission planning

0 引言

随着载人深空探测活动的深入推进, 未来航天器的任务在高可靠性和高安全性的基础上, 面临着系统规模庞大且功能复杂、信息类别多且数据量大、天地通信约束严苛(高时延、带宽有限、通信窗口限定)、地面操控难度大等新形势。现有航天器的局部自主的技术水平难以满足后续任务需求, 在轨航天器需要自主规划并执行命令, 不断根据实施的情况和环境的变化做出合理的调整, 最终完成目标任务。多个航天器配合的任务尤其需要航天器的自主运行能力。

目前国内外的研究中, 针对航天器的任务规划的研究主要针对单航天器的研究和应用, 如 NASA 应用在观测卫星 EO-1 的 CASPER 系统^[1], 应用在天文望远镜 HST、

JWST 的 SPIKE 系统^[2]、ESA 应用在无人探测器 Rosetta 的 RSSC 系统^[3], 国内应用在某遥感卫星的基于有向图的任务指令生成算法^[4]。针对多航天器的任务规划研究和应用较少^[5-6]。本文在分析多航天器任务特点基础上, 重点研究多航天器的协同任务规划问题, 对后续大规模航天活动具有理论和应用意义。

1 任务分析

多航天器任务按照航天器交互方式, 可分为两种: 一种是同构式航天器交互, 多航天器作为有机整体, 接到统一任务后, 将任务按协商分配或者招投标的方式向单个航天器派出任务; 一种是异构式航天器交互, 多航天器之间有机交互共同完成任务。

同构式航天器一般选择某一航天器作为决策航天器, 其余航天器为成员航天器: 决策航天器接收地面的统一任务, 负责与各成员航天器通信、协商, 并发布任务信息, 其作为任务分配的组织者和决策者, 拥有共享的任务信息。成员航天器负责接收由决策航天器发来的任务需求, 根据

收稿日期: 2018-11-12; 修回日期: 2018-12-10。

基金项目: 载人航天器领域预先研究项目(17140114)。

作者简介: 陈丹(1983-), 男, 高级工程师, 主要从事航天器信息系统设计方向的研究。

自身实际情况进行约束评估, 决定是否申请任务。决策航天器根据各成员航天器申请任务情况确定完成任务的航天器并进行握手确认^[7]。同构式航天器交互方式框图见图 1 所示。

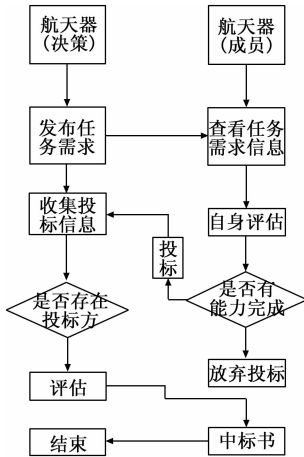


图 1 同构式航天器交互框图

异构式航天器需要多航天器间互相配合完成共同的任 务, 可采用分层级的方式进行协同任务: 高层级的航天器接收地面的统一任务, 根据统一任务需求进行约束评估, 并向低层级的航天器发布任务信息。低层级的航天器接收任务信息后, 根据自身实际情况进行约束评估, 决定是否接收任务。高层级航天器根据低层次航天器任务接收情况进行最终的任务确认和发布。异构式航天器交互方式框图见图 2 所示。

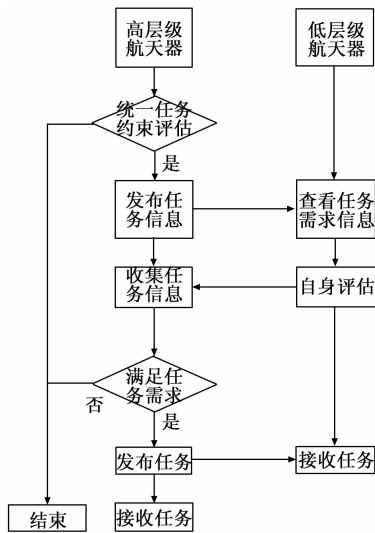


图 2 异构式航天器交互框图

2 多航天器协同的任务结构设计

2.1 多航天器协同任务结构分析

多航天器系统作为一个整体自主完成任务目标, 多航天器之间的任务协同, 可将单个航天器视为单一的 Agent,

借鉴多 Agent 系统 (Multi-Agent System, MAS) 架构进行航天器间的任务协商。多航天器系统的协同任务结构见图 3 所示。

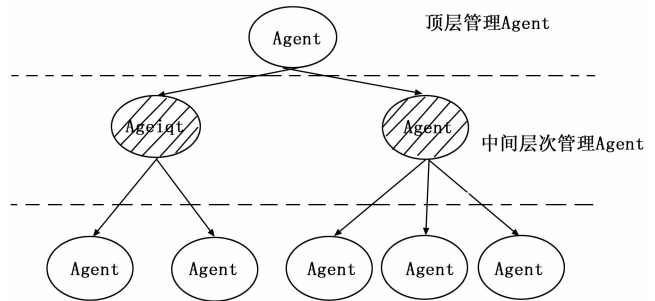


图 3 集中式多 Agent 系统示意图

2.2 多航天器任务结构分析

多航天器协同的自主任务体系结构从逻辑上分为三层: 底层针对航天器各分系统, 由各个自主的分系统 Agent 完成状态监测和局部控制; 中层针对单个航天器整体, 完成航天器的系统级自主; 顶层是多航天器协同, 决策、规划调度任务。多航天器协同的自主任务体系结构见图 4 所示。

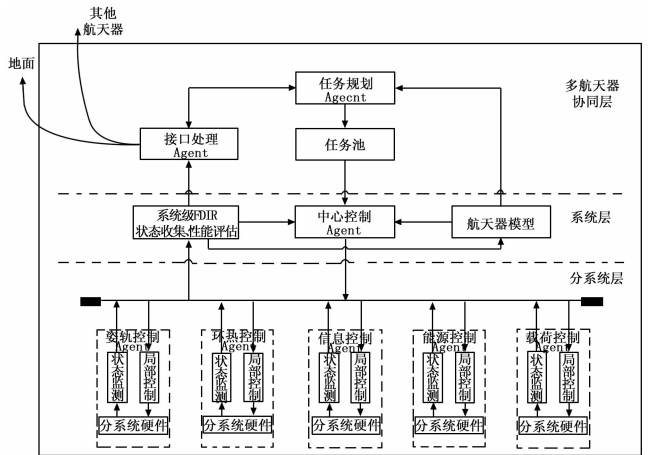


图 4 多航天器协同的自主任务体系结构图

2.2.1 多航天器协同层

多航天器协同层进行与各航天器的交互协同, 完成航天器的任务规划, 包含任务规划 Agent、接口处理 Agent 和任务池。

接口处理 Agent 负责航天器间信息的传递, 一方面接收其它航天器 Agent 的信息, 另一方面将接收到的系统级 FDIR 实时状态信息传递给其它的航天器 Agent。此外, 接口处理 Agent 接收地面命令, 保证地面的控制介入能力。接口处理 Agent 接收系统级 FDIR 实时反馈的状态信息,

任务规划 Agent 接收来自接口处理 Agent 的任务, 与航天器模型交互进行任务约束检查, 完成任务规划和分解, 主要包括多航天器系统协同任务向航天器任务的分解, 系统重构过程中的任务重新规划。任务规划 Agent 还完成对任务的调度, 当多个相互冲突的任务同时发生时, 任务规

划 Agent 根据任务优先级进行排序, 达到消解冲突的目的。

任务池存放任务规划 Agent 的合规性任务, 既作为下层 Agent 执行任务的输入, 也接收更新任务的执行完成情况。任务池接收地面的管理操作, 执任务删除、调整等命令。

2.2.2 系统层

航天器系统层接收多航天器协同层的任务信息, 对航天器进行整体协调和控制。航天器系统层包括中心控制 Agent、系统级 FDIR 和航天器模型。

中心控制 Agent 是航天器内部任务规划的控制中心, 也是多航天器协同层与分系统层的纽带, 将多航天器协同层分配的任务进行分解, 根据航天器模型以及资源使用情况将任务解析为动作, 直至解析到详细的控制指令序列, 并传递给下层的分系统 Agent。同时, 协调各个分系统 Agent, 解决各个分系统自主工作引起的资源冲突。

系统级 FDIR 从整体上对航天器进行状态监测、性能评估和故障诊断, 并在出现故障时给出解决方案。

航天器模型反映航天器设计状态, 实时接收各分系统 Agent 的状态信息进行模型状态对比, 将结果反馈上层的任务规划 Agent 和本级的中心控制 Agent 用于任务规划和解析。

2.2.3 分系统层

航天器通常由多个分系统组成, 它们具有相对独立的功能, 能完成各自的子任务。分统层由分系统 Agent 组成, 各分系统 Agent 的功能如下:

- 1) 姿轨控制 Agent, 进行姿态捕获、姿态和轨道确定及控制;
- 2) 环热控制 Agent, 测量航天器温度、湿度、压力、浓度并进行调节;
- 3) 信息控制 Agent, 实现航天器的数据管理、传输和链路控制;
- 4) 能源控制 Agent, 管理航天器发电和配电;
- 5) 载荷控制 Agent, 对有效载荷进行监测和控制。

各分系统 Agent 的结构相同, 均包含状态监测、局部控制、系统软硬件三个部分。各分系统 Agent 一方面接受上层控制指令并完成相应的功能; 另一方面对外部环境的变化做出反应, 处理异常情况, 实现自我管理。

3 自主任务规划算法设计

自主任务管理的核心算法是求解规划问题, 目前用于规划问题的求解算法主要有图规划、基于启发式状态空间搜索的规划、约束可满足规划、分层规划和基于模型检测的规划等。任务规划可分为正常任务规划和动态任务规划, 正常任务规划可采用 SGPlan 算法用于单个任务的解析, 动态任务规划可用于紧急或故障后的任务排序。

SGPlan 是一款基于目标分解技术的规划器, 他由 Yixin

Chen 等人联合开发, 最开始在 2004 年提出, 并陆续优化^[6]。SGPlan 的基本思想将原始问题分解为多个子问题, 每个子问题都有对应的子目标, 然后利用改进的 Metric-FF 来求解各个子问题, 最后消解子目标间的冲突。

假定单个航天器接收到某一任务, 任务需要相关功能 A、B、C 协同完成。按照 SGPlan 算法, 将任务 P 分解为子任务 SA、SB、SC, 子任务 SA、SB、SC 包含动作 $[a_1] \sim [a_n]$ 、 $[b_1] \sim [b_n]$ 、 $[c_1] \sim [c_n]$ 以及状态 $a_1 \sim a_n$ 、 $b_1 \sim b_n$ 、 $c_1 \sim c_n$, 具体见图 5 所示。

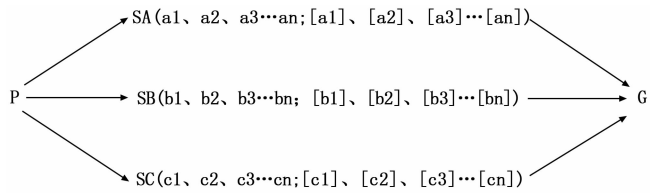


图 5 任务分解到子任务

航天器完成任务规划即是从初始状态 P 在满足各种约束的条件下走向目标状态 G 的可行路径。约束指的是状态、动作之间的相互逻辑关系。对于比较确定的任务, 可以将 P 到 G 的路径提前确定, 航天器在轨规划时只需根据当前的任务参数求解路径中的动作时间。对于仅明确约束的任务, 可采用路径搜索的方法规划各种路径, 然后与约束匹配, 得出可行路径。

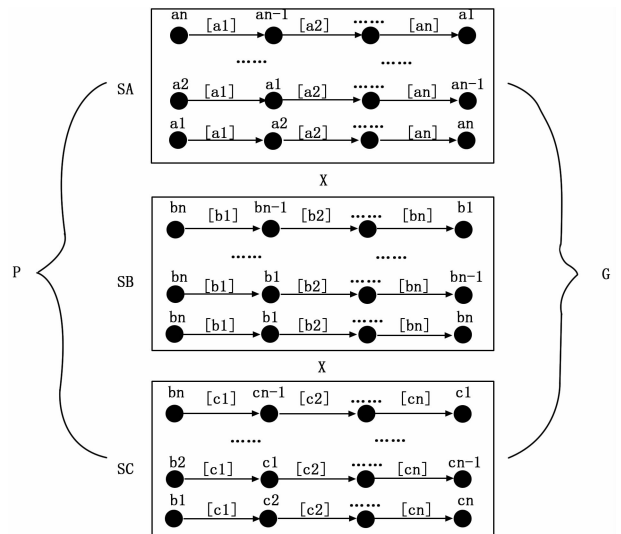


图 6 任务路径求解搜索示意图

对于航天器发生故障或者有临时任务需要调整任务执行的, 可采用启发式搜索方法, 将任务排序关注的设定为启发评估值 $H(a)$, 例如将 $H(a)$ 设定为采用 R_m 任务后可执行任务的数量。算法示意图如图 7 所示。

图中, S 为起始节点, 其余每个节点均有 $H(a)$ 用于评估节点值, 从 S 节点出发向下扩展节点, 从子节点中选取 $H(a)$ 为最大的节点, 并将该节点设为新的初始节点,

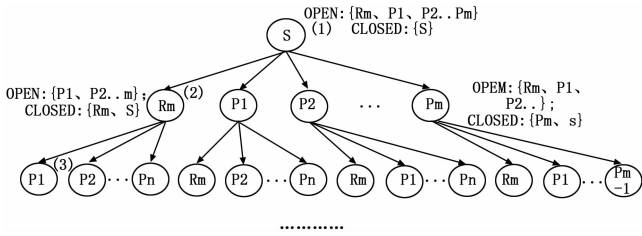


图 7 加入修正任务计划序列后的任务排序算法示意

不断循环，最终求解出任务序列。

4 仿真算例

为了对多航天器协同的自主任务规划方法进行验证，选取三个航天器协同在轨进行推进剂补加的应用场景进行仿真，包括核心控制航天器 A、携带燃料的补加航天器 B、待补加航天器 C。航天器 A 为停靠、中转平台；航天器 C 在轨大部分时间为独立飞行，可与航天器 A 进行对接，停靠后开展燃料补加、维修等任务；航天器 B 为补加航天器，由地面按需发射，入轨后与航天器 A 对接，用于向航天器 A、C 提供推进燃料等资源。

推进补加需要 A、B、C 三个航天器协同完成，因此采用异构式航天器的合作式协同方式（见图 2）。仿真场景设定为：航天器 C 推进剂剩余量低于门限值，向航天器 A 发出补加协同请求，航天器 A 进行整体地统一任务约束评估通过后，向航天器 B 和 C 发布补加任务信息；航天器 B、C 接收任务后进行自身资源及健康状态评估，航天器 A 根据各航天器的评估结果确认是否满足本次任务需求，并向航天器 A、B、C 发布任务；航天器 A、B、C 根据统一任务指令协同开展任务规划，进一步分解为航天器内部的子任务、动作和指令。任务规划过程中，若单个航天器的状态不满足任务要求时，再提交航天器 A 进行统一的再协同，直至任务通过或任务终止。

针对推进补加仿真算例，建立航天器 Agent 及航天器内部分系统 Agent。每个航天器内部包括任务规划 Agent、接口处理 Agent、中心控制 Agent、系统 FDIR，并根据任务需求配置姿轨控制 Agent、能源控制 Agent 和推进补加 Agent，见图 8 所示。

仿真任务过程包括交会、对接、推进补加三个过程，其中航天器 A、航天器 C 全程参与三个过程，航天器 B 仅参加推进补加一个过程。因此，航天器 A、C 的功能 Agent 包含了能源控制 Agent、姿轨控制 Agent 和推进补加 Agent，航天器 B 仅建立能源控制 Agent 和推进补加 Agent。

采用基于 FIPA 规范^[9]的 JADE 平台^[10]对航天器 A、航天器 B、航天器 C 的任务协同过程进行仿真，通过仿真，各航天器的中心控制 Agent 均接受了航天器级的任务，仿真结果见图 9 所示。图中，REQUEST 表示请求通信方式，INFORM 表示通知通信方式。

航天器完成各航天器间的协同后，中心控制体 Agent

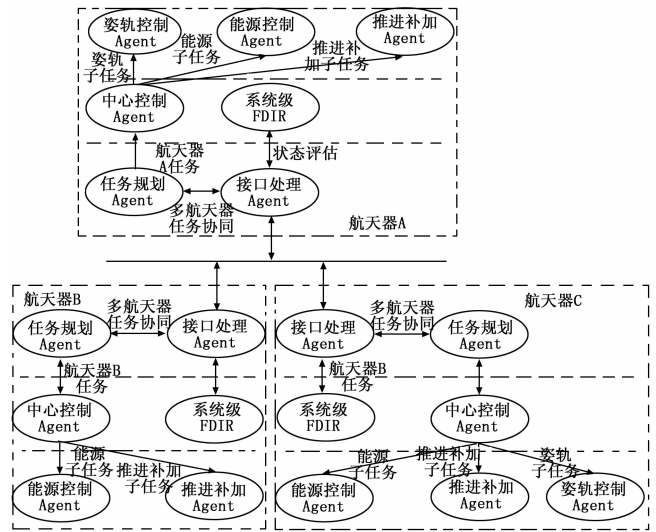


图 8 推进补加多 Agent 仿真模型

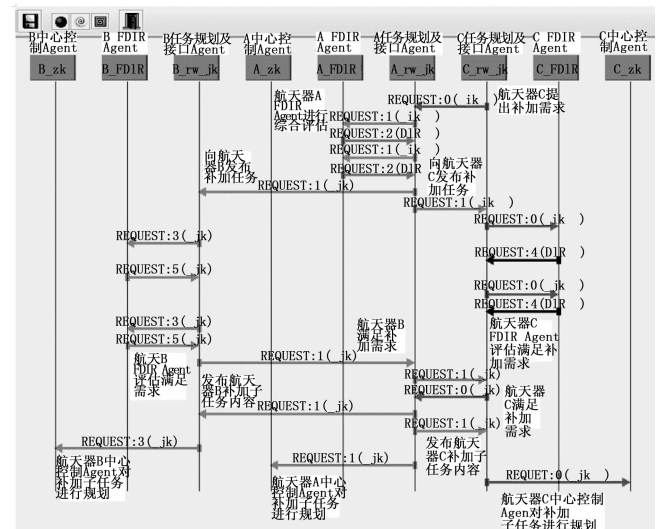


图 9 推进补加过程中各 Agent 间协同过程

进行任务解析，以航天器 A 在补加过程的任务解析为例，补加任务解析为能源子任务、姿轨控制子任务、补加子任务，每个子任务又可解析为相关动作，见图 10 所示。图中，能源子任务 a2 是姿轨子任务进行姿态调整动作的前置约束，姿轨子任务 b2 是能源子任务进行帆板跟踪的前置约束，b3 是补加子任务进行真空放气动作的前置约束。

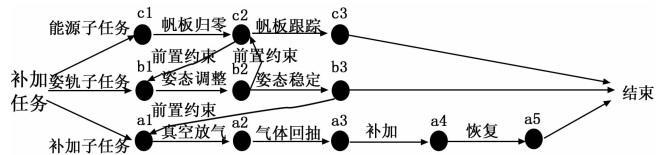


图 10 航天器 A 补加任务解析图

将任务的起始时间代入，按照任务解析图的逻辑关系将动作解析成指令，完成任务到指令的规划。采用 Matlab 进行仿真，结果见图 11、图 12 所示。

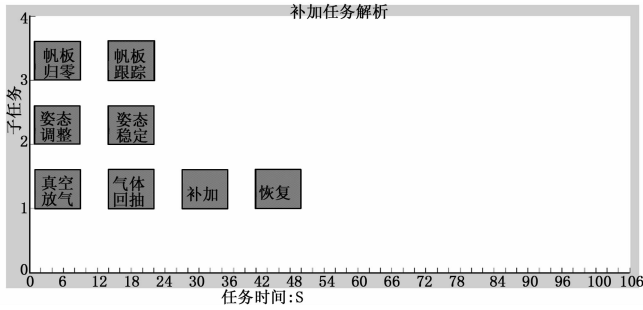


图 11 补加任务解析前动作图 (初始)

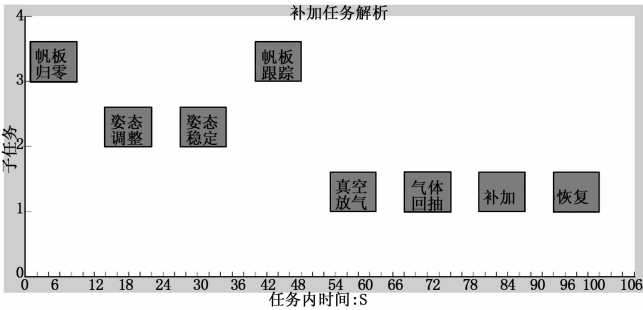


图 12 补加任务解析前动作图 (完成)

由图可见, 程序按照图 10 进行规划解析, 运算得到: ①帆板归零、②姿态调整、③姿态稳定、④帆板跟踪、⑤真空放气、⑥气体回抽、⑦补加、⑧恢复的动作指令序列。

5 结论

本文提出了一种多航天器协同的自主任务体系结构, 给出了基于 Agent 的体系结构的基本框架和功能描述, 并进行了算法设计, 最终以三个航天器协同开展推进剂补加为例进行了验证。仿真结果证明, 本文提出的基于 Agent 的

协同架构以及基于 SGplan 的算法可以满足多航天器协同自主任务规划需求。

参考文献:

[1] Chien S, Rabideau G, Knight R, et al. ASPEN—Automating space mission operations using automated planning and scheduling [A]. SpaceOps2000 [C]. Colorado, 2000.

[2] Calvani H M, Berman A F, Blair W P, et al. The Evolution of the FUSE Spike Long Range Planning System [A]. Proceedings of the 4th International Workshop on Planning and Scheduling for Space [C]. Darmstadt, 2004; 25-31.

[3] Chien S, Rabideau G, Tran D, et al. Scheduling Science Campaigns for the Rosetta Mission: A Preliminary Report [A]. International Workshop on Planning and Scheduling for Space (IWSS 2013) [C]. Moffett Field, 2013.

[4] 田志新, 崔晓婷, 郑国成, 等. 基于有向图模型的卫星任务指令生成算法 [J]. 航天器工程, 2014, 23 (6): 54-60.

[5] 颜 军, 李 勇. 多 Agent 自主系统结构及其在分离模块航天器中的应用 [J]. 航天控制, 2013, 31 (2): 49-55.

[6] 张 健, 戴金海. 面向多星协同的卫星自组织自主控制体系结构 [J]. 国防科技大学学报, 2005, 27 (5): 95-98.

[7] 郝会成, 姜 维, 李一军, 等. 基于 Multi-Agent 敏捷卫星动态任务规划问题 [J]. 国防科技大学学报, 2013, 35 (1): 53-59.

[8] Chen Y, Wah B W, Hsu C W. Temporal Planning using Subgoal Partitioning and Resolution in SGPlan [J]. Journal of Artificial Intelligence Research, 2006, 26: 323-369.

[9] Foundation for Intelligent Physical Agent [EB/OL]. <http://www.fipa.org>.

[10] 程志锋, 张 蕾, 陈佳俊, 等. 基于 JADE 的多 Agent 系统开发 [M]. 北京: 国防工业出版社, 2013.

[11] 李 凌. 云计算服务中数据安全的若干问题研究 [D]. 北京: 北京科学技术大学, 2012.

[12] 冯朝胜, 秦志光, 袁 丁. 云数据安全存储技术 [J]. 计算机学报, 2015, 38 (1): 150-163.

[13] 高 原, 吴长安. 云计算下的信息安全问题研究 [J]. 情报科学, 2015, 33 (11): 48-52.

[14] 林 闯, 苏文博, 孟 坤. 云计算安全: 架构、机制与模型评价 [J]. 计算机学报, 2013, 36 (9): 1765-1784.

[15] 熊 俊. 基于改进密钥的云计算数据安全存储策略设计 [J]. 现代电子技术, 2016, 39 (20): 31-34.

[16] William Stallings. Network Security Essentials Applications And Standards (Fifth Edition) [M]. 北京: 清华大学出版社, 2014.

[17] 所 辉. 基于云计算的信息安全防护方案研究 [J]. 制造业自动化, 2012, 34 (5): 113-115.

(上接第 202 页)

参考文献:

[1] (美) Thomas Erl, (英) Zaigham Mahmood. 云计算概念、技术与架构 [M]. 北京: 机械工业出版社, 2016.

[2] 涂云杰, 白 杨. 基于 Hadoop 和双密钥的云计算数据安全存储策略设计 [J]. 计算机测量与控制, 2014, 22 (8): 2629-2631.

[3] 郎登何. 云计算基础及应用 [M]. 北京: 机械工业出版社, 2016.

[4] 傅颖勋, 罗圣美, 舒继武. 安全云存储系统与关键技术综述 [J]. 计算机研究与发展, 2013, 50 (1): 136-145.

[5] 杨 建, 汪海航, 王 剑, 等. 云计算安全问题研究综述 [J]. 小型微型计算机系统, 2012, 33 (3): 12-14.

[6] 刘婷婷. 面向云计算的数据安全保护关键技术研究 [D]. 北京: 北京信息工程大学, 2013.

[7] 王希忠, 吴 琼, 黄俊强, 等. 云计算数据安全关键技术研究