

基于行为树与 Docker 的无人集群应用快速集成验证方法

于宁 张立旺 周晨磊^(通讯作者) 陈培锴 刘涛

天津（滨海）人工智能创新中心，天津，300457；

摘要：针对无人集群应用开发中集成效率低、验证成本高、难以适应复杂场景等难题，本文提出一种基于行为树与 Docker 的快速集成验证方法。该方法利用行为树对任务逻辑和行为建模，清晰呈现各无人平台协作与执行流程；利用 Docker 容器化技术，完成应用及运行环境的封装、隔离与快速部署。本文详细阐述了该方法的原理、架构及关键技术，并基于 Gazebo 引擎，通过城市安保典型应用进行验证。结果表明，该方法在提升集成效率、降低验证成本和增强场景适应性等方面表现卓越，为无人集群应用的高效开发与验证开辟了新路径。

关键词：无人集群；行为树；Docker；快速集成；城市安保

DOI：10.64216/3080-1508.25.09.054

引言

随着科技的飞速发展，无人集群在军事和民用领域潜力尽显。军事上，其能执行侦察、打击等任务，凭借分布式协同作战提升作战效能、减少人员伤亡；民用里，可投身于城市安保、物流配送等场景^{[1][2]}，提高工作效率与服务质量。但无人集群应用开发面临诸多挑战：一方面，多个无人平台协作复杂，任务逻辑和行为流程设计与实现难度大；另一方面，不同场景对功能和性能要求不同，开发需频繁集成验证，传统方法集成效率低、验证成本高，还难以快速适配新场景。因此，研究一种高效的无人集群应用快速集成验证方法具有重要的现实意义^[3]。

行为树是以树形结构表示系统行为逻辑的图形建模语言，可清晰描述系统行为与任务逻辑，在无人集群中能明确各平台任务行为及协作关系，便于理解和维护^[4]。Docker 是流行容器化技术，可将应用及其依赖打包成可移植容器，实现快速部署与隔离运行，解决兼容性问题。Gazebo 是开源 3D 动态仿真平台，支持构建高逼真仿真环境等，为复杂系统提供低成本、高安全性仿真验证环境，降低研发风险与成本^[5]。DDS 是基于发布-订阅模式的分布式实时数据通信中间件，有低延迟等特性，便于构建分布式系统^[6]。将行为树与 Docker 深度融合，结合 Gazebo 的验证环境和 DDS 的通信机制，有望为无人集群应用的快速集成验证提供创新性解决方案。

1 系统架构设计

本文提出的基于行为树与 Docker 的无人集群应用快速集成验证方法系统架构如图 1 所示，该系统依托 Linux/Windows 操作系统及服务器集群，由无人集群应用

集成中心管理节点、Docker 容器化管理机制、Gazebo 仿真平台及人机交互软件平台四部分构成，各部分均可通过 Docker 容器独立管理与部署，实现高效协同。其中，无人集群应用集成中心管理节点是核心，集成行为树建模^[7]、任务切换控制等功能。基于行为树可构建复杂任务模型，实现子任务动态切换；通过机载算法管理模块配置轻量化软件；利用 DDS 分布式通信及通信桥接模块保障通信并转换消息。Docker 容器化管理机制作为柔性部署引擎，能快速部署与管理不同软件运行环境的算法，提供启停控制接口，与 DDS 机制配合，实现算法快速部署、隔离运行与交互通信，提升集成效率、降低成本。Gazebo 仿真平台是核心支撑载体，可构建高逼真三维仿真环境，集成多类型无人系统及任务载荷仿真模型，提供统一接口，可视化特性强，为复杂系统提供低成本、高安全性的验证环境。人机交互软件平台是动态操控中枢，搭建用户与系统交互桥梁，通过可视化界面实现全流程的协同任务管理。

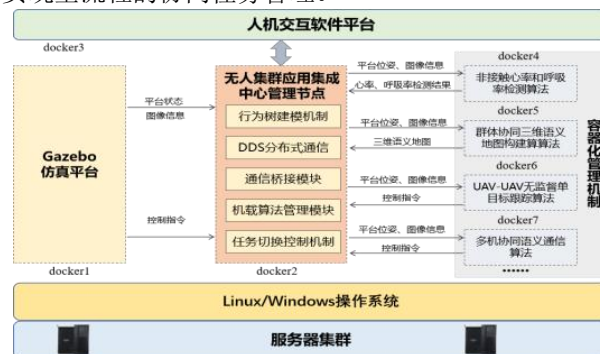


图 1 基于行为树和 Docker 的无人集群应用快速集成方法系统架构

2 关键技术实现

2.1 双层次多范式行为树任务建模机制

在无人集群应用快速集成验证场景中,任务的合理拆解与高效建模是实现系统功能的基础。行为树建模机制凭借其清晰的层次化结构、良好的任务扩展性,成为适配无人集群复杂任务逻辑表达的有效手段。基于此,本文提出如图 2 所示的双层次多范式行为树任务建模机制,上层通过主任务行为树把控整体流程,下层兼容行为树、状态机、过程式编程^[8]等不同建模范式,构建起灵活且规范的任务描述体系,支撑无人集群应用任务的有序执行与快速集成验证。

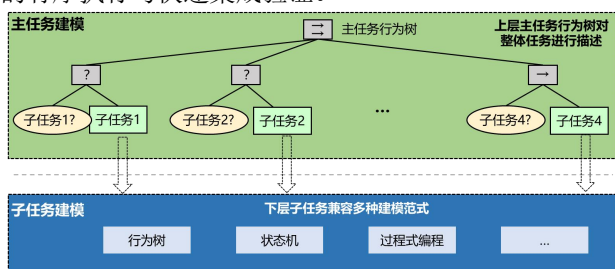


图 2 双层次多范式行为树任务建模机制

2.1.1 主任务行为树建模机制

主任务行为树作为无人集群应用整体任务的顶层抽象,以树形结构对任务进行拆解与组织。树的根节点代表无人集群应用的总任务目标,分支节点通过不同序列、选择、并行等逻辑关系将总任务分解为一系列有序的子任务集合,清晰描述从任务启动到目标达成的完整流程,使复杂任务逻辑变得直观、可追溯。

2.1.2 兼容不同范式的子任务建模机制

无人集群任务类型多样,单一建模方式难以周全适配。针对子任务建模需多模态支持,依特性灵活择取不同范式,彼此互补以满足多元需求。结构复杂、含多层嵌套与条件分支的子任务,用行为树范式,细分成小节点,依逻辑有序执行,可精细管控目标侦察等子任务;执针对行逻辑以状态转换为主的子任务,选状态机范式,定义状态、转换条件与动作,精准控制行为,保障任务稳定;针对逻辑简单、需快速响应或线性执行的,则用过程式编程范式,拆解为代码片段,低成本实现确定性执行效果。

本文提出的双层次多范式行为树任务建模机制优势突出,通过层次化结构将无人集群任务从主到子逐步拆解,把复杂逻辑转化为可视化树形结构,开发人员可快速梳理流程,维护时也能迅速定位环节、明晰逻辑,大幅提升了任务逻辑的可读性与可维护性。在无人集群应用迭代中,主任务行为树能通过添加、修改分支节点

灵活扩展任务流程,子任务多范式建模支持成熟子任务模型在不同项目直接复用,减少重复开发,加速新功能集成验证,适配快速迭代的技术需求。

2.2 Docker 容器化管理机制

在无人集群应用的快速集成验证框架中,Docker 容器化管理机制是实现异构子任务高效集成的核心支撑技术。该机制通过标准化容器封装、轻量化环境隔离和自动化生命周期管理,有效屏蔽不同子任务在运行环境、开发语言和部署方式上的差异,为多模态任务的快速集成提供统一运行载体,解决了多模态子任务在异构环境中的部署兼容性问题,为复杂系统的快速集成验证提供了工程化解决方案。

2.2.1 容器化封装策略

针对上层行为树分解的多模态子任务,设计分层级容器化封装策略:一是基础镜像标准化,采用 Linux/Windows 轻量级镜像作为基础环境,内置 Python/ROS/C++ 等主流开发框架运行时环境,通过 Dockerfile 定义统一的依赖安装脚本,确保不同编程语言实现的子任务具备一致的运行基底;二是功能模块原子化,遵循“一个容器一个进程”原则,将每个子任务的软件实体(可执行文件、配置文件等)封装为独立容器单元;三是支持环境变量注入,通过 `--env-file` 参数注入任务专属配置(IP 地址、端口号等),支持容器在不同验证场景下的参数化部署,避免硬编码带来的环境耦合问题。

2.2.2 容器生命周期动态管理

为支撑容器生命周期动态管理,设计基于 Docker SDK 的全周期管理系统,实现主任务对原子容器的精准控制,包括启动控制、状态监控和任务停止三个阶段。其中,在启动控制阶段,主任务行为树节点通过 `docker run` 接口创建容器并资源限制参数;在状态监控阶段,实时采集 CPU 使用率、内存占用等运行指标,异常时将自动触发重启策略;在任务停止阶段,采用渐进式双重终止策略,首先发送 `SIGTERM` 信号通知进程优雅退出,等待 5 秒后若未终止则发送 `SIGKILL` 强制终止,避免僵尸进程残留。在大规模集群场景中,结合 Docker Compose 工具定义任务依赖关系,通过 `depends_on` 参数确保子任务容器按顺序启动。

2.2.3 资源隔离与调度优化

借助 `cgroups` 技术,可对 Docker 实施细粒度资源控制,为每个子任务容器设定独立资源配额。对于计算

密集型任务,像运动学/动力学解算容器,用`--cpuset-cpus="0,1"`绑定特定CPU核心,减少上下文切换开销。在分布式集群环境里,结合Kubernetes^[9]调度器实现容器动态部署。利用标签和污点机制,把状态机容器调度到内存优化节点,过程式编程容器部署到低延迟I/O优化节点,形成差异化资源分配。通过深度绑定子任务范式特性与节点硬件特性,Kubernetes能按差异化需求精准投放与动态均衡容器级资源,为无人集群多范式任务并行执行提供底层资源支撑。

2.3 基于独立进程隔离的任务切换控制机制

任务切换控制机制的核心目标是实现主任务行为树对下层多模态子任务的动态启停调度,为进一步降低跨平台依赖与部署复杂度,将行为树、状态机、过程式编程等多样化子任务的启动和停止抽象为子进程的启动和停止,通过Shell脚本进行统一封装,屏蔽不同类型子任务的技术细节差异,实现多模态子任务的无差别操控与轻量化管理,支撑无人集群任务的灵活切换。

在子任务启动控制阶段,针对不同任务建模类型和部署方式,Shell脚本可灵活解析为不同执行逻辑:若子任务采用行为树建模,则启动行为树解释器并加载子任务的行为树描述文件;若采用状态机建模,则调用状态机解释器解析子任务的状态描述文件;若采用过程式编程建模,对于原生二进制程序,直接执行对应可执行文件;若结合了本文所述Docker容器化管理机制,则调用容器命令启动相应容器并执行容器内的二进制程序,从而屏蔽环境依赖差异。

在子任务停止控制阶段,通过操作系统接口实现渐进式双重终止策略:在优雅终止阶段,向子任务进程发送SIGTERM信号,允许子任务自主完成资源清理(如保存中间数据、关闭网络连接);在强制终止阶段,若子任务在超时时间内未响应,则发送SIGKILL信号强制终止,避免僵尸进程残留。此外,若子任务为Docker容

器化部署,通过调用`docker stop`命令,利用Docker的进程管理机制确保容器内程序优雅退出,同时释放容器占用的CPU、内存等系统资源。

3 实验验证

为验证本文基于行为树和Docker的无人集群应用快速集成方法的有效性,依托Gazebo仿真引擎,以无人集群城市安保典型应用开展系统验证。通过行为树建模机制,将城市安保上层主任务拆解为集结、抵近、探索、巡逻、处置、返航等六个子任务节点(如图5所示),下层采用过程式编程生成子任务可执行文件;利用Docker容器化管理,将心率和呼吸率检测、三维语义地图构建等算法封装至独立容器,实现环境隔离与资源复用;借助任务切换控制机制,通过shell脚本统一管理容器与程序,主任务行为树以独立进程启停子任务,完成任务动态调度。

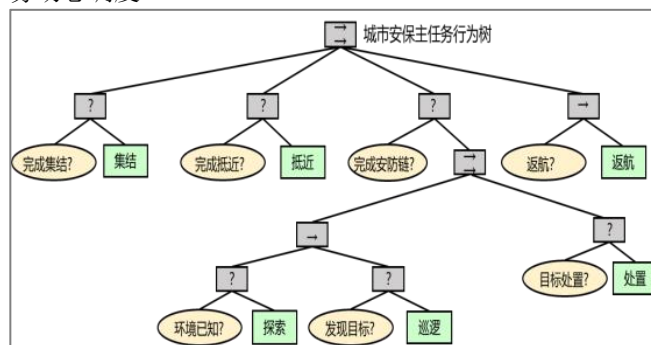
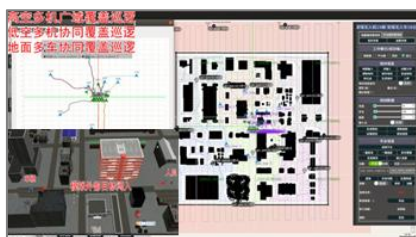
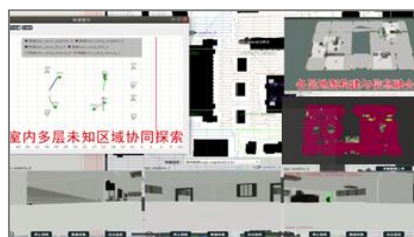


图5 城市安保主任务行为树

城市安保典型应用实验验证过程截图如图6所示,外部区域通过“高空-低空-地面”协同巡逻驱离目标,建筑内部依次开展环境侦察、地图构建、特情检测与处置,最终按指令返航。结果表明,本文方法通过Docker解耦算法与环境,用双层次行为树建模实现多模态任务快速集成,可跨平台部署且无需重配环境,模块化设计增强场景适应性,在提升集成效率、降低试验成本、适应复杂场景上表现优异。



(a)



(b)



图 6 城市安保典型应用实验验证。(a)空地协同外部区域巡逻；(b)室内未知区域协同探索与建图；(c)着火点、被困人员等特情目标侦察；(d)特情目标协同处置。

4 结论与展望

针对传统无人集群应用开发中集成效率低、验证成本高、场景适应性差等难题，本文提出了一种基于行为树与 Docker 的快速集成验证方法。该方法以行为树对无人集群任务逻辑与协作关系可视化建模，将复杂流程转为直观树形结构，增强任务逻辑可理解性与可维护性；借助 Docker 容器化技术，一体化封装、隔离并快速部署应用及其运行环境，保障应用在不同场景稳定运行。经 Gazebo 仿真引擎及城市安保应用验证证明，此方法可大幅提升集成效率、降低验证成本，场景适应性强。项目团队未来将对行为树模型进行优化，增强其对动态复杂场景的描述能力，并推动 Docker 与边缘计算融合提升部署效率，拓展应用场景测试范围，加速无人集群技术创新发展。

参考文献

- [1]Tong X, Hu J, Han S, et al. Optimization of Urban Unmanned Logistics Distribution Path with Multiple Distribution Centers [J]. SSRN Electronic Journal, 2022.
- [2]Moreno W F, Tangarife H I, Escobar Díaz A. Image analysis applications in precision agriculture [J]. Visión electrónica, 2018, 11(2): 200 - 210.
- [3]An X, Yu X, Song W, et al. A Software-Defined Distributed Architecture for Controlling Unmanned Swarm Systems [J]. Electronics, 2023, 12(18): 3739.
- [4]史殿习, 苏雅倩文, 李宁, 等. 基于行为树调度的多无人机未知室内空间探索方法[J]. 计算机科学, 2022,

49(S2): 83-93.

- [5]晏杰, 王彦臻, 林彬, 等. 基于 Gazebo 和 Docker 的群体机器人仿真系统设计[J]. 组合机床与自动化加工技术, 2021, (11): 105-108+113. DOI:10.13462/j.cnki.mmtamt.2021.11.025.
- [6]彭思远, 王文辉, 徐玉强, 等. 基于 DDS 的异构仿真系统互联方法[J/OL]. 指挥控制与仿真, 1-6[2025-06-04]. <http://kns.cnki.net/kcms/detail/32.1759.tj.20250225.1909.004.html>.
- [7]王琪玮, 张琪, 杨硕, 等. 动态任务环境下智能体健壮行为树控制架构设计[J]. 系统仿真学报, 2025, 37(01): 54-65.
- [8]Protsenko V. Specification of the Procedural Programming Language [J]. NaUKMA Research Papers. Computer Science, 2023, 5: 4 - 11.
- [9]Carrión C. Kubernetes Scheduling: Taxonomy, Ongoing Issues and Challenges [J]. ACM Computing Surveys, 2022, 55(7): 1 - 37.

作者简介：于宁（1994—），女，汉族，河北省邢台市，硕士，工程师，研究方向：群体智能、机器人操作系统。

张立旺（1994—）男，汉族，天津市蓟州区，硕士研究生，工程师，研究方向：群体智能、机器人操作系统。
周晨磊（1987—），男，汉族，河北省唐山市，硕士，工程师，研究方向：群体智能、机器人操作系统。
陈培锴（1995—），男，汉族，河北省邯郸市，硕士，工程师，研究方向：目标检测与跟踪技术。
刘涛（1998—），男，汉族，山东省烟台市，硕士，助理工程师，研究方向：群体智能、机器人操作系统。