
机器人操作系统的实践与思考

戴华东@AIRC-NIIDT

2019.06

背景

- PC计算和互联网计算时代：Windows + Intel
- 移动互联网计算时代：Android + ARM
- 机器人和智能计算时代：操作系统？ + 微处理器？



- 比尔·盖茨在《科学美国人》杂志上撰文：机器人将重复个人电脑崛起的道路，走进千家万户，而**机器人操作系统**是关键

智能化时代
智能操作系统+智能芯片



PC时代
Windows+Intel



移动互联网时代
Android+ARM



机器人时代操作系统的挑战

□ 挑战一：机器人如何与人共融？

- 现状：每一类机器人都有大量专用软件，这些专用软件把机器人看作“工具”或“用具”
 - ❖ 满足特定领域需求
 - ❖ 适应特定环境
 - ❖ 执行固定任务
- 困难：机器人具有资源多域异构特征，难以满足跨域协同的需求
 - ❖ 机器人的管理和控制软件，本质上既是操作系统，也是协同作业系统
 - ❖ 通过软件定义，把机器人变成“伙伴”和“朋友”
 - ❖ 除了管理传统资源，还要管理机器人的行为，管理作业中的流程和任务规划

机器人时代操作系统的挑战

□ 挑战二：复杂环境下机器人的**自主行为控制与协同**

➤ 现状：采用“感知—规划—行动”控制回路

- ❖ 起源于个体机器人的智能控制
- ❖ 多用于简单合作环境
- ❖ 适用于执行特定具体任务

➤ 困难：作业环境、个体行为及自身状态复杂多变，具有不确定性

- ❖ 适应环境的能力：感知、学习、推理、抽象
- ❖ 行为控制回路：观察、判断、决定、行动
- ❖ 群体机器人协同和人机协同的通信网络
- ❖ 机器人的应急指挥与控制模型

机器人时代操作系统的挑战

□ 挑战三：群体智能

➤ 现状：着力提升单体智能，实现更高自主性

- ❖ 具有一定的自主能力
- ❖ 具有对相对简单环境的感知能力
- ❖ 具有初级人机互理解能力

➤ 困难：单体智能提升的基础上，如何聚合多个单体智能形成群体智能

- ❖ 群体自主执行复合多样的任务
- ❖ 复杂情境的协同感知
- ❖ 一对多、多对多的机-机/人-机互理解、互遵守

机器人操作系统：机器人的“新脑”

- “打破烟囱林立，实现互连、互通、互操作”靠“平台”，平台的支撑靠操作系统
- 管理是自主行为控制的基础，操作系统的内涵就是“管理”
- 软件是承载智能的主体，操作系统是各类智能软件的基础

应对三大挑战，必须设计新的基础软件

——群体机器人操作系统

机器人操作系统的实践与思考

□ 面向行为的概念抽象

——机器人操作系统的核心概念

□ 多态分布的体系结构

□ 适应环境的群体智能

从计算机操作系统谈起

□ 计算机操作系统核心基础概念

➤ 基于“**进程/线程**”的**控制抽象**

❖ 程序的并发执行导致

- ✓ 程序和计算机执行程序的活动不再一一对应

❖ 进程是可并发执行的程序在一个数据集合上的运行过程

- ✓ 进程是系统进行资源分配的一个独立单位

❖ 线程是系统进行调度的一个独立单位

➤ 基于“**文件**”的**数据抽象**

❖ 数据的存储介质复杂多样，访问方式各不相同

- ✓ 磁带、磁盘、光盘、U盘等

❖ 文件：实现对数据的一致抽象，提供统一访问视图

- ✓ 文件名、属性等

❖ 基于文件实现操作系统高级功能

- ✓ 设备文件、内存映射机制、虚拟存储机制等

机器人操作系统的核心概念

□ 基于“角色”的控制抽象——“他”是“伙伴”

- “角色”群体在一个场景中完成一个或多个作业
- 不同机器人可以扮演不同的 群体“角色”
 - ❖ 领队者、跟随者
 - ❖ 自我中心角色、任务角色、维护角色
 - ❖ “长机-僚机”、“指挥舰-护卫舰-辅助舰”
- 机器人的不同载荷可以承担不同的功能“角色”
 - ❖ 定位、导航、通信、图传、机械、动力
- 有人、无人协同扮演不同“角色”



机器人操作系统的核心概念

□ 基于“角色”的控制抽象

➤ 角色 (actor) 被抽象为: 机器人完成给定任务的一次执行的行为

- ❖ 资源、插件、任务、结构
- ❖ 状态与调度: 扮演、激活、切换、迁移、...

➤ 两类用户开发接口

- ❖ 任务开发接口 (XML格式角色状态机)
- ❖ OODA算法开发接口 (插件)

➤ 五大核心类/模块

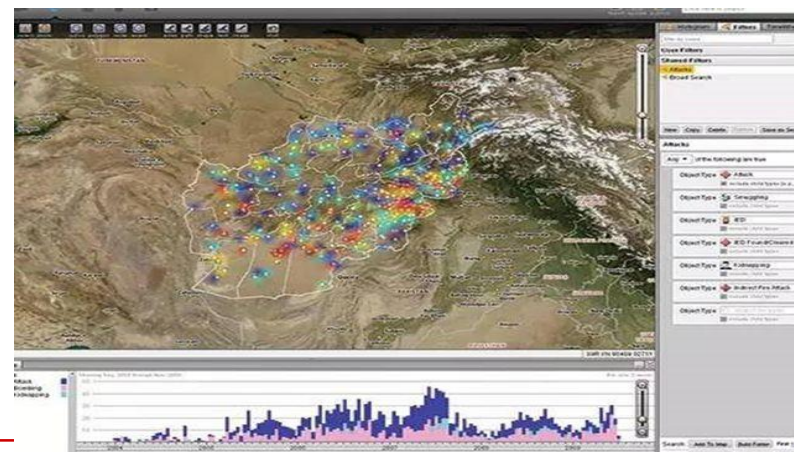
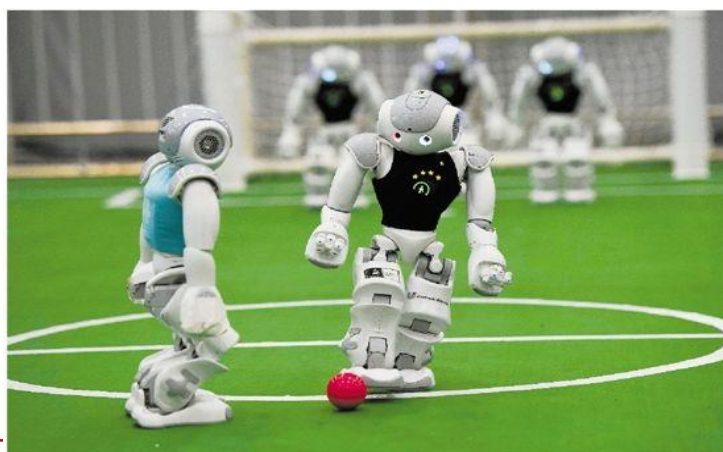
- ❖ 角色匹配
- ❖ 任务解析
- ❖ 角色调度
- ❖ 角色状态机管理
- ❖ 群组织结构管理

```
class ACB {
public:
    //actorID and name
    int64_t actorID;
    std::string name;
    //task info
    int64_t taskID;
    std::string taskName;
    TaskInfo taskInfo; |
    //ACB tree info
    ACB* pParent;
    ACB* pChild;
    ACB* pSibling;
    //actor state
    int16_t state;
    //resources and configurable parameters
    std::vector<SensorActuatorInfo> sensors;
    //platform information
    PlatformInfo* pPlatformInfo;
    //swarm information
    SwarmInfo* pSwarmInfo;
    //priority
    int32_t prio;
    //formation position
    int32_t formationPos;
    //formation type
    std::string formationType;
};
```

机器人操作系统的核心概念

□ 基于“场景”的数据抽象——“他”的“舞台”

- 一个或多个作业活动的时域与空域
 - ❖ 多平台、多视角、多尺度、多图层
 - ❖ 按时间、空间、时空混合方式来组织和检索信息
- 不同的角色在一个场景中共享信息，协同完成作业
- 可以在文件系统、数据库、地理信息系统、SLAM之上进一步封装



机器人操作系统的核心概念

□ 基于“场景”的数据抽象

- 提供基本的机器人数据管理功能
- 支撑机器人的观察、判断、决定、行动等行为
 - ❖ 基于角色的数据融合
 - ❖ 面向任务的情境感知
 - ❖ 人-机协同判断等

□ 群体“场景”一致性

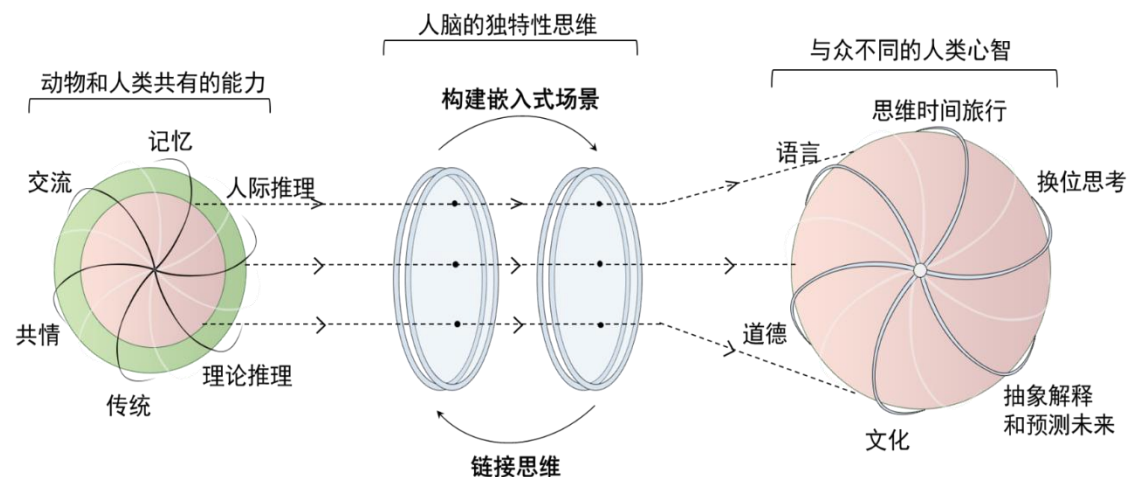
- 分布式存储，全局数据的一致抽象访问

□ 场景描述 - 层次化结构

- 角色：群体/单体/传感器的属性、动作、状态等
- 图层：包含若干角色和时空约束

□ 构建嵌入式场景

- 想象不同的场景和后果，将其嵌入到更大场景中



机器人操作系统的核心概念

□ 通用任务指令 (UTO)

- 目的性系统
- 类似传统计算机操作系统的界面抽象 “shell/API”
- 基于UTO方案化指令切实将任务规划、地面站、虚拟/半实物仿真环境和机器人群体串接起来，形成一个整体



```
<_UTOConfig>
  <actor name="TakeOff">
    <param name="TargetRegionType" value="Circle"/>
    <param name="TargetRegionCenter" value="100 100 10"/>
    <param name="TargetRegionRadius" value="20"/>
    <transition name="finish_event">
      <successor>Assemble/Follower</successor>
    </transition>
  </actor>
  <actor name="Assemble/Follower">
    <transition name="lost_leader_event" sysNum="1" barrierKey="0">
      <successor>Assemble/Leader</successor>
    </transition>
    <transition name="finish_event" sysNum="10" barrierKey="1">
      <successor>Patrol/Follower</successor>
    </transition>
    <transition name="egress_cmd" sysNum="10" barrierKey="7">
      <successor>Egress</successor>
    </transition>
  </actor>
  <actor name="Assemble/Leader">
    <transition name="finish_event" sysNum="10" barrierKey="1">
      <successor>Patrol/Leader</successor>
    </transition>
    <transition name="egress_cmd" sysNum="10" barrierKey="7">
      <successor>Egress</successor>
    </transition>
  </actor>
  <actor name="Patrol/Leader">
    <transition name="finish_event" sysNum="10" barrierKey="6">
      <successor>Assemble/Leader</successor>
    </transition>
  </actor>
</_UTOConfig>
```

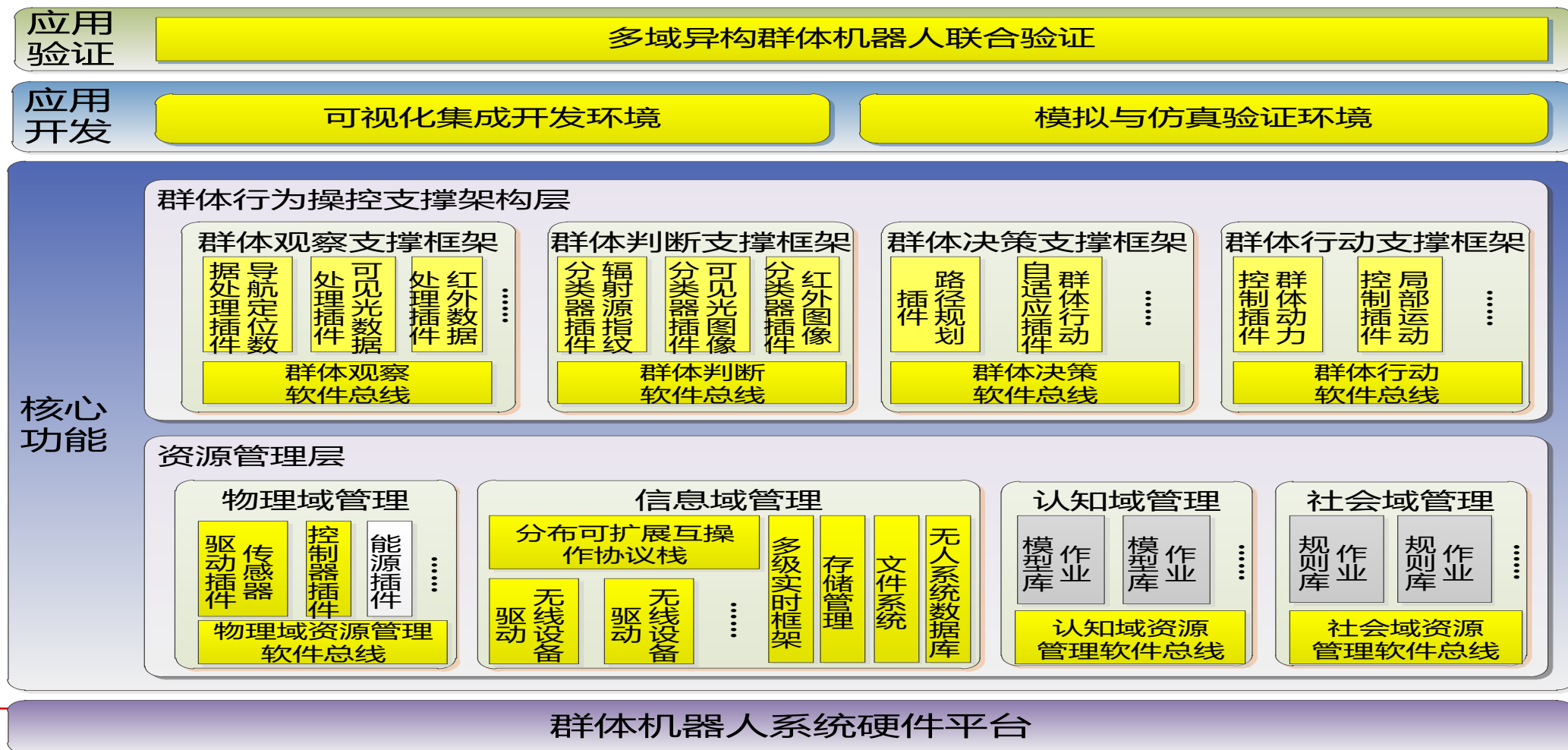
起飞
集结
同步
地面站操控
巡逻

机器人操作系统的核心概念

- 面向行为的概念抽象
- 多态分布的体系结构
——机器人群体如何组织管理
- 适应环境的群体智能

多态分布的体系结构

□ 纵向看：层次式结构



多态分布的体系结构

□ 物理域资源管理

➤ 实现物理域资源管理软件总线，抽象出逻辑设备

- ❖ 实现与物理设备的解耦
- ❖ 实现行为操控的一致访问
- ❖ 实现异构平台的软件重用

➤ 驱动插件

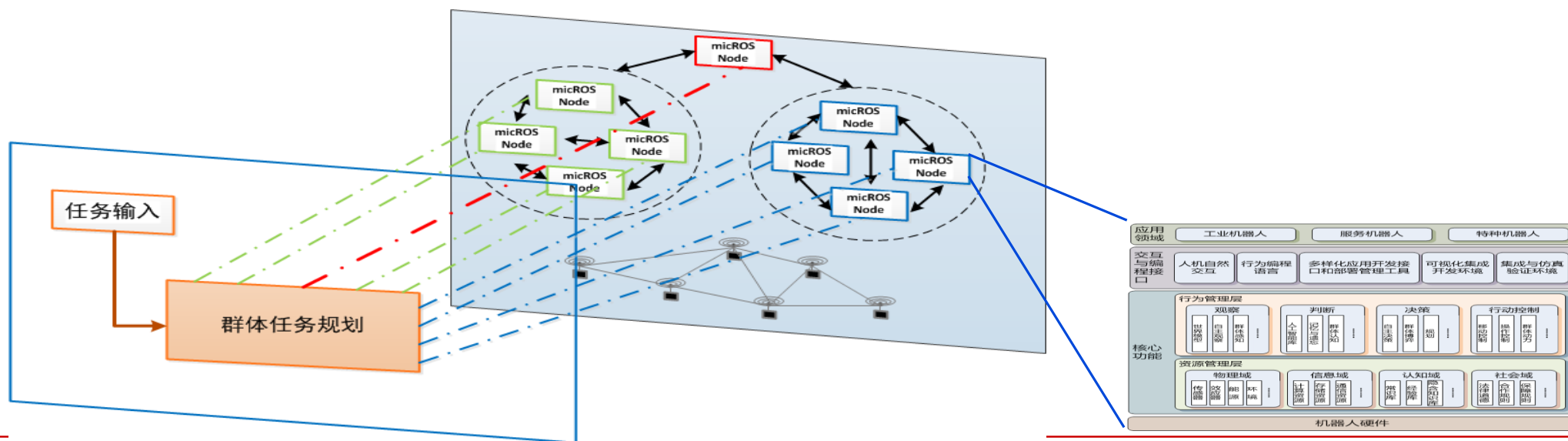
- ❖ 设备驱动
- ❖ 总线驱动
- ❖ 控制器驱动



多态分布的体系结构

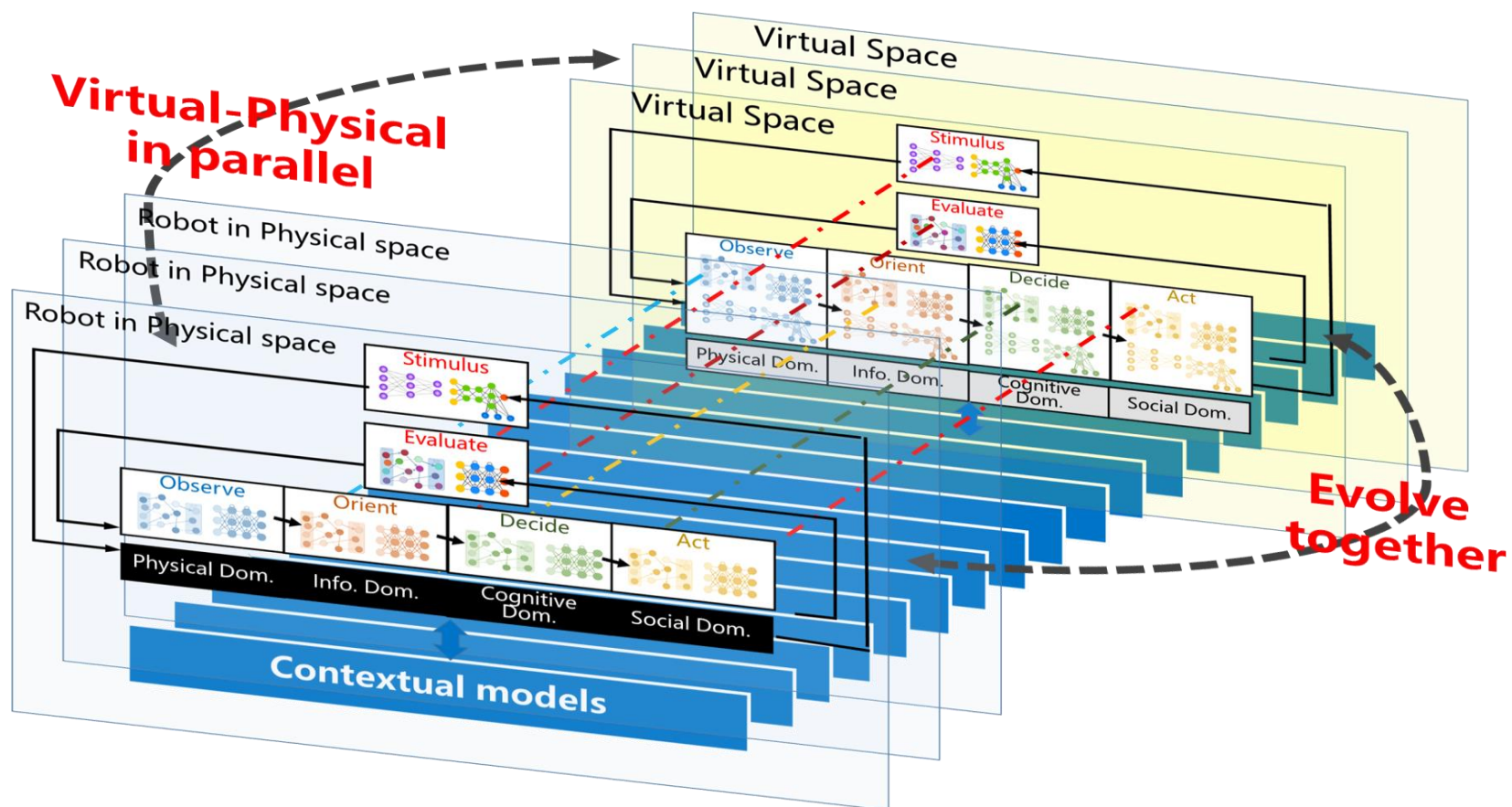
□ 横向看：分布式结构

- 按“四条链”完成机器人自主作业
 - ❖ 任务链、行动链、协同链、保障链
- 基于“四条链”支撑场景集
- 进一步开展基于5G移动自组网的规模化实验



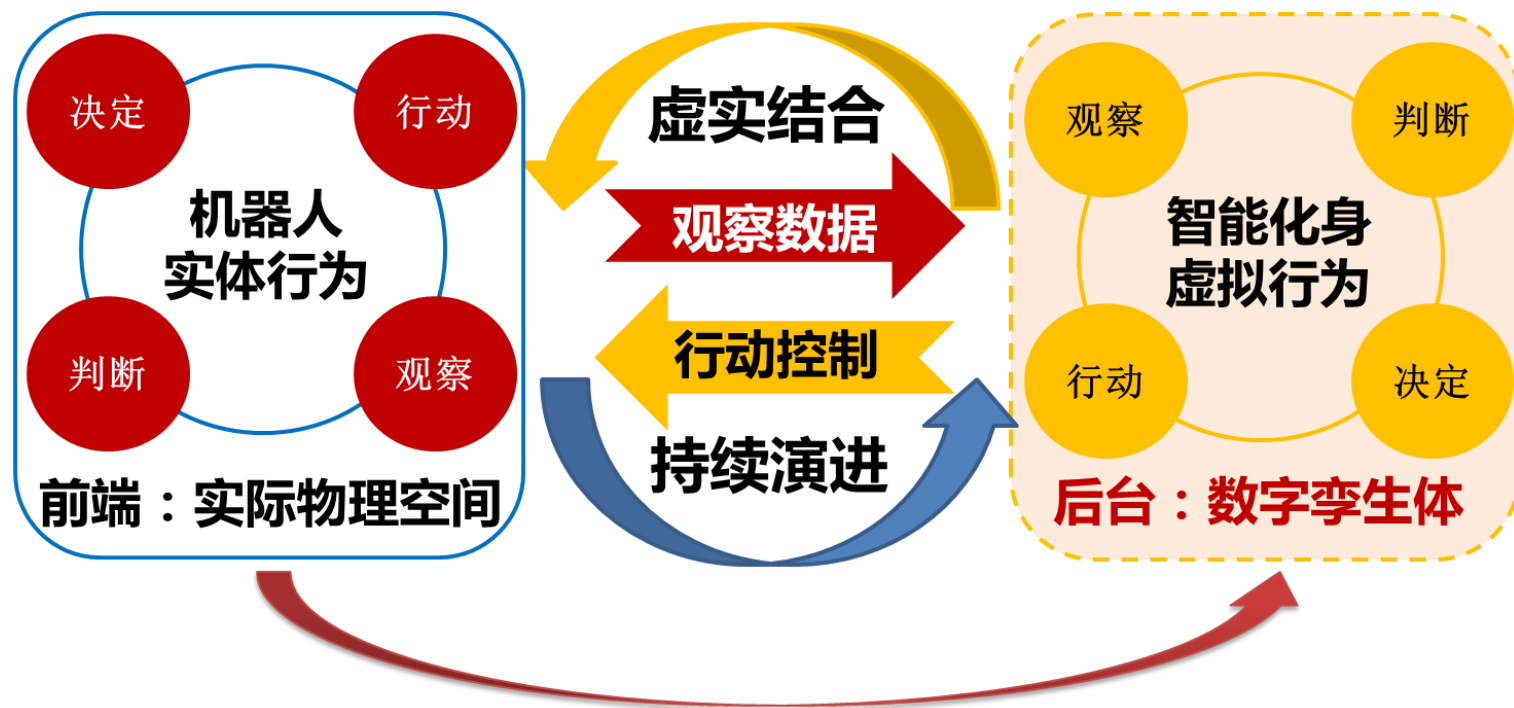
多态分布的体系结构

□ 深向看：可持续自主学习的平行架构



多态分布的体系结构

□ 数字孪生是虚实联动、持续演进的基础



数字孪生：充分利用物理模型、传感器更新、运行历史等数据，集成多学科、多物理量、多尺度、多概率的数字映射系统

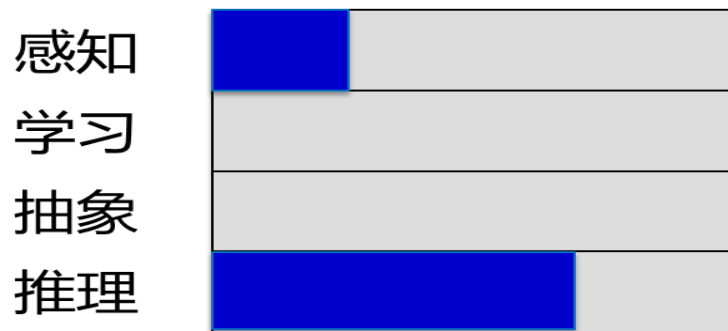
机器人操作系统的实践与思考

- 面向行为的概念抽象
- 多态分布的体系结构
- 适应环境的群体智能
——人工智能的下一个突破口

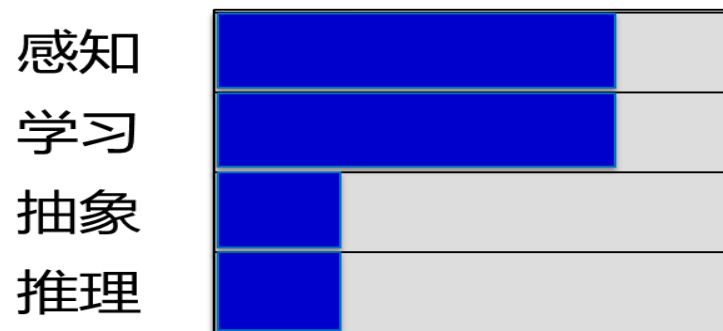
三次浪潮

□ DARPA认为人工智能发展经历了三次浪潮

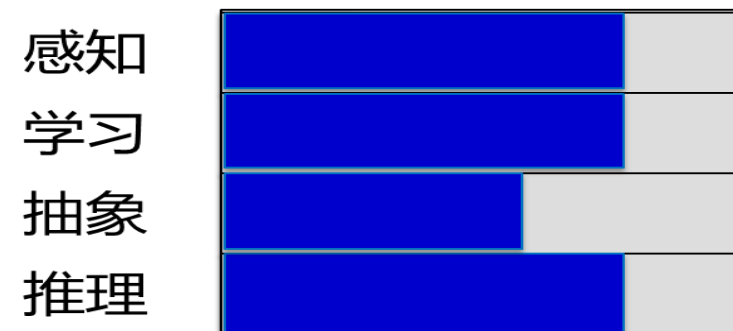
- 第一次浪潮：手工知识(Handcrafted Knowledge)
- 第二次浪潮：统计学习(Statistical Learning)
- 第三次浪潮：适应环境 (Contextual Adaptation)



手工知识



统计学习



适应环境

适应环境

□ 一种适应环境的**持续自主学习系统**的架构模型



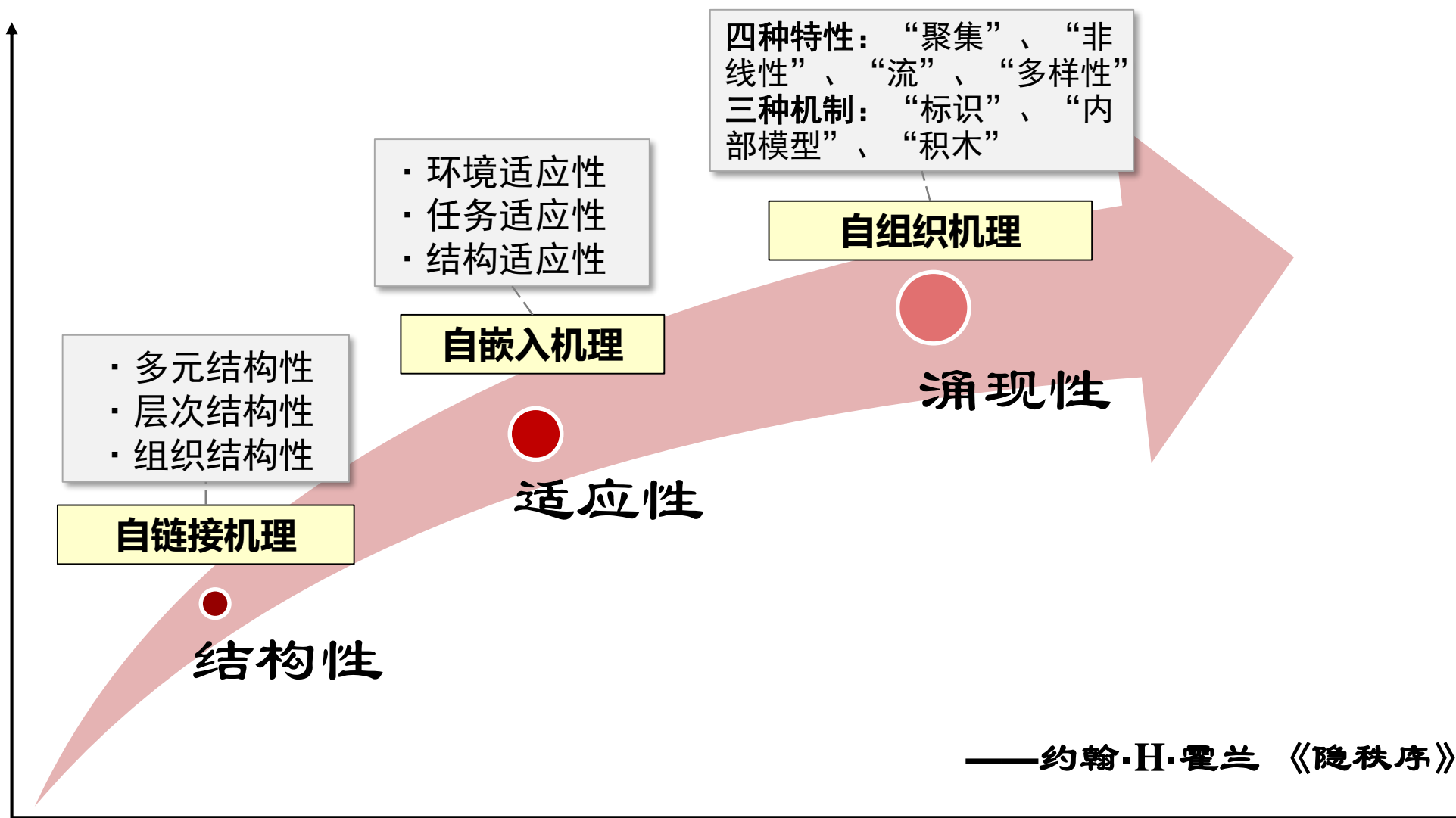
- 构建环境模型的思路：**场景 + 语义**
- 基于环境模型可以进行**观察、判断、决定、行动**
- 用平行控制架构实现**感知、学习、抽象、推理**

内嵌数理知识的人工智能

□ DARPA发布《颠覆性机遇特别通知—物理人工智能(PAI)》建议征询书

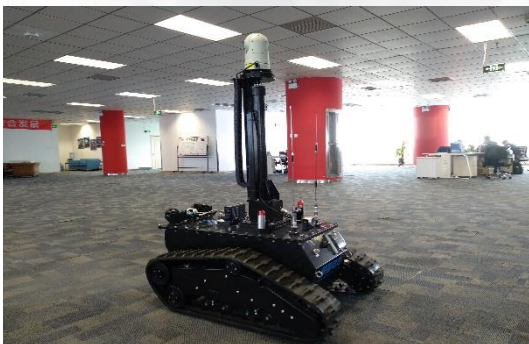
- 通过“嵌入”与具体任务相关的物理、数学知识，克服信息的不完整性
- 利用观测、试验和模拟数据等知识开发AI原型，克服稀疏、嘈杂或不完整数据的局限性
- 使用模拟和/或实际数据演示AI原型，例如卫星或雷达图像处理或人机协作
- 解决AI系统的计算要求和基本性能限制，提高AI系统的预测准确性

群体智能



机器人操作系统的初步实践

□ 无人平台适配



Morphable, Intelligent and Collective Robot Operating System



机器人操作系统的初步实践

□ 无人平台适配

- 计算控制板、通信板
- 多款可见光传感器，包括Kinect one, DVS 相机, ZED双目相机, Realsense 相机, Bumblebee 相机等
- 多款激光雷达，包括Velodyne 16/32/64线激光雷达, 2D激光雷达等
- 多款红外相机，包括Flir双光相机, Kinect 360等
- 多款GNSS/IMU，包括差分GPS, Xsens IMU等



与传统计算机操作系统的区别

	计算机系统	micROS
层次架构	分核内/核外两层，实现资源管理	分资源管理、行为管理两层，实现资源与行为管理
分布架构	互连、互通、互操作，实现分布计算	互连、互通、互操作、互理解、互遵守，实现群体智能
控制抽象	进程/线程模型	基于角色的“观察—判断—决定—行动”执行链，虚实互动的平行学习机制
数据抽象	文件	场景
功能	管理硬件、软件、数据等资源	除传统信息资源外，还管理物理域、认知域、社会域资源，和机器人、无人系统的行为，以及群体行为
人机交互	输入命令和数据，输出结果数据	输入任务和环境，输出行动

请批评指正