

Lec13-2 IOT平台设备管理

什么是设备管理？

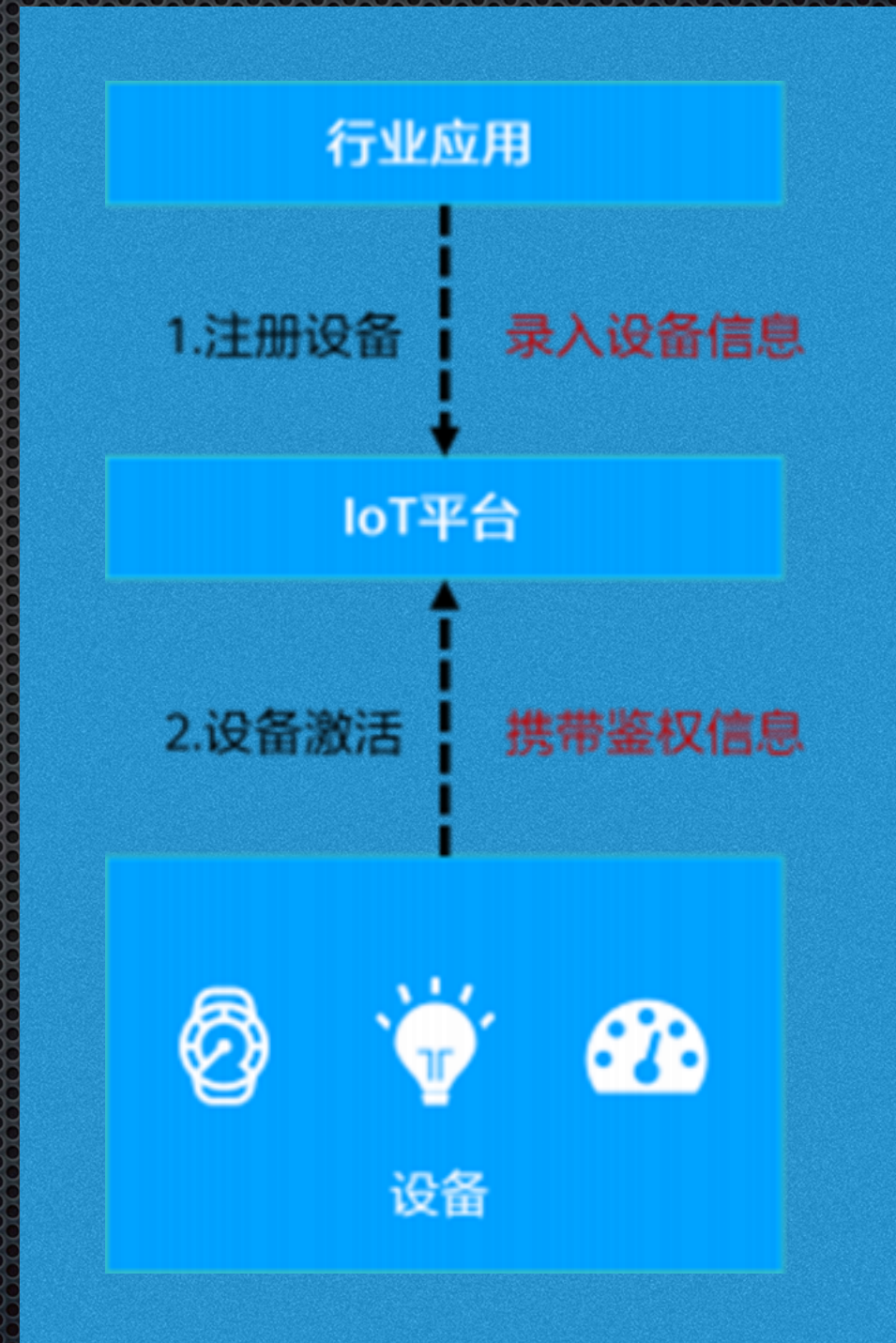
- 设备管理服务, 可以帮助对所有连接的设备, 在全球范围内进行规模化的注册、查看及远程管理
 - 可以在 IoT 平台内查看到所有设备队列, 并进行信息跟踪、运行状态查看、管理连接、查看日志数据等动作
 - 通过固件标准上报的版本信息, 还可以定位到每个设备当下运行的固件信息, 便于确认设备的稳定并排查问题
 - 可以通过无线下载技术 (OTA), 发送固件升级, 从而确保设备始终运行最新版本软件
- 设备管理在设备接入基础上, 提供了更丰富完备的设备管理能力, 简化海量设备管理复杂性, 节省人工操作, 提升管理效率

主要功能

- 物联网平台提供功能丰富的设备管理服务，包括不限于：
 - 生命周期
 - 设备分组
 - 设备影子
 - 物模型
 - 数据解析
 - 数据存储
 - 在线调试
 - 固件升级
 - 远程配置
 - 实时监控等

设备注册&接入鉴权

- 为了保证接入物联网平台的设备是安全可信的，需要进行设备注册和设备接入鉴权操作
 - 设备注册，指用户通过控制台或调用注册设备API在IoT平台中注册设备信息，平台中存在设备信息后，再接入真实的实体设备，这样平台与终端实体设备可以实现连接和通信
 - 设备接入鉴权，是指IoT平台对接入平台的设备进行鉴权认证，用于保障设备接入信息的完整性和安全性、设备与平台消息传输完整性和安全性



命令下发

- 设备的產品模型中定义了IoT平台可向设备下发的命令，平台向设备下发命令，修改设备的服务属性，实现对设备的控制
- 平台向设备下发命令包括立即下发和缓存下发两种情况，如下

命令下发机制	定义	适用场景
立即下发	不管设备是否在线，平台收到命令后立即下发给设备。如果设备不在线或者设备没收到命令则下发失败。支持给本应用的设备和被授予权限的其他应用的设备下发命令。	立即下发适合对命令实时性有要求的场景，如路灯开关灯，燃气表开关阀。使用立即下发时，命令下发的时机需要由应用服务器来确定。
缓存下发	物联网平台在收到命令后先缓存，等设备上线或者设备上报数据时再下发给设备，如果单个设备存在多条缓存命令，则进行排队串行下发。支持给本应用的设备和被授予权限的其他应用的设备下发命令。	缓存下发适合对命令实时性要求不高的场景，如配置水表参数，

设备联动规则

- 规则引擎的使用对象是终端用户，系统已经预置支持的规则场景，终端用户通过方便、易理解的友好界面在自有设备下制定自动化规则
- 规则可以和设备、应用、告警绑定，当绑定的信息满足条件时，规则可以自动化执行响应动作
 - 规则引擎定位处理各种事件，利用规则引擎可以完成异常事件的及时通知和快速处理，帮助终端用户维护设备、监控设备，保证系统业务的及时恢复
 - 阈值超限、范围超限、位置跟踪等事件，也可定义为规则引擎输入条件，并关联对应的处理动作

设备联动规则示例

触发条件

根据时间定义条件

如定义8:00钟关闭走廊的灯



根据设备上报的数据定义条件

如定义当温度计的温度值大于25度时打开空调



根据系统事件定义条件

如当系统生成Sensor异常告警时，发送通知到指定手机



响应动作

响应动作1

系统给传感器下发需要执行的动作，8:00准时关闭走廊的灯



响应动作2

系统给传感器下发需要响应的命令，温度值大于25度时打开空调



响应动作3

发送通知，包含邮件和短信的方式。Sensor异常告警时，发送通知到指定手机



设备批操作

- 设备批操作向物联网平台提供对终端设备统一管理的通道，能提升对终端设备的管理效率，很好地满足用户批量管理设备的需求
- 当用户接入的设备数量过多，或者需要对全部或某一个群组的设备进行相同的操作时，可以采用设备批操作
- 物联网平台支持对设备的批量操作，包括批量注册设备、批量命令下发、批量位置上传、批量设备配置和批量软固件升级
 - 批量注册设备：因注册设备数量过多而导致注册时间太长，可采用批量注册的方式注册设备
 - 批量命令下发：当物联网平台需要对批量设备下发命令时，可通过北向接口创建批量命令下发任务，在物联网平台上，可查看任务的执行状态、操作者和成功率等信息
 - 批量位置上传：当物联网平台需要对设备的位置进行批量上传时，可采用批量位置上传操作。这里的设备主要指安装位置相对固定的终端设备，如水表
 - 批量设备配置：当物联网平台需要对设备进行批量配置时，可采用批量设备配置操作
 - 批量软固件升级：当物联网平台需要对设备的固件或者软件进行批量升级时，可采用批量软固件升级操作

设备管理—物模型

- 物模型（Thing Model）是对设备在云端的功能描述，包括设备的属性、数据、服务和事件
- 物联网平台通过定义一种物的描述语言来描述物模型，称之为 TSL（即 Thing Specification Language），采用JSON格式，可以根据 TSL 组装上报设备的数据

物模型的定义-阿里

功能类型

说明

是什么

属性 (Property)

一般用于描述设备运行时的状态，如环境监测设备所读取的当前环境温度等；属性支持GET和SET请求方式。应用系统可发起对属性的读取和设置请求

做什么

服务 (Service)

设备可被外部调用的能力或方法，可设置输入参数和输出参数。相比于属性，服务可通过一条指令实现更复杂的业务逻辑，如执行某项特定的任务

事件 (Event)

设备运行时的事件，事件可以被订阅和推送，一般包含需要被外部感知和处理的通知信息，可包含多个输出参数，如，某项任务完成的信息，或者设备发生故障或告警时的温度等

可对外提供什么

<https://www.alibabacloud.com/help/zh/doc-detail/73727.htm>


```

{
  "schema": "物模型结构定义的访问URL",
  "profile": {
    "productKey": "产品ProductKey"
  },
  "properties": [
    {
      "identifier": "属性唯一标识符（产品下唯一）",
      "name": "属性名称",
      "accessMode": "属性读写类型：只读（r）或读写（rw）。",
      "required": "是否是标准功能的必选属性",
      "dataType": {
        "type": "属性类型：int（原生）、float（原生）、。。。。",
        "specs": {
          "min": "参数最小值（int、float、double类型特有）",
          "max": "参数最大值（int、float、double类型特有）",
          "。。。。。。。。。}}}
        ],
      "events": [
        {
          "identifier": "事件唯一标识符（产品下唯一，其中post是默认生成的属性上报事件。）",
          "name": "事件名称",
          "desc": "事件描述",
          "type": "事件类型（info、alert、error）",
          "required": "是否是标准功能的必选事件",
          "outputData": [],
          "method": "事件对应的方法名称（根据identifier生成）"
        }
      ],
      "services": [
        {
          "identifier": "服务唯一标识符",
          "name": "服务名称",
          "desc": "服务描述",
          "required": "是否是标准功能的必选服务",
          "callType": "async（异步调用）或sync（同步调用）",
          "inputData": [],
          "outputData": [],
          "method": "服务对应的方法名称（根据identifier生成）"
        }
      ]
    }
  ]
}

```


设备影子

- 物联网平台提供设备影子功能，用于缓存设备状态
 - 设备在线时，可以直接获取云端指令
 - 设备离线后，再次上线可以主动拉取云端指令
- 设备影子是一个 JSON 文档
 - 存储设备上报状态
 - 应用程序期望状态信息
- 每个设备有且只有一个设备影子，设备可以通过MQTT获取和设置设备影子来同步状态
 - 该同步可以是影子同步给设备，也可以是设备同步给影子

应用场景

- 场景1:
 - 网络不稳定，设备频繁上下线
- 场景2:
 - 多程序同时请求获取设备状态
- 场景3:
 - 设备掉线
 - 指令携带时间戳


```
{
  "state": {
    "desired": {
      "color": "RED",
      "sequence": [
        "RED",
        "GREEN",
        "BLUE"
      ]
    },
    "reported": {
      "color": "GREEN"
    }
  },
  "metadata": {
    "desired": {
      "color": {
        "timestamp": 1469564492
      },
      "sequence": {
        "timestamp": 1469564492
      }
    },
    "reported": {
      "color": {
        "timestamp": 1469564492
      }
    }
  },
  "timestamp": 1469564492,
  "version": 1
}
```


关于模板

- 模板定义了一类设备的schema，Schema中定义了各设备各属性的显示名称、类型、默认值等信息
 - 通过模板，我们定义了设备的视图(view)
 - 此外，需要支持模板的CRUD操作，IoT Device Management Schema API 主要包含管理设备模板的相关接口--<https://cloud.baidu.com/doc/IOT/s/Bjwvy7j1b>
- 例如：
 - 物模型模板--<https://open.iot.10086.cn/doc/oes/book/ecp/objectmodelmanagement.html>
 - Device template library in IoT Central--<https://azure.microsoft.com/en-us/blog/device-template-library-in-iot-central/>

数字孪生

- 数字孪生，英文名叫Digital Twin（数字双胞胎），也被称为数字映射、数字镜像
- 起源
 - 工程实践：
 - 2009年，AFRL（美国空军研究实验室）发起了一个“**机身数字孪生**”项目
 - NASA，2010年首次提出了数字孪生（Digital Twins）的概念
 - 更有理论色彩：
 - 2002年Michael Grieves在密歇根大学提出了PLM（产品生命周期管理）概念模型
- 数字孪生的概念，起源于制造业，现在已广泛应用到了智慧城市、智慧交通、智慧农业、智慧医疗、智能家居等行业。简而言之，数字孪生无处不在。

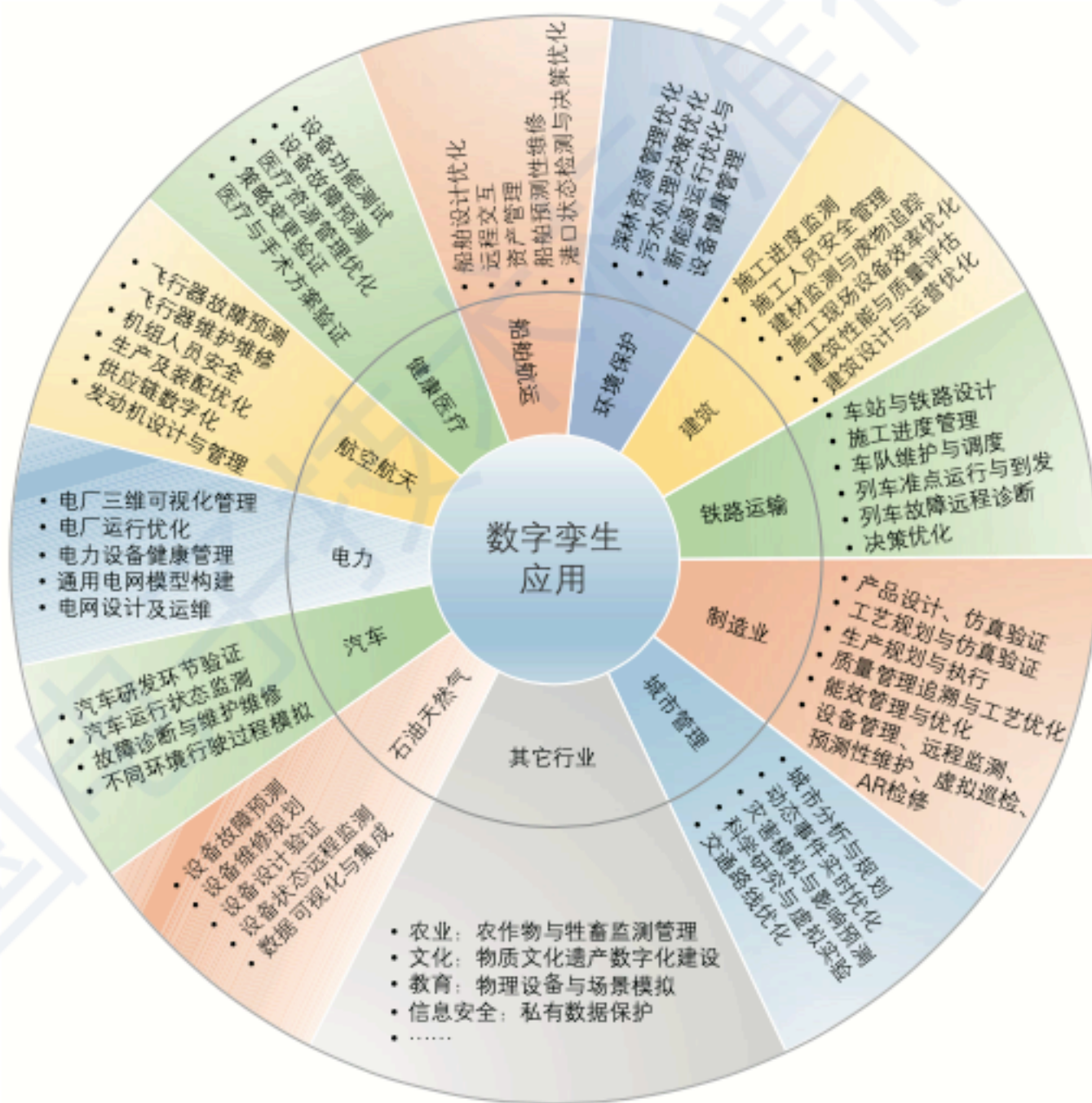


图 2 数字孪生行业应用

数字孪生

- 官方定义：
 - 数字孪生，是充分利用物理模型、传感器更新、运行历史等数据，集成多学科、多物理量、多尺度、多概率的仿真过程，在虚拟空间中完成映射，从而反映相对应的实体装备的全生命周期过程

数字孪生的价值

- 数字孪生的概念非常大。简单的看一下这个概念所提供的典型的好处:
 - 可见性: 数字孪生能够实现机器操作的可见性, 以及制造工厂或者机场中大型的互联系统的可见性
 - 预测性: 使用多种建模技术(基于物理和基于数学的), 数字孪生模型能够用于预测机器未来的状态
 - 假设分析: 通过适当设计的接口, 可以很容易的与模型进行交互, 并且对模型询问假设问题, 来模拟现实中无法创建的各种条件
 - 对行为进行理解和解释的记录与沟通机制: 数字孪生模型能够作为一种沟通和记录机制, 能够对单独的机器或者机器的集合的行为进行理解和解释
 - 连接不同的系统: 比如后端的业务应用
 - 如果设计的正确, 数字孪生模型能够用来连接后端的业务应用, 在供应链运作中实现业务成果, 包括制造、采购、仓储、运输、物流、现场服务等

数字孪生与仿真(Simulation)的区别

■ 仿真技术

- 是将包含了确定性规律和完整机理的模型转化成软硬件的方式来模拟物理世界的方法
- 应用仿真硬件和仿真软件通过仿真实验，借助某些数值计算和问题求解，反映系统行为或过程的模型技术
- 仿真技术仅仅能以离线的方式模拟物理世界，不具备分析优化功能，因此不具备数字孪生的实时性、闭环性等特征

■ 数字孪生

- 需要依靠包括仿真、实测、数据分析在内的手段对物理实体状态进行感知、诊断和预测，进而优化物理实体，同时进化自身的数字模型
- 仿真技术作为创建和运行数字孪生的核心技术，是数字孪生实现数据交互与融合的基础
- 在此基础上，数字孪生必需依托并集成其他新技术，与传感器共同在线以保证其保真性、实时性与闭环性

数字孪生与信息物理系统(CPS)的异同

- 数字孪生与 CPS 都是利用数字化手段构建系统为现实服务
- 不同：
 - CPS 属于系统实现，而数字孪生侧重于模型的构建等技术实现
 - 相比于综合了计算、网络、物理环境的多维复杂系统 CPS，数字孪生的构建作为建设 CPS 系统的使能技术基础，是 CPS 具体的物化体现

数字孪生与元宇宙

- 在维基百科中这样描述“元宇宙”：元宇宙是通过虚拟增强的物理现实，是呈现收敛性和物理持久性特征的、基于未来互联网的具有连接感知和共享特征的3D虚拟空间
- 共同点：
 - 都以数字技术为基础，再造高仿真的数字对象和事件，进行可视化感知交互和运行
 - 底层支撑技术可通用
- 不同的是
 - 元宇宙可以现实物理世界为数字框架，也可以完全塑造全新的理念数字世界，终极形态是基于数字世界实现的原生社会
 - 而数字孪生则是以信息世界严格、精确映射物理世界和事件过程为框架和基础，无论是工业制造，还是城市管理，基于实时客观数据的动态进程，与人工智能结合的挖掘分析和深度学习；并进一步模拟情境和决策，以改进现实或更好适应现实，最终实现自动控制或自主决策控制，终极形态是自主孪生

行业实现

1. 简单的设备模型

- 这些实现通常使用包含两个主要属性集的JSON文档:
 - 一组观测值或者报告值：通常，设备上的探测器读取当前值，并更新这些观测属性。比如，一台机器当前观察的转速(比如 1000 RPM)
 - 一组期望值：这是控制程序希望在设备上设置的值。比如，一个应用能够设置引擎转速到1200 RPM
- 除了这两组主要的属性，这些实现也在JSON文档中存储了相关的信息，比如设备的名字或者序列号，或者当前的位置
-

行业实现

2.工业孪生

- 这类实现通常被工业物联网供应商所采用，它包括PLM工具设计机器的信息和一台设备的模型，一些工业供应商关注物理属性、设计信息和实时数据，并且将他们展现在一个资产/设备模型图当中
- 值得注意的是，这些模型通常都基于机器的物理属性



例如：预测孪生

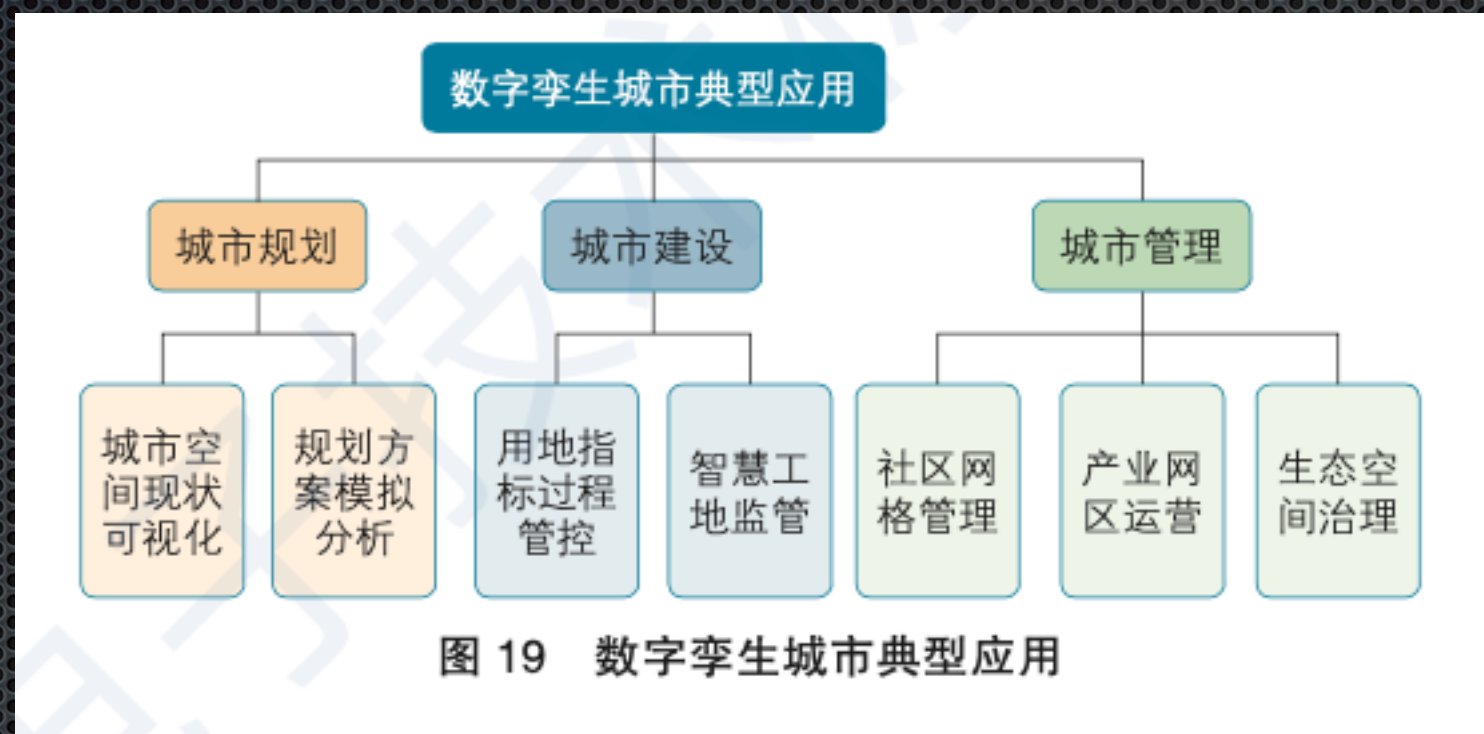
- 行为和预测建模可以通过两种方式实现：
 - 基于物理方式：使用物理的方式建模，可以利用资产的精确设计的知识和制造参数。像有限元分析之类的技术，经常用于建立高精度模型，来回答“*What-if*”之类的问题。比如，使用这种模型，用户能够根据给定的负载条件评估机器设备各个部分的应力模式
 - 基于分析/计算模型：预测模型能够使用机器学习技术来进行创建，不需要引入原始的设计人员。数据分析师仅仅基于对机器设备的外部观察就能够创建一个预测模型

智慧城市领域数字孪生应用

- 2019 年中国新型智慧城市规模超过 9000 亿元，未来几年将保持较快速度增长，预计到 2023 年市场规模将超过 1.3 万亿元
- 当前，安全综治、智慧园区、智慧交通是智慧城市建设投入的重点，三大细分场景规模占智慧城市建设总规模的71%，而城市级平台、机器人等新技术和产品则在快速落地，被更多城市建设方采纳和应用



典型应用场景介绍



参考文献

- Challenges and Opportunities in Edge Computing.pdf
- edge and fog computing for IoT A survey on current research activities future directions
- <https://iot-analytics.com/digital-twin-market-analyzing-growth-emerging-trends/>