

SD474408

## 结合参数化设计和 **Design Automation** 云计算功能，快速 搭建立体物流仓的云 **WMS** 服务平台

马晓音

杭州杉帝科技有限公司

### 学习目标

- 在本课程结束时，受众将能够了解 iLogic 与 DA 的关系
- 在本课程结束时，受众将能够借助 iLogic 完成参数化建模制图
- 在本课程结束时，受众将能够借助 Forge 的 DA 实现云端三维建模制图
- 在本课程结束时，受众将能够借助 Forge 的 View 整合实现 WMS 系统的三维可视化

### 描述

我们构建一个自动化仓库建模系统，系统通过用户将建仓数据、仓库布局数据等信息导入系统，系统借助 Forge 的 Design Automation 云计算服务完成自动化建模，传统 WMS 通过调用由系统生成的三维模型并经过简便的系统功能升级后，能够在客户端呈现一个三维可视化版本的 WMS 系统。

### 讲师

马晓音

杭州杉帝科技有限公司联合创始人之一，目前是公司的 CTO。主要负责项目相关的平台构建及软硬件产品研发。

一个经历过中国互联网技术浪潮的 80 后技术人，从嵌入式开发到云端大数据，从互联网 B2B 到安防领域，一直在技术的道路上摸爬滚打，在经历了技术所带来的翻天覆地变革之后，想把互联网思维带入最为传统的中国制造业。

## 介绍

这次行业演讲提供了关于为什么以及如何将 **Design Automation**（以下简称 **DA**）作为设计和实施智能建仓项目的解决方案的见解。通过展示此次在智能建仓项目中的获得的实施经验和教训，本讲义能够为任何希望通过 **DA** 解决自动建筑建模提供参考。我们会把重点聚焦在 **DA** 使用的制作、部署、调试流程，以及如何联动 **Forge** 的 **Data Management**、**Model Derivative** 和 **Viewer** 功能。

## 引入 **DA** 技术的条件和背景

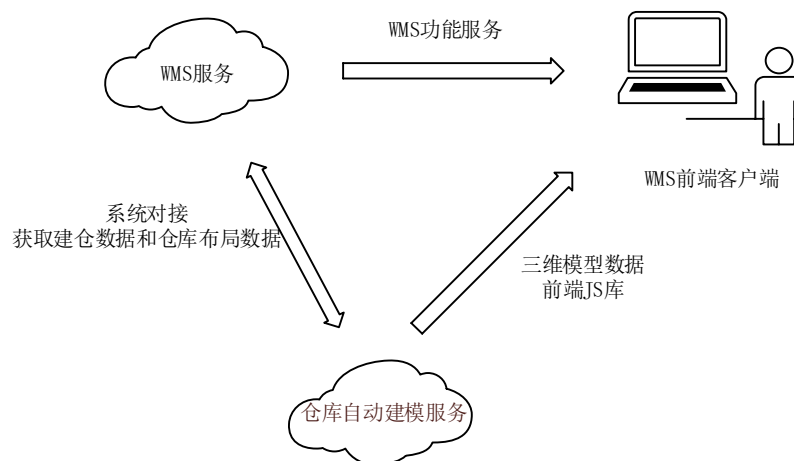
智能建仓项目是由一个专业制作 **WMS** 系统的技术团队所领导的系统开发项目，意图通过三维可视技术来解决传统 **WMS** 在复杂仓库中不易操作的使用难点。项目方决定引入三维展示技术是一次大胆的技术尝试，技术团队定义了基础功能和需求用于进行评估。

首先我们联合技术团队针对仓库的三维建模工作进行工作评估，评估的结果并不乐观，不管是项目建设方还是系统使用方都没有任何人员有三维建模经验，而且仓库的定制化程度很高，所以为仓库绘制三维模型成为一个很大的工作量，需要一个操作极其简单的仓库三维建模工具进行辅助，为此我们提出了拖拽式的布局工具进行快速三维建模。但在需求调研的后期我们发现，在三维建模这个需求上不管是技术团队还是运维团队的需求是一致的，由于需要建模的仓库数量很多，多到无法用人力完成这个建模工作，就算提供布局工具这种便捷三维建模工具所需要的人员和工作量也是极其巨大的，所以需要实现从已有的 **WMS** 系统中抽取所有仓库的建仓数据和布局信息用来自动建模。

其次我们对技术集成方式进行可行性评估，由于现有 **WMS** 系统已经成熟使用，系统本身功能完善且能够提供完善的数据查询接口，只需要将仓库三维展示功能无缝嵌入到 **WMS** 系统操作界面中，用较少的开发代价完成数据绑定和操作对接就能满足项目设计需求，而 **Web** 技术将是最好的低成本嵌入方案。最终方案是针对仓库模型提供一个集三维展示、仓位渲染、空间定位和图层操作的前端三维 **JS** 库，帮助项目方技术团队快速集成各种基与三维展示的功能开发。

最终我们提供了一个帮助项目方技术团队实现预期需求的“一站式”解决方案：**WMS** 后台系统服务调用我们自动化建模服务进行建模，而 **WMS** 的前端客户端通过调用由自动化建模服务发布

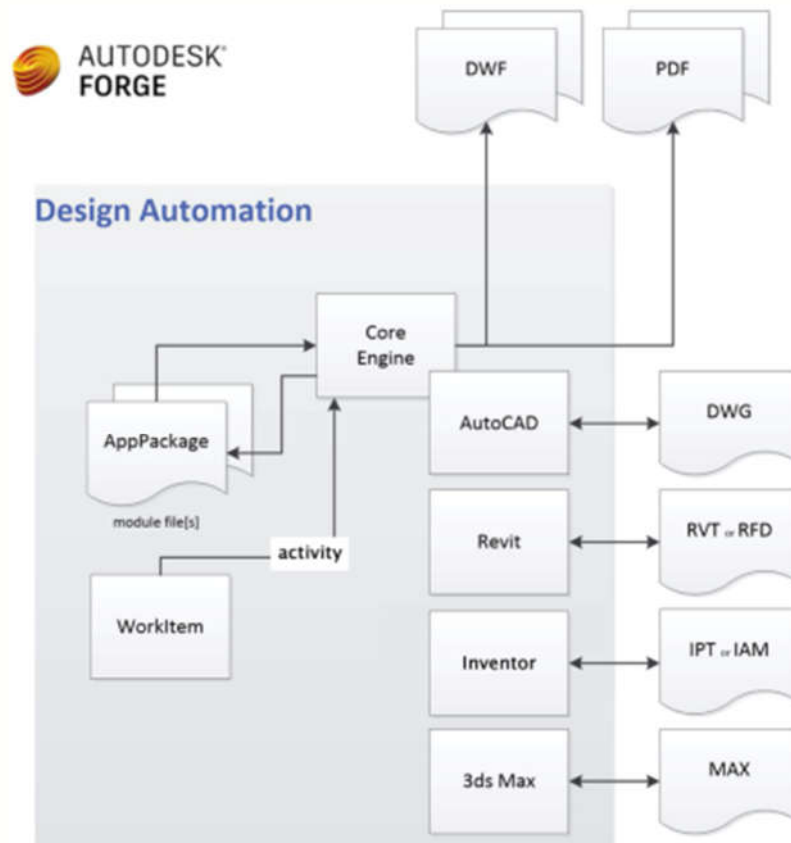
的与模型配套的前端 JS 库实现三维展示和功能操作。这种解决方案的最大好处在于模型生成和三维数据部署都实现了无人值守的全程自动化，而为三维展示功能所带来的开放也被限制在后台数据对接和前台的展示调用，大大降低项目方技术团队的开发难度和成本，而且实施周期很短，整体研发也能够控制在 4 周以内。



解决方案整体架构图

### 梳理 iLogic 和 DA 的关系

首先我们先了解 Forge DA 这个云计算服务，DA 的全称是 Design Automation 即设计自动化。它在云端提供了多种 CAD 计算引擎用于进行大规模自动化作业，这些作业任务可以是枯燥重复的、频繁的又或者是需要高强度计算的，都可以通过 DA 这个计算平台进行处理。目前 DA 的 API 支持 AutoCAD, 3dsMax, Inventor 和 Revit，在该项目中我们采用了 Inventor 这个 CAD 引擎作自动建模服务的核心造型器，Inventor 不仅能够驱动三维模型的参数化设计，又能够通过 iLogic 程序实现一些复杂装配逻辑，非常符合该项目对建模设计的要求。



DA 引擎及模块说明图

为了让自动建模模块能够正常运作，其最为重要的核心点在于如何将含参模型与建仓数据、布局数据进行整合，实现自动化三维建模。为此我们首先需要借助参数化设计帮助我们可以根据货架参数快速生成各种尺寸、不同型号的货架，之后将仓库的布局数据通过 iLogic 的灵活逻辑处理进行动态模型生成和装配，最终将这些动作通过程序和参数进行固化（DA 中的 AppBundle 和 Activity）后交由 Forge 进行批量处理并等待输出，至此 Forge DA 的流程走完，综上所述我们可以将 DA 的本质理解为将预先设计好的 iLogic 程序通过业务逻辑进行自动化执行。之后就可以进行各种操作，例如模型文件获取或者直接对接 Forge 的 Model Derivative 服务进行文件格式转换或者轻量化压缩操作。

### DA 相关的专业术语描述

术语	描述
<b>AppBundle</b>	<p>一个封装好的动态连接库模块，用来执行特定功能，通常情况下是一个 dll 文件或是 CAD 引擎的插件文件。</p> <p>例如：使用 Inventor 引擎，我们通过定义一个插件模块来操控 Inventor 从配置文件中读取参数列表，同时打开需要处理的含参模型，重新设定模型中的参数数值导出并保存这个重新绘制的新模型文件。</p>
<b>Activity</b>	<p>通过配置，可以为不同任务指定引擎、AppBundle 以及所需要的配置参数，这些配置参数可以是：输入输出的路径配置和载入和输出的方式定义；AppBundle 模块所自定义的执行参数；CAD 引擎的启动参数等。</p> <p>例如，针对同一文件需要实现不同版本输出，我们可以指定不同版本的 CAD 引擎，执行同一个 AppBundle 来实现。</p>

## 借助 iLogic 完成参数化建模制图

在‘梳理 iLogic 和 DA 的关系’章节中解释了 DA 的核心本质是执行 iLogic 程序，那么如何设计这个能够自动化执行的 iLogic 尤为重要，在项目中我们把整个自动化建模业务拆分成四个阶段进行实施：

1. 为所有货架、传送带、AGV 小车建立含参模型库。
2. 编写能够识别仓库布局数据并根据数进行仓库模型装配的 iLogic 程序。
3. 构建一个 DA 自动化程序，实现识别建仓数据，并将仓库中用到的各种模型通过之前准备的含参模型库进行自动化构建备用，构建完成后调用能够自动化装配的 iLogic 程序进行模型组装。

在这章节里我们着重梳理一二两个实施阶段。

4. 在云端部署服务，提供安全可靠的 API 接口用于 WMS 系统后台服务的调用，至此云服务交付完成。

### 针对物流仓储行业装备模型的特殊性进行参数化设计

在仓储行业中，往往用到的装备数量很多但品类并不多，不同仓库用到的同一装备存在细节差异往往只是尺寸、层高等参数，这些参数就是我们在制作含参模型时选择的关键参数。在本项目中，拿货架来讲，货架类型只有三种（平板货架、托盘货架和自动托盘货架），变化点只有长、



宽、高、层数、列数和排数等，这种类型的三维建模非常适合参数化后进行自动模型构建，同理我们根据项目需求还针对传送带、AGV 小车等物流仓储行业常用模型进行参数化构建。通常除了需要在三维模型中体现这些行业特殊装备外，还有而其他通用办公设施需要表现，为此我们直接对接 FDU 的含参模型库，为不同建模要求提供多种自动建模服务。

### 编写自动化建仓模块

这部分的工作相对容易，WMS 系统本身用来存储建仓和布局信息的数据就是已二维表的方式进行存储，几乎可以无缝转换成为一张 Excel 表。随后我们借助 Inventor 强大的 iLogic 功能，通过构建一个装配文件，并在这个文件中编写一个能够从 Excel 表中读取仓库布局数据，并根据这些数据从已经通过参数化构建好的装备模型进行模型的组装，一个步骤就能够实现仓库建模自动化。

小技巧，iLogic 可以直接打开 Excel 文件，并进行从中读取数据，操作流程如下：

- 打开一个 excel 文件，并选择 Sheet1 页：`GoExcel.Open("excel_file_name.xlsx", "Sheet1")`
- 读取指定单元格的数据：`GoExcel.CellValue("G1")`
- 用完之后记的要关闭这个文件：`GoExcel.Close()`

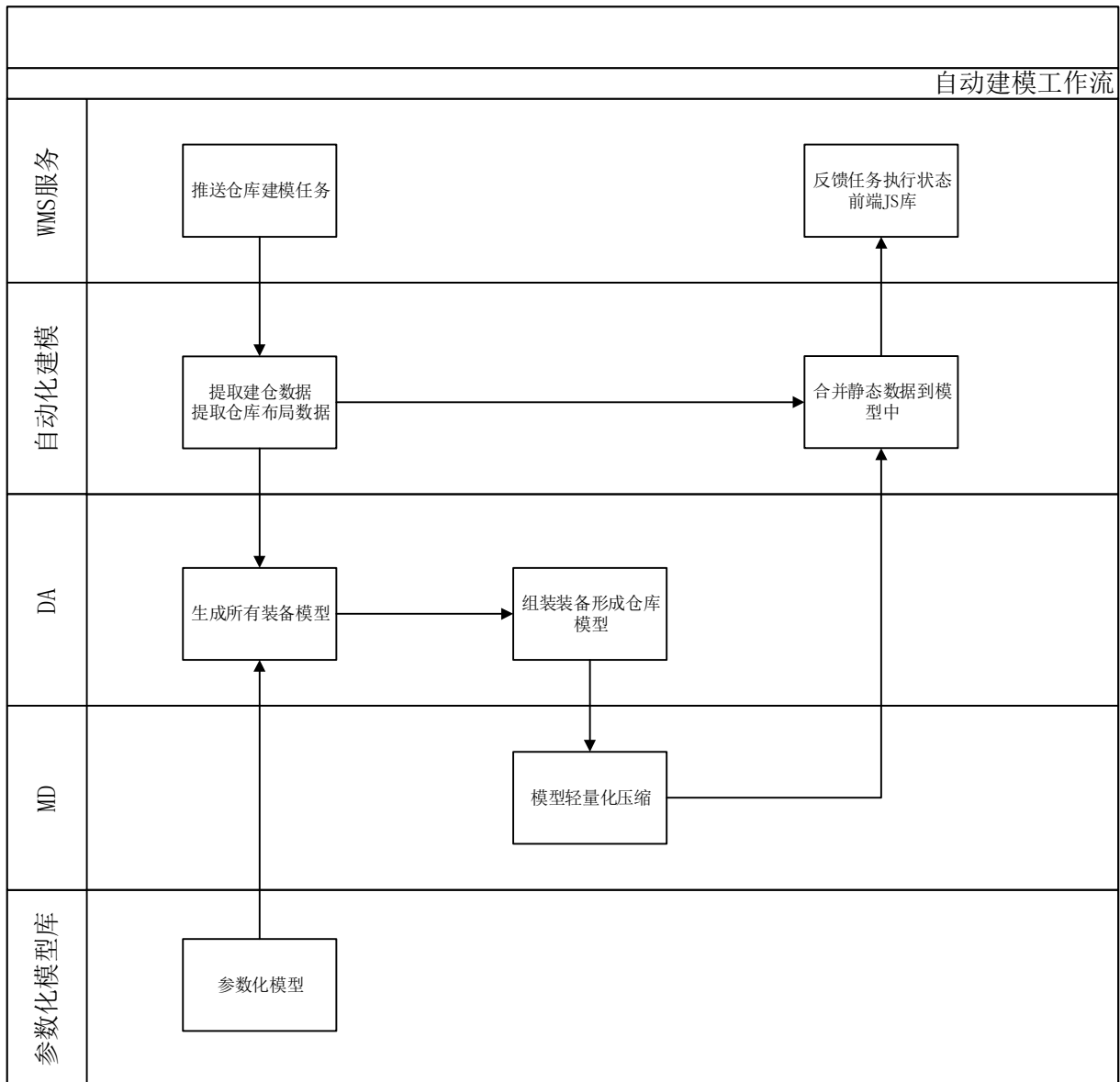
### 借助 Forge 的 DA 实现云端三维建模

之前的准备工作完成后，我们可以通过 Inventor 进行技术验证，查看整个 iLogic 程序是否按照设计要求进行正确运行。同时我们需要检查我们的 iLogic 程序是否符合 DA 的运行要求，我们需要注意的就是需要去掉所有带 UI 交互的操作，例如 Confirm 确认以及 MessageBox 提示呼出都是不被允许的，原因就是当这部分 iLogic 程序在 DA 上执行的时候没机会进行这些人机交互去进行这些确认操作。

接下来的工作就开始水到渠成了，通过 DA 我们把原本只能本地单机计算的手动建仓程序升级成为一个能够海量处理建仓数据的自动化建仓云计算服务。基于这个云计算服务，我们为项目技术团队提供安全可靠的 API 接口，技术团队通过这些 API 接口可以轻松实现系统级的数据对接，帮助技术团队花费最少代价实现仓库的自动化三维建模工作。

至此，整个自动化建模业务逻辑就完成了，在图中（自动化建模服务在项目中的工作流示意图）描述了自动化建模服务是如何与 WMS 进行系统级对接的。WMS 系统的后台服务端发起建模要求，通过 API 接口发送任务请求到自动化建模服务上，任务请求包含建仓信息、布局数据和仓位关联等数据，当收到建模请求之后，自动化建模服务会反馈唯一任务 ID 号，之后的所有建模、数据绑定以及前端展示用的 JS 库生成等任务都由自动化建模服务器通过 Forge 或本地计算进行协调处理生成。而 WMS 服务器只需用之前由自动化建模服务反馈的唯一任务 ID 进行任务进度跟踪，当任务处理完成后，调用下载接口下载与之配套的前端展示 JS 库，推送给需要前端客户浏览器后就完成 WMS 的三维化接入了。

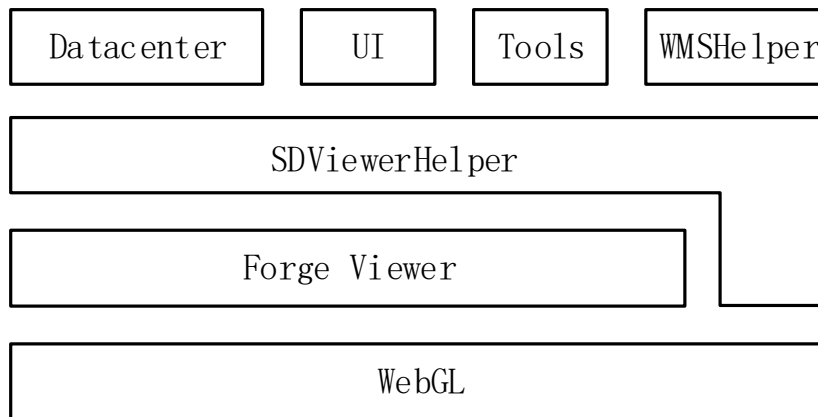




自动化建模服务在项目中的 workflow 示意图

### 整合实现 WMS 系统的三维可视化

在解决仓库三维建模问题后，我们还需要帮助项目技术团队解决技术整合问题，由于技术团队没有三维可视化的相关技术储备，为此我们需要将各种基于三维可视化的操作进行功能封装，尽可能降低技术团队在引入新技术后所带来的学习成本，为此我们提供了一套基于 Forge Viewer 二次开发的 WMS 三维可视化前端 JS 库—SDViewerHelper.JS。



前端 JS 库架构

### 数据是模型的核心

就三维可视化功能来讲，三维展示仅仅只是很小的一个功能点，最大的功能是将数据在三维层面上进行数据可视化，而如何将数据绑定在三维模型中的各个部件是一个技术需要解决的难题。所以 **SDViewerHelper.JS** 库除了提供基础的三维展示功能外，系统还自动将所需要的数据绑定在三维模型中，通过一个名为 **Datacenter** 的子集进行数据管理和操作。在这个项目中系统在构建模型的同时，自动将数据构建到于此匹配的 **JS** 库，将建仓信息中的仓位数据自动绑定在三维模型中，使用者不用理解任何三维编程原理，单纯引用这个 **JS** 库之后，只需要通过仓位号和 **JS** 库提供的接口就可以快速定位模型中该仓位，并针对该仓位进行各种三维操作动作。除了提供这种静态数据的绑定功能之外，**JS** 库还提供动态数据的绑定，在这个项目中，我们把仓库的仓位状态信息通过不同的颜色动态显示在三维模型中。

### 操作是三维的关键

当操作界面从传统的二维进入到三维时，操作方式就有更多的体现方式，原本只能通过文字查询的粗旷方式，我们可以在模型上直接点击，通过点击模型的不同部位进行不同的功能操作，为此，我们在 **SDViewerHelper.JS** 库中提供一个 **UI** 子集，专门用于提供在三维场景中的人机交互功能，提供了悬停、点击、绑定菜单、弹窗等交互方式。在该项目中，我们改变了原本 **WMS** 系统中需要通过表格查询各仓位中存放的物品信息的方式，采用鼠标右键仓位进行弹窗进行直观查看。