



# 泛亚汽车技术中心

Pan Asia Technical Automotive Center

## 探索“智能+”发动机数字化研发之路

许智勇, 泛亚驱动系统部



MATLAB EXPO

# 泛亚汽车技术中心的发动机数字化研发探索之路

## 挑战

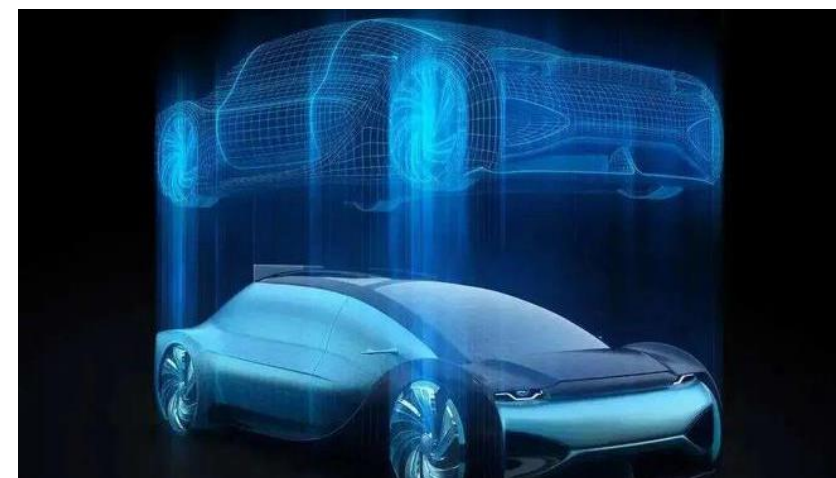
汽车市场竞争激烈，车型更新迭代快，车辆驱动系统的新产品、新技术层出不穷，因此提高研发效率、缩短设计周期对提高发动机产品的竞争力尤为重要。

## 解决方案

- 通过实践，从研发方法、知识、工具等几个方面入手，将开发的公式、流程、知识、经验等内容**模型化和代码化**；
- 利用MATLAB系列组件，定义了发动机**架构和技术策略开发框架**，搭建了发动机技术**策略库智能决策和优化模型**；
- 利用**智能算法**解决了发动机开发中的多目标决策问题，并开发了实用的发动机研发APP，集成了各领域的算法和模型。

## 结果

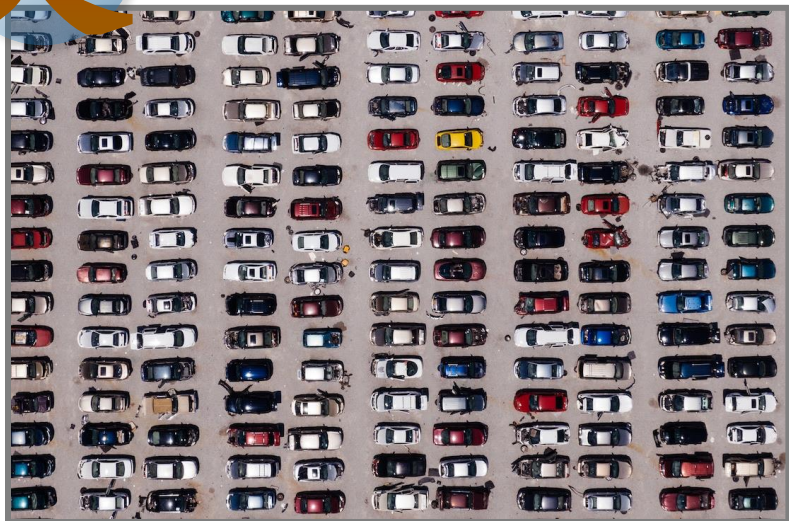
- 为发动机的数字化研发制定了完整的框架和实施方式
- 智能算法的应用提高了决策过程的科学性
- 研发APP成为模型和代码的管理及应用平台，便于日常使用及维护更新，提高了研发工作的效率。



# 背景

## 当前市场汽车产品的特点

# 炙



竞争焦灼  
风起云涌

# 智



面向未来  
以“智”赋能

# 质



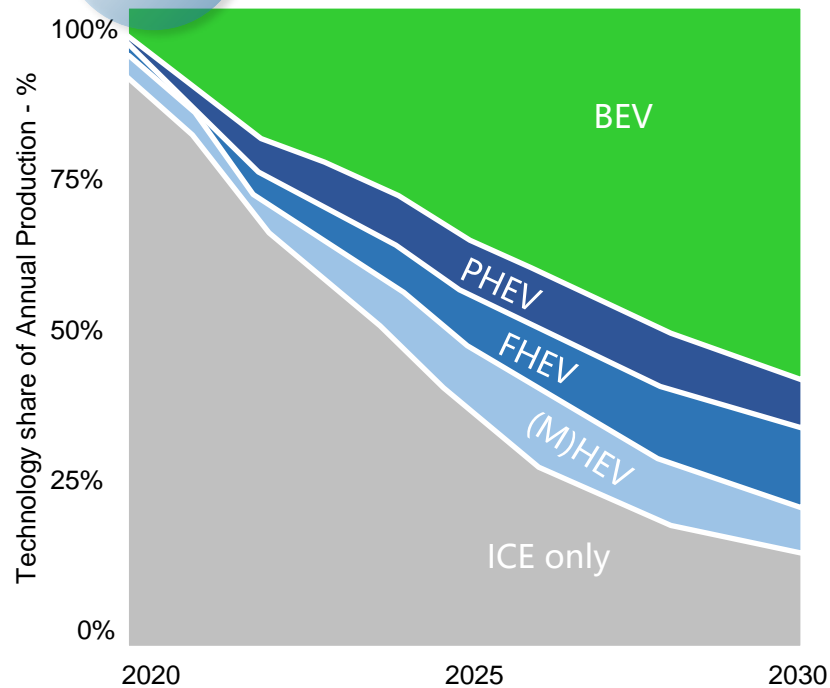
剑指高端  
以“质”为本



# 背景

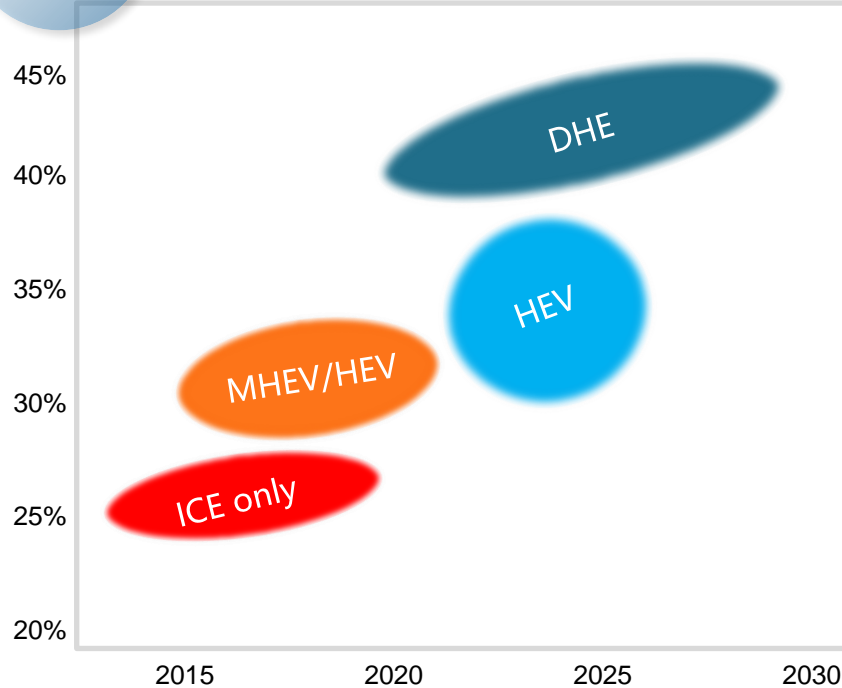
车辆驱动系统技术一日千里

## 多元化



内燃机/电机/电池技术各领风骚

## 高效化



内燃机效率最高达45%~46%

## 零碳化

- 柴油 Diesel
- 汽油 Gasoline
- 液化石油气 LPG
- 压缩天然气 CNG
- 合成柴油 CTL/GTL/BTL

**可再生能源**  
Renewable Energy

- 生物能 Biomass
- 水能 Hydraulic
- 风能 Wind Energy
- 太阳能 Solar Energy
- E-Fuel 合成燃料

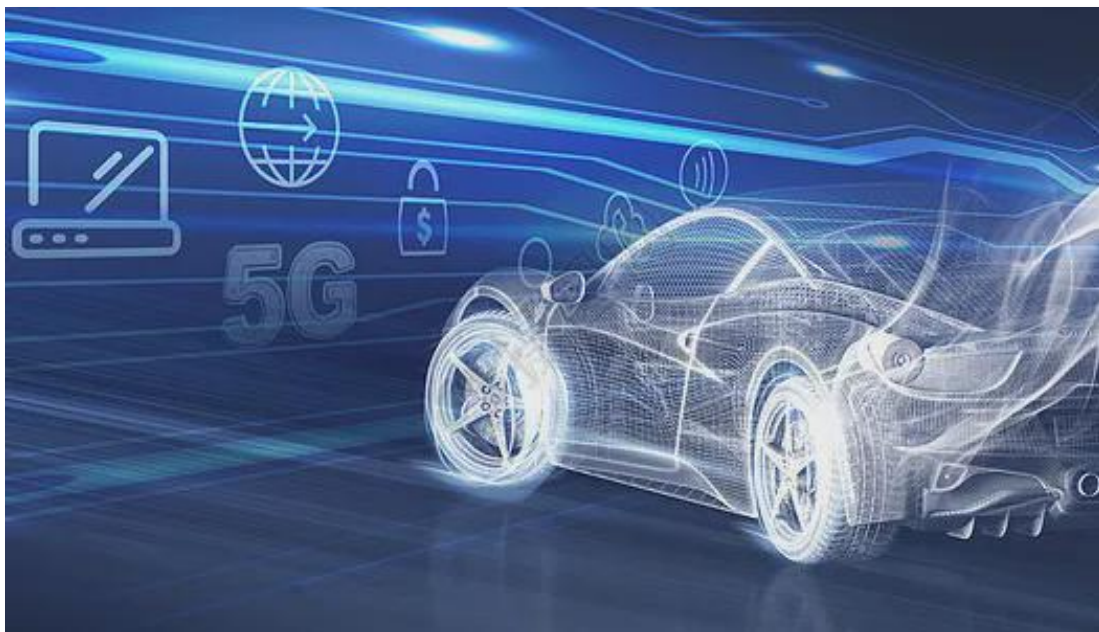
氢气 Hydrogen

电 Electricity

低碳/零碳燃料的开发与使用

# 背景

研发数字化是缩短开发周期，降低研发成本，提高研发效率，提升交付质量的有效路径。  
泛亚发动机研发数字化，涵盖知识动态化、决策智能化、设计自动化、流程可视化等内容



## 机遇

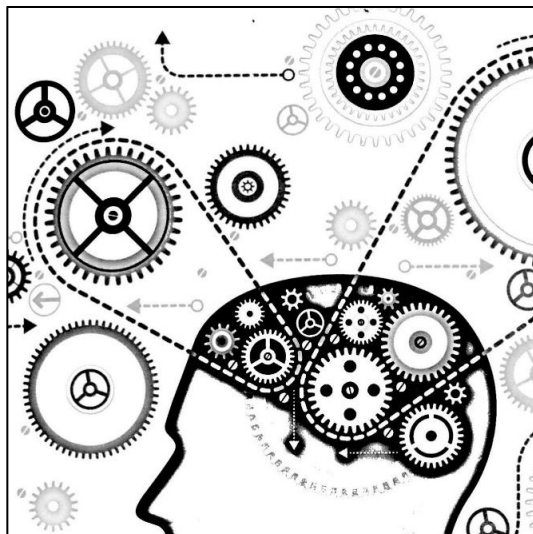
- 数字化转型是各行各业大势所趋
- 人工智能技术的发展奠定了良好的技术基础

## 挑战

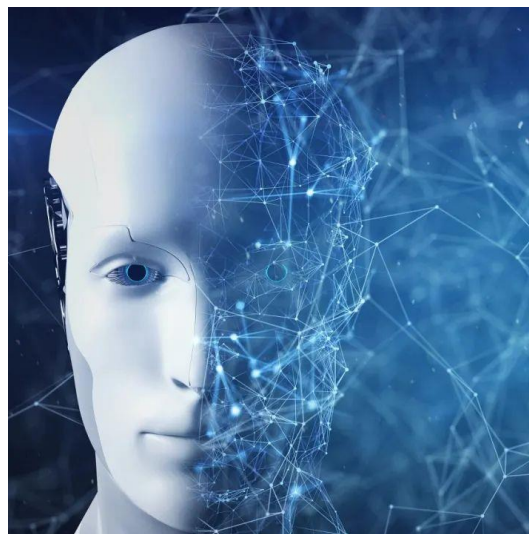
- 不同组织的数字化转型工作不能生搬硬套理论，结合具体业务才能实现最终落地
- 初始阶段的投入回报比较低，忌浅尝辄止，需要统一思想，坚持不懈。



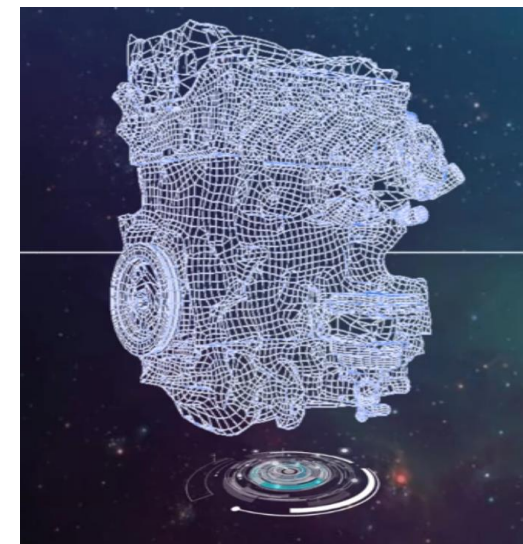
# 内容概要



数字化的思考与实现  
研发数字化  
数字化研发



“智能+”决策  
知识工程  
智能决策/自动运行



模型与平台  
模型的集成及使用

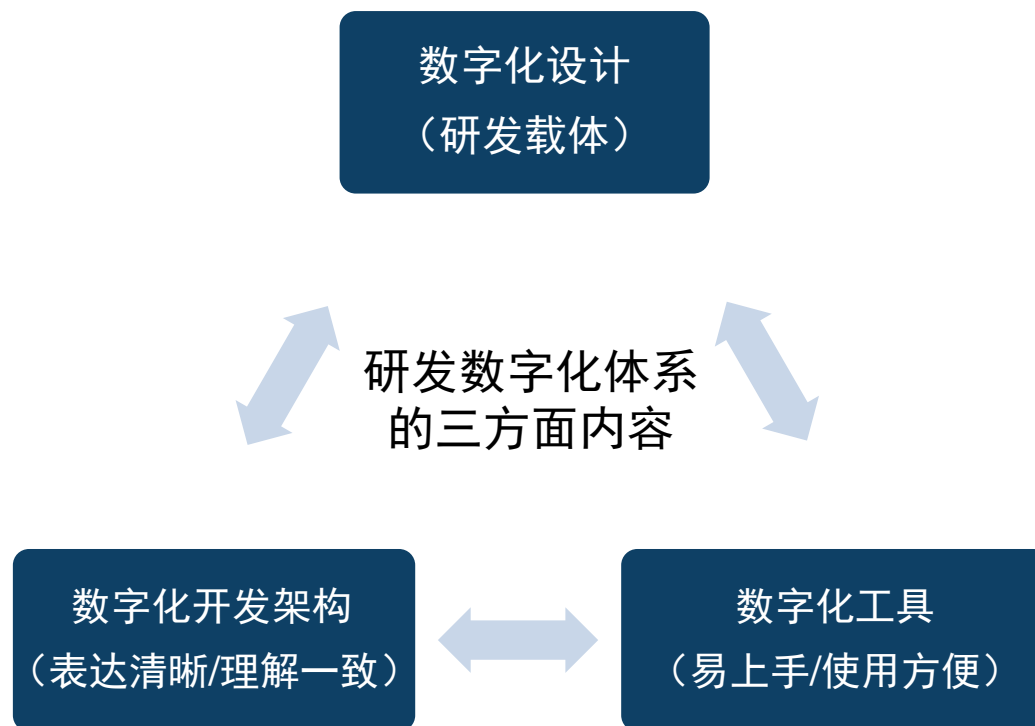


# 数字化研发愿景



# 研发数字化的愿景及过程解耦

- 研发数字化是知识沉淀的创新路径
- 将**研发经验、研发文档、研发活动、研发策略**等内容，根据一定的**方法论**，借助**软件开发平台**，形成有层次、相关联、可运行的函数、模型或者代码，以实现在虚拟空间完成研发任务的目标。
- 研发数字化价值
  - 实现知识继承
  - 解决人才断层
  - 提升研发效率





# 数字化研发架构及技术探索

## 研发 workflow 架构



传统业务流的开展依赖于需求文档、流程文档、技术文档等文档类知识

分析、评审、验收等工作需要有经验的专家团队来完成

研发架构数字化，即建立工作业务的流转顺序和定义任务间的数据交互

# 数字化研发架构及技术探索

## 研发 workflow 架构

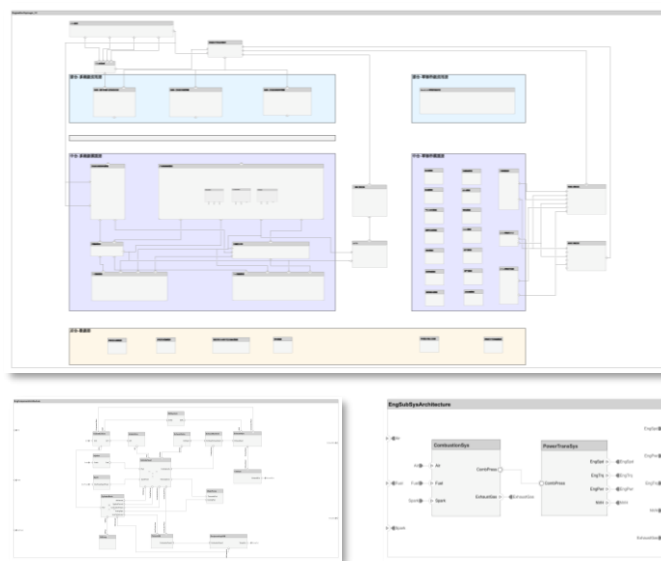
传统 workflow 大量依赖“文档”及“人”



使用 MATLAB Requirement 定义需求

ID	需求描述	需求类型	需求状态	需求来源
1.1	系统级需求	Functional	Approved	System Architect
1.2	子系统级需求	Functional	Approved	System Architect
1.3	组件级需求	Functional	Approved	System Architect
1.4	接口需求	Functional	Approved	System Architect
1.5	性能需求	Functional	Approved	System Architect
1.6	可靠性需求	Functional	Approved	System Architect
1.7	安全性需求	Functional	Approved	System Architect
1.8	兼容性需求	Functional	Approved	System Architect
1.9	可维护性需求	Functional	Approved	System Architect
1.10	可扩展性需求	Functional	Approved	System Architect
1.11	可测试性需求	Functional	Approved	System Architect
1.12	可部署性需求	Functional	Approved	System Architect
1.13	可移植性需求	Functional	Approved	System Architect
1.14	可配置性需求	Functional	Approved	System Architect
1.15	可定制性需求	Functional	Approved	System Architect
1.16	可升级性需求	Functional	Approved	System Architect
1.17	可降级性需求	Functional	Approved	System Architect
1.18	可恢复性需求	Functional	Approved	System Architect
1.19	可审计性需求	Functional	Approved	System Architect
1.20	可追溯性需求	Functional	Approved	System Architect
1.21	可验证性需求	Functional	Approved	System Architect
1.22	可证明性需求	Functional	Approved	System Architect
1.23	可解释性需求	Functional	Approved	System Architect
1.24	可理解性需求	Functional	Approved	System Architect
1.25	可沟通性需求	Functional	Approved	System Architect
1.26	可协作性需求	Functional	Approved	System Architect
1.27	可集成性需求	Functional	Approved	System Architect
1.28	可互操作性需求	Functional	Approved	System Architect
1.29	可兼容性需求	Functional	Approved	System Architect
1.30	可移植性需求	Functional	Approved	System Architect
1.31	可配置性需求	Functional	Approved	System Architect
1.32	可定制性需求	Functional	Approved	System Architect
1.33	可升级性需求	Functional	Approved	System Architect
1.34	可降级性需求	Functional	Approved	System Architect
1.35	可恢复性需求	Functional	Approved	System Architect
1.36	可审计性需求	Functional	Approved	System Architect
1.37	可追溯性需求	Functional	Approved	System Architect
1.38	可验证性需求	Functional	Approved	System Architect
1.39	可证明性需求	Functional	Approved	System Architect
1.40	可解释性需求	Functional	Approved	System Architect
1.41	可理解性需求	Functional	Approved	System Architect
1.42	可沟通性需求	Functional	Approved	System Architect
1.43	可协作性需求	Functional	Approved	System Architect
1.44	可集成性需求	Functional	Approved	System Architect
1.45	可互操作性需求	Functional	Approved	System Architect
1.46	可兼容性需求	Functional	Approved	System Architect
1.47	可移植性需求	Functional	Approved	System Architect
1.48	可配置性需求	Functional	Approved	System Architect
1.49	可定制性需求	Functional	Approved	System Architect
1.50	可升级性需求	Functional	Approved	System Architect
1.51	可降级性需求	Functional	Approved	System Architect

使用 system composer 定义任务、接口、部分评审流程



研发架构数字化，利于整理思路、任务分配、明确模块目标、易于检查完整性

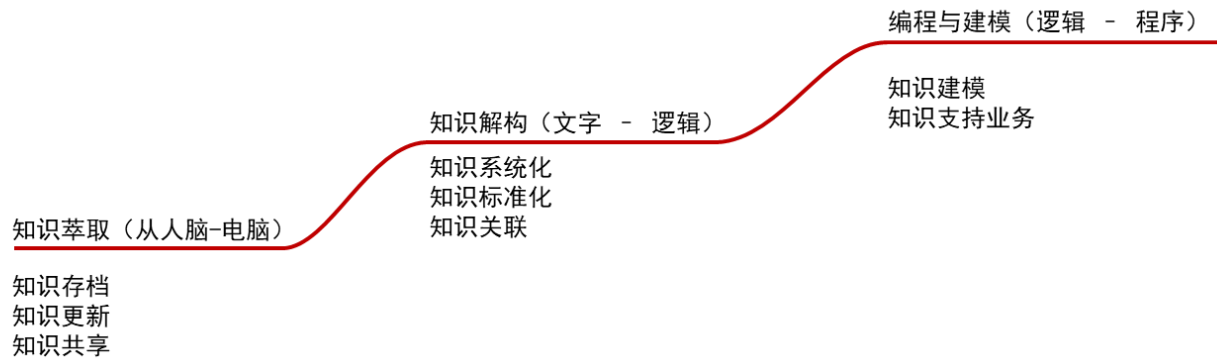
使用 Requirement Manager 和 MATLAB System Composer 搭建数字化的研发架构模型，表达清晰、接口明确，易于信息准确传递、避免误解，迭代更新方便快捷

研发架构的数字化

# 使用数字化工具解决决策问题

- 自动化运行的关键步骤——决策，即：把静态的知识库变为动态知识工具
- 决策基础：知识工程
- 决策三要素：**决策指标、决策流程、决策算法和工具**

## 知识工程的实施步骤



知识文档化

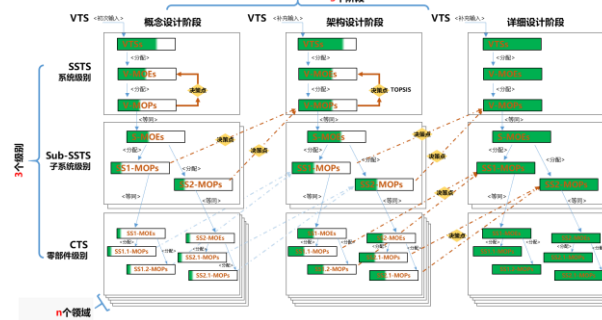


知识数字化

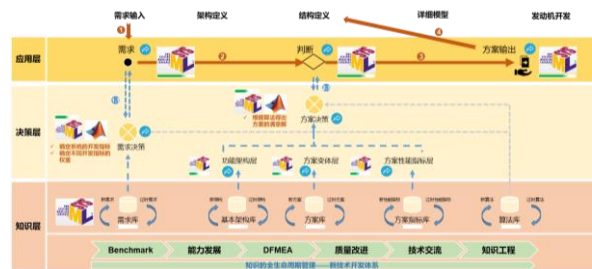


知识智能化

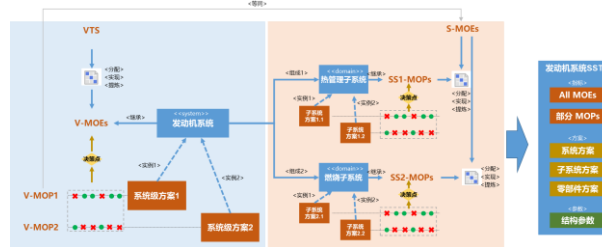
## 决策指标 定义决策依据和目标



## 决策流程 定义决策点



## 决策算法





# 使用数字化工具解决决策问题

- 自动化运行的关键步骤——决策，即：把静态的知识库变为动态知识工具
- 决策基础：知识工程
- 决策三要素：决策指标、决策流程、决策算法和工具

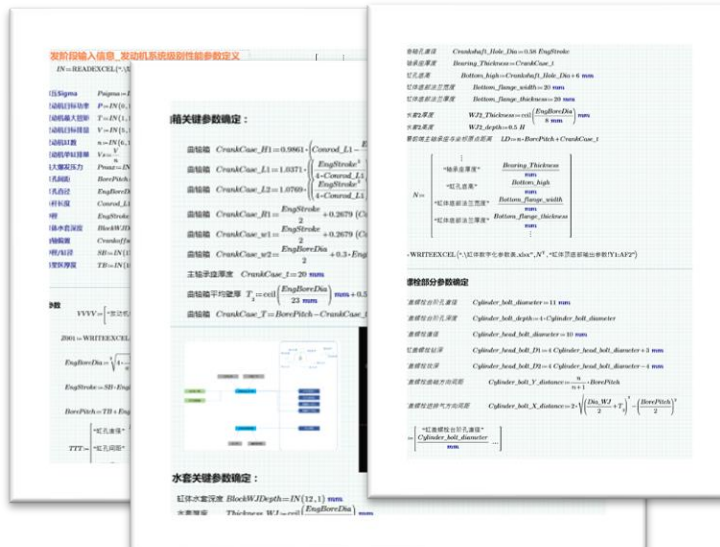
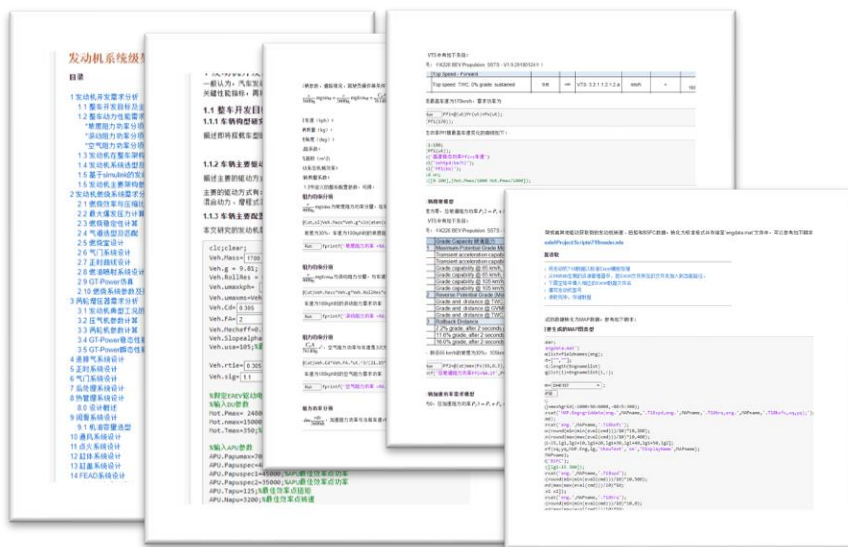
静态的知识  
大脑中的知识



使用MATLAB Live Script沉淀领域知识

使用MathCAD沉淀领域知识

利用Live Script和  
MathCAD转化为动态知识



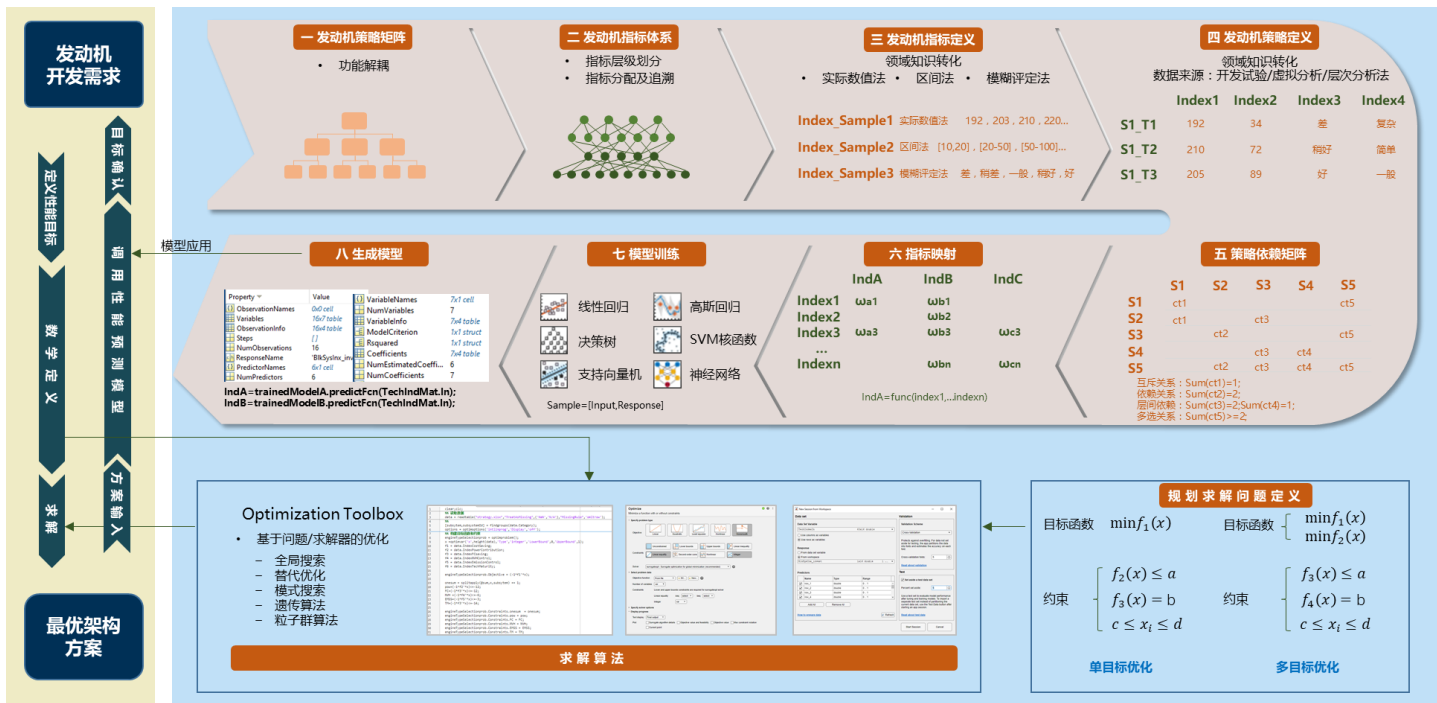
知识动态化、可运行

人与计算机均可识别  
利于知识迭代更新

不同领域的动态文档  
可联合进行参数优化

# “智能+” 算法赋能决策算法

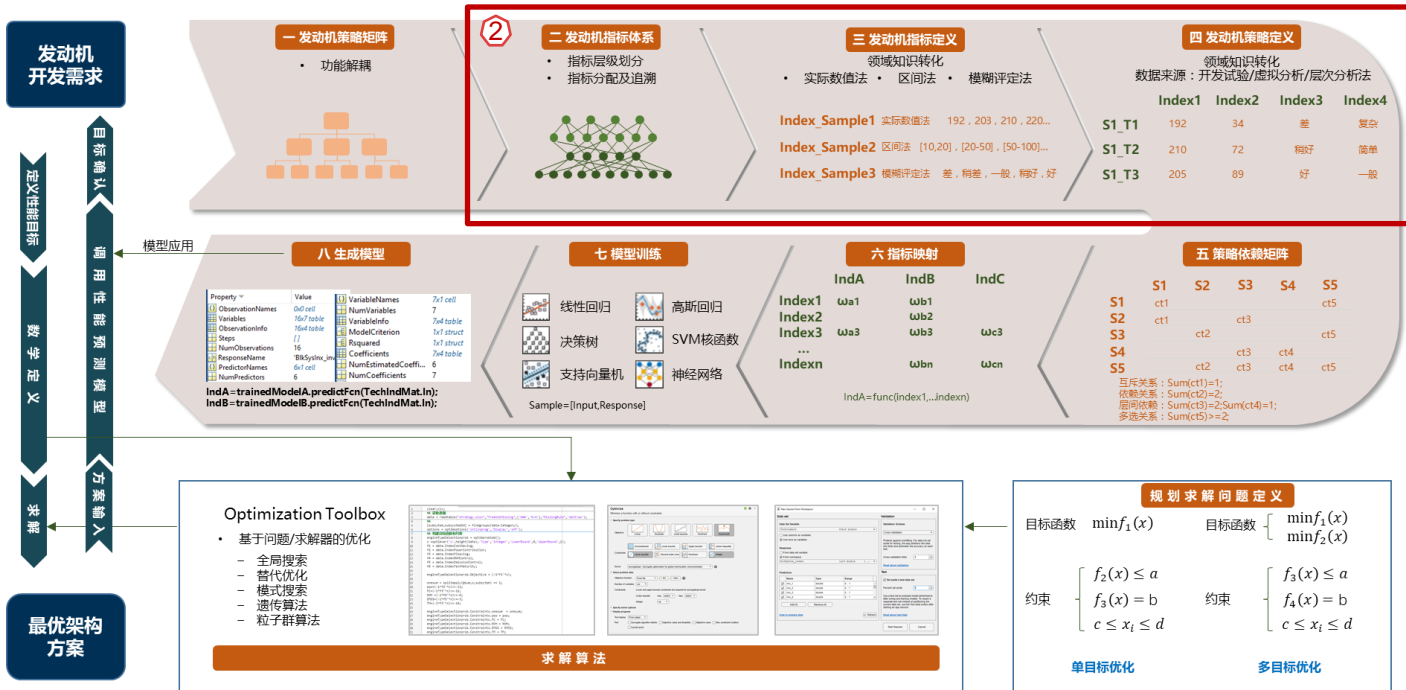
- 决策算法——决策自动化的核心步骤，以数学方式解耦决策过程
- 决策三要素：**决策指标、决策流程、决策算法和工具**
- 数字化工具的应用：问题建模——单目标/多目标优化工具箱**Optimization Toolbox**，拟合和优化——**线性规划函数**拟合工具箱
- 决策算法价值：**科学性、一致性、升级潜力**



- 1 隐性知识向显性知识转化，让知识看得见
- 2 运用系统性、全面性的指标体系，利于知识表达标准化、规范化
- 3 应用基于模型的技术 (Model Base Technology) 表达清晰无异议，确保变更的正确转递
- 4 决策模型依赖于指标体系、依赖矩阵、映射矩阵、训练模型等环节，既包含领域知识、又运用了数据科学，随着数据的丰富，模型质量也可以获得提升

# “智能+” 算法赋能决策算法

- 决策算法——决策自动化的核心步骤，以数学方式解耦决策过程
- 决策三要素：**决策指标、决策流程、决策算法和工具**
- 数字化工具的应用：问题建模——单目标/多目标优化工具箱**Optimization Toolbox**，拟合和优化——**线性规划函数**拟合工具箱
- 决策算法价值：**科学性、一致性、升级潜力**

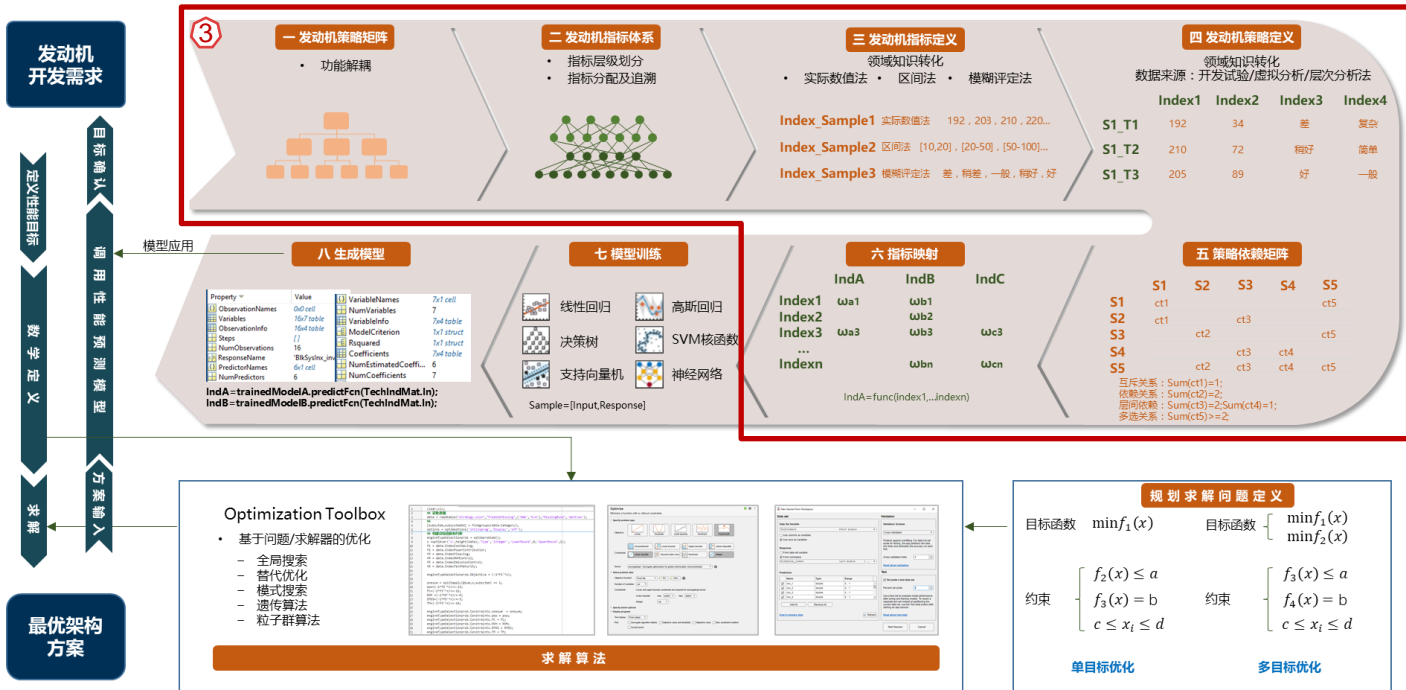


- 1 隐性知识向显性知识转化，让知识看得见
- 2 运用系统性、全面性的指标体系，利于知识表达标准化、规范化
- 3 应用基于模型的技术（Model Base Technology）表达清晰无异议，确保变更的正确转递
- 4 决策模型依赖于指标体系、依赖矩阵、映射矩阵、训练模型等环节，既包含领域知识、又运用了数据科学，随着数据的丰富，模型质量也可以获得提升



# “智能+” 算法赋能决策算法

- 决策算法——决策自动化的核心步骤，以数学方式解耦决策过程
- 决策三要素：**决策指标、决策流程、决策算法和工具**
- 数字化工具的应用：问题建模——单目标/多目标优化工具箱**Optimization Toolbox**，拟合和优化——**线性规划函数**拟合工具箱
- 决策算法价值：**科学性、一致性、升级潜力**



1 隐性知识向显性知识转化，让知识看得见

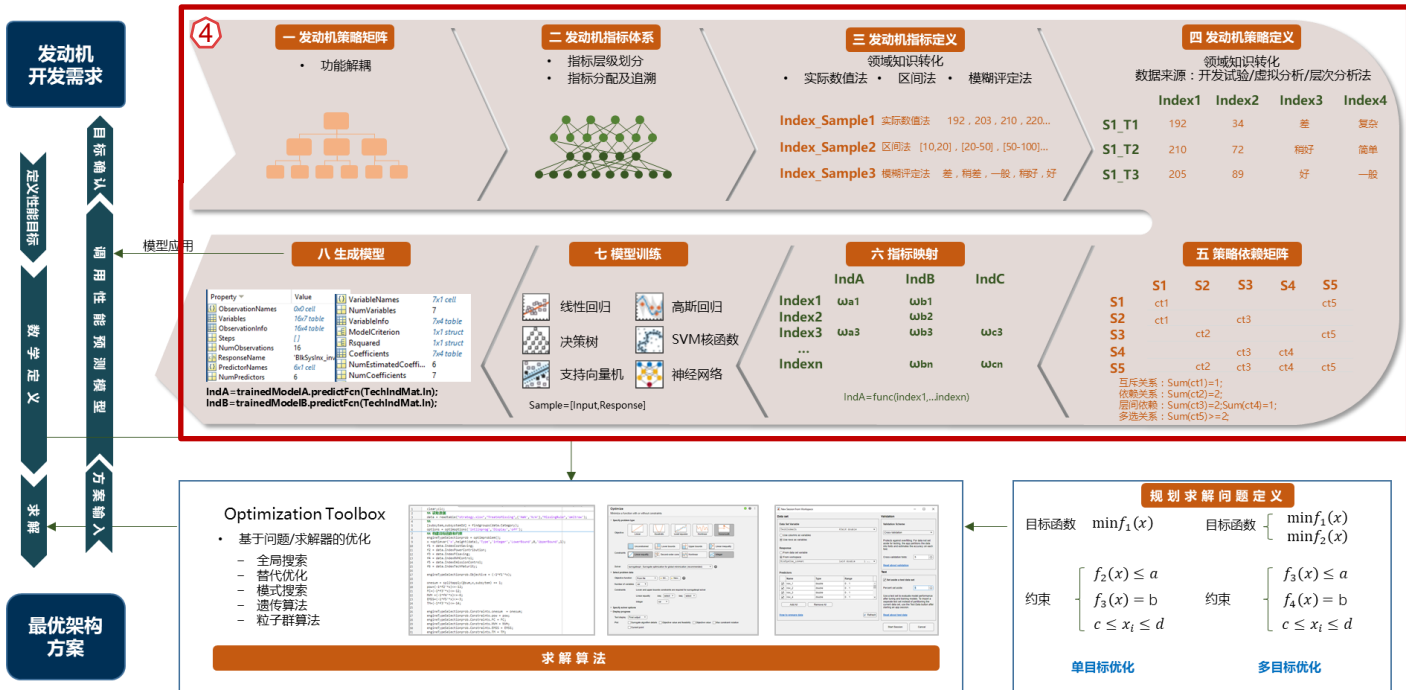
2 运用系统性、全面性的指标体系，利于知识表达标准化、规范化

3 应用基于模型的技术 (Model Base Technology) 表达清晰无异议，确保变更的正确转递

4 决策模型依赖于指标体系、依赖矩阵、映射矩阵、训练模型等环节，既包含领域知识、又运用了数据科学，随着数据的丰富，模型质量也可以获得提升

# “智能+” 算法赋能决策算法

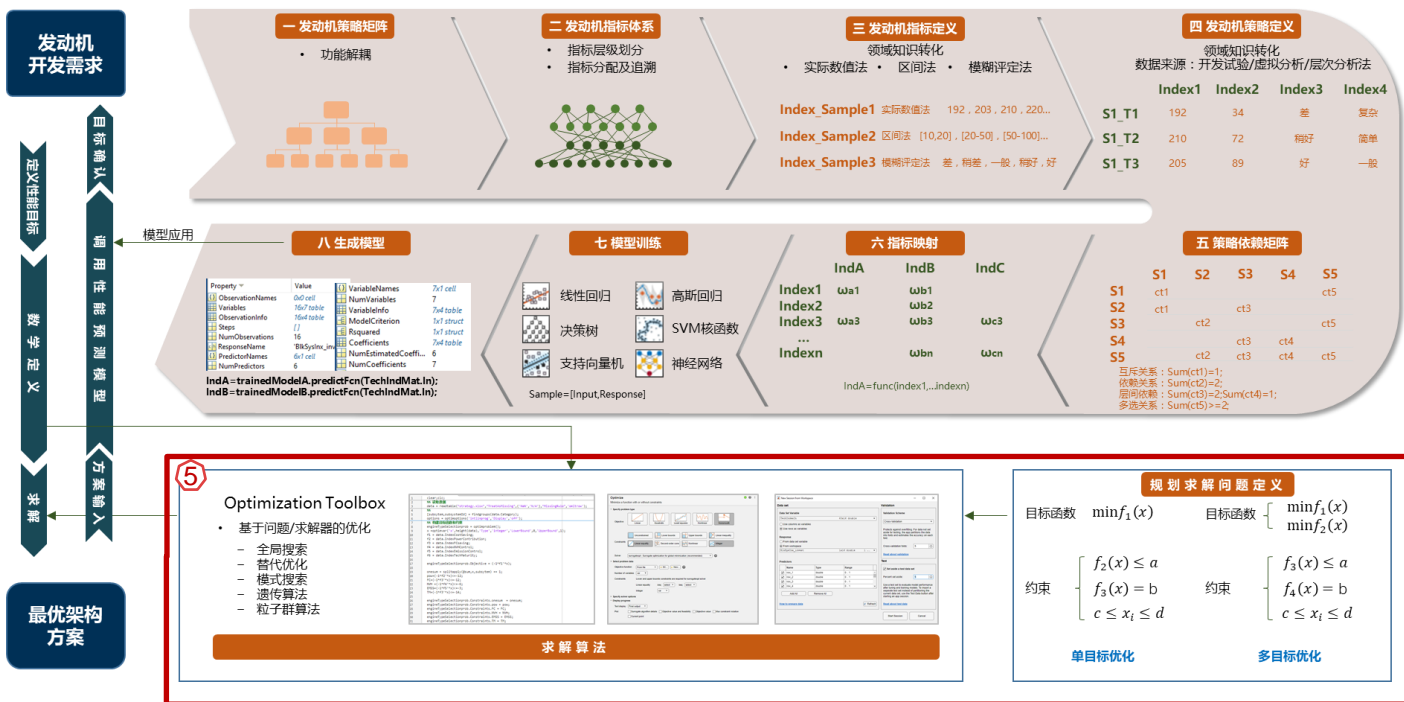
- 决策算法——决策自动化的核心步骤，以数学方式解耦决策过程
- 决策三要素：**决策指标、决策流程、决策算法和工具**
- 数字化工具的应用：问题建模——单目标/多目标优化工具箱**Optimization Toolbox**，拟合和优化——**线性规划函数**拟合工具箱
- 决策算法价值：**科学性、一致性、升级潜力**



- 1 隐性知识向显性知识转化，让知识看得见
- 2 运用系统性、全面性的指标体系，利于知识表达标准化、规范化
- 3 应用基于模型的技术 (Model Base Technology) 表达清晰无异议，确保变更的正确转递
- 4 决策模型依赖于指标体系、依赖矩阵、映射矩阵、训练模型等环节，既包含领域知识、又运用了数据科学，随着数据的丰富，模型质量也可以获得提升

# “智能+” 算法赋能决策算法

- 决策算法——决策自动化的核心步骤，以数学方式解耦决策过程
- 决策三要素：**决策指标、决策流程、决策算法和工具**
- 数字化工具的应用：问题建模——单目标/多目标优化工具箱**Optimization Toolbox**，拟合和优化——**线性规划函数**拟合工具箱
- 决策算法价值：**科学性、一致性、升级潜力**



**5 求解问题**

- 最优发动机架构方案生成，本质是个组合优化问题
- 方案空间的数量为 $\sim 2^{73}$ 个，穷举优化不可行
- 智能算法如遗传算法、粒子群算法的应用，极大的提高优化效率，单次优化时间 $< 5\text{min}$
- 应用MATLAB Optimization Toolbox工具箱进行算法开发，降低了智能算法的开发使用门槛



# “智能+” 算法赋能决策算法

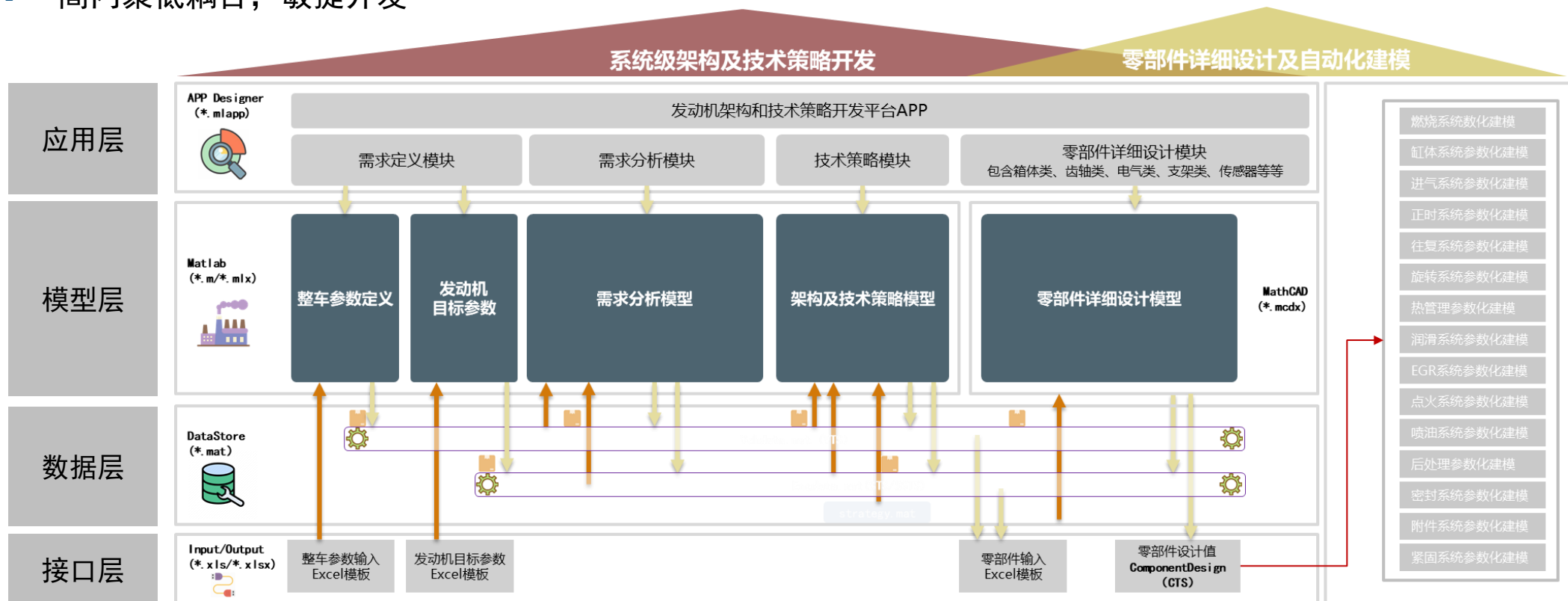
- 决策算法——决策自动化的核心步骤，以数学方式解耦决策过程
- 决策三要素：**决策指标、决策流程、决策算法和工具**
- 数字化工具的应用：问题建模——单目标/多目标优化工具箱**Optimization Toolbox**，拟合和优化——**线性规划函数**  
拟合工具箱
- 决策算法价值：科学性、一致性、升级潜力

任务	专家能力类型	任务目标	使用的工具
定义策略矩阵	架构专家/领域专家	定义系统的物理架构/虚拟架构	System Composer
定义指标体系	架构专家/领域专家	定义领域的指标体系	System Composer/Profile
定义指标类型	领域专家	确保指标数值标准化	System Composer/Profile
定义策略指标数值	领域专家	确保正确表达策略	System Composer/Profile
定义依赖矩阵 指标映射	领域专家/数据专家	知识/经验的转化	Function Define
模型训练	数据专家	建模求解	Regression Learner
规划求解	数据专家	决策模型的管理和运用	Global Optimization Toolbox

架构专家/领域专家/数据专家融合——促进工程师转型

# 模型与平台——集成界面、平台管理

- 领域知识、决策点的分散且庞杂，为了管理和使用更加方便，需要在统一平台上进行管理
- 整合开发流程、开发参数的输入输出、各领域计算和决策模型等
- 高内聚低耦合，敏捷开发



# 模型与平台——集成界面、平台管理

- 领域知识、决策点的分散且庞杂，为了管理和使用更加方便，需要在统一平台上进行管理
- 整合开发流程、参数接口、计算和决策模型、项目管理、报告生成、用户管理等功能
- 高内聚低耦合，敏捷开发



# 模型与平台——模型耦合，迭代更新

- 加速发动机技术策略研发速度
- 以设计为中心，打通虚拟分析/生产制造/质量管理的边界

**发动机架构和技术策略APP**

STEP102	STEP103
发动机1D模型建立	需求分解策略
STEP104	STEP105
需求策略分解	性能需求分解
STEP106	STEP107
性能需求分解	策略计算
STEP108	STEP109
策略需求分解计算	核心策略计算
STEP110	STEP111
策略需求分解定义	需求策略分解计算
STEP112	STEP130
策略需求分解计算	策略需求分解
STEP140	STEP150
策略需求分解分析	策略需求分解

这是我们团队设计研发的发动机架构和策略App



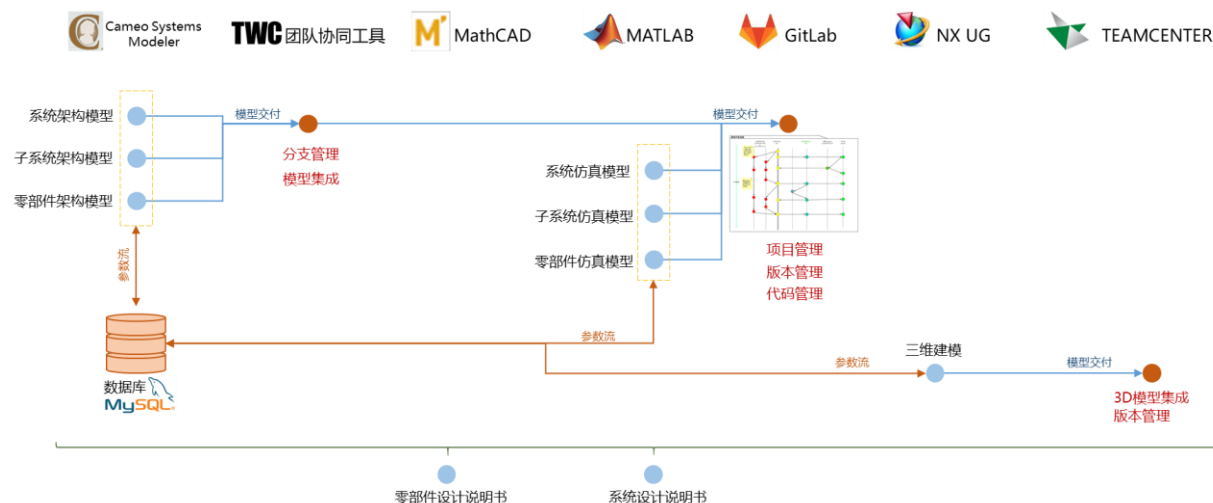
# “智能+” 推进团队开发能力转型

- 从领域专家到模型专家、数据专家、代码专家，领域知识模型化、代码化
- 规划数据资产的产生管理（计划中）

## 数字化业务下的专家角色定义

	数字化工作角色													
	管理						集成				开发			
	统一建模工具和软件版本	统一开发环境	持续更新和集成工具链	建立基本算法库	定义分工和工作流	把控模型质量	制定本级别模型架构	制定次级别交付物规范	制定次级别建模、命名规范	定义次级别模型颗粒度	完成本领域建模	完成本领域建模文档	持续更新和细化模型	响应其他领域的需求
<b>领域专家</b>														
<b>TFO</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>SFO</b>							✓	✓	✓	✓	✓	✓	✓	✓
<b>BFO</b>										✓	✓	✓	✓	✓

## 数据资产的生产链和存储链



# 总结

## 数字化设计

数字化需求、数字指标体系、数字定义产品、智能算法，达成隐形知识显性化，实现知识沉淀与继承

## 发动机数字化开发架构

利用System Composer，构筑出联结知识、模型、代码的立体网络



## 数字化开发工具

App Designer、Live Script、规划工具箱、优化求解工具箱等灵活运用，降低了开发门槛，提高开发效率

# MATLAB EXPO

Thank you



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.