# Autosar开发方法论

## 汽车电子开发流程介绍

**概念介绍：OEM、tier1、tier2**

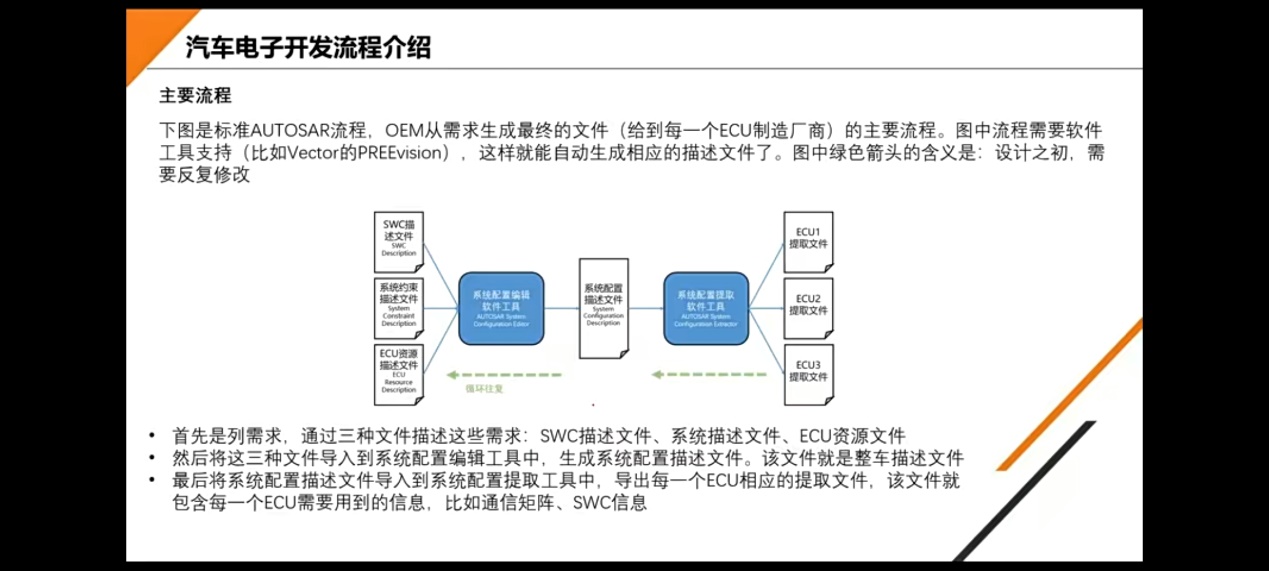
|  |  |  |  |
| --- | --- | --- | --- |
| 简称 | OEM | TIER1 | TIER2 |
| 中文名 | 代工厂（整车厂） | 一级供应商 | 二级供应商 |
| 例子 | 奔驰、宝马、奥迪等（主要做整车的装配工作） | 大陆、博世等（主要给OEM供应ECU、钣金件等） | 英飞凌、NXP等（主要给TIER1供应零件，比如ECU上的芯片、MOS管和电路板等） |

OEM属于甲方爸爸，好多应用层算法是由OEM自行完成；Tier1生产制造ECU，其中ECU的状态就需要受到OEM的约束，比如UDS、XCP、下线检测等功能要求；针对Tier2就是芯片等厂家，主要负责芯片硬件提供及开发方式提供推荐，受到TIER1、OEM的约束；工具链厂家主要提供开发工具链给tier1、OEM进行产品开发（vector、matlab、劳德巴赫等）

Autosar方法论是开发满足Autosar架构的ECU软件的开发流程。

**主要流程**

下图是标准AutoSAR流程，OEM从需求生产最终的文件（给到每一个ECU制造厂商）的主要流程。图中流程需要软件工具支持（比如Vector的PREEvision），这样就能自动生成相应的描述文件了。图中绿色箭头的含义是：设计之初，需要反复修改。



首先是列需求，通过三种文件描述这些需求：SWC描述文件、系统描述文件、ECU资源文件。

然后将这三种文件导入到系统配置编辑工具中，生成系统配置描述文件。该文件就是整车描述文件。

最后将系统配置描述文件导入到系统配置提取工具中，导出每一个ECU相应的提取文件，该文件就包含每一个ECU需要用到的信息，比如通信矩阵、SWC信息。

**描述文件**

1. **swc描述文件（应用层软件组件描述文件）**

描述每个软件组件需要的资源（比如存储、CPU时间等）

SWC直接的接口

运行机制

1. **系统约束描述文件（整车公共资源描述）**

网络拓扑

通信矩阵

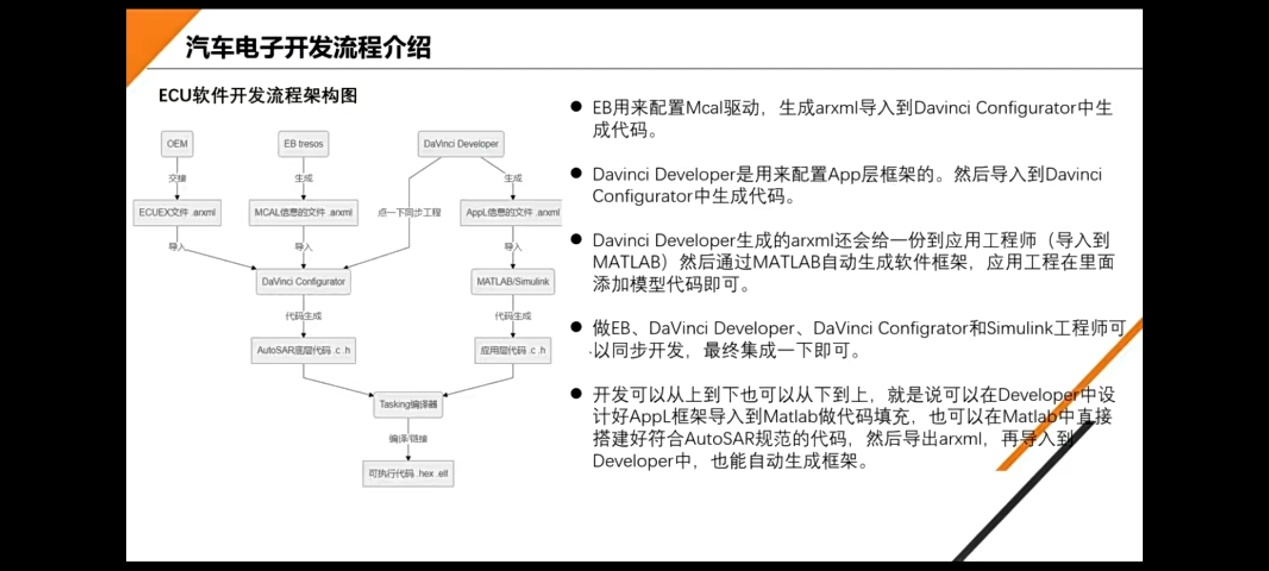
总线波特率

各种协议

1. **ECU资源描述文件（描述每个ECU都需要实现什么功能，系统设计者通过该文件将不同功能的SWC分配到对应的ECU中）**

传感器、执行器、存储器、引脚分配

1. **系统配置描述文件（上面3中文件的汇总）**
2. **ECU提取文件（将系统配置描述文件信息分配给单个ECU，是单个ECU获取属于自己的信息）**



EB用来配置MCAL驱动，生成ARXML导入到Davinci Configurator中生成代码。

Davinci Configurator是用来配置App层框架的。然后导入到Davinci Configurator中生成代码。

Davinci Developers生成的arxml还会给一份到应用工程师（导入到MATLAB）然后通过MATLAB自动生成软件框架，应用工程在里面添加模型代码即可。

做EB、DaVinci Developer生成的arxml还会给一份到应用工程师（导入到MATLAB）然后通过MATLAB自动生成软件框架，应用工程在里面添加模型代码即可。

做EB、DaVinci Developer、Davinci Configrator和Simulink工程师可以同步开发，最终集成一下即可。

开发可以从上到下也可以从下到上，就是说可以在Developer中设计好Appl框架导入到Matlab做代码填充，也可以在matlab中直接搭建设计好符合Autosar规范的代码，然后导出arxml，再导入到Developers中，也能自动生成框架。