

前言

近些年来，我国已经涌现了大量的制造型企业，随着竞争的发展，对产品质量的要求越来越高，特别是在电子制造型企业里，对于制造流程控制的要求不断提升。许多企业为了增强竞争力，不惜重金实施 ERP 系统，以求缩减成本控制、缩短供货周期。但在内部生产控制上，却没有一个完善成熟的平台。MES 制造执行系统就是在这种背景上，受到大家越来越多的关注，越来越多的企业在考虑或实施自己的 MES 系统。

本书系统地介绍了 MES 的功能模块和实现的逻辑。本人在国内大型的电子产品制造企业担任多年的工艺工程师和工厂控制系统业务分析师，对制造的流程非常熟悉，参与设计了所在企业的 MES 系统。本书是作者多年经验的总结。

希望本书有助于大家理解 MES 系统，有助于国内制造企业的流程控制和质量提升。

杭州 张涛云

tallrain@gmail.com

2006-12-1

目录

前言.....	1
第一章 什么是制造执行系统.....	1
1.1MES 的功能.....	1
1.2MES 与 ERP 的关系.....	1
第二章 产品、工单、序列号.....	2
2.1 产品(Product).....	2
2.2 工单(Work Order).....	2
2.3 序列号(Serial Number).....	3
2.4 条形码.....	4
第三章 工艺路线.....	5
3.1 工位(Step).....	5
3.2 工艺路线(Route).....	5
3.3 当前工艺状态(Route Status).....	6
3.4 工艺检查(Route Check).....	7
3.5 工艺记录(Route Update).....	8
3.6 工艺历史(Route History).....	9
3.7 工艺初始化(Route Initial).....	10
3.8 工艺结束(Route Close)与工单关闭.....	10
3.9 故障与维修.....	11
3.10 自动测试.....	11
3.11 产品、工单与工艺路线.....	11
第四章 装配.....	13
4.1 装配(assembly)与子装配(sub-assembly).....	13
4.2 序列号关联表的设计.....	15
4.3 新增序列号关联.....	16
4.4 序列号关联替换.....	16
4.5 序列号关联注销.....	17
4.6 序列号关联查询.....	18
第五章 包装.....	19
5.1 包装的概念.....	19

第六章 材料跟踪.....	22
6.1 为什么需要跟踪材料.....	22
6.2 定义材料.....	22
6.3 定义物料.....	22
6.4 关联.....	23
第七章 架构.....	25
7.1 模块.....	25
7.2 角色.....	26
7.3 数据库.....	26
7.4 服务器.....	27
7.5 客户端.....	27
第八章 外部数据管理.....	28
8.1 外部数据.....	28
8.2 数据分类.....	28
8.3 数据传输.....	29
8.4 数据处理.....	29

第一章 什么是制造执行系统

1.1 MES 的功能

在传统的加工生产企业中，往往用流程卡来进行生产的跟踪控制。在流程卡上印刷工序列表，每经过一个工位，则由作业人员打勾、签名、标注日期。这样产品在生产线上流转时，我们可以检查产品是否经过了所有的必须工序。

MES 全称 **Manufacturing Execution System** 制造执行系统，它的核心功能与流程卡类似，即通过流水号控制生产工序。**MES** 通过创建产品的序列号来取代流程卡号，生产过程中的任何一个工序通过序列号访问数据库，得到所有相关的控制信息。

流程卡在生产完成之后即作废，而序列号却一直保存在企业的数据库中，所以可以作为产品今后的跟踪依据，方便质量管理。

简单地说，**MES** 有两大核心功能：根据序列号进行工序的控制，根据序列号进行历史数据的跟踪。其它功能还有装配、包装、物料跟踪，我们在具体的模块中详细介绍。

1.2 MES 与 ERP 的关系

ERP 即 **Enterprise Resource Planning** 企业资源计划，是生产企业的核心模块，其主要功能是物料管理和成本控制。

所有的生产都是由 **ERP**(通过工单)发起、由 **ERP**(通过发货单)结束的。一些 **ERP** 系统也包含了简单的生产模块。我们可以把 **MES** 看成 **ERP** 生产模块的专业化。因此 **MES** 与 **ERP** 的接口应保持逻辑上的一致性。

第二章 产品、工单、序列号

2.1 产品(Product)

MES 的流程通常是由 ERP 系统发起的，因此对于产品的定义必须与 ERP 保持一致。通常，用 item_no(产品编码)或者 item_no 和 item_revision(产品版本)的组合来定义一个产品(product)。

从 ERP 的角度而言，产品等同于物料编码。一个完整的产品周期可能包含以下环节，这些环节都与产品相关，如图 2-1 所示：

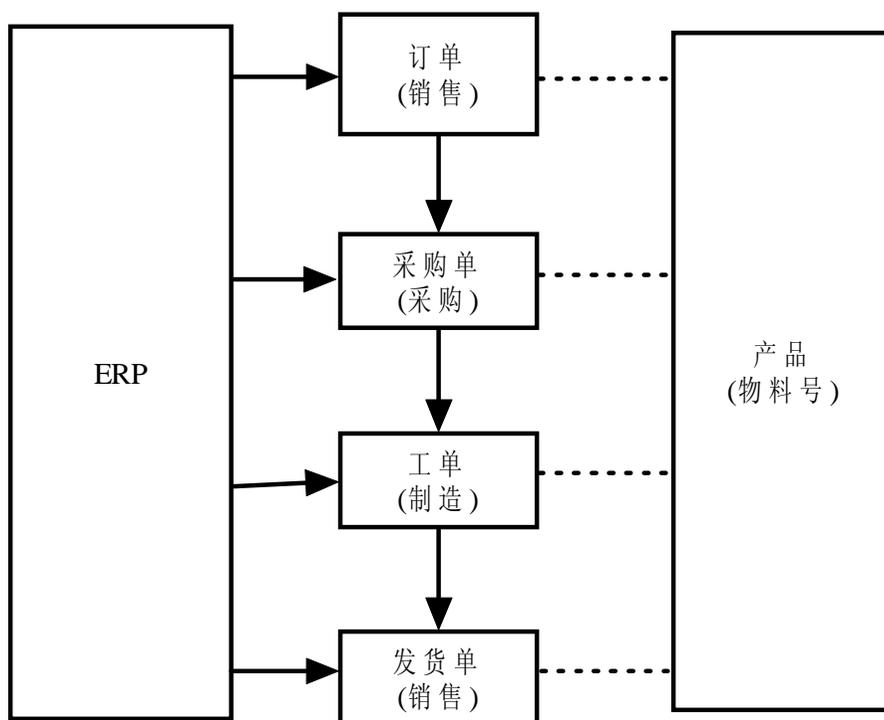


图 2-1

对于 MES 而言，产品除了物料属性以外，还有一些其它与生产密切相关的属性，如工艺流程、装配结构、测试规范等。我们在后面的章节中将陆续展开。

2.2 工单(Work Order)

工单又称任务单、计划单。

一个工单定义了一次加工作业，包含以下信息：待加工产品、数量、加工起点、原材料输出仓库、成品输入仓库，以及工单状态、计划完成时间等辅助信息。

一个标准的 MES 系统是由工单驱动的。首先由计划员在 ERP 中定义工单，然后将此工单刷新到 MES，车间的主管将工单落实到具体的产品工艺路线，操作人员按照工艺路线进行作业。如图 2-2：

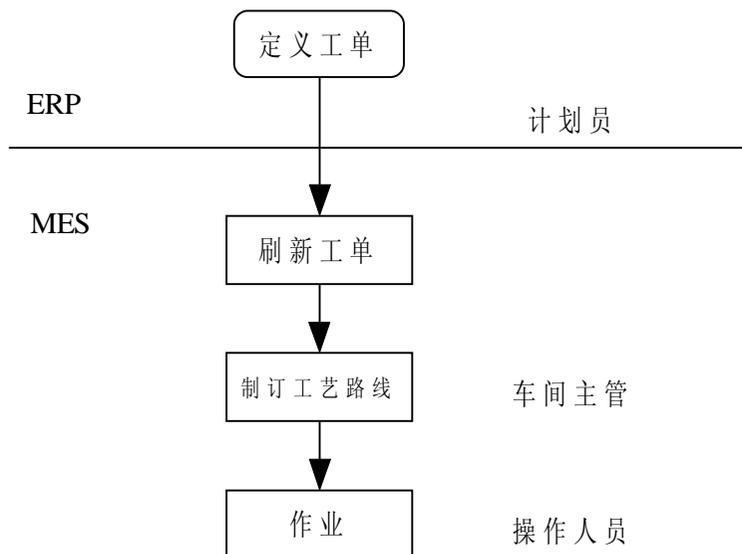


图 2-2

2.3 序列号(Serial Number)

在传统的生产过程中，我们往往用流程卡来指导产品的生产。在 MES 系统中，我们用序列号来代表一个具体的产品。

序列号代表系统里唯一的一个产品，它具有 item_no 属性(ERP 物料号)，此外，它必须与工单关联，这样才能够用工单进行驱动。

对于制造件和装配件，它们的序列号规则是不一样的。对于制造件，其序列号可以全部由 MES 系统分配产生，而采购件的序列号是供应商产生的，只能作为外来数据导入 MES，同时赋予其 item_no 属性。

在一些情况下，序列号是企业的重要资源。如对于网卡 MAC 地址，其前 6 位是企业代号，后 6 位是企业内部的 MAC 流水号，MAC 地址的申请需要向相关组织缴纳费用。另如中国 GSM 手机的 IMEI 号码，是由政府信

息产业部分分配的。

在某些情况下，序列号并不代表 ERP 中明确定义的产品。如有一款手机产品 G10，其 ERP 物料号为 10010，MES 系统会为其分配一序列号 20061100001(item_no 属性为 10010)，然而在 ERP 的 BOM 中并没有定义 IMEI，为了保持序列号格式的一致，我们为 IMEI 定义一个“虚拟”(ERP 中没有此物料号)的 item_no: IMEI。这样此手机在生产完成之后，至少有 2 个序列号，1 个是 10010 对应的序列号，1 个是 IMEI 对应的序列号。

车间的操作人员在作业时，将序列号与工单建立关联关系，这样工艺路线就落实到每一个具体的序列号了。

产品、工单、序列号三者之间的关系如图 2-3 所示，序列号继承自产品，工单通过与序列号关联的方式驱动产品。

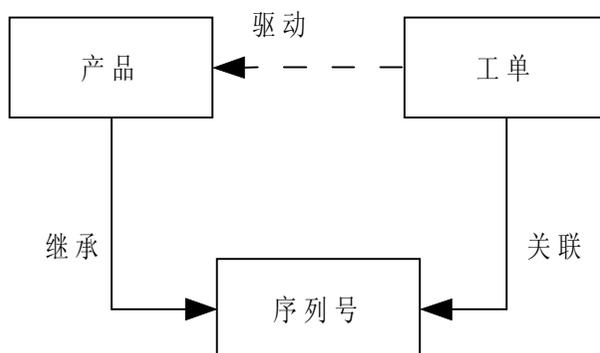


图 2-3

2.4 条形码

为了提高序列号输入的效率，MES 系统采用条形码(以下简称条码)来记录序列号。

条码是用条码打印设备，把序列号打印在尺寸裁切好的不干胶贴上，然后用激光扫描设备识别。

常用的条码打印设备有 zebra 公司的系列条码打印机。常用的激光扫描识别设备有 symbol 公司的手持激光扫描枪系列。

常用的条码编码规范有 39 码和 128 码，在一些包装工厂，还用二维码来记录大量数据信息。

第三章 工艺路线

3.1 工位(Step)

在现代化生产中, 批量生产多采用流水作业, 流水线由若干个独立作业的工位组成。

工位是产品生产周期中的一个点, 或者说一次作业。

工位是一个逻辑概念, 一个工位可以对应若干个实际的工作台, 比如电脑的主板装配工位, 可以由 4 个工人、在 4 个工作台上做相同的装配作业, 那么我们说这 4 个工作台对应于同一个工位。

在 MES 系统中, 工位的定义可以和实际的作业有一定的出入。还是举主板装配工位为例, 在实际生产中可能作进一步的细化, 分成定位、焊接、固定 3 个工位, 但在 MES 系统中只定义一个工艺路线控制点, 那么我们可以看作是一个工位。

3.2 工艺路线(Route)

工艺路线是指产品的生产方法。通常我们用流程图来表示工艺路线, 流程图由当前工位、结果、下一工位组合而成, 如图 3-1。

工艺路线表达了产品完整的生产周期。

一个产品可以定义若干个工艺路线, 如试生产工艺路线、量产工艺路线、返工工艺路线等。但是一个工单只能驱动一条工艺路线, 因此工单从 ERP 刷新到 MES 后, 在生产之前, 必须由车间主管或生产线组长指定工单到某一工艺路线。

简单地说, 工艺路线是产品的一种属性, 由工位组成, 受工单驱动。

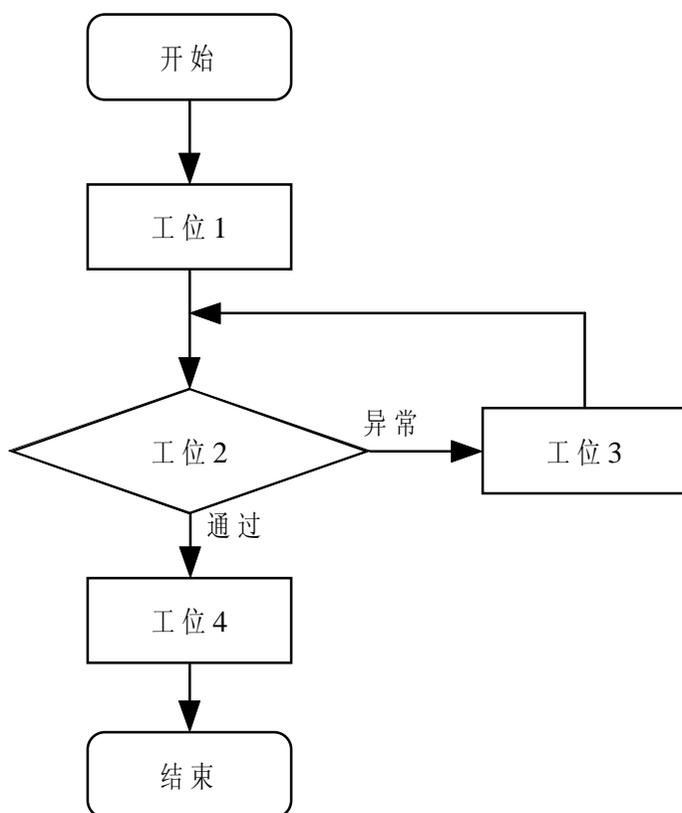


图 3-1

3.3 当前工艺状态(Route Status)

生产控制的目的，是为了实现“在指定的工位，做指定的产品”。

我们在工艺路线里已经定义了当前工位、结果、下一工位的组合关系，因此我们只要再记录某个序列号对应的产品当前的工位和结果，就可以得到下一工位，由此可以进行工艺检查和工艺记录。

对当前工位的定义方法有两种。

第一种定义刚刚结束的工位是当前工位。这样在工艺状态表里记录了序列号、当前工位名、当前工位结果。

第二种定义下一工位是当前工位。这样在工艺状态表里记录了序列号、当前工位名。

3.4 工艺检查(Route Check)

工艺检查是判断序列号是否出现在正确的工位。方法如下：

按第一种定义，前台录入工单、序列号、作业工位。后台根据工单得到工艺路线；根据序列号得到当前工位和结果；然后根据工艺路线和当前工位、结果，得到下一工位；比较下一工位和作业工位，得到工艺检查的结果。如图 3-2 所示：

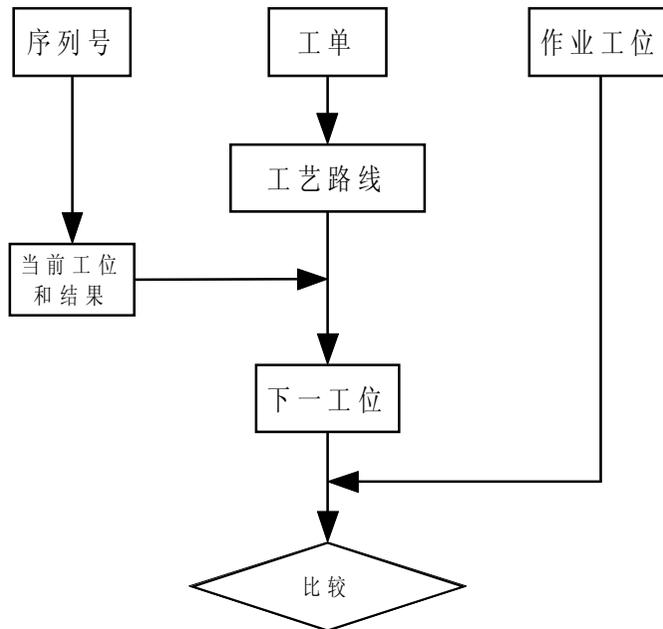


图 3-2

按第二种定义，前台录入序列号、作业工位。后台根据序列号得到当前工位；比较当前工位和作业工位，得到工艺检查的结果。如图 3-3 所示：

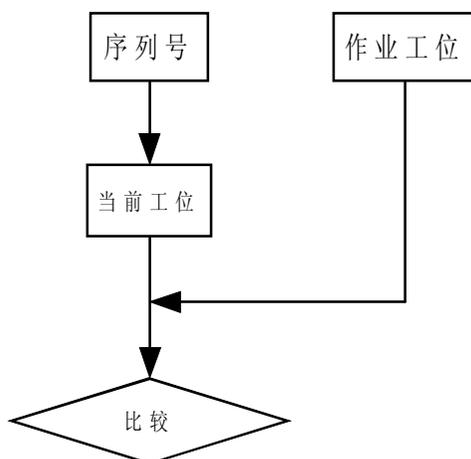


图 3-3

3.5 工艺记录(Route Update)

工艺记录是指工位作业完成后，对工艺的状态进行更新。

按第一种定义，前台录入序列号、作业工位、结果，后台直接将此数据更新到工艺状态表。如图 3-4 所示：

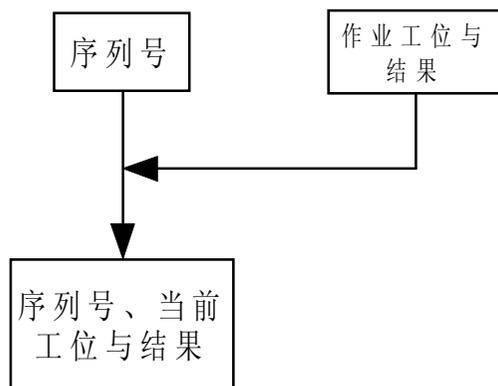


图 3-4

按第二种定义，前台录入工单、序列号、作业工位、结果；后台根据工单得到工艺路线；根据工艺路线、作业工位、结果，得到下一工位；根据序列号、下一工位更新工艺状态。如图 3-5 所示：

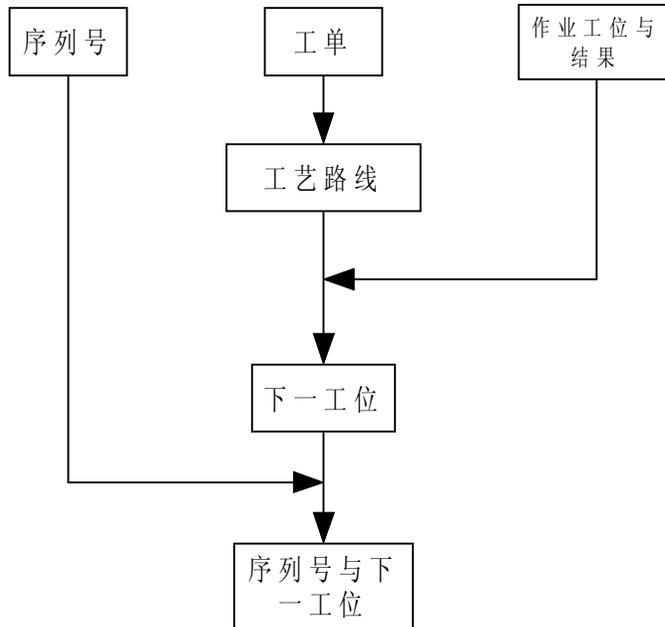


图 3-5

我们可以看出，两种方法的区别是：

第一种方法在 route check 时引用工艺路线，第二种方法在 route update 时引用工艺路线。

第一种方法在 route check 时较为繁琐，但是适合以下这种特殊情况，即当前工位和结果的组合，对应的下一工位有多条记录，如图 3-6：

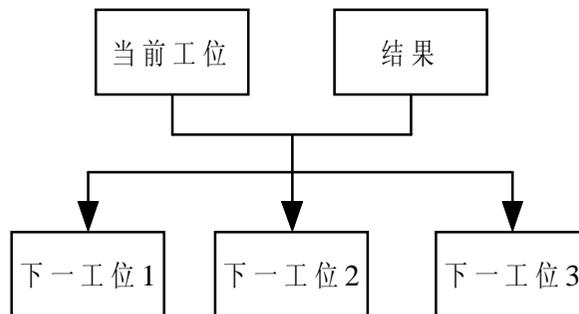


图 3-6

3.6 工艺历史(Route History)

在进行 Route Update 的时候，同时产生一条 log 记录，包括作业时间、

作业人员、作业工单、序列号、当前工位、结果。当生产结束的时候，一个序列号有多条 `route log` 记录，这些记录就构成了一个序列号的工艺历史 `Route History`，也就是一个完整的生产周期的记录。

3.7 工艺初始化(Route Initial)

工艺的起始点比较特殊，因为有了它之后才能进行工艺检查。为了设计的方便，我们在所有的工艺路线中，均设计其起始点为工艺初始化点，此点之后的工位就可以使用 `route check` 和 `route update` 方法了。

初始化的方法有两种。

第一种是在序列号创建之时就进行初始化。

第二种是在 `route check` 时检查是否有 `route status` 记录，如果没有，则先进行初始化。

初始化是一种特殊的 `route update`，其工位名是 `initial`。

3.8 工艺结束(Route Close)与工单关闭

工艺结束点也是较为特殊的一个工位，它是所有工艺路线的最后一个工位。

按照 `route update` 的第二种方法，则系统通过 `route update` 自动产生 `close` 工位。关闭工单时，系统得到此工单驱动的所有序列号；然后 `route check` 其当前工位是否都是 `close`；如果全部 `close` 则允许关闭工单，否则需强行关闭工单。

按照 `route update` 的第一种方法，则 `route history` 里不包括 `close` 工位，所以必须手动关闭。关闭工单时，系统得到此工单驱动的所有序列号；然后 `route check` 其当前工位是否都是 `close`；如果全部 `close` 则允许关闭工单，否则需强行关闭工单。可以在关闭工单时，将其驱动的序列号 `route update` 到 `close` 工位。

3.9 故障与维修

故障工位与维修工位是较为特殊的 route 工位，因为它们在 route update 的同时，还要记录相应的故障与维修记录。

故障与维修信息可以看成 route update 的扩展信息，每一次的故障与维修记录，都通过外键关联到一条 route log 历史记录，因此 route 与故障、维修的动作就接合在一起，可以通过序列号找到它们之间的关联。

3.10 自动测试

在现代企业的生产中，为保证产品的质量，往往在生产周期中设置若干个测试点，在大批量的生产中，这些点往往采用自动测试完成。具体的操作是：首先根据产品与工位定义 test profile 测试模板，然后输入实测数据，生成测试数据文件，并关联到序列号。由于测试文件与测试模板是关联的，因而能够根据 test profile 进行分类统计。

自动测试是一个特殊的 route 工位，操作前需要 route check，完成后进行 route update 并生成 route log。

测试数据与 route log 进行关联，因而可以根据 route history 直接定位测试数据。

3.11 产品、工单与工艺路线

产品、工单与工艺路线的关系，见图 3-7，装配结构、包装结构、材料跟踪的定义参见后面的章节。

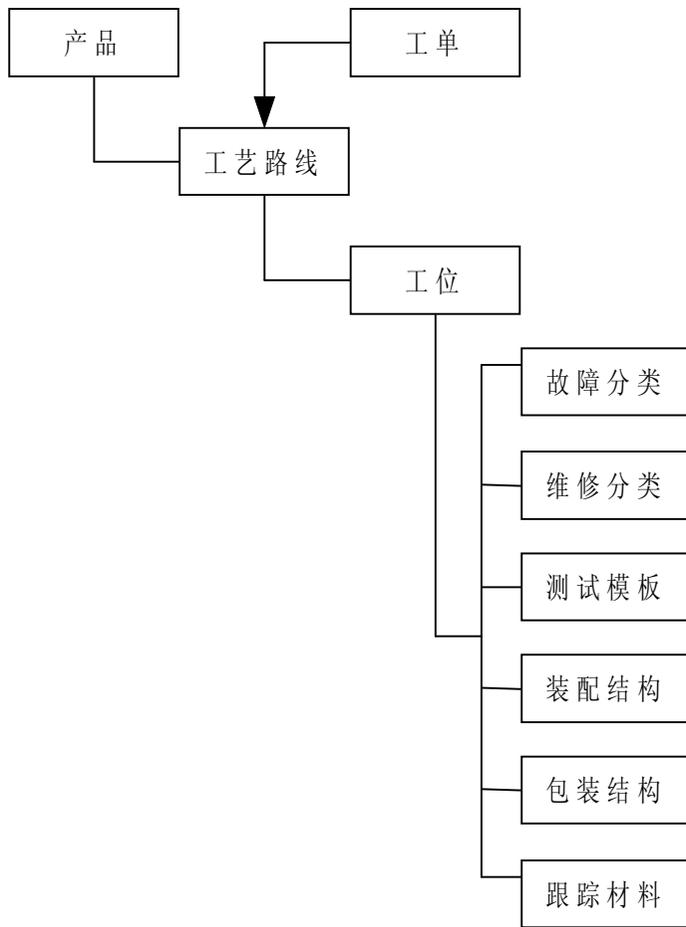


图 3-7

第四章 装配

4.1 装配(assembly)与子装配(sub-assembly)

ERP 中的产品不仅是物料，它同时还是一个装配件，通过 BOM 表达它的装配结构。一个装配件由若干个子装配件组成。

比如一台 PC，包含的子装配件有主板、CPU、显卡、硬盘、显示器等，其 BOM 的结构如图 4-1：



图 4-1

这个结构表示一台 PC 由 1 件主板、1 件 CPU、1 件显卡、2 件硬盘、1 件显示器、1 件机箱组成。这种结构可以看成 PC 这种产品的一个属性。在 MES 系统中，这种结构通过产品工艺路线的若干工位体现，如图 4-2：



图 4-2

图 4-2 表示 PC 的装配工艺共分散在 4 个工位中。

为了记录产品与其子装配之间的关系，我们需要将产品的序列号与子装配的序列号进行关联。具体的操作如表 4-3：

装配工位	装配序列号
主板装配工位	PC 序列号 机箱序列号 主板序列号 CPU 序列号
显卡装配工位	PC 序列号 显卡序列号
硬盘装配工位	PC 序列号 硬盘 1 序列号 硬盘 2 序列号
显示器装配工位	PC 序列号 显示器序列号

表 4-3

关联完成之后，MES 数据库里就保留了产品与其装配件之间序列号的关联关系，如图 4-4：

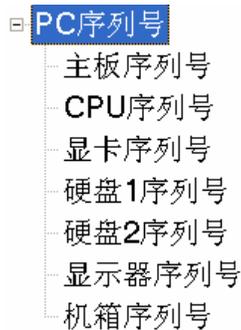


表 4-4

从技术的角度而言，有两种方法记录序列号的关联关系。

第一种方法是在数据库里保存产品的结构，然后各相关序列号指向这种结构。

第二种方法是直接记录各序列号之间的关联关系，形成一棵“树”。

比较而言，第一种方法方便形成批量数据关联关系的报表，而第二种方法隔离了序列号数据与产品结构，所以产品的结构变了也不会影响历史数

据。

我们更倾向于采用第二种方法，因为这对于后面要提到的序列号关联替换、注销等操作更为方便。

4.2 序列号关联表的设计

考虑到关联的结构，用父-子节点来定义树状结构是非常方便的，效率也比较高。如图 4-5 所示的结构：

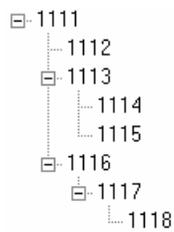


图 4-5

用表 4-6 表达：

ID	SN	Father_ID
1	1111	-2
3	1112	1
4	1113	1
5	1114	4
6	1115	4
7	1116	1
8	1117	7
9	1118	8

图 4-6

ID 字段是序列号所在的节点，SN 是序列号，father_id 是序列号的父节点，如为负数则表示此父节点并没有对应的序列号存在，也就是说此 SN 是该节点数上的根节点，为方便，我们称负数的节点为虚节点。

4.3 新增序列号关联

按照节点的设计,任何一次关联动作可以看成把某一节点设置为另一节点的父节点,为方便叙述,我们统一称为把 SN1 的节点设置为 SN2 的父节点,或者把 SN2 关联到 SN1 上。

新增序列号关联的方法为:更新序列号关联表,设置 SN2 的父节点等于 SN1 的节点。

为避免逻辑上的混乱,我们在关联之前必须进行检查,规则为:

- 1) SN2 的父节点必须为虚节点,也就是说 SN2 是根节点,这样做是为了保证任何一个 SN 只能有一个父节点。
- 2) SN2 的父节点不能是 SN1 所在树上的根节点。

例如,将 2221 关联到 1116 上,如图 4-7

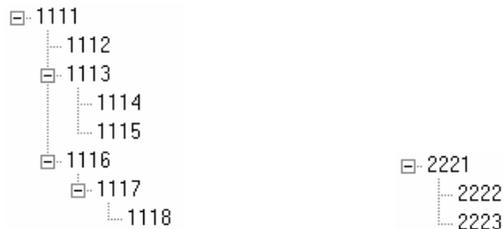


图 4-7

则 2221 必须没有父节点,这样不会造成大于 1 个父节点。而且 2221 必须不等于 1116 树上的根节点(1111)。

4.4 序列号关联替换

输入旧序列号 SN_old、新序列号 SN_new,以进行替换。

替换的前提条件:

1) SN_new 的父节点必须是虚节点,也就是说,SN_new 必须是根节点。

2) SN_new 不能等于 SN_old 所在树上的根节点。

替换的方法为:

1) 设置 sn_new 的父节点等于 sn_old 的父节点。

2) 设置 sn_old 的父节点为一个新的虚节点,这样 sn_old 仍可以看成以 sn_old 为根的一棵树。

如图 4-8 的已有关联:

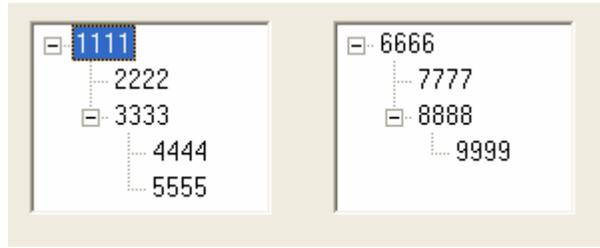


图 4-8

以 6666 替换 3333 的结果如图 4-9:

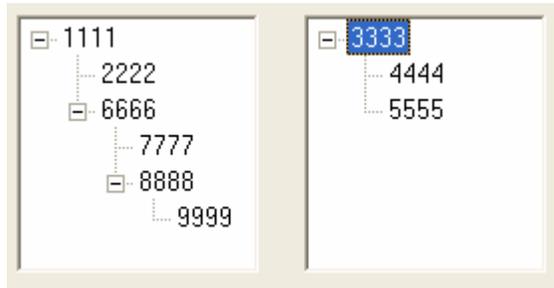


图 4-9

4.5 序列号关联注销

对于在序列号关联表中已有记录的序列号，它不仅有自己的节点，而且也可能是其它序列号的父节点，因此在注销时可能对其它序列号产生影响。

序列号关联注销的动作为：

- 1) 定义该序列号的父节点为一新的虚节点；
- 2) 更新所有该序列号的子节点，分别定义其父节点为新的虚节点。

例如：注销 1113，如图 4-10 所示：



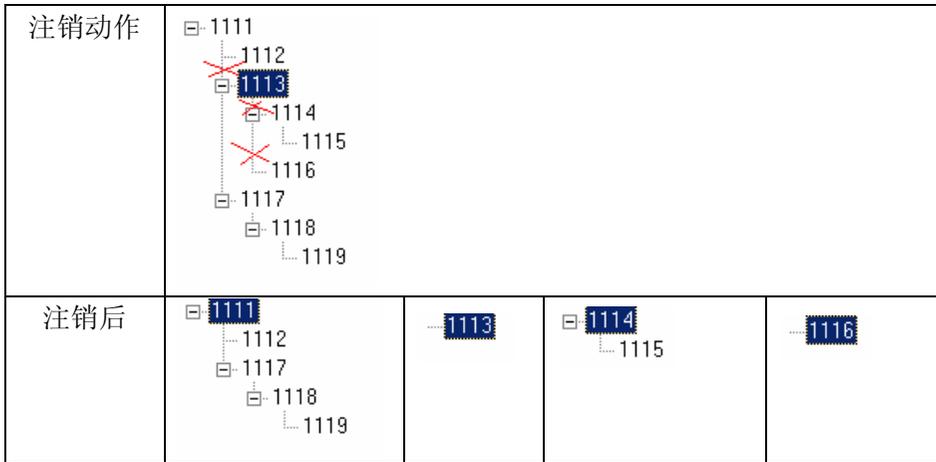


图 4-10

4.6 序列号关联查询

输入单个序列号进行关联查询的方法为：

- 1) 追溯至此序列号的根节点。
- 2) 以根节点做递归查询，列出所有树状结构的子节点。
- 3) 在每个节点上，通过关联表得到序列号的相关信息。

这种方法很容易还原生成序列号关联关系的树，而且也不存在效率的问题。

但是在实际的生产中，经常需要做大批量的查询，并且生成格式统一的报表，比如根据工单输出所有的关联序列号。象这种情况，如果还要逐个序列号进行迭代查询的话那么效率就会有很大的影响。解决方法是，在查询之前，系统定期对关联数据进行后台处理，把树表展成平表，这样查询时就无需迭代，一条 SQL 就可以跑出所有的数据。具体的实现方面，需要数据库设计方面的技巧，也可以参考数据仓库的原理。

第五章 包装

5.1 包装的概念

在 MES 系统中，我们定义包装的概念为，将一个产品放在某一个不能代表此产品的容器中的动作。

举个例子，将手机放入礼品盒的动作不是包装，因为礼品盒序列号与手机的序列号是一一对应的，可以用礼品盒的序列号来描述此手机，因此两者之间是序列号关联关系。而将多件手机礼品盒打包到托运箱的动作是包装，因为一个包装箱里包含了若干只手机，因此不能用包装盒来代表某一件具体的手机产品。

我们可以将包装理解为在产品生产完成之后，为发货而准备的一个动作。

图 5-1 是一个典型的包装例子：



图 5-1

5.2 包装的实现

就技术的角度而言，包装和关联实现的方法基本一致，都是用数据库储存树状的结构。

但是由于包装箱和产品的属性还是有较大的差异，因此我们有必要分别处理。具体的来说，首先我们定义包装箱的属性，如尺寸、重量、物料号等，然后建立产品序列号与包装箱序号的引用关系。

图 5-1 的包装箱属性，如表 5-2 所示：

箱号	属性
大箱 1	包装箱属性 1
小箱 1	包装箱属性 2
小箱 2	包装箱属性 3

表 5-2

引用关系如表 5-3 所示:

序列号	属性	箱号
小箱 1	包装箱	大箱 1
产品 1	产品	小箱 1
产品 2	产品	小箱 1
小箱 2	包装箱	大箱 1
产品 3	产品	小箱 2
产品 4	产品	小箱 2
产品 5	产品	小箱 2

表 5-3

包装的基本动作是装箱与拆箱,对应于数据库层面的新增引用关系和删除引用关系。

5.3 包装与发货

发货意味着产品生产周期的结束,在许多企业里,往往有一个独立于 ERP 与 MES 的发货管理系统。在执行发货的动作时,作业人员录入箱号,然后从 MES 系统里得到此箱所有产品的序列号,以及按照物料汇总的统计信息;然后将序列号信息录入到发货管理系统,将物料汇总信息录入到 ERP。

为了发货的方便,通常将一些发货的信息反映到包装箱上,如张贴 A4 报表,或者贴二维条码,以及反映相应的发货单号等。由于许多信息是从 MES 系统中取的,为了提高效率,往往将这些动作提前放在包装时进行。也就是说,在包装时就预先记录了部分发货信息。

MES 与 ERP、发货系统的关系,如图 5-4 所示:

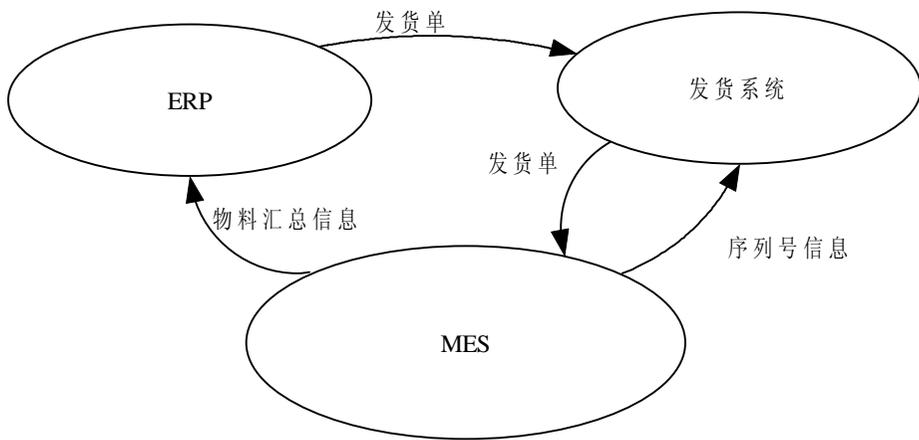


图 5-5

第六章 材料跟踪

6.1 为什么需要跟踪材料

在许多情况下，我们需要对生产过程中用到的材料进行跟踪，特别是针对一些关键的采购件。比如 PC 生产商发现一批 CPU 有严重质量缺陷，那么必须知道这批 CPU 已经被应用到哪些产品中，也就是说必须根据 CPU 的批次信息，找到与之关联的产品序列号。

MES 中的材料跟踪模块，就是为实现这个功能而设计的。

6.2 定义材料

这里所说的材料，是指产品 BOM 中要跟踪的零部件列表，通常分散在若干个工位中。

要跟踪的零部件列表是产品的一种属性，定义方式如图 6-1：

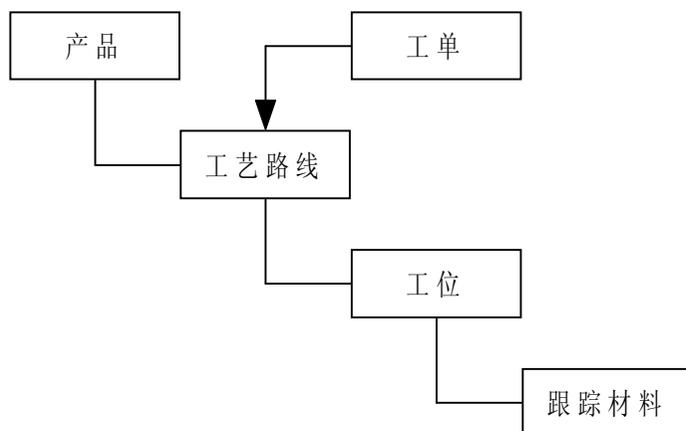


图 6-1

6.3 定义物料

材料跟踪与装配的不同之处在于，子装配件都有独立的序列号，而待跟踪材料没有独立的序列号，只有物料和批次信息。

比如对于常用的电阻电容，由于同一批次内的产品差异性不大，我们不

会追踪到每一个独立的物料。通常对物料的定义建立在最小包装单位上的，如对于盘状包装的贴片元器件，其最小包装单位是盘，也就是说每一个单独的包装盘，都有一个唯一的最小包装物料编号，在此包装单位的所有物料具有完全的可替代性。

最小包装序列号包含以下属性：ERP 物料号、单位、数量、供应商信息、进料日期、失效日期。

6.4 关联

物料的关联通过一个特殊的 route 工位进行，具体的操作是关联产品序列号与最小包装物料号。

关联的实现方法与装配的方法较为类似，区别在于，装配的时候，所有的序列号之间是一一对应的，而物料关联时，一个最小包装物料号允许关联到多个产品中。如图 6-2：

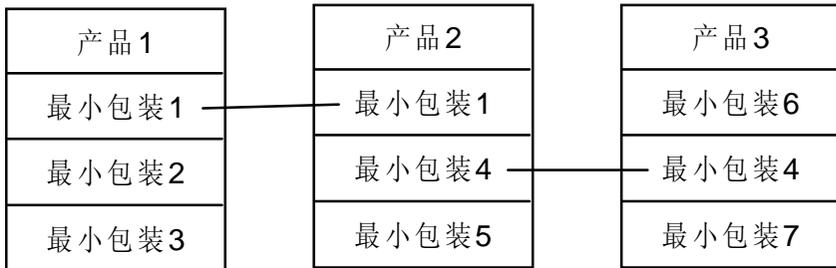


图 6-2

由图可以看出，最小包装 1 被应用到产品 1、产品 2 中，最小包装 4 被应用到产品 2、产品 3 中。

假设产品 1 应用到的物料 1 数量等于 2，那么这个物料有可能是在同一个最小包装里的，也有可能是在不同的包装里的，那么怎么区分定义呢？

在贴片加工的工艺中，除了要定义物料的编号外，还要定义物料的位置，相同物料在同一位置出现的数量只能是 1，因此在跟踪时只要记录产品关联的最小包装物料号与位置即可。

而在非贴片加工的工艺中，往往对位置的信息并不敏感，因此在关联时只要记录产品序列号，以及与之关联的最小包装物料号和数量即可。

6.5 跟踪

通常有两种跟踪方式。

第一种是根据序列号跟踪物料。根据工单或其它产品信息，得到产品序列号的集合，然后根据序列号集合得到物料的批次信息。

第二种是根据物料跟踪序列号。

跟踪的依据都是最小包装物料号和序列号的关联关系。

第七章 架构

7.1 模块

MES 主要由资源管理、流程控制、报表查询 3 个大的模块组成，如图 7-1 所示：

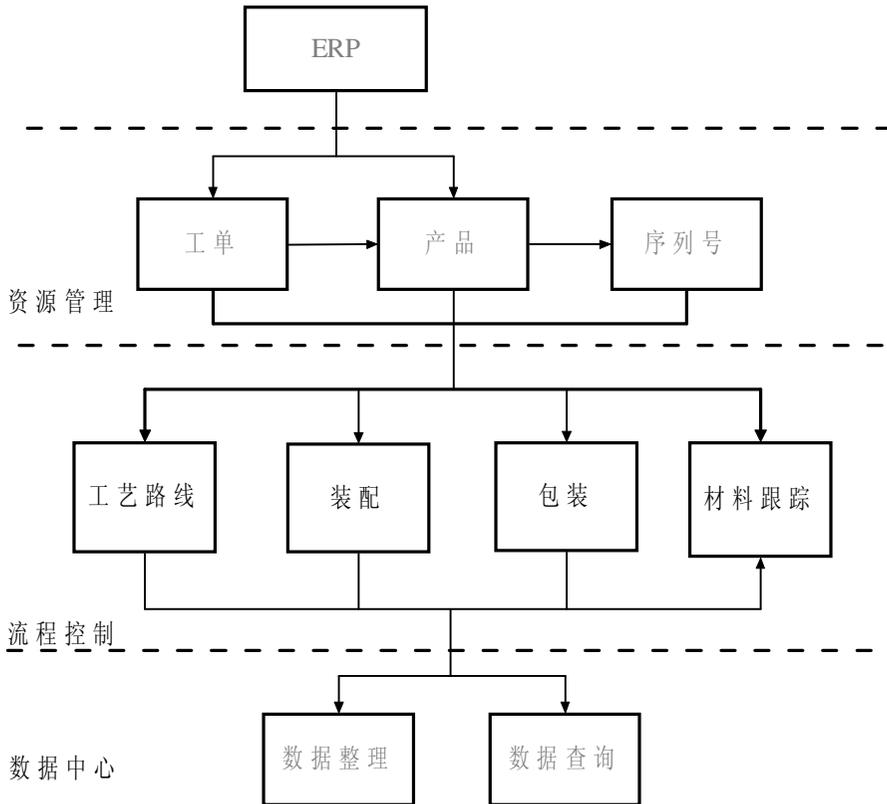


图 7-1

MES 中的资源包括工单、产品、序列号。工单从 ERP 导入，产品的基本信息也在 ERP 中定义。序列号根据工单和产品信息，在 MES 系统本地产生。

流程控制模块包括工艺路线控制、装配控制、包装控制、材料跟踪控制。这些控制模块都根据产品定义，然后基于序列号进行控制。尽管工单、产品、序列号几乎贯穿于所有流程控制模块，但是我们会将它们的扩展属性与基本属性分离，并拆分到各相应的控制模块中，这样就保证了各控制模块的逻辑分闭。

数据中心模块包括数据整理和数据查询。此模块将资源管理和流程控制中的数据，定期在后台按照一定的规则进行整理归类，然后提供统一的接口进行分析查询。由于 MES 中的数量非常庞大，各类数据之间的引用关系又较为复杂，数据不整理的话，进行统计查询的效率会非常低。

7.2 角色

表 7-2 列出了 MES 系统中的基本角色及其功能。

角色	功能
计划员	定义工单。
制造工程师	定义产品工艺路线。
	定义产品装配结构。
	定义产品包装结构。
	定义产品物料跟踪结构。
	定义产品测试模板
生产主管	激活工单，将工单指定到具体的工艺路线。
生产作业人员	生成序列号。
	工艺检查与更新。
	装配。
	包装。
	物料信息录入。
	测试。
报表查询人员	各类报表。

表 7-2

7.3 数据库

由于 MES 系统的数据量庞大，连接数和并发量都比一般的系统大得多，所以必须采用支持这种特性的数据库，如 oracle 和 sql server。

7.4 服务器

从逻辑上来说，MES 系统应包含以下 3 种服务台：**http** 服务器、**ftp** 服务器、**email** 服务器。

http 服务器提供以下服务：**webservice**、权限管理、报表查询、其它后台数据处理。

ftp 服务器提供以下服务：文件存储、版本更新。

email 服务器提供以下服务：与外界接口、系统消息发送。

7.5 客户端

系统采用三层架构，客户端通过 **webservice** 进行事务处理。

客户端根据角色的不同，提供不同的功能。

对于制造工程师，客户端提供的功能是定义产品的工艺路线。工艺路线由工位组成，而装配结构、包装结构、物料跟踪结构、测试模板均定义在工位上，因此客户端的核心功能是对工位的定义。

对于生产主管而言，客户端提供的功能是激活工单，即将工单关联到具体的序列号。

对于生产作业人员而言，客户端提供的功能是产生序列号和实现各工位的具体操作，因此其核心功能是解析系统对工位的定义，并展现相应的结构，实现数据录入和流程控制。

第八章 外部数据管理

8.1 外部数据

目前生产外包的现象已经非常普遍。为了方便生产数据的综合分析，必须将外部数据集中到企业的中央数据库。图 8-1 是一个典型的外部数据管理流程。

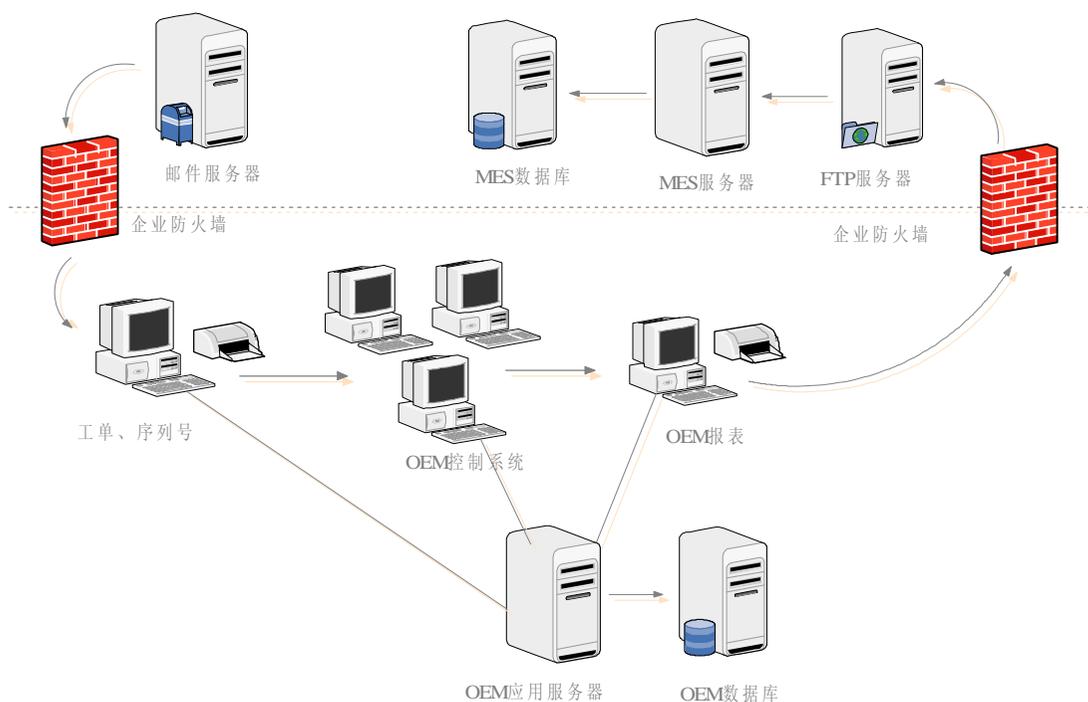


图 8-1

由图 8-1 可以看出，OEM 控制系统是一个逻辑封闭的系统，与 MES 的中央数据库通过接口进行数据交换。

OEM 的数据经过整理之后进入数据中心，就可以象本地的数据一样，采用统一的报表进行查询分析。

8.2 数据分类

为了方便对 OEM 数据的统一管理，应该分析数据的特性，按照数据类

型进行分类，经过分类的数据类型适用于所有的 OEM 厂商。OEM 最基本的数据类型有：序列号基础信息、装配信息、包装信息、测试信息等。对于已定义的数据类型，所有的 OEM 厂商都以相同的格式存储数据。

8.3 数据传输

最方便的数据传输方式是通过 FTP 传输。

OEM 的数据首先在本地汇集到一台 PC 上，然后通过 FTP 上传到企业的 FTP 服务器上。

可以在 FTP 上，根据数据类型的不同，创建若干个子目录，以方便数据的后续处理。

由于数据类型和 OEM 厂商无关，所以所有的 OEM 厂商都可以用同一个数据传输程序进行数据上传，有助于数据安全和控制。

8.4 数据处理

MES 的程序从 FTP 服务器上取数据，经过整理之后导入数据中心。

由于数据已经按照特性进行了分类，所以可以很方便地编写导入程序，定期在后台运行。

数据一旦进入 MES 数据库，就和本地产生的数据没有本质的差别，可以应用相同的规则进行数据整理，因此不需要开发额外的报表。