

文波 | 周汇报 | 2016年9月4日

关于张晨晖师兄的工作的学习

周汇报

尊敬的杨老师:

您好，经过本周的学习，我初步完成了晨晖师兄工作环境的搭建，现在将本周的学习情况汇报给杨老师：

# 论文的学习：

关键词：复杂事件处理CEP、事件驱动架构EDA

文章中通过引入复杂事件处理和事件驱动软件框架，通过面向事件的数据处理方式改善了MES的设计方式，提高了MES系统的的自适应、自处理能力，实现了MES的“物理感知—信息融合—决策响应”。[好像在晨晖师兄的这一块工作并没有接入MES系统？？？小方博士说基础事件的维护和复杂事件的处理没有配入当前的MES系统，所以这一块的资料还是只能从论文上获取了]

文中比较详细的介绍了研究的背景和意义，同时介绍了mes和cep的研究现状，在第二章中介绍了系统的架构设计（需求分析—总体结构设计—功能模块设计—业务流程设计—信息模型的构建—人机交互设计等几个模块展开了论述）；在第三章中介绍了复杂处理中的事件定义和分析；第四章则对其处理方法进行了探讨，最后一章则是系统验证。

# 软件的学习：

根据师兄的的论文，他做的主要包含下面几个模块，前面标灰色的，应该是没有被接入系统的，只有现场执行管理（标绿）被接入到了我们的MES系统：

简单事件的管理：

复杂事件模式定义：<模式模板定义、配置方式>

动作基础信息维护及主题订阅：

复杂事件处理模块：

现场执行管理：

1、前端--在Dorado7中<也可以在Android studio上做app以适配各种型号的Android设备，同时也可以>

在com.shoplink.entity下新建需要的java对象🡪在com.mes.shopLink下新建\*.touch.xml视图,在Model下建立DataType数据类型，并与刚建立的java对象对应<右击新建的datatype,选择通过java类型创建propertyDef>，在view下建立DataSet(dataProvider指的是提供数据的DataAccess类下的get或save方法，dataType即数据类型) 🡪在com.shopLink.dataAccess下建立DataAccess.java类，并写上相应的get或save方法

2、后台--在MES系统中<Myeclipse6.6+dorado5+jboss4.2.2>

在接口mmc\_svn.src.com.shoplink.service.iface.ShopLinkService.java中写入相应的get或save方法🡪在实现类ShopLinkServiceImpl.java中调用Biz.java中的方法🡪在shoplink.biz.\*Biz.java中用sql语句或hql语句实现save或get读取操作。

3、数据库服务<oracle11g>

数据：<使用那些表？如何得到这些表？怎样进行增删改操作？>

# 关于软件安装：

不得不说每次安装MES系统都会消耗大量的时间。由于暑期回来的时候，这台电脑中病毒了，D盘打不开，导致电脑被卡死，开机要花半个小时，开机也是各种卡顿，所以我就重装了个系统，因为现在都是比较流行Win10的系统的，所以我就在这台电脑上装了win10 ，win10的体验感还是很好的，这就为接下来的装MES系统不成功埋下了隐患。在各种方法用尽，在宝安师兄和小方同志的共同努力下，还是不能安装成功。后来在经历了重装系统，再重装win7才最终正常安装的，但是周勇师兄说在他的win10电脑上能配置成功，这就是很纠结的问题了。安装成功后，根据出错原因，我和小方博士思考出来的结果很有可能是win10系统内的springframework的一些包过比我们实验室用的更新，导致了冲突。

所以都建议实验室以后安装MES的系统都未win7旗舰版32位版本（虽然说win10系统真心比win7好用一些，而且界面看起来更加漂亮），这样子能省去很多不必要的麻烦。

自我总结：学习进度还是略显缓慢，目前有些方法还没有掌握，但是大致知道了整个数据流的过程，Dorado7 touch的相关知识还需要加强学习。

希望杨老师能给予批评指导，来鞭策我学习。

此致

敬礼!

                                                                                  文 波

 2016年9月4日

# 论文学习笔记：

MES简单事件划分成五大类，分别为任务事件类型、设备事件类型、物料事件类型、人员事件类型和周转箱事件类型。

TaskEvent =（eventUid，location，timestamp，taskUid，taskState，completeQty，lateStart，lateFinish，actualStart，actualFinish，planStart）

DeviceEvent =（eventUid，location，timestamp，deviceUid ，deviceState，planStart，maintainTaskState，lateStart，lateFinish，actualStart，actualFinish）

MaterialEvent =（eventUid，location，timestamp，materialUsn ，materialState，materialQty，materialType）

EmployeeEvent =（eventUid，location，timestamp，employeeId ，employeeState，deptUid）

BoxEvent =（eventUid，location，timestamp，boxId，boxState，boxType，boxQty）

一些接口：

SimpleEventManager：简单事件类型维护接口，主要用于管理简单事件类型，包括对预定义的简单事件类型进行修改、查询等，并根据实际情况扩展需要的简单事件类型，这里的扩展主要是针对子事件类型。

ComplexEventManager：复杂事件维护接口，主要用于管理复杂事件模式列表，包括对预定义复杂事件的修改、查询等，同时也支持复杂事件的扩展，根据业务需求定制相应复杂事件模式。该接口也提供了将预定义的复杂事件模式列表生成相应的规则文件，以支持系统的事件处理功能。

EventAttributeManager：事件属性维护接口，主要用于管理事件的属性列表，包括对预定义属性的修改、查询等，同时也支持属性的扩展。

ActionManager：动作维护接口，主要用于管理动作列表，包括对预定义动作的修改、查询等，同时也支持动作的扩展，以实现更多丰富的业务操作。

EventActionManager：事件动作维护接口，主要用于管理复杂事件模式与动作列表的关联关系，支持用户为预定义的复杂事件模式匹配动作、查询等，以实现相关业务触发动作。

ExceptionManager：异常管理接口，主要用于管理被实时记录的生产异常事件，包括面向用户的实时查询、面向用户的异常处理、生产异常的自动统计等功能，以支持一些需要人工决策的处理场景。

对于事件处理体系部分，主要包括以下核心业务接口以及实现类。

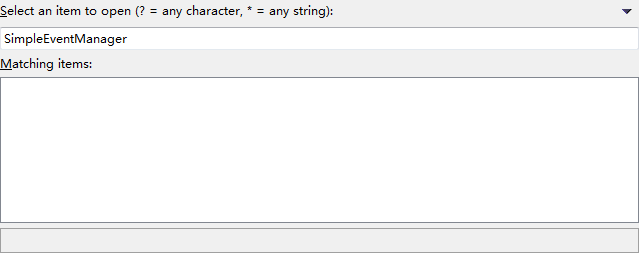
EventProcessManager：事件处理实现类，该类的实现主要依赖于Drools Fusion提供的接口及实现类。主要实现的功能包括读取已预处理的原始数据流、简单事件转化、利用CEP引擎执行复杂事件处理等。该类是事件处理流程实现的关键，当CEP引擎模式识别成功时，会发布相关实例事件，由EventPublisher完成。

JMSService：JMS工具类服务组件，该类的实现依赖JMS接口。主要用于封装会话连接、主题获取、关闭连接等通用功能，为事件发布组件和订阅组件服务。

EventPublisher：事件发布组件实现类，该类的实现依赖JMS接口。主要完成实例事件在相关主题上的发布工作。

BasicSubscriber：事件订阅组件实现类，该类的实现依赖JMS接口，其中最重要的是MessageListener接口，用于实现对相关主题上事件的监听任务。当有事件到来，即会触发动作。该类包括TaskSubscriber、DeviceSubscriber、BoxSubscriber、MaterialSubscriber和EmployeeSubscriber五个具体实现子类，用于分别处理不同主题上的实例事件。

ActionExecutor：动作执行组件实现类，主要封装了对预定义动作列表的方法实现，当订阅组件监听到事件时会自动调用相关方法执行预定义的业务操作



**表 1 MES简单事件列表**

|  |  |  |
| --- | --- | --- |
| 事件类型 | 子事件命名 | 子事件语义 |
| 任务类型事件  （TaskEvent） | TaskDispatch | 任务派工 |
| TaskReady | 任务就绪 |
| TaskStart | 任务开始 |
| TaskCompleteFirst | 完成首件 |
| TaskCancel | 取消任务 |
| TaskComplete | 完成任务 |
| TaskInterrupt | 任务中断 |
| 设备类型事件  （DeviceEvent） | DeviceRun | 设备运行中 |
| DeviceFree | 设备空闲 |
| DeviceFault | 设备故障 |
| DeviceMaintain | 设备维修中 |
| 物料类型事件  （MaterialEvent） | MaterialApply | 物料申请 |
| MaterialSign | 物料签收 |
| MaterialCheck | 物料检验 |
| 人员类型事件  （EmployeeEvent） | EmployeeReady | 人员就绪 |
| EmployeeAbsence | 人员未到位 |
| 周转箱类型事件  （BoxEvent） | BoxAddition | 周转箱进入 |
| BoxCurrent | 周转箱存在 |
| BoxStop | 周转箱停留 |
| BoxLeave | 周转箱离开 |

**表 4 复杂事件模式列表**

|  |  |  |
| --- | --- | --- |
| MES功能 | 复杂事件模式命名 | 复杂事件模式语义 |
| 任务作业管理 | cep\_TaskStartControl | 任务开始监控 |
| cep\_TaskTimeout | 任务已拖期 |
| cep\_TaskProcessControl | 任务进度监控（中断等） |
| cep\_NewTaskAdded | 设备新增任务 |
| cep\_TaskComplete | 完成加工任务 |
| cep\_TaskFinishCheck | 任务完成首检 |
| 物料管理 | cep\_MaterialSign | 物料签收控制 |
| cep\_MaterialCheck | 物料检验 |
| cep\_MaterialApply | 缺料需求申请 |
| 设备管理 | cep\_DeviceFault | 设备故障监控 |
| cep\_DeviceRecover | 设备恢复可用 |
| cep\_DeviceMaintainControl | 设备维修进度 |
| 周转箱管理 | cep\_BoxDelay | 周转箱滞留 |
| cep\_BoxPassInv | 周转箱出入库 |
| cep\_BoxCardError | 周转箱与组箱单一致性校验 |
| 人员管理 | cep\_EmployeePass | 人员进出监控 |

事件处理相关接口

1）EventManager：系统中与事件相关的维护和操作都有该接口完成，主要包括事件属性维护、事件监听、复杂事件产生等。

2）StateManager：该接口用来维护与业务对象相关的状态，包括增加、修改和删除等操作的实现。

量化评价相关接口

量化评价模块主要包括两部分，一是基于证据理论的事件对状态的置信评价，二是基于属性域权重的事件对状态相关属性的置信评价。

1）DSManager：该接口对主要对证据理论所涉及的相关方法进行了实现，主要包括动态证据体矩阵创建，Dempster组合规则实现，状态置信评价规则实现；

2）AttriWeightManager：该接口主要用于处理相关的条件事件的属性域权重，在创建复杂事件时调用，用来维护复杂事件对状态相关属性域的评价。

规则创建相关实现类

1）RulePatternManager：该接口用于维护复杂事件处理模式，主要包括两个部分，一是事件操作符的维护，二是与事件操作符相映射的复杂事件处理模式的维护

2）RuleActionManager：该接口用于维护规则匹配所需要执行的动作列表，动作列表主要包括复杂事件的生成动作以及对业务对象状态信息进行维护的动作；

3）RuleManager：该接口用于维护基于复杂事件处理模式和执行动作构建的规则实例，对规则实例的相关属性进行维护。提供对规则实例的添加、修改、删除和查询操作。

4）DroolsUtil：用于创建规则上下文的工具类，该类调用Drools CEP规则引擎的相关接口，管理规则上下文的生命周期，并提供规则编译和执行出现异常的捕获机制。

1）RfidService：为了实现RFID中间件系统与现有信息系统的集成而构建的接口，该接口利用Web Service技术对外发布，供外部RFID中间件系统调用，同时将RFID阅读器采集的标签信息封装成XML格式字符流并作为回掉函数的参数进行主动反馈。

2）ShoplinkService：为了实现为工位实现业务操作而创建的接口，并通过Web Service相关技术进行发布，移动终端反馈系统通过创建客户端接口调用ShoplinkService接口，从而实现了终端反馈系统与现有MES系统的集成，进一步EventManager通过对反馈数据的捕获获得了来自移动终端的数据流。