**会议记录表**

|  |  |  |  |
| --- | --- | --- | --- |
| 时间 | 2023年4月14日 | 星期 | 第九周 |
| 队伍 | 啊对对队 | 会议地点 | 线上会议 |
| 进行时长 | 2小时 |
| 参会人员 | 孟小凡、尹国泰、黄泰达、吴百川 | | |
| 会议主题 | Petri网模型的构建以及动态建模工具的学习、完善项目SRS | | |
| 会议内容 | 议题1：阅读 “SYSTEM MODELLING WITH PETRI NETS”，进一步理解Petri网知识  （尹国泰）  Petri网（Petri Net）是一种常用于描述并发系统的数学模型，由德国数学家Carl Adam Petri在20世纪60年代提出。Petri网由有向图和一组规则组成，可以表示系统中的状态、事件和转换。  Petri网通常包含两种节点：库所（Place）和变迁（Transition）。库所用圆圈表示，表示系统中的状态，例如机器的存储空间、系统的缓冲区等。变迁用长方形表示，表示系统中的事件或操作，例如进程的创建、消息的发送等。库所和变迁之间用有向边连接，表示事件和状态之间的关系。每条边上有一个权重，表示在发生事件时库所中的资源数量的变化。  Petri网可以用于建模各种并发系统，如软件系统、工业自动化系统、流程控制系统等。具体应用包括生产线的调度、进程的同步、通信协议的设计等。Petri网可以通过可视化工具进行建模和仿真，以验证系统的正确性和性能。  Petri网的建模方法包括静态建模和动态建模。静态建模用于描述系统的结构和静态属性，例如系统中的状态和事件；动态建模用于描述系统的行为和动态属性，例如系统的并发执行和状态转换。建模时需要考虑系统的实际需求和限制，选择合适的Petri网模型和规则，以保证模型的准确性和可用性。  （吴百川）  在一个标准的Pctri网模型中，其主要结构如下几个元素：位置（place，或称库所）、变迁（transition）、弧（arc）。位置主要用于描述系统元素所处的状态；变迁主要用于描述系统元素变化的事件；弧主要是通过其指向来确定元素局部状态和事件之间的变化关系。Pctri网模型中，标记主要应用于在库所中，其在库所中的动态变化表示系统元素的不同状态。  定义：三元组N=（P，T；F），称为有向网的充分必要条件是：  （1）P∩T=Ø（二元性）  （2）P∪T≠Ø（网非空）  （3）F⊆P×T∪T×P  （4）dom（T）∪cod（F）=P∪T  在Pctri网模型中，P用于描述N的库所集，T用于描述N的变迁集，F用于描述流关系（flow relation）.库所集和变迁集是Pctri网的基本成分，流关系是库所集和变迁集构造出来的。  Petri网模型描述的是基于图的数学形式化模型，对于分析离散的系统有好的效果。Petri网模型还可用来描述非因果的行为和不同步的因果关系。Petri网理论研究的主要内容是系统模型的各种行为特征，主要包含以下特性：有界性、可逆性、可覆盖性、活性、可达性、公平性等。  （黄泰达）  数据流图 (DFD) 用于表示业务信息系统中的数据流，它表达了系统中的据传从输入到存储间所涉及的程序。  数据流图可以分为逻辑形和物理形。逻辑数据流图描述了用以完成某业务功能所涉及的、业务层面的数据流动， 物理数据流图则描述系统层面的数据流动。  11:05  数据流图的优点：  在数据流图中没有任何具体的物理元素，它只是描绘信息在软件中流动和被处理的情况。  因为数据流图是系统逻辑功能的图形表示，即使不是专业的计算机技术人员也容易理解它，所以是分析员与用户之间极好的沟通工具。  设计数据流图时只需考虑系统必须完成的基本逻辑功能，完全不需考虑怎样具体地实现这些功能，因此，它也是今后进行软件设计很好的出发点。  数据流图可以在任何抽象层次上被用来表示系统或软件。事实上，数据流图可以被分层次地画，层次越低（在数据流图中1层比0层的层次更等）表现出的信息流细节和功能细节也越多。数据流图既提供了功能建模机制、也提供了信息流建模机制。  （孟小凡）  OCL(Obest Constraint Language)  OCL是约束（Constraint）语言和查询（Query）语言：  一个约束就是对一个（或部分）面向对象模型或者系统的一个或者一些值的限制。UML类图中的所有值都可以被约束，而表达这些约束的方法就是 OCL。在UML2标准中，OCL不仅用来写约束，还能够用来对UML图中的任何元素写表达式。每个OCL表达式都能指出系统中的一个值或者对象。因为 OCL表达式能够求出一个系统中的任何值或者值的集合，因此它具有了和SQL同样的能力，也就是说OCL也是一种查询语言。  OCL是基于数学的，但没有使用数学符号：  OCL的基础是数学中的集合论和谓词逻辑，并且它有一个形式化的数学语义，但是它并没有使用某种数学符号。因为虽然数学符号能够清晰的、无歧义的表达事物，但是只有极少的专家可以看懂。所以数学符号并不适合用于一个广泛应用的标准语言。  自然语言是最易懂的，但是它是含混不清晰的。OCL取了自然语言和数学符号的折中方案，使用普通的ASCII字符来表达数学中同样的概念。如果你不喜欢当前的OCL表达方法，OCL规范还允许你定义自己的OCL符号集，这点是可以理解的，因为OCL有一个清晰的数学语义。  强类型的语言：  OCL是一个类型语言，任何表达式的值都是属于一个类型的。这个类型可以是预定义的标准类型例如Boolean或者Integer，也可以是UML图中的元素例如对象。也可以是这些元素组成的集合，例如对象的集合、包、有序集合等等。  议题2：构建的一部分动态建模（初步构建，非最终模型）  假设我们的人才招聘系统有以下几个步骤：   1. 招聘管理员发布职位信息。 2. 求职者申请职位。 3. 招聘管理员筛选申请，决定是否邀请面试。 4. 求职者接受邀请，面试。 5. 招聘管理员根据面试情况决定是否录用。 6. 招聘结束。   为了控制整个招聘过程，我们可以使用Petri网对其进行建模。首先我们需要定义一些库所和变迁：  库所：   * 职位信息库所（Job Posting）：表示职位信息 * 申请人数库所（Applicants）：表示已申请的求职者数量 * 面试邀请库所（Interview Invites）：表示已发出的面试邀请数量 * 面试结果库所（Interview Results）：表示已完成面试的求职者数量 * 录用结果库所（Hires）：表示已录用的求职者数量   变迁：   * 发布职位信息变迁（Post Job）：表示招聘管理员发布职位信息 * 申请职位变迁（Apply for Job）：表示求职者申请职位 * 筛选申请变迁（Screen Applicants）：表示招聘管理员对申请进行筛选 * 面试变迁（Interview）：表示求职者进行面试 * 决定录用变迁（Make Hire Decision）：表示招聘管理员根据面试情况决定录用求职者 * 面试失败变迁（Failed Interviewer）：表示面试结果为失败，求职者未被录用   确定库所和变迁之间的关系后，初步绘制的Petri图如下    此外，我们绘制了项目的顶层图和零层图的雏形      议题3：SRS结构的设计（采用简易结构，非国标）  我们决定参考下面的案例  <https://www.omegaxyz.com/2019/07/23/software-specification/>  确定我们的SRS结构如下  [一、 引言](https://www.omegaxyz.com/2019/07/23/software-specification/#%e4%b8%80%e3%80%81-%e5%bc%95%e8%a8%80)  [1.1 定位与目标](https://www.omegaxyz.com/2019/07/23/software-specification/#1-1-%e5%ae%9a%e4%bd%8d%e4%b8%8e%e7%9b%ae%e6%a0%87)  [1.2 对象](https://www.omegaxyz.com/2019/07/23/software-specification/#1-2-%e5%af%b9%e8%b1%a1)  [1.3 软件需求分析理论](https://www.omegaxyz.com/2019/07/23/software-specification/#1-3-%e8%bd%af%e4%bb%b6%e9%9c%80%e6%b1%82%e5%88%86%e6%9e%90%e7%90%86%e8%ae%ba)  [1.4 软件需求分析目标](https://www.omegaxyz.com/2019/07/23/software-specification/#1-4-%e8%bd%af%e4%bb%b6%e9%9c%80%e6%b1%82%e5%88%86%e6%9e%90%e7%9b%ae%e6%a0%87)  [二、 需求概述](https://www.omegaxyz.com/2019/07/23/software-specification/#%e4%ba%8c%e3%80%81-%e9%9c%80%e6%b1%82%e6%a6%82%e8%bf%b0)  [2.1 项目背景](https://www.omegaxyz.com/2019/07/23/software-specification/#2-1-%e9%a1%b9%e7%9b%ae%e8%83%8c%e6%99%af)  [2.2 需求概述](https://www.omegaxyz.com/2019/07/23/software-specification/#2-2-%e9%9c%80%e6%b1%82%e6%a6%82%e8%bf%b0)  [2.3 系统结构](https://www.omegaxyz.com/2019/07/23/software-specification/#2-3-%e7%b3%bb%e7%bb%9f%e7%bb%93%e6%9e%84)  [三、 系统功能需求](https://www.omegaxyz.com/2019/07/23/software-specification/#%e4%b8%89%e3%80%81-%e7%b3%bb%e7%bb%9f%e5%8a%9f%e8%83%bd%e9%9c%80%e6%b1%82)  [3.1 功能总览](https://www.omegaxyz.com/2019/07/23/software-specification/#3-1-%e5%8a%9f%e8%83%bd%e6%80%bb%e8%a7%88)  [3.2 业务流程图](https://www.omegaxyz.com/2019/07/23/software-specification/#3-2-%e4%b8%9a%e5%8a%a1%e6%b5%81%e7%a8%8b%e5%9b%be)  [3.3 数据流分析](https://www.omegaxyz.com/2019/07/23/software-specification/#3-3-%e6%95%b0%e6%8d%ae%e6%b5%81%e5%88%86%e6%9e%90)  3.4 数据字典  [3.5 E-R图](https://www.omegaxyz.com/2019/07/23/software-specification/#3-5-e-r%e5%9b%be)  [四、 软硬件及外部系统接口需求](https://www.omegaxyz.com/2019/07/23/software-specification/#%e5%9b%9b%e3%80%81-%e8%bd%af%e7%a1%ac%e4%bb%b6%e5%8f%8a%e5%a4%96%e9%83%a8%e7%b3%bb%e7%bb%9f%e6%8e%a5%e5%8f%a3%e9%9c%80%e6%b1%82)  [4.1 用户界面](https://www.omegaxyz.com/2019/07/23/software-specification/#4-1-%e7%94%a8%e6%88%b7%e7%95%8c%e9%9d%a2)  [4.2 硬件需求](https://www.omegaxyz.com/2019/07/23/software-specification/#4-2-%e7%a1%ac%e4%bb%b6%e9%9c%80%e6%b1%82)  [4.3 运行环境](https://www.omegaxyz.com/2019/07/23/software-specification/#4-3-%e8%bf%90%e8%a1%8c%e7%8e%af%e5%a2%83)  [五、 可靠性与可用性需求](https://www.omegaxyz.com/2019/07/23/software-specification/#%e4%ba%94%e3%80%81-%e5%8f%af%e9%9d%a0%e6%80%a7%e4%b8%8e%e5%8f%af%e7%94%a8%e6%80%a7%e9%9c%80%e6%b1%82)  [5.1 性能需求](https://www.omegaxyz.com/2019/07/23/software-specification/#5-1-%e6%80%a7%e8%83%bd%e9%9c%80%e6%b1%82)  [5.2 安全性需求](https://www.omegaxyz.com/2019/07/23/software-specification/#5-2-%e5%ae%89%e5%85%a8%e6%80%a7%e9%9c%80%e6%b1%82)  [六、 参考文献](https://www.omegaxyz.com/2019/07/23/software-specification/#%e5%85%ad%e3%80%81-%e5%8f%82%e8%80%83%e6%96%87%e7%8c%ae) | | |
| 备注 | 无 | | |