

****

**面向对象的分析与设计个人报告**

|  |  |
| --- | --- |
| 课程名称 | 面向对象的分析与设计 |
| 姓名 | LS2121201 卜永祺 |

目录

[一、目标模型简介 3](#_Toc766)

[二、目标模型元模型 4](#_Toc6412)

[三、目标模型图形建模 5](#_Toc6106)

[四、用户故事到EARS的转换 7](#_Toc23162)

## 一、目标模型简介

用户故事：

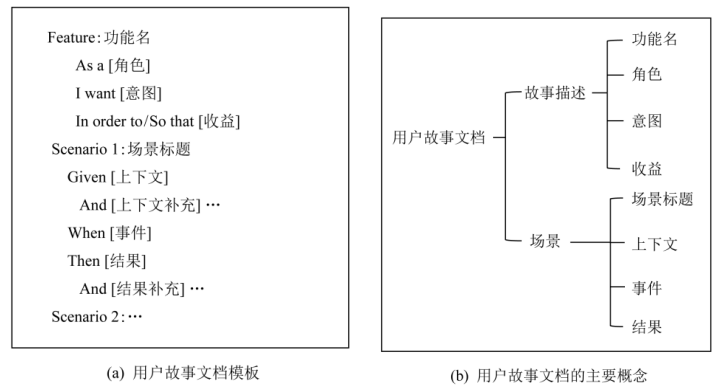
一般的用户故事是从用户的角度来描述用户渴望得到的功能。一个好的用户故事包括三个要素：

角色：谁要使用这个功能。

活动：需要完成什么样的功能。

商业价值：为什么需要这个功能，这个功能带来什么样的价值。

比较简单所以我们使用了带场景的用户故事，包括功能(feature)和场景(scenario)两部分信息，表达用户故事实例化需求，该模板及主要概念。其中，“Feature”后面是一个故事陈述,表示一 个独立的功能，即以某个“角色”为视角,提出对使用软件系统的“意图或功能需要”以达成某种“目的∕收益”，其中,“Scenario”后面是特征的实例化描述。在某个状态和条件满足的情况下,执行某个动作或发生某个事件，系统达成了某种结果状态。一个功能。包括至少一个或多个场景，用来探索一些边界情况, 帮助实现该功能需求的正确性检验和测试。其中，所有场景都遵循同样的模式:将系统置于某种特定的。上下文(Given和And描述的部分)；戳一下(或者点一下)系统(When 描述的部分)；系统置于新状态 (Then和And描述的部分)。



EARS:

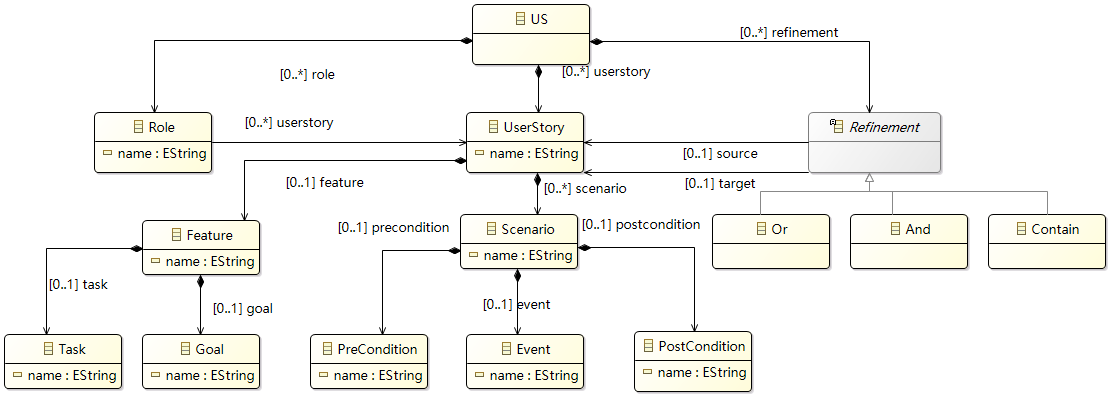
在软件工程中，软件需求规范（SRS）是定义软件及其功能的 文档,而且SRS通常是用自然语言（NL）编写的。NL需求文档中可能出现的一些问题：歧义、模糊性、复杂性、重复、需求的不恰当、不 可测试性。为了消除这些问题，引入了半结构化的自然语言Easy Approach to Requirements Syntax（EARS）。EARS于2009年在劳斯莱斯（Rolls-Royce）公司的Alistair Mavin创建。其最初目的是为其产品（包括航空发动机控制系统）创 建更好的需求文档。

EARS是半结构化自然语言，相对简单，易于学习。EARS是一种使用五个简单模板编写SRS文档的方法。EARS应用广泛，而且已经应用到很多工业界。 Alistair Mavin于2009年创建EARS并应用于航空发动机控制系统[。Alistair Mavin和Eero Uusitalo等人将EARS应用于核能领域的监管需求，核电厂安全设计法规指南 . Brendan Hal和Devesh Bhatt 等人将EARS用于霍尼韦尔（Honeywell）航天航空，采用约束语言增强的需求方法（CLEAR）方法。由于EARS语法的简易性还应用于需求培训课，英特尔（Intel）的Requirements Authors Mentoring Program(RAMP)需求作者指导计划,做用EARS重写需求的实验取得很好的效果。

## 二、目标模型元模型

用户故事：

参考用户故事和带场景的用户故事，绘制了新的用户故事元模型，该元模型将Rola（角色）从Feature（功能）中提取出来，让它与UserStory（用户故事）属于同一层级，并与相应的用户故事有userstory连接，使得一个Rola（角色）可以对应多个UserStory。Feature（功能）可以创建Task（完成这个功能需要采取的行动）和Goal（解释为什么要完成当前功能或者可以获得的利益）。Scenario（上下文）可以创建PreCondition（前置条件）、PostCondition（后置条件）以及Event（补充）。Refinement用于描述UserStory之间存在的各种关系。



EARS:

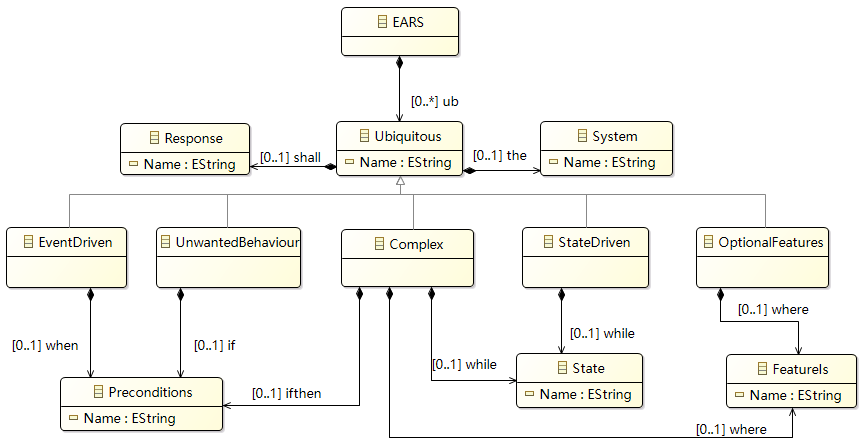
EARS通用需求语法为:

<optional preconditions> <optional trigger> the <system name> shall <system response>

为表示不同类的需求，通用需求语法被五类（普遍存在的、事件驱动的、状态驱动的、不需要的行为和可选的特性），具体描述如下：

1. Ubiquitous requirements(普遍存在、无处不在的)；没有任何先决条件或触发器。它不会由在系统检测到的事件或处于定义的系统状态而调用，但它是始终处于活动状态。
2. Event-driven requirements(事件驱动)：只有当在系统检测到触发事件发生时，系统才有的响应。关键字“when”用于事件驱动的需求。
3. Event-driven requirements(事件驱动)：当系统处于定义的某个状态时，系统才有的响应。
4. Unwanted behaviour requirements(不想要的行为需求)：涵盖所有不良情况。这包括故障，干扰，与所需用户行为的偏离以及交互系统的任何意外行为。不想要的行为是早期需求遗漏的主要来源，因此需要进行昂贵的返工。使用关键字If和Then来指定不想要的行为。
5. Optional feature requirement(可选的特性需求):在系统中，用于指定特性，用关键字Where表示。
6. Complex(复杂的需求):可以将以上五种组合成第6个模式表示复杂的需求。

根据以上语法创建了EARS元模型。

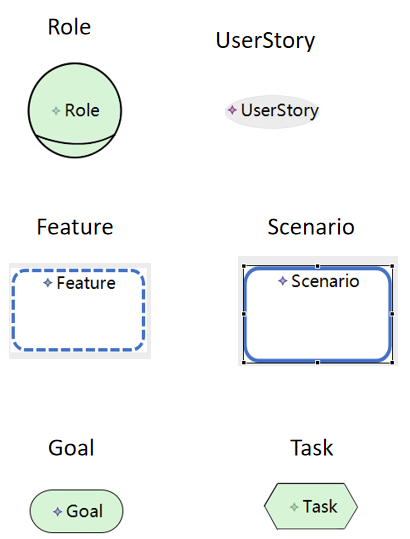


源码链接：[Object-oriented/第一次独立作业元模型/卜永祺\_ls2121201 at main · YongQi123/Object-oriented (github.com)](https://github.com/YongQi123/Object-oriented/tree/main/%E7%AC%AC%E4%B8%80%E6%AC%A1%E7%8B%AC%E7%AB%8B%E4%BD%9C%E4%B8%9A%E5%85%83%E6%A8%A1%E5%9E%8B/%E5%8D%9C%E6%B0%B8%E7%A5%BA_ls2121201)

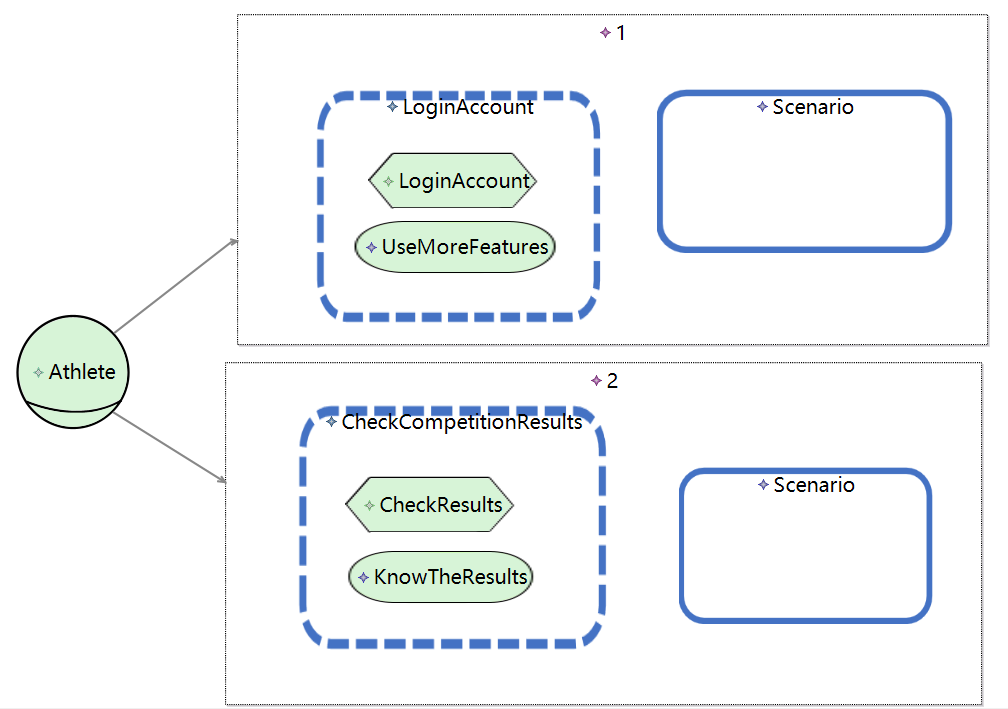
## 三、目标模型图形建模

用户故事：

关于用户故事工具的制作，其中Roal、UserStory、Task、Goal目前沿用了iStar2.0标准的图符进行了映射，之后会在后续工作进行改善。Feature被定义为蓝色虚线边框的长方形，Scenario被定义为蓝色线边框的长方形。



实例：

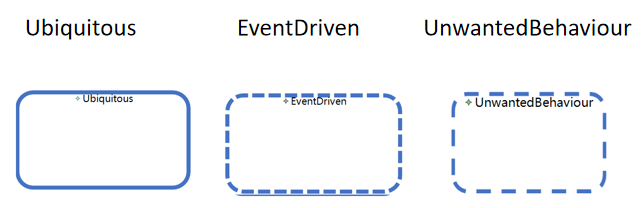


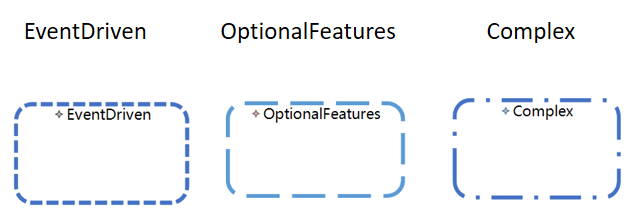
1.作为运动员，我想登录账号以便使用更多功能。

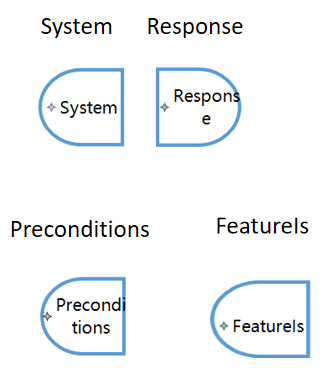
2.作为运动员，我想查询比赛成绩以便知道比赛结果。

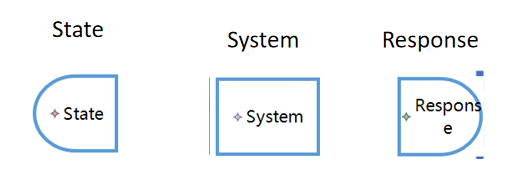
EARS:

为描述EARS的6种不同类型需求语法，我为每种类型设置了不同的边框，用来表示不同的不同的含义。由于在不同情况下，其内容系统和系统响应可能为两个或者三个元素，在为两个时会以两个半圆进行表示，如果为三个则是两个半圆和一个长方形进行表示。



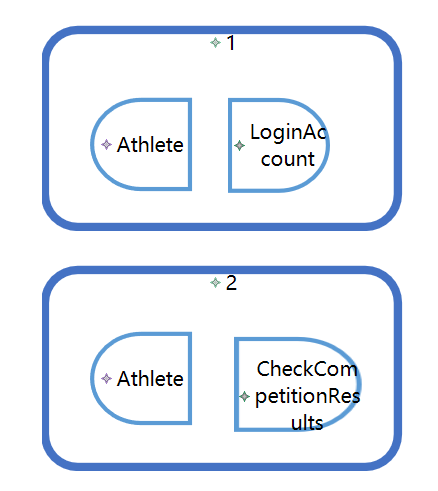






源码及视频演示链接：[Object-oriented/第二次独立作业Sirius/卜永祺-LS2121201 at main · YongQi123/Object-oriented (github.com)](https://github.com/YongQi123/Object-oriented/tree/main/%E7%AC%AC%E4%BA%8C%E6%AC%A1%E7%8B%AC%E7%AB%8B%E4%BD%9C%E4%B8%9ASirius/%E5%8D%9C%E6%B0%B8%E7%A5%BA-LS2121201)

实例：

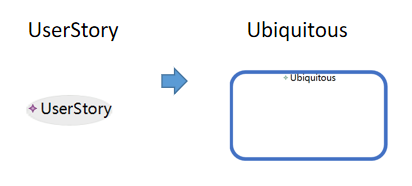


1.运动员账号需要登录

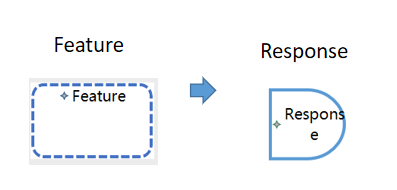
2.运动员可以查询成绩

## 四、用户故事到EARS的转换

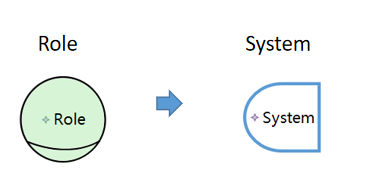
转换说明：转换规则目前有三个：



用户故事UserStory到EARS中Ubiquitous（普遍事件）的转换。

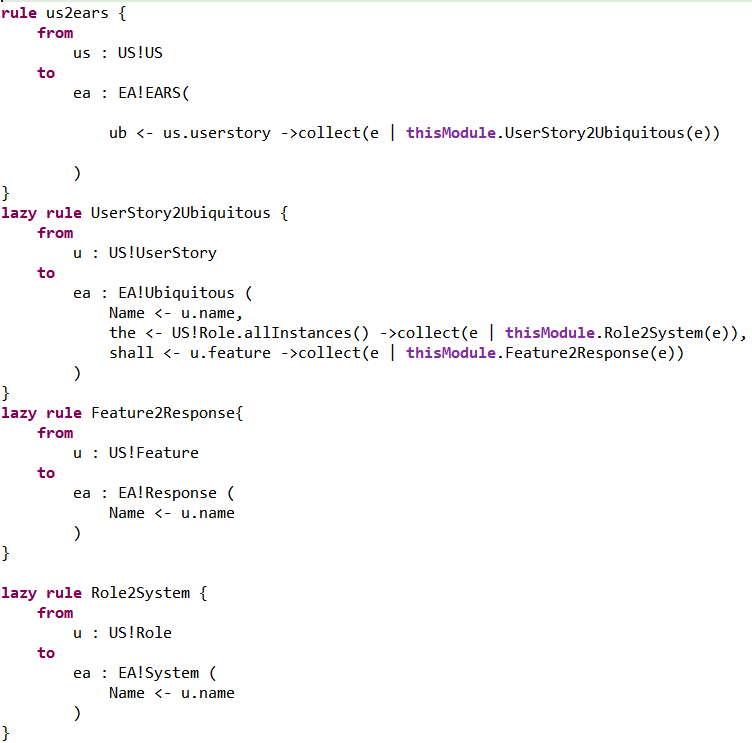


用户故事Feature（功能）到EARS中Response（系统响应）的转换。

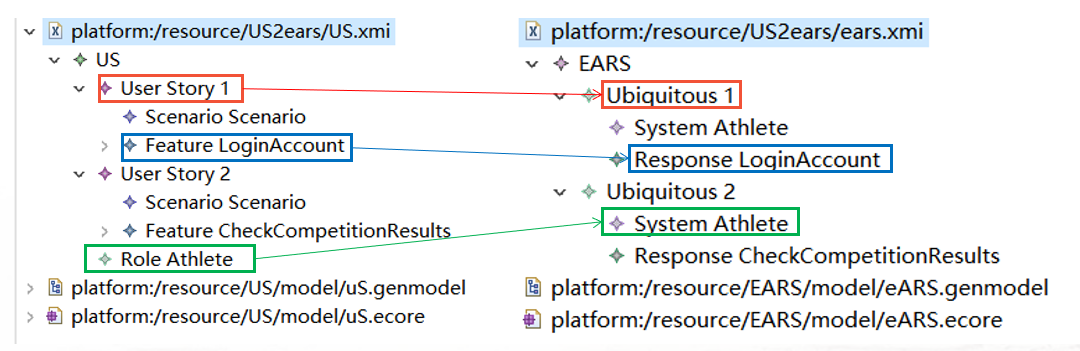


用户故事Role到EARS中System（系统）的转换。

转换代码：



转换规实例：



转换结果的XMI映射关系。

源码及视频演示链接：[Object-oriented/第三次独立作业ATL/卜永祺-LS2121201 at main · YongQi123/Object-oriented (github.com)](https://github.com/YongQi123/Object-oriented/tree/main/%E7%AC%AC%E4%B8%89%E6%AC%A1%E7%8B%AC%E7%AB%8B%E4%BD%9C%E4%B8%9AATL/%E5%8D%9C%E6%B0%B8%E7%A5%BA-LS2121201)