面向对象的分析与设计

课程报告

姓名 李阳

学号 SY2121109

学院 软件学院

时间 2021年12月15日

目录

[第一章 目标模型简介 1](#_Toc90633503)

[1.1 拓展的需求模型定义 1](#_Toc90633504)

[1.2 拓展需求模型的意义 1](#_Toc90633505)

[第二章 元模型定义 2](#_Toc90633506)

[2.1 元模型的定义 2](#_Toc90633507)

[2.2 拓展后的元模型 2](#_Toc90633508)

[第三章 目标模型图形建模 4](#_Toc90633509)

[3.1 元模型和图形映射关系说明 4](#_Toc90633510)

[3.2 元模型生成图形模型演示 4](#_Toc90633511)

[第四章 模型转换 5](#_Toc90633512)

[4.1 目标元模型 5](#_Toc90633513)

[4.2 待转换模型 6](#_Toc90633514)

[4.3 转换结果 6](#_Toc90633515)

[4.4 源码 7](#_Toc90633516)

1. 目标模型简介

1.1 拓展的需求模型定义

需求模型由概念类图、系统顺序图、用例图以及系统合约组成。RM2PT实现了由需求模型到原型系统的自动生成。近些年来，随着微服务架构的日益流行，微服务的拆分成为困扰人们的难题。基于这个前提对需求模型进行拓展，为其中的概念类图、用例图引入微服务边界，从而实现功能和数据的拆分。通过定义拓展后的需求模型可以实现从原始需求模型到微服务拆分的需求模型的转化。通过定义转换规则实现了对一个单一架构系统的功能的拆分。

1.2 拓展需求模型的意义

现有的微服务拆分方式主要是领域驱动设计（DDD）：领域驱动设计是一种业务层面的划分，该过程主要分为两个阶段：领域模型设计和领域模型驱动软件设计。领域模型既包含了实体概念又包含了过程概念，同时包含属性与方法，是一组抽象概念的集合。领域模型是一种设计模型，因此从需求模型到领域模型的转化需要领域专家的参与。其设计步骤包含初步场景分析、领域建模、微服务设计与拆分。

尽管领域驱动设计在业界被广泛接受，但该微服务拆分方法仍然存在以下问题：1）DDD首先需要进行专业的领域划分，该过程需要领域专家的参与来定义实体、聚合以及领域模型，然而现实是在很多微服务架构场景中该步骤实施具有难度。 2）领域驱动设计理论性强，由于无法同时满足理论中所有的条件，建立一个完整且自封闭的领域模型十分困难，这就导致了很少有项目能够将领域驱动设计完美落地。3）域模型通常会省略推论微服务所必需的信息，除此之外它还缺少基础结构组件，因此DDD需要设计人员投入大量精力。

基于上述情况提出的一种基于需求模型的微服务拆分能够自动、快速、直接地生成微服务架构，从而解决了DDD方法复杂性、概念性较强的缺点。基于需求模型的微服务拆分方法用到的需求模型由用例图、系统顺序图、概念类图、系统操作合约组成，其中的概念类图与领域模型相似，但其中只包含属性。从需求直接进行微服务的拆分省去了建立领域模型的步骤，使微服务架构的建立能够自动生成从而不再需要领域专家的参与，节省开销的同时提高了建立微服务架构的效率。

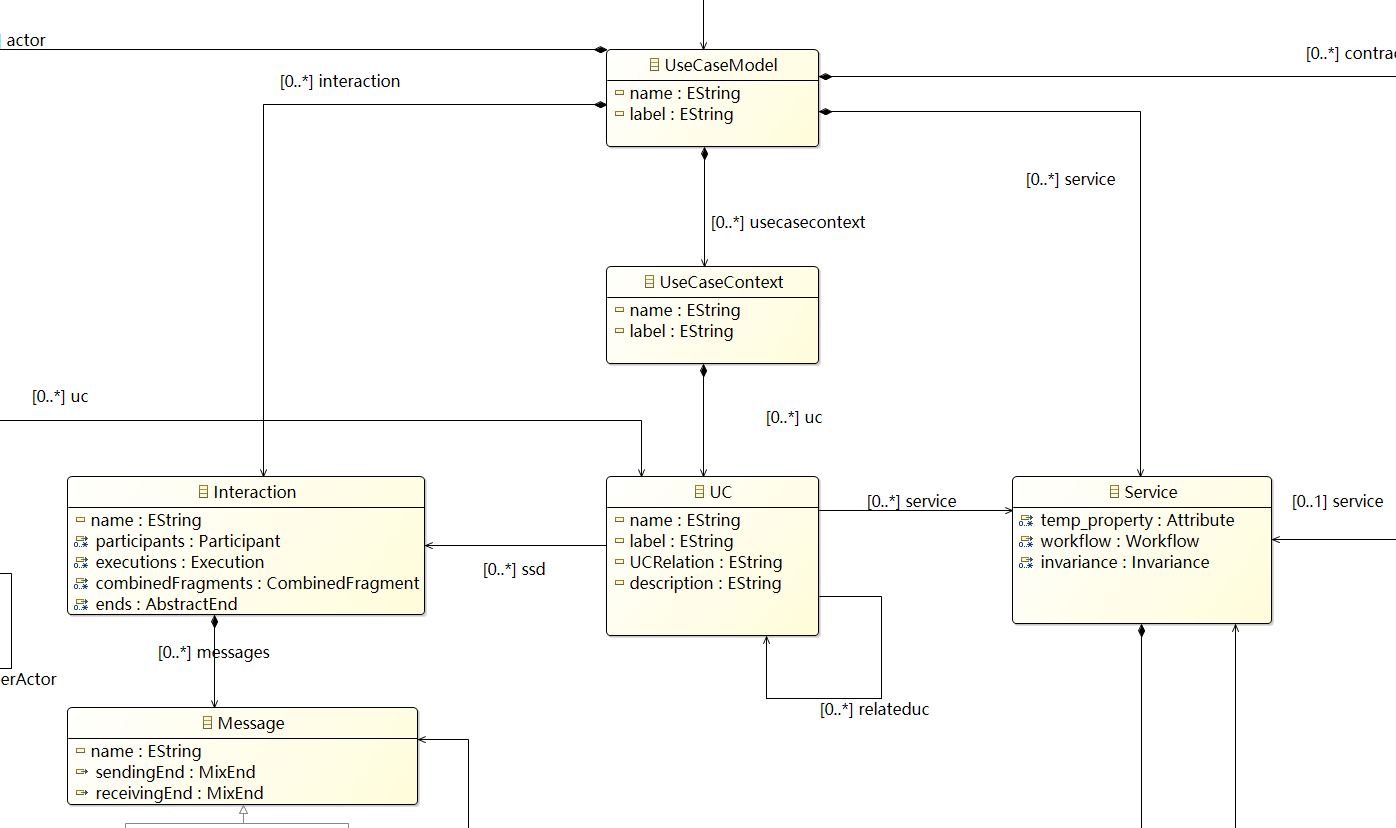
1. 元模型定义

2.1 元模型的定义

元模型（MetaModel）是关于模型的模型。元模型定义了描述某一模型的规范，具体来说就是组成模型的元素和元素之间的关系。元模型是相对与模型的概念，每个模型都有一种元模型来解释它，同时离开了模型元模型也就没有了意义。元模型往往用来在某一特定的领域定义一个基础的通用的语言，来讨论和描述该领域的问题及解决方法。例如，可以将元模型想象成为某种形式语言，模型就是一篇用该语言描述的文章，其中元模型中的元素就是该语言的词汇，元素之间的关系就是该语言的语法。

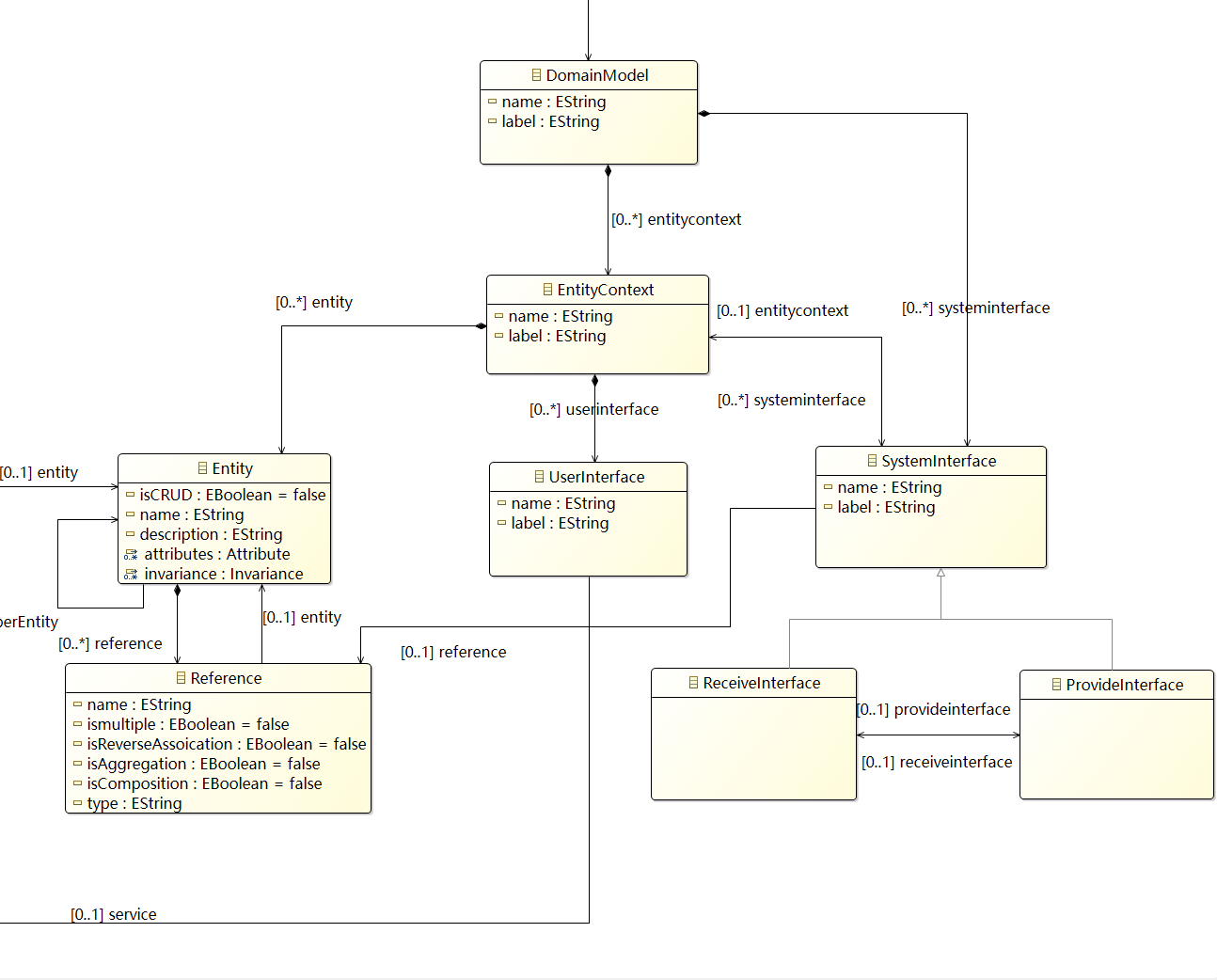
2.2 拓展后的元模型

在Remodel元模型上添加UseCaseContext、EntityContext、SystemInterface等元素，并添加元素之间对应的关系,用以表示微服务的拆分结果。



图一

首先应该定义用例图边界，需要在UC与UseCaseModel之间增加UseCaseContext元素来表示边界上下文。这个边界用来表示在将来的系统中只有边界内的实体类才会被包含在单个微服务系统之中。该类边界实现了对功能的拆分。

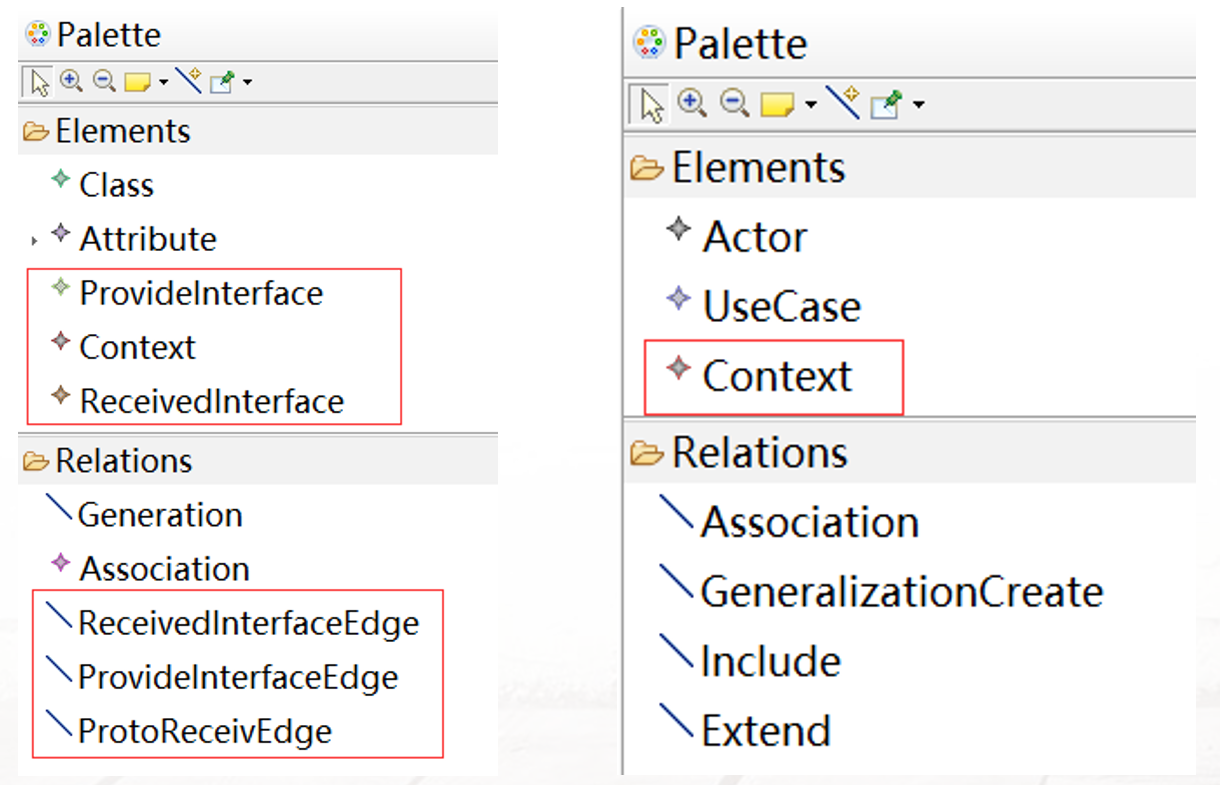


图二

之后需要定义如图二所示的概念类图的边界，概念类图边界定义在DomainModel和Entity之间，用来对实体类进行划分，将其切分为单个的微服务集合。同时我们需要定义系统接口用来表示不同微服务之间的交互关系。如果在之前的概念类图中两个类之间存在依赖关系，而这两个类被分到了不同的微服务中，那么就需要在两个微服务边界上定义系统接口，通过系统接口实现信息的传递。同时UserInterface依赖于service，表示微服务向外接提供的功能。

1. 目标模型图形建模

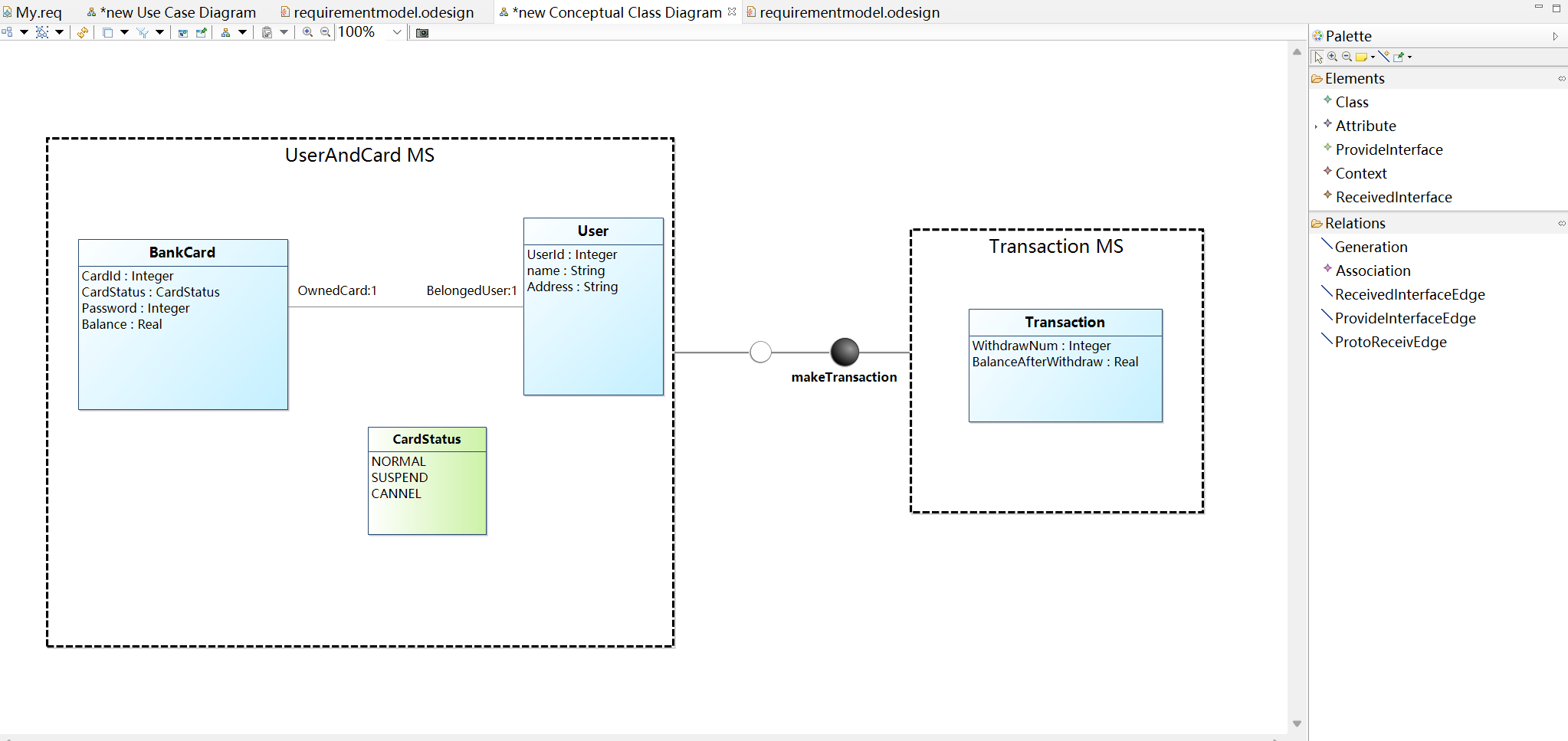
3.1 元模型和图形映射关系说明

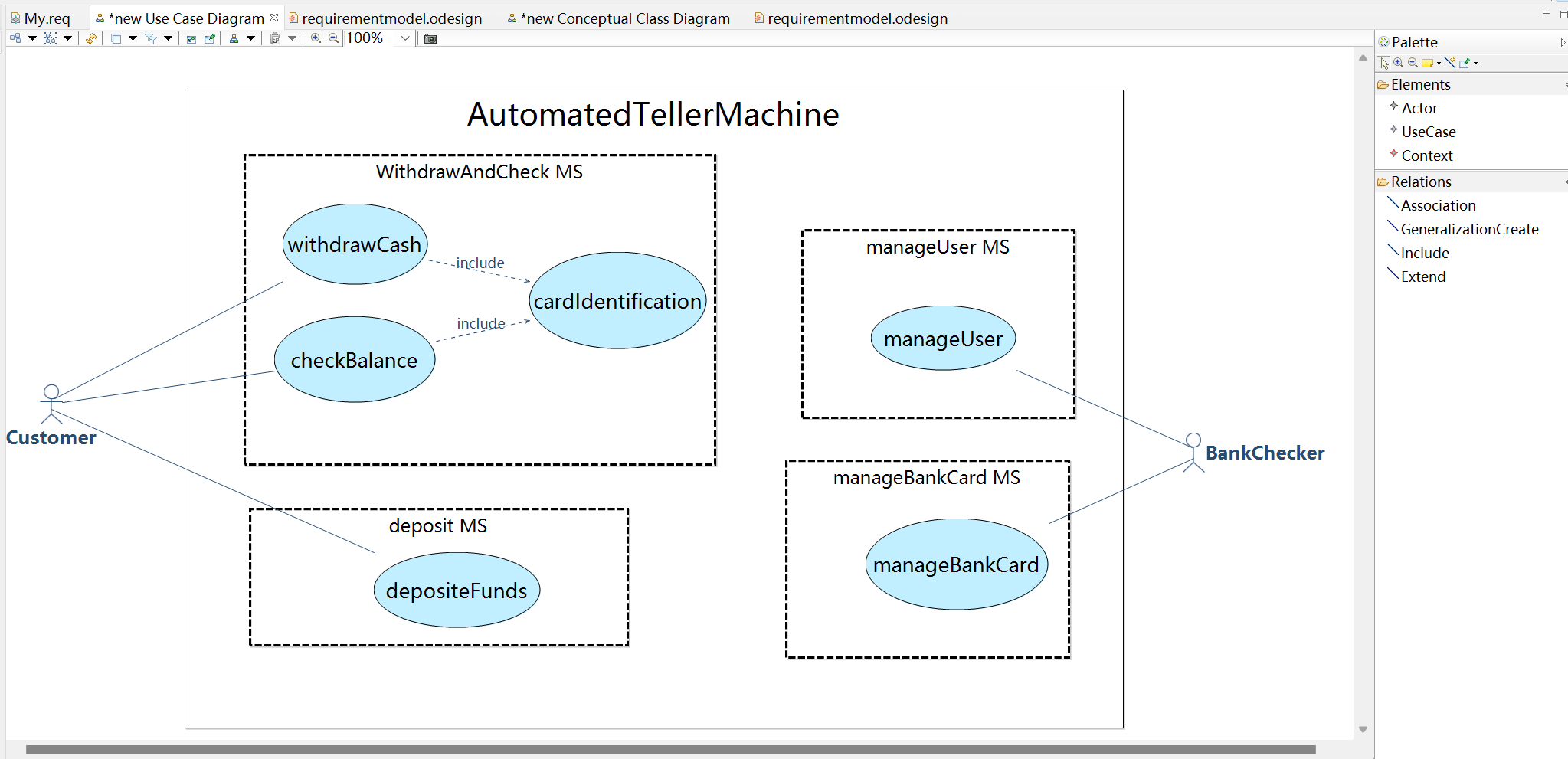


图三

两类边界都用虚线框来表示。在用概念类图中需要定义接口提供节点、接口接收节点，以及微服务边界与它们之间的两类连边、两类接口节点之间的连边。在用例图中仅需要定义边界。最终的图形如3.2节所示。

3.2 元模型生成图形模型演示





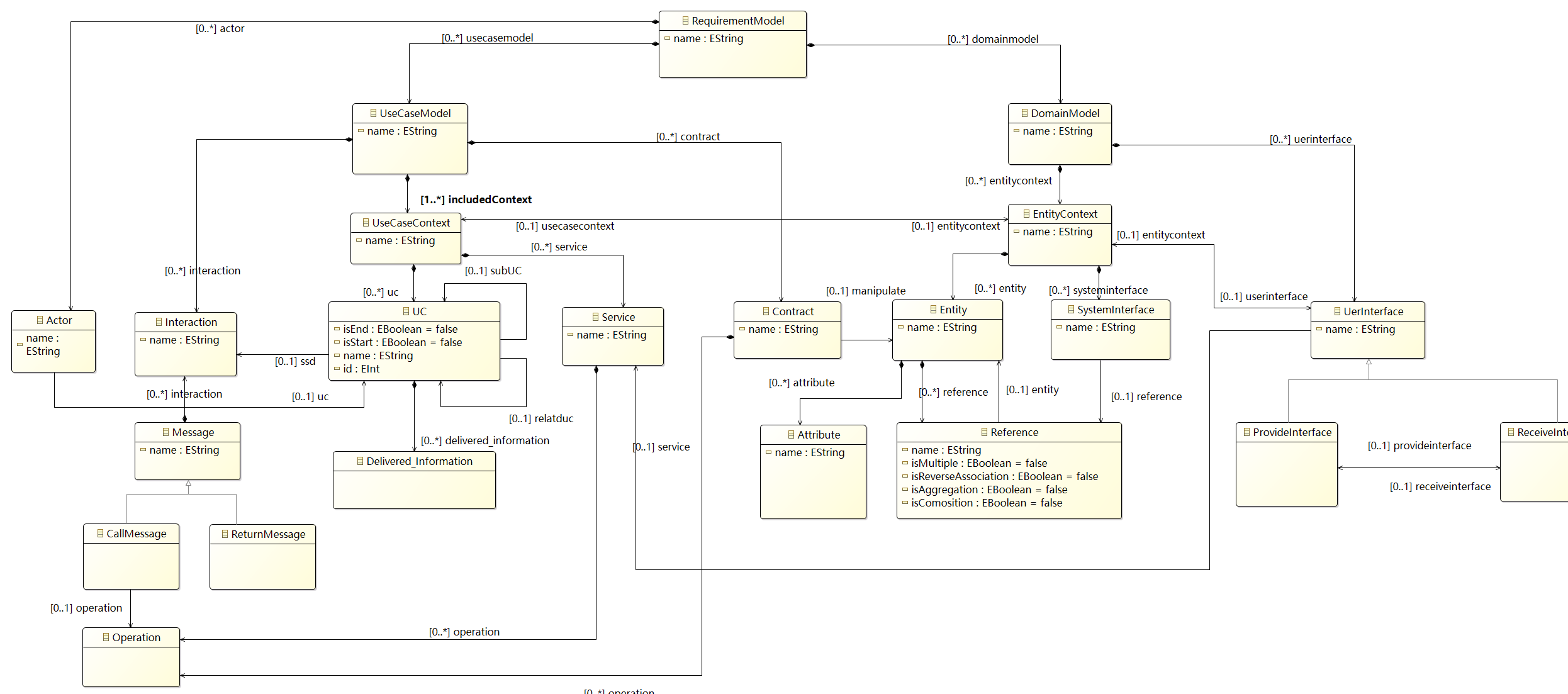
该节描述了对ATM系统的建模结果，用例图根据拆分原则得到了四个微服务。

第四章 模型转换

模型转换的目标是从原始的需求模型转化为拆分后带边界的需求模型，涉及输入包括待转换元模型、目标元模型、ATL转换规则。本章只进行了用例图的转换。

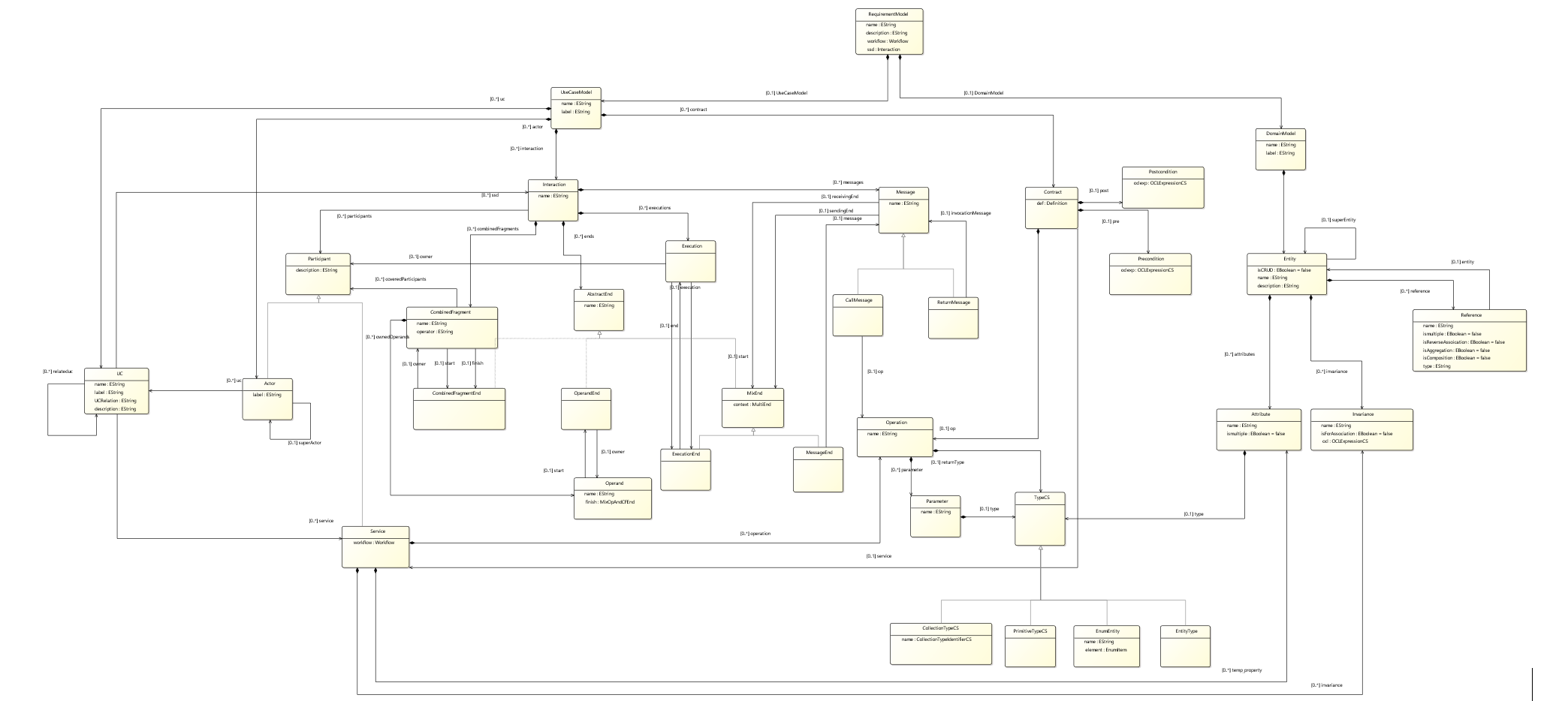
4.1 目标元模型

目标元模型为：该模型为拓展后的需求模型元模型



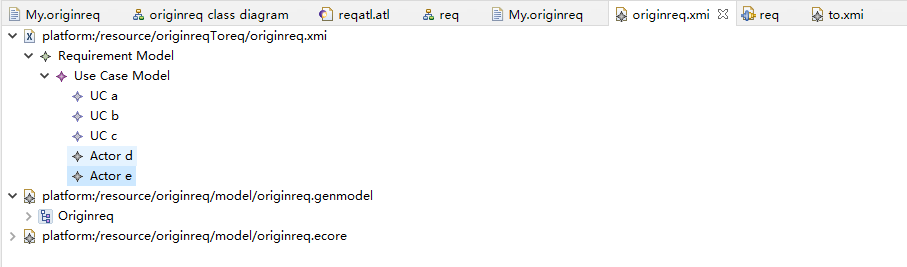
4.2 待转换模型

待转换模型为Remodel模型：

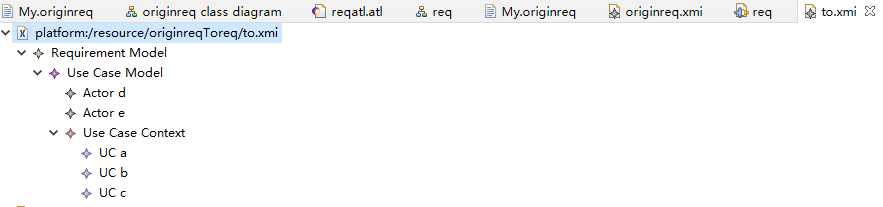


4.3 转换结果

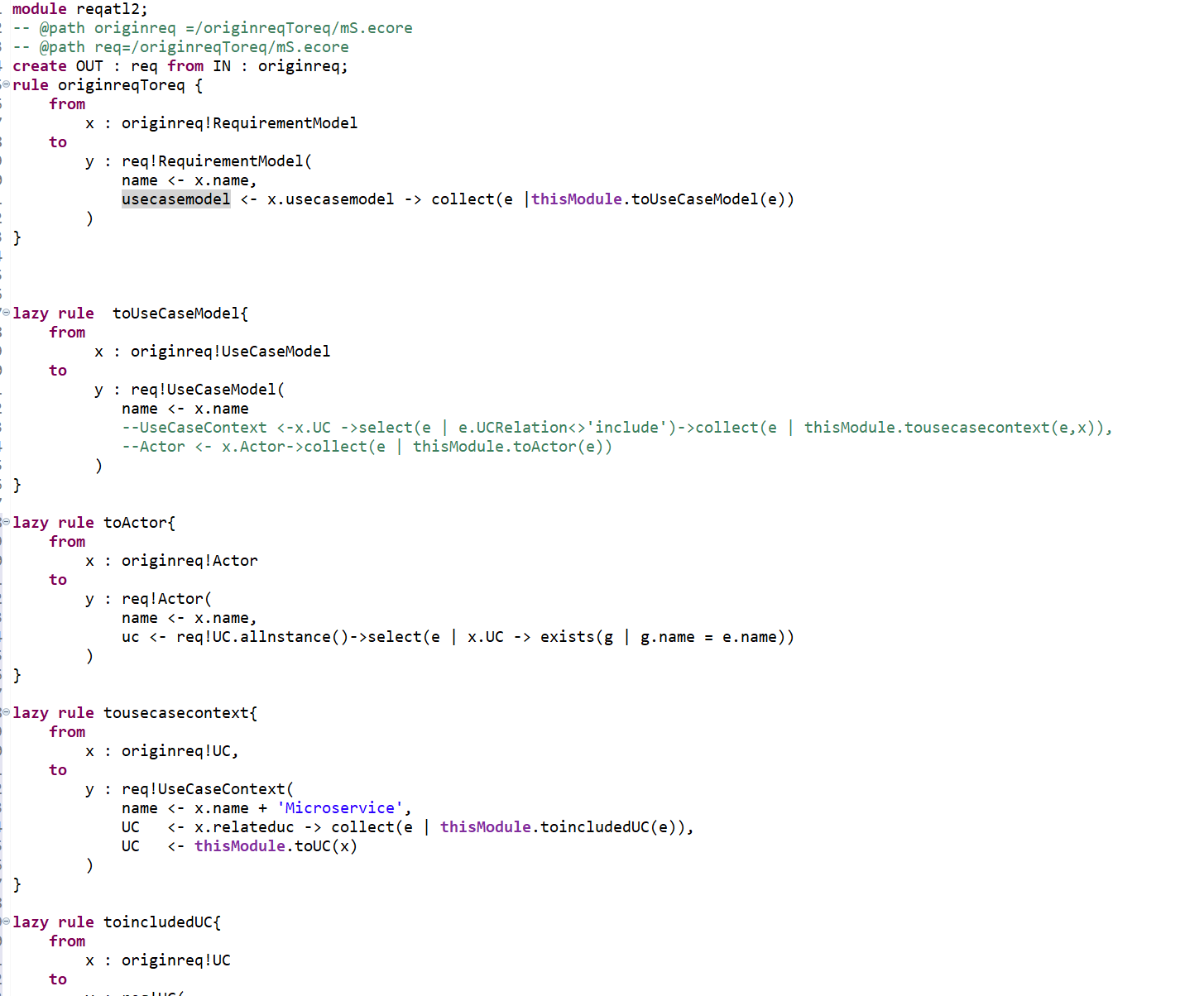
转换输入



转换输出



4.4 源码



第五章：总结

该实验从数据和过程独立进行微服务的划分，最后再将两种划分结果进行汇总得到最终的微服务架构。可以快速、高效地生成高质量的微服务框架，使开发人员只需将主要精力投入到需求模型生成上，提高建立微服务架构的效率。