面向对象的分析与设计

课程报告

姓名 吴成琦

学号 SY2139222

学院 网络空间安全学院

时间 2021年12月23日

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。一般来说，区块链是一个由分布式网络中的节点维护的不可篡改的账本。这些节点通过执行被共识协议验证过的交易来各自维护一个账本的副本，账本以区块的形式存在，每个区块通过哈希和之前的区块相连。

第一个被广为人知的区块链应用是加密货币[比特币](https://en.wikipedia.org/wiki/Bitcoin)，而其他应用都是从它衍生出来的。以太坊是另一种加密货币，它采用了不同方法，整合了许多类似比特币的特征，但是新增了智能合约为分布式应用创建了一个平台。比特币和以太坊属于同一类区块链，我们将其归类为公共非许可(Public Permissionless)区块链技术。这些基本上都是公共网络，允许任何人在上面匿名互动。

智能合约，在 Fabric 中称之为“链码”，作为受信任的分布式应用程序，从区块链中获得信任，在节点中达成基本共识。它是区块链应用的业务逻辑。

有三个关键点适用于智能合约，尤其是应用于平台时：

多个智能合约在网络中同时运行，

它们可以动态部署（很多情况下任何人都可以部署），

应用代码应视为不被信任的，甚至可能是恶意的。

大多数现有的具有智能合约能力的区块链平台遵循顺序执行架构，其中共识协议：

验证并将交易排序，然后将它们传播到所有的节点，

每个节点按顺序执行交易。

几乎所有现有的区块链系统都可以找到顺序执行架构，从非许可平台，如 [Ethereum](https://ethereum.org/)（基于 PoW 共识）到许可平台，如 [Tendermint](http://tendermint.com/)、[Chain](http://chain.com/) 和 [Quorum](http://www.jpmorgan.com/global/Quorum) 。

采用顺序执行架构的区块链执行智能合约的结果一定是确定的，否则，可能永远不会达成共识。为了解决非确定性问题，许多平台要求智能合约以非标准或特定领域的语言（例如 [Solidity](https://solidity.readthedocs.io/en/v0.4.23/)）编写，以便消除非确定性操作。这阻碍了平台的广泛采用，因为它要求开发人员学习新语言来编写智能合约，而且可能会编写错误的程序。

此外，由于所有节点都按顺序执行所有交易，性能和规模被限制。事实上系统要求智能合约代码要在每个节点上都执行，这就需要采取复杂措施来保护整个系统免受恶意合约的影响，以确保整个系统的弹性。

1.2 拓展需求模型的意义

现有的智能合约模型需要我们通过对区块链的五层架构和交易模型进行建模和分布式处理。智能合约定义业务对象的不同状态，并管理对象在不同状态之间变化的过程。智能合约很重要，因为它们允许架构师和智能合约开发人员定义在区块链网络中协作的不同组织之间共享的关键业务流程和数据。

尽管区块链智能合约开发基于传统的区块链开发，使得开发效率和开发进程大大提高，但基于模型的开发使得设计和自动生成智能合约代码变得较为容易，使得开发效率得以快速提高。

1. 元模型定义

2.1 元模型的定义

元模型（或称替代模型）是“模型的模型”， 元建模则指建立元模型的过程（meta-physics 为”形而上学“，即“科学的科学”）。因此，元建模的工作包括：分析、构建和开发一套用于给某类指定问题建模的框架、规则、约束、模型和理论等。在[软件工程](https://zh.wikipedia.org/wiki/%E8%BD%AF%E4%BB%B6%E5%B7%A5%E7%A8%8B" \o "软件工程)和[系统工程](https://zh.wikipedia.org/wiki/%E7%B3%BB%E7%BB%9F%E5%B7%A5%E7%A8%8B" \o "系统工程)中，顾名思义，这正符合元和建模的概念。元模型种类多样，应用广泛。

元建模是在一特定领域内构建“概念”(事件、条件等)集合。模型是真实世界中现象的抽象，元模型又是另一种抽象，关注模型本身的属性，所以可以把一个元模型看做对模型的抽象。模型应该服从它元模型的定义，如同一种计算机程序服从这种编程语言语法定义一样。

元模型的应用：

作为需要交换或存储的语义数据的概要

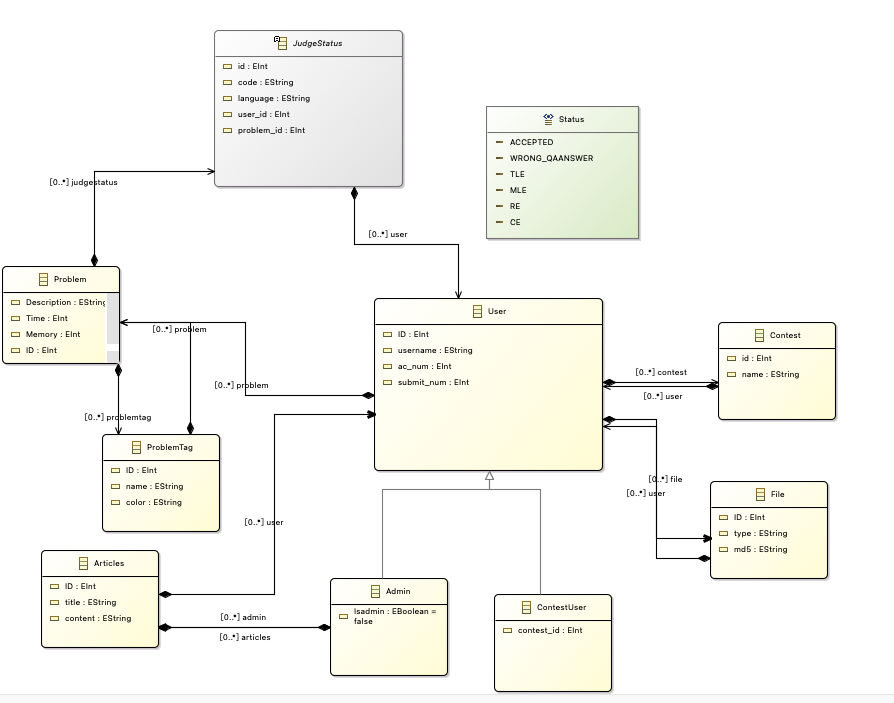
作为支持特定的方法或过程的语言

作为表达现有信息额外的语义的语言

因为元建模的“元”特性，其实践和理论都与科学学，元哲学，超理论和系统学，和元意识相关。这一概念可应用于数学，并已实际应用于计算机科学和计算机工程/软件工程的相关领域。

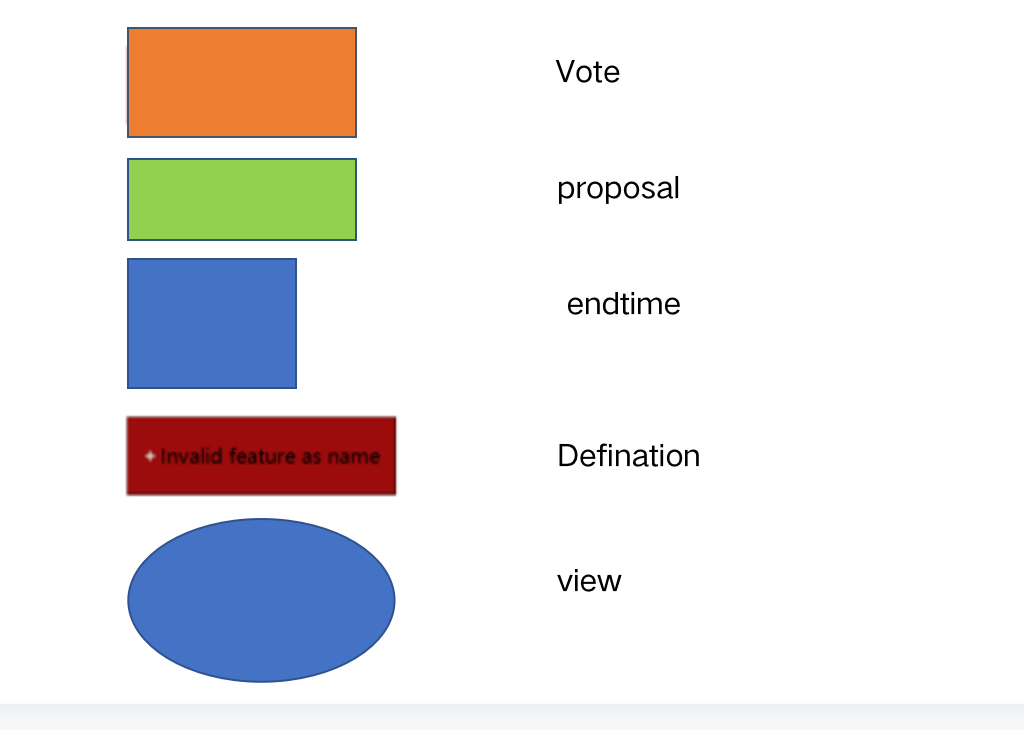
2.2 拓展后的元模型

按照下图的方式定义元模型。



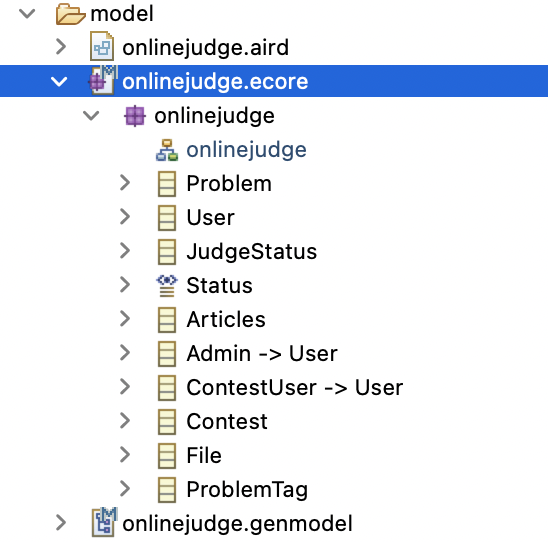
1. 目标模型图形建模

可以创建如下的几个类：

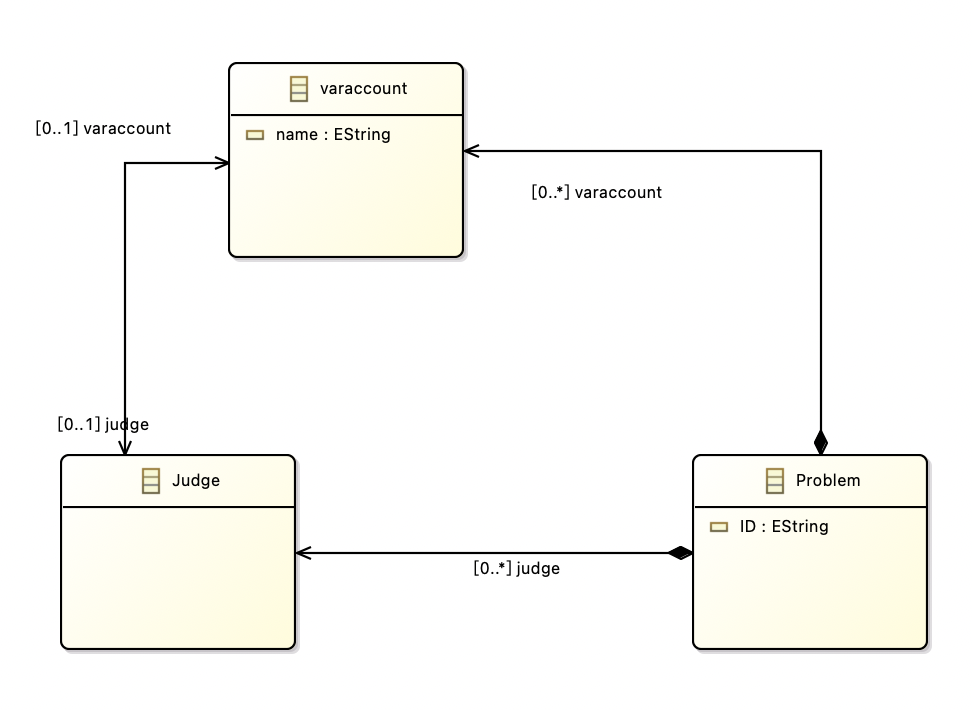


1. 模型转换

输入端模型：



输出端模型：



输入段示例：



输出端示例：

