第16讲 | 深入区块链技术（八）： UTXO与普通账户模型

2018-04-30 陈浩

深入浅出区块链

进入课程 >

讲述：黄洲君

时长 10:31　大小 6.03M

我们在第 2 讲“区块链到底是怎么运行”一文中，提到了村长给张三转账的例子，那里村长的例子就是 UTXO 模型的一个简化版本。

评论区里有不少留言在问：“为什么不直接记余额呢？”看来很多人都对这个问题很感兴趣，今天我们就来聊一聊这个话题。

区块链网络中有两种记账模式，除了 UTXO 模型还有 Account Based 结构，也就是普通账户模型，也叫账户余额模型，□前者□在比特币系的数字货币中被广泛使用，后者更多是用在智能合约型的区块链上。

普通账户模型

我们先从传统的账户模型□出发来聊聊是如何记账的，假设我们现在有一个支付系统，在这个支付系统中有村长和张三两个账户，村长账户里有 100 万，现在要转账给张三 10 万，这其中涉及的操作是这样的：

1. 检查村长的账户余额是否大于 10 万；
2. 把村长的账户扣除 10 万变成 90 万，然后发送一笔转账消息给张三的账户；
3. 张三的账户接受到转账消息，将张三的账户余额加 10 万。

我们可以发现，无论是村长还是张三，都具有一个余额作为状态，即当前余额是记录在某个地方的，只需要读出来即可，这种设计我们叫做账户余额模型。

如果以上三个步骤是在一个中心化系统中，甚至在同一个数据库中，那将非常简单，会直接退化成一个事务，我们见到的银行账户、信用卡系统、证券交易系统、各种电商类应用，理财类应用基本都是一个中心化系统中的，最多也就是跨表跨数据库。

想必这类场景下的设计，各位工程师对此应该是了如指掌的。

如果以上的步骤中，村长和张三的账户分属两个不同的系统，例如从 A 银行到 B 银行，就需要经过人民银行支付系统，即可信任的中心化第三方来做中介。

你可能发现了，在跨行转账的这种情况下，是没有办法做事务的，所以 1 和 3 是不同步的，如果 3 操作失败，还需要从 2 倒退到 1 的状态，这个情况叫做冲正交易。

□普通账户模型具有自定义数据类型的优点，但是却需要自己设计事务机制，就是上述所说的冲正交易。而接下来所讲的 UTXO 模型则恰恰相反。

UTXO 模型

UTXO 全称是：“Unspent Transaction Output”，这指的是：未花费的交易输出。这里面三个单词分别表示 “未花费的” “交易” “输出”，接下来我来详细讲解一下 UTXO 的含义。

UTXO 的核心设计思路是无状态，它记录的是□交易事件，□而不记录□最终状态，也就是说只记录变更事件，用户需要根据历史记录自行计算余额。

有点像□MySQL 中的 Binlog，主从模式的情况下，按照 Binlog 来更新数据，Redis 的 AOF 模式备份模式也是如此，UTXO 也是类似的思路。

下面我们按照□按照普通账户中的例子来重新讲解一遍。

如果要记录□交易本身，那么我们可以构造一笔交易，□这笔交易中村长转账 10 万给张三的同时，90 万转给自己。

如下所示：

```
村长   100 万 --> 张三   10 万
                --> 村长   90 万
```

这里其实有三条子记录，左边□一条，右边两条，左边叫做输入，右边叫做输出。

输入和输出组成了交易，输入和输入需要满足一些约束条件：

1. 任意一个交易必须至少一个输入、一个输出；
2. 输入必须全部移动，不能只使用部分，所以才产生了第二个输出指向村长自己；
3. 输入金额 = 输出金额之和 + 交易手续费，这里必须是等式。

对于村长来说，首先构造交易的输入输出，满足上述条件，然后广播到全网，接收方自行判断交易是否属于自己。这里满足约束条件构成的交易模型，也就是村长记录的三条转账事件就是 UTXO 模型。

账户余额模型与 UTXO 的比较

我们可以归纳出 UTXO 与普通账户模型的一些区别。

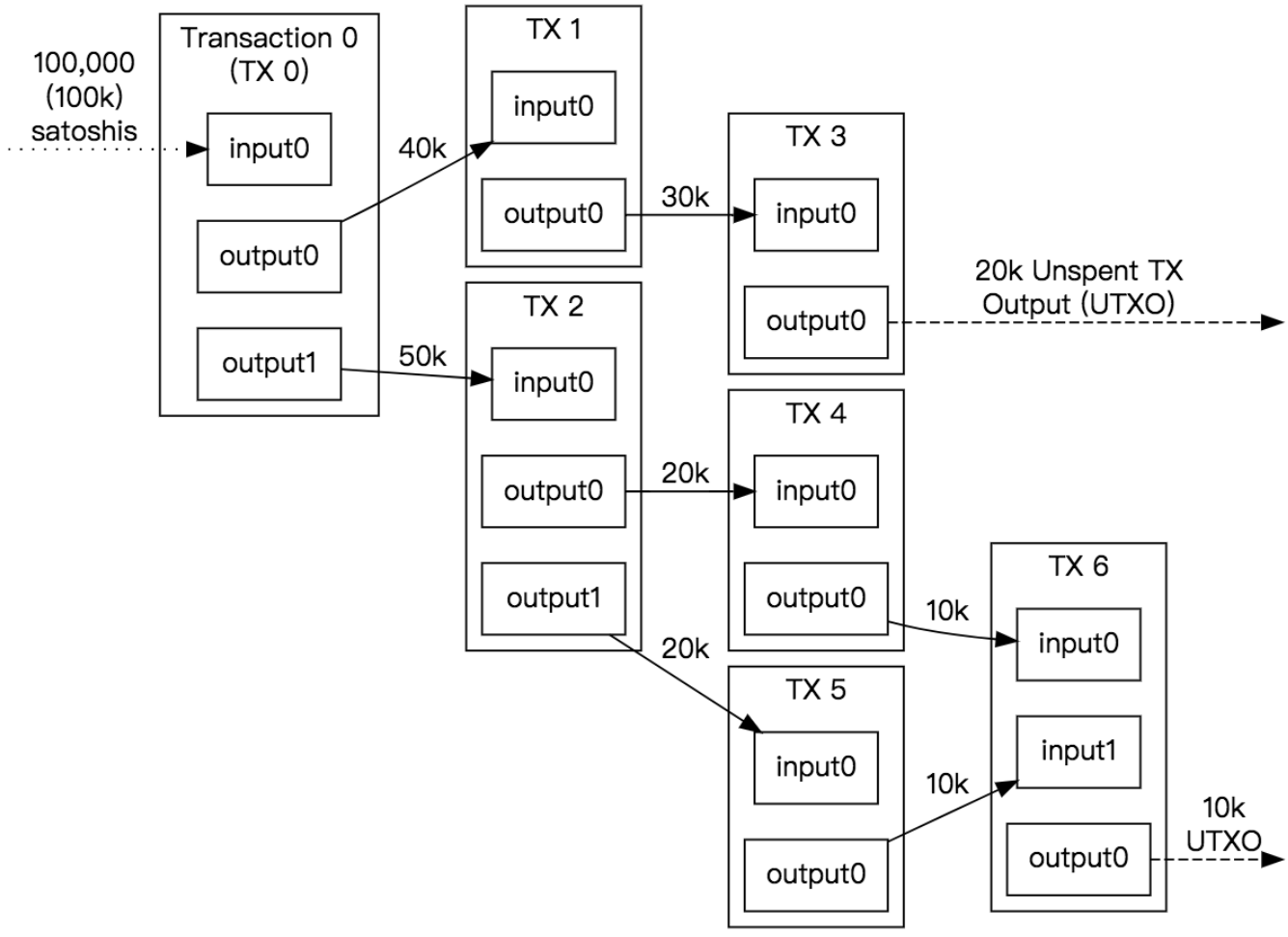
1. 存储空间，UTXO 占用空间比账户模型高，因为账户模型只记录最终状态。
2. 易用性，UTXO 比较难处理，账户模型简单容易理解。例如 UTXO 在使用上，还需要配合高效的 UTXO 组装算法，这个算法要求尽可能降低输入输出的个数，还要让“零钱“归整，算法的复杂度相比账户余额无疑要高。
3. 安全性，UTXO 比账户模型要高，UTXO□本身具备 ACID 的记账机制，而账户模型需要自行处理，例如重放攻击。

普通账户模型具有较高的自由度，可以让智能合约有更好的发挥空间，并且它避免了 UTXO 的复杂组装逻辑，精度控制上也更为得心应手。

UTXO 似乎天然是为数字货币设计的，具有较高频次跨账户转移场景都使用 UTXO 会比较好，考虑到智能合约的普适性，UTXO 与智能合约并不能很好地兼容，但是这也对开发者的自身水平提出了更高的要求。

区块链中的 UTXO 模型

我们借用比特币开发者文档中 UTXO 模型的□图示，来看看 UTXO 实际的构造形式。



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

上图中，所有的交易都可以找到前向交易，例如 TX5 的前向交易是 TX2，TX2 中的 Output1 作为 TX5 中的 Input0。

意思就是 TX2 中的付款人使用了 Output1 中指向的比特币转移给 TX5 中的收款人，接着 TX5 中的人又把收到的比特币转移给了 TX6 中的收款人，成为了 TX6 中 Output0。

我们也可以发现，TX6 中的收款人□还没有产生 TX7 交易，也就是说 Output0 还没有被花费，

这时候我们终于得到了 UTXO 的真正语义：Unspent Transaction Output，未花费的交易输出。

我们这时候可以发现 UTXO 也同样能表示余额，不过是重演计算的方式，它用不同的方式表达了余额，我们把一个地址上所有的 UTXO 全部□找出来，就是这个地址总的余额了。

我们还可以发现，无论是 TX5 还是 TX2，都已经成为历史交易，□它们□都忠实客观地记录了两笔交易，这两笔交易代表的是事件，而不是余额状态转移，□这是我们看到的最直观的区别。

我们再来看看一个□真实的交易例子。

这是区块链上一笔真实交易的例子，它记录了一笔 450ETP 的转账记录。

左边是输入，右边是两笔输出，其中第二个输出是给自己的账户，这和我们村长转账给张三的例子是一样的。

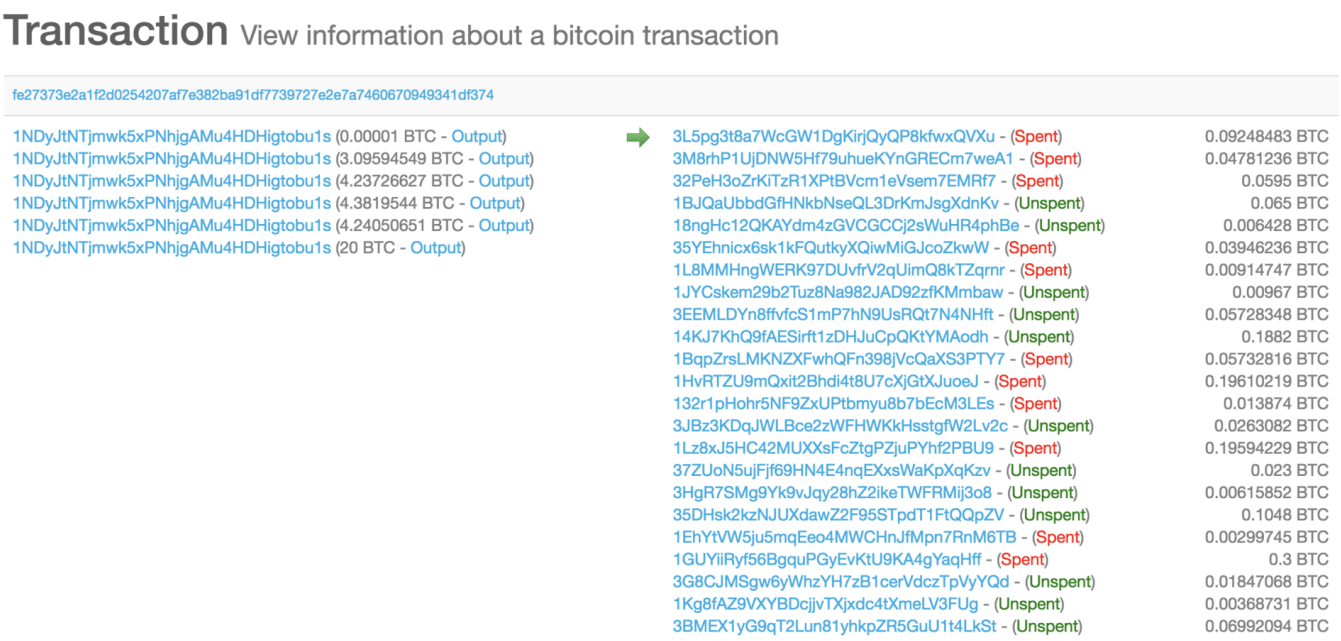
下图是交易解码为 JSON 格式的样子，可以看到 Previous_output 是放到 Inputs 数组里的，意思就是前向输出作为本次的输入。

复制代码

```
{
"hash" : "89e80e14db07c4904a57e2c1efb689bccbbf43942103c1a92166d5c0f27ea3d2",
"height" : 1093399,
"inputs" :
[
    {
        "address" : "MLWtmjwCtmK44FMwJMSfAkHaEvnnb2N6HX",
```

```
        "previous_output" :
        {
            "hash" : "770a72f35d3e3a78bd468949bad649f03b241cf7e2a84cc2d6fdabacdcc47f06"
            "index" : 0
        },
        "script" : "[ 304402202b21d7a79276985dc99777b70fd5095796dad58f35e29a019d2cb6cca
        "sequence" : 4294967295
    }
],
"lock_time" : "0",
"outputs" :
[
    {
        "address" : "MGz9yjLLn4AqyraRjSpiP2GmTWKnT3yfiL",
        "attachment" :
        {
            "type" : "etp"
        },
        "index" : 0,
        "locked_height_range" : 0,
        "script" : "dup hash160 [ 63ab0013d183f2592e4b46a358df01e88a09c0b8 ] equalverify
        "value" : 45000000000
    },
    {
        "address" : "MLWtmjwCtmK44FMwJMSfAkHaEvnnb2N6HX",
        "attachment" :
        {
            "type" : "etp"
        },
        "index" : 1,
        "locked_height_range" : 0,
        "script" : "dup hash160 [ 8a63941b392771c40f1c15e4374808f6bb464cba ] equalverify
        "value" : 118082150283
    }
],
"version" : "2"
}
```


我们再看看比特币上的例子：



这一笔比特币交易包含 6 个输入，几十个输出，交易一共 3.5kb，交易的输入输出会影响交易大小，比特币的交易费是根据字节收费的，交易尺寸越大越贵，而交易尺寸主要和输入输出的个数有关，也就是说，算法上并不规定输入输出的个数，而只有□区块尺寸限制。

在比特币中将小于 100kb 的交易称为标准交易，超过 100kb 的称为非标准交易。它的前向 input 以及生成一个 out 约占用 161~250 bytes 。所以在比特币中，大约的 inputs/ouputs 的最大数目限制为 100KB/161B ~= 600 个。

UTXO 的特性及缺点

从计算的角度来说，UTXO 具有非常好的并行□支付能力，也就是我们上文中所说的如果没有尺寸限制，一笔交易可以包含任意笔输入输出，同时也没有次序要求，在一笔交易中哪一个 UTXO 在前，哪个在后面不影响最终结果。

从存储的角度来说，UTXO 具有较好的可裁剪特性，可裁剪性指的是 UTXO 类型的交易，如果从最老的那一笔 UTXO 开始截断数据库，那么之前的数据可以删除掉了。

如果想进一步压缩数据□尺寸，□可以在任意位置截断，记录 UTXO 对应的交易哈希即可，然后□从其他节点获取并校验 UTXO，这也是 SPV 轻钱包工作的基础之一。

以太坊中并没有使用比特币的这种 UTXO 设计，这与以太坊的宗旨有关，以太坊的目标是构建通用计算，而比特币是数字货币，需求不同导致设计的不同。

V 神指出了 UTXO 的缺陷，一共有三类。

1. 可表达的状态少 。

UTXO 只能是已花费或者未花费状态，这就没有给需要任何其它内部状态的多阶段合约或者脚本留出生存空间，这也意味着 UTXO 只能用于建立简单的、一次性的合约，UTXO 更像是一种二进制控制位。

2. 区块链盲点（Blockchain-blindness）。

UTXO 的脚本只能看到自己这条历史轨迹，无法看到区块链的数据的全貌，这导致了功能性扩展受到了限制，我们在花费比特币的过程中需要小心翼翼的组合 UTXO，这也导致了系统状态逻辑复杂，不适合设计成智能合约的基础结构。

3. 价值盲点（Value-blindness）。

UTXO 脚本不能提供非常精细的金额控制，基于账户模型的余额在花费过程中，可以任意的按值存取，它仅取决于程序能表示的最小精度。

而 UTXO 要求必须全部移动，如果要□满足一个目标值金额，对组合 UTXO 算法的要求会比较高，采用许多有不同面值的 UTXO，一方面要求尽可能地□精确，另一方面又要求输入输出的数量尽可能的小。

UTXO 是比特币上的原生设计，在区块链以前是没有这种逻辑数据结构，UTXO 的出现给了人们看待数据转移的不同视角，但 UTXO 不是所有区块链所必需的，公链开发过程中的是否选用 UTXO 模型可以根据业务场景进行判断。

总结

好了，今天我们分别介绍了普通账户模型和 UTXO 模型，并从不同角度比较了二者的优劣。

从技术选择上来看，比特币□选择 UTXO 是为了满足支付的安全性，以太坊选择普通账户模型是为了智能合约的自由度。

最后留给你一个问题，历史上 UTXO 或账户模型是否引发过比较严重的使用缺陷呢？你可以给我留言，我们一起讨论，感谢你的收听，我们下期再见。

上一篇 第15讲 | 深入区块链技术（七）：哈希与加密算法

下一篇 第17讲 | 去中心化与区块链交易性能

精选留言 (12)

写留言

Geek_05a05... 2

2018-05-03

最近以太坊的漏洞问题，估计V神自己也会发现UTXO的设计是有道理的

展开

Mrsong721 1

2018-05-16

老师你好。如果我的账户里有三条输入交易，分别为1，2，3，那么我现在要转给他人账户5.5，那么可以三个输入对应一个输出么？

作者回复: 可以的，剩下的0.5就会作为交易费付给矿工了

阿痕 1
2018-05-01

历史上账户模型产生的严重安全问题很多，一类是中心系统的bug，比如前几年发生的光大乌龙事件；另一类是内部腐败，银行内部人员偷偷修改账户的事件也不少见

展开

gopherliu
2018-10-31

你好，问个问题。矿工收到的交易费也要计算到整个的UTXO集合吗?

展开

Super~琪...
2018-08-29

有点不明白，数据库截断那里，原先的数据删除只保留古老的输入。那不会造成数据丢失吗?

作者回复: 不会，因为已经是共识了，只要重新生成创世哈希即可，重点在未花费的输出（UTXO）

活泼君
2018-07-17

在比特币中将小于 100kb 的交易称为标准交易，超过 100kb 的称为非标准交易。它的前向 input 以及生成一个 out 约占用 161~250 bytes 。所以在比特币中，大约的 inputs/ouputs 的最大数目限制为 100KB/161B ~= 600 个。
---请问这个kb等于KB吗?

作者回复: 你好，是的。

Hesher
2018-05-17

找零的那条输出，如果忘了，这部分找零会作为交易费用。那这个找零给自己的动作，一般是钱包构造好的吗？

作者回复: 是的，钱包提供的，一般都有找零，否则无法控制交易费

Mrsong721
2018-05-16

老师你好。如果我的账户里有三条输入交易，分别为1，2，3，那么我现在要转给他人账户5.5，那么可以三个输入对应一个输出么？

作者回复: 可以的，输出是5.5，剩下的0.5会被矿工拿走。

总书记
2018-05-07

UTXO 的脚本只能看到自己这条历史轨迹，无法看到区块链的数据的全貌，是指无法看到整个区块的货币用量？

作者回复: 对的，是指无法感知其他UTXO，所以组装交易只能用户的钱包来做，而账户模型则没有这个限制，可以是代码处理的。

FF
2018-05-02

如果想进一步压缩数据尺寸，可以在任意位置截断，只记录交易哈希即可
————————
截断后原输入输出是怎么找回的？任意截断不会导致断片的数据不完整吗

展开

gracegao
2018-05-02

讲的很透彻，受教！

展开

郑涛

2018-05-01

以太坊的代币算是账户模型的吗

展开

作者回复: 是的，账户模型。