架构师的六脉神剑-领域建模

P9工作法
分布式架构、上百人团队管理、全球支付实践与AI技术

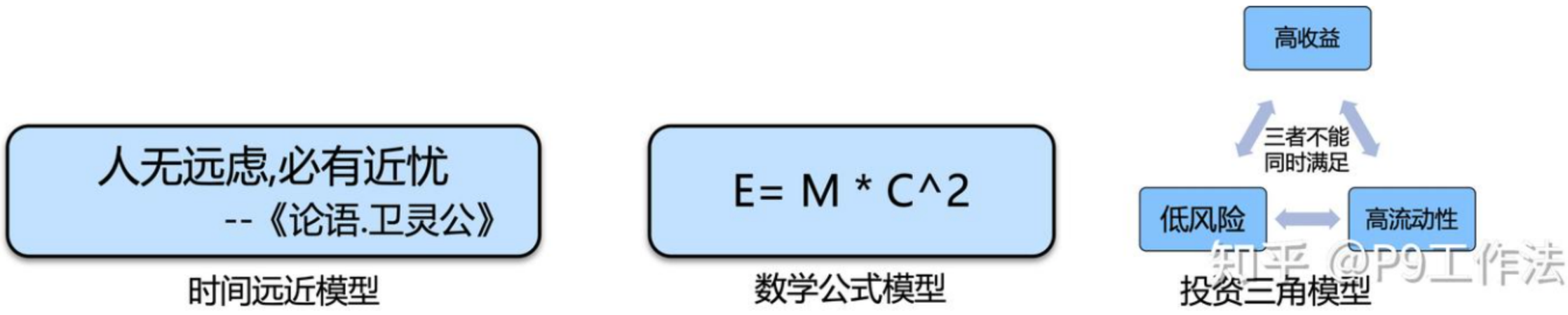
已关注

5 人赞同了该文章

在当下互联网分布式架构中，基于领域模型驱动做设计即DDD（Domain Driven Design）是解决软件复杂度的核心利器。但是实践DDD确实非常难的。领域模型不是一幅具体的图，也不是一个领域专家脑海中的知识，而是经过严格逻辑推导并选择性抽象出的知识，一旦掌握该方法就像是掌握内功心法一样，对业务的理解、结构设计、关键细节把控等就会愈加炉火纯青。

什么是领域建模

模型是经验的抽象集合，如谚语、公式、定理、方法等实际上都是模型，如下图。模型是我们理解世界的方式。例如，在文化领域，“人无远虑，必有近忧” 是模型:在数学领域，“正态分布”是模型；在科学领域，“爱因斯坦场方程” 是模型；在经济领域，“边际收益递减”是模型；在金融领域，“投资不可能三角” 是模型在计算机领域，架构模型、算法模型、网络协议模型等更是多得难以计数。

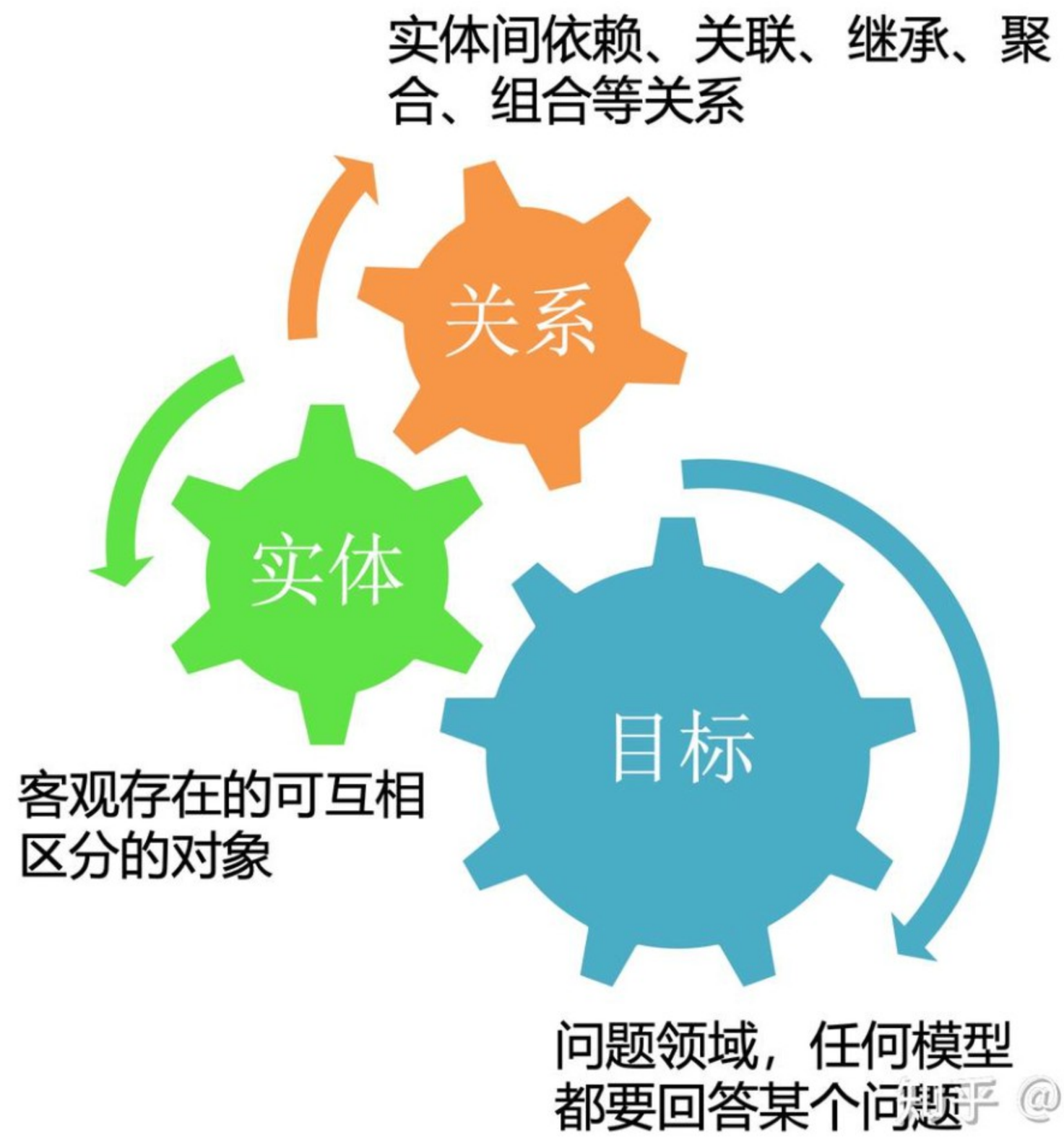


时间远近模型　　数学公式模型　　投资三角模型

模型是对现实世界的简化、抽象和形式化，是处理特定问题有价值经验的积累。所以领域建模的核心就是针对特定问题的所有相关方面的模型抽象，统一问题描述语言，抓住问题本质。将复杂网状事务进行结构化整理分析，建立实体间关系，定义实体结构和行为，准确表达业务含义的过程。

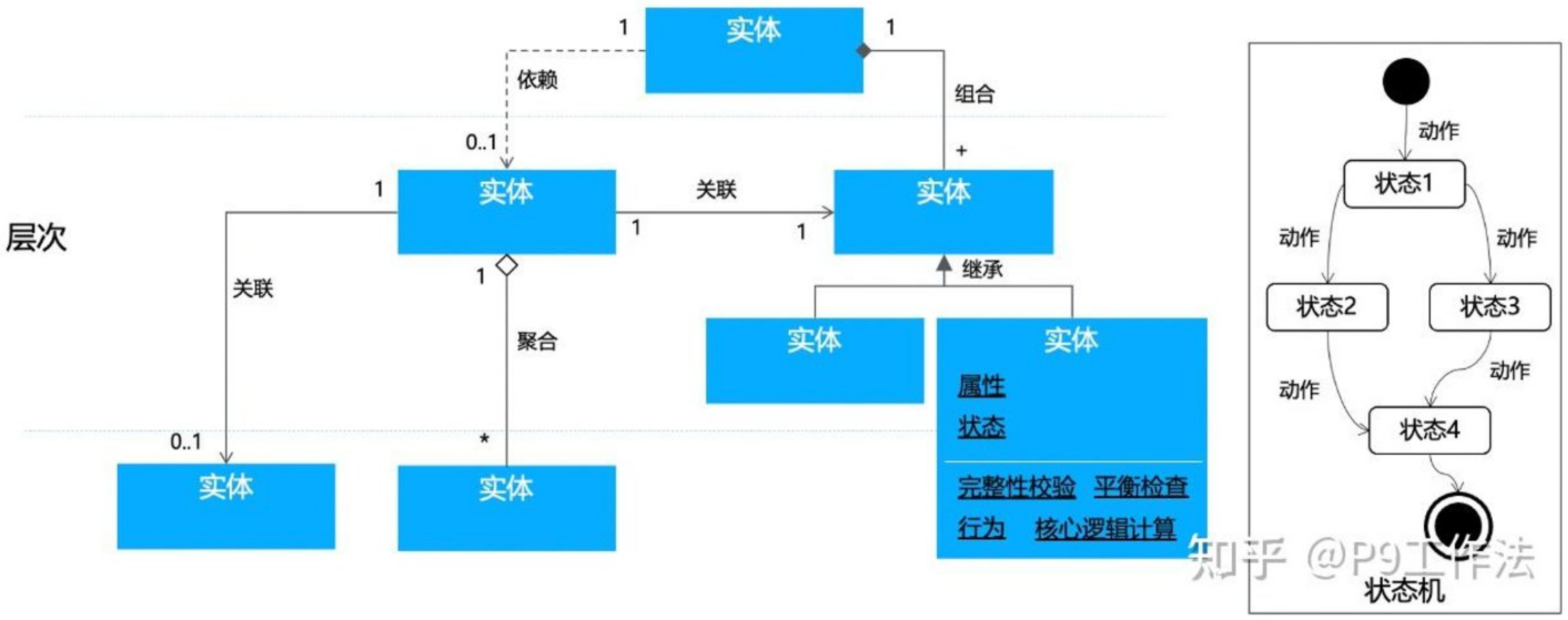
什么是领域模型

领域模型包含三个核心要素：目标、实体和关系。如下图所示，**目标**指的是问题域，即任何模型都是为了回答某个问题，这也意味着领域模型是有边界的。**实体**是客观存在的、可相互区分的对象，强调客观性，一方面是为了准确描绘客观世界，另一方面是因为只有真实存在的事物才能被精准认知。**关系**则是指实体之间的依赖、关联、继承、聚合、组合等。关系使得实体不再孤立，也使得实体成为更高层次有机体的组成部分。



实体构成必备的六大件

实体是领域模型的基本构成单元，其定义的准确性直接影响领域模型的质量。我们也许没办法一下子回答“某个实体对不对”的问题，但我们能够快速回答“某个实体坏不坏”的问题。如果实体没有包含全六大要素，那它大概率就是坏的。这六大要素包括属性、行为、状态机、完整性检查、平衡性检查，以及核心逻辑计算。

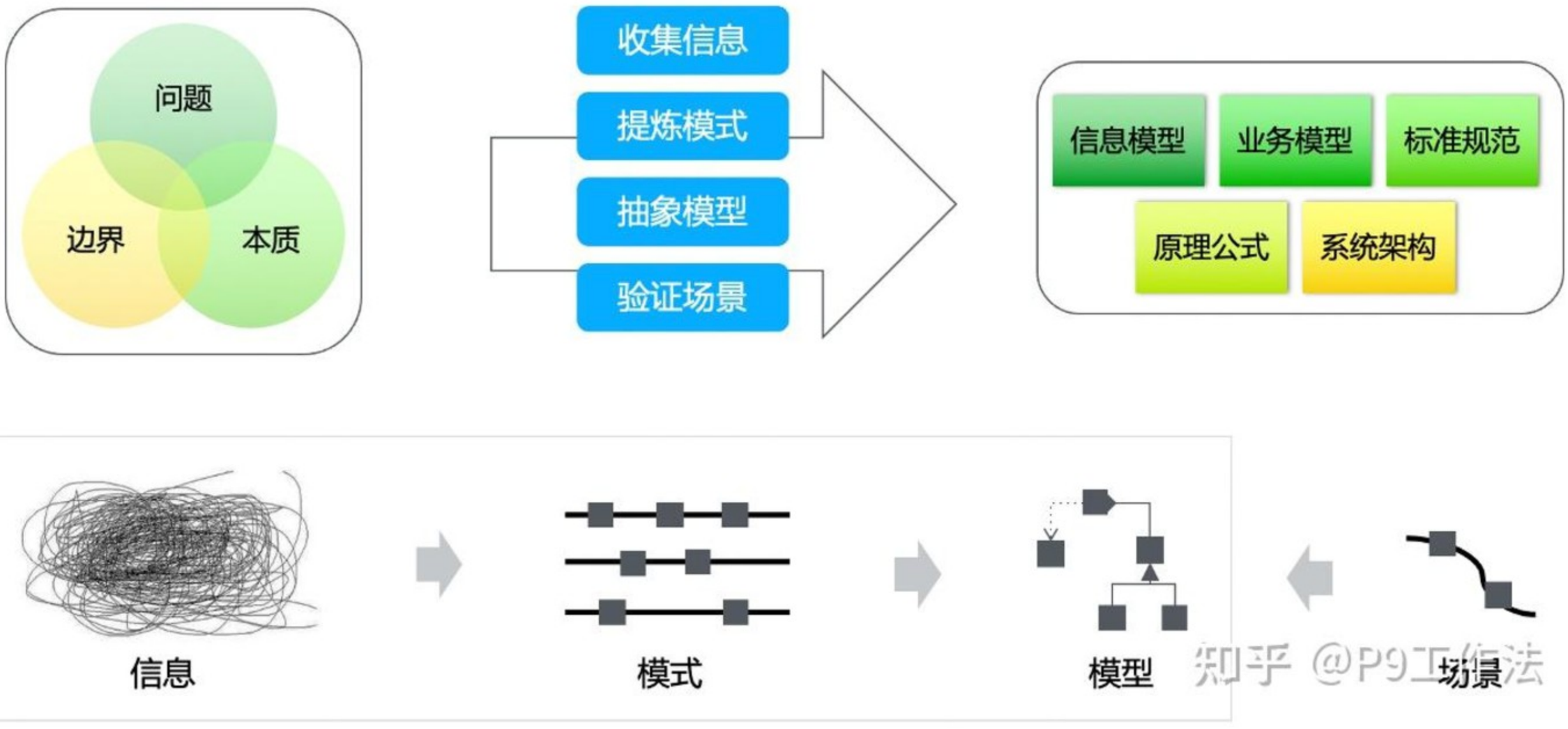


- **属性**：实体的特征值与真实世界相呼应，并使用业务语言进行描述。属性可以进一步分为自有属性和关联属性。自有属性定义实体自身的特征，而关联属性则定义实体与其他实体之间的关系。例如，汽车的自有属性包括颜色、尺寸等，而其关联属性则可能包括轮子、发动机等。
- **行为**：实体的行为应与真实世界相对应，并使用业务语言进行描述。只有明确了行为，实体才能响应问题并实现目标。由于实体需要对外界产生影响，因此行为特别强调可交互性和完整性。以汽车为例，汽车的行为包括启动、挂挡、 行驶、制动、关停等。这些行为是任何驾驶员都能理解和操作的，它们共同构成一个完整的行为闭环，缺一不可。
- **状态机**：这是实体的内在秩序，它决定了某个行为是否可以进行，以及行为完成后会发生什么变化。例如，未启动的汽车不允许挂挡，更不能行驶；已挂挡的汽车不能进入待行驶状态等。
- **完整性检查**：这是实体的行为保证，它确保行为的输入和输出的完整性、合法性、准确性等。例如，汽车行驶前必须确保有轮子，并且这些轮子都要被充满气。在汽车启动时，系统会对模型的关键点进行自检，以确保模型的完整性，才提供行驶服务。
- **平衡性检查**：这是实体的整体性保证。实体通常包含多个属性，并关联多个其他实体。作为一个有机整体，实体需要能够检查自身是否处于正确状态。例如，档位与车速之间必须满足特定的匹配关系。
- **核心逻辑计算**：这是实体的内核。实体的性能取决于这一内核。例如，汽车的动力系统驱动就是其内核之一。

领域建模四招大法

领域建模是一个过程，首先针对特定问题的所有相关方面进行抽象，其次， 统一问题描述语言，以抓住问题的本质。最后，将复杂网状的事务进行结构化整理， 建立实体间的关系，定义实体的结构和行为，以准确地表达业务含义。

尽管不同的人面临的问题和环境各不相同，但领域建模的基本方法是普遍适用的。。我将这些方法定义为领域建模的普适四招，如下图。



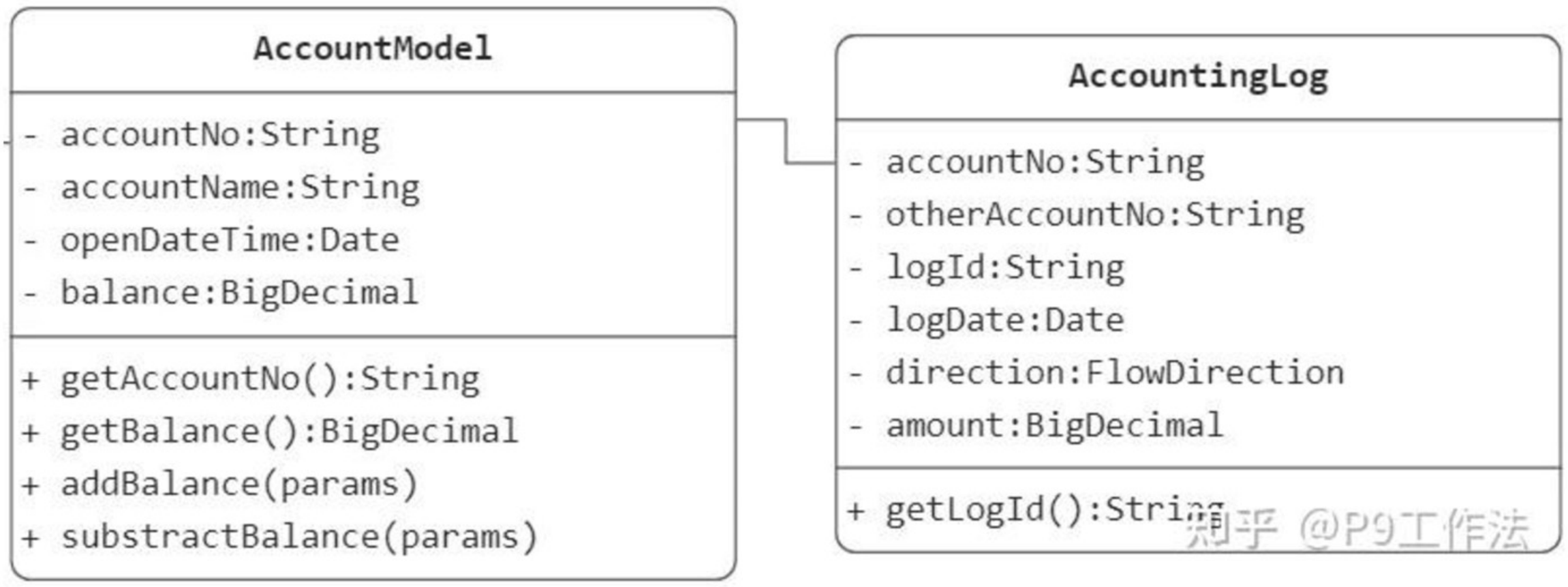
第一招：收集信息

有人将认知分为四个层次:数据、信息、知识和智慧。数据是原始的、未经加工的事件、经历和现象，它们缺乏意义、组织和结构。信息是经过分类、结构化、有目的和加工处理的数据。知识是模型化的、有组织的信息，能够解释和预测现象。智慧是对知识的选择性应用，懂得在多种可能性中选择最佳模型。我们的第一招是从问题出发，收集信息。这个过程包括以下五个步骤。

1. **明确问题：**问题的范围可以大，也可以小，但必须真实存在。
2. **描述问题：**收集与问题相关的所有数据，力求全面。
3. **定义信息：**在确认数据完备的基础上，对数据进行分类，并从多个维度和视 角进行定义。关键在于确保数据在同一维度上的统一性。
4. **统一信息：**确保信息的标准化，与实际业务一致，并采用统一的表达方式。
5. **还原问题：**通过将问题重新呈现，检查信息是否能够清晰地描述问题，这是保证信息逻辑自洽的有效方法。

以设计账务记账系统为例，缺乏账户模型会导致多种问题。我们可以从问题出发，明确模型的关键属性。例如，缺乏账户模型可能导致账户余额的记录不准确，需要通过记账明细进行汇总才能得出余额；确定资金变动的相关方也可能变得困难。通过解决这些问题，我们可以明确账户模型的关键属性，包括账号、余额、记账明细，以及明细中的时间戳、对方账户等信息，如下图。

AccountModel 是账户的基础模型，AccountingLog 是记账明细模型。



第二招：实例推演，提炼模式

围绕具体问题，从问题的实例出发，观察实例中各参与者的核心特征和行为。通过对这些特征和行为的归类，可以提炼出具有普遍适应性、稳定性和可操作性的模式。在提炼模式时，应从多个维度进行展开，抓住关键点，避免只看到局部而忽视整体。

模式提炼完成后，一种有效验证方法是与多方进行对话，并尝试将新模式应 用于新的案例中进行验证。以账户模型为例，在运行过程中可能会发现存在热点账户问题，即某个账户由于频繁记账而成为性能瓶颈。通过对系统中所有存在账户热点的账户的特征和行为进行分析，可以找到突破点。

从记账金额来归类，

1. 一些账户可能表现出频繁一进一出，但在日终时，其实际余额并没有显著变化。
2. 另一些账户以频繁的资金流入为主要特征，资金流出可能表现为周期较长的大额流出，总体上流出金额小于流入金额。
3. 还有一些账户的记账模式不规律，部分时段流入金额大于流出金额， 而其他时段则相反。

从账户属性来归类:

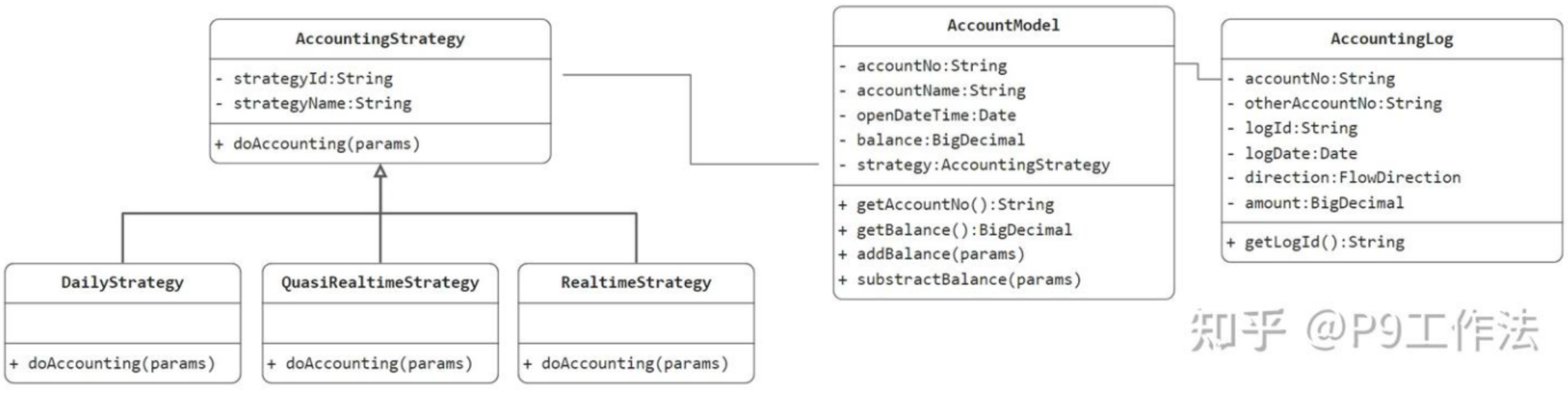
1. 记账频率极高的账户，通常是值得信赖的大客户或用于内部资金管理的账户。这类账户具备一定的资金安全保障。
2. 大多数账户属于记账频率一般的普通账户。

通过整合记账金额归类和账户属性归类这两个维度的信息，可以推演出一个可能的操作模式：记账未必完全需要实时更新余额。对于某些场景下的特定账户， 可以采取延迟更新余额的策略，以在余额准确性和记账效率之间找到平衡点。

第三招：结构分析，模型抽象

模式提炼后，我们对要回答的问题就有了基本认知。从模式的参与者互动中，我们可以识别出实体，以及它们之间的关系，并构建实体层次结构。在筛选实体时，应选择那些最接近真实、最简单的内容。找到那些稳定不变、必需的元素，并遵循“如无必要，勿增实体”的原则。针对上述账户模型，我们可以进一步结构化并针对记账时效，抽象定义以下模型：

- **实时记账模型**：这是所有账户默认遵循的记账策略，风险最低，但记账性能较差。
- **准实时记账模型**：针对账户资质较好的账户，采用准实时记账策略，平衡记账风险和记账性能。
- **日末记账模型**：针对内部资金管理类账户，采用日终汇总记账策略，以实现 记账性能的最大化，但这种策略风险较高，因此必须严格控制。



第四招：真实还原，验证场景

领域模型建立后，具备了解释真实世界的能力。以提炼出的模式为指导，从真实案例出发，用领域模型演绎多种场景。场景可以具有一定的扩展性。关键在于在边界范围内尽可能全面地覆盖。

在完成新的模型抽象后，需要重新回到真实的业务场景中，从源头验证领域建模的准确性，避免陷入纯逻辑推演。将记账策略模型应用于真实场景时，需要 从策略配置的源头进行论证:

- 谁在何时配置记账策略?
- 谁来判断哪些账户适合哪种记账策略?
- 如何防范记账策略配置错误带来的风险?

从这些源头进行论证后，我们会发现仅拥有核心模型是不够的，还需要有配套的配置模型来规避风险。如图通过引入配置模型来判断每个账号的适用记账策略。配置模型中记录当前账号的记账策略生效或失效状态，并通过记录表来记录操作人和变更历史，确保变更记录的完整性。同时，在激活记账策略时，通过账户模型中的账户类型来统计对应类型的历史动账记录规律，判断是否符合当前策略，从而实现记录变更和策略生效的检查。