

众包供应链知识平台研究

章茜 邵鹏飞 吴耀辉*

浙江万里学院 信息与智能工程学院, 浙江宁波, 315100;

摘要: 众包供应链面临数据安全和隐私保护挑战, 本文通过构建本体模型和运用语义规则优化管理。研究利用 Protege 平台构建模型明确概念关系, 用 SWRL 写规则描述约束, 再以 OWL 语言设限制条件, 设计知识融合系统保障数据安全透明、实现知识融合自动化公正。该研究解决传统众包供应链数据安全信任问题, 高效共享融合知识, 提升供应链管理效率和可靠性, 为管理提供新思路方法。

关键词: 众包供应链; Protege 平台; 本体模型; 知识融合

DOI:10. 69979/3041-0673. 25. 02. 057

1 引言

1.1 研究背景

传统众包模式通过整合分布式资源提升任务执行效率, 但其依赖中心化平台的管理方式存在显著缺陷: 数据存储与传输环节易受攻击导致隐私泄露风险, 且平台可信度高度依赖运营方, 存在单点信任危机^[1]。近年来, 区块链技术凭借去中心化、可追溯性等特性, 为构建透明、安全的供应链协作机制提供了新思路, 但其在动态知识融合与语义一致性保障方面的潜力尚未充分挖掘。

1.2 研究现状

在本体推理与规则验证方面, 陈瑞翔等^[2]提出基于本体推理的智能合约漏洞检测方法, 验证了语义规则在自动化决策中的有效性, 但研究聚焦安全领域, 未扩展至供应链协作场景。

吉原^[3]系统梳理了区块链众包平台的技术框架, 指出现有系统多关注任务分发效率, 缺乏对多源知识冲突的融合机制。在去中心化知识融合实践方面, 朱向荣团队^[4]开发的 FactChain 系统通过智能合约实现跨组织知识共享, 但其动态约束规则的设计局限于粗粒度资源管理, 难以适配众包供应链的细粒度协作需求。

2 关键技术介绍

2.1 使用 SWRL 建立规则

SWRL (Semantic Web Rule Language) 是一种用于在语义网上表达规则的语言, 可以与 OWL 本体相结合。在众包供应链知识本体中, 可以使用 SWRL 编写规则来描述接包方、发包方、包、裁判等之间的复杂关系和约

束条件。例如, 可以编写规则来表示“如果接包方的信誉评分低于某个阈值, 则不能承接某个特定的包”。

2.2 使用 OWL 语言设置限制

OWL (Web Ontology Language) 是一种用于定义本体的语言, 可以用来设置类的限制条件。在众包供应链知识本体中, 可以使用 OWL 设置接包方、发包方、包、裁判等类的限制条件, 如“接包方必须具有至少一个成功的项目经验”、“发包方的预算必须大于某个特定的金额”等。

3 众包供应链本体平台设计

3.1 Protege 平台构建

Protege, 作为一款开源的本体编辑和知识管理平台, 由斯坦福大学精心打造。在语义网、知识图谱和数据集成等领域, Protege 得到了广泛的应用。

首先, Protege 支持创建和编辑 OWL (Web Ontology Language) 本体。平台提供了类层次结构、对象属性、数据属性和注释的可视化管理, 使用户能够直观地构建和维护本体。

其次, Protege 集成了多种推理引擎, 如 Hermit、Pellet 和 FaCT++, 能够进行逻辑推理和一致性检查。这些推理引擎帮助用户自动推导出隐含的类关系和个体关系, 从而发现和纠正本体中的潜在错误。这种推理支持提高了本体的准确性和完整性。

最后, Protege 还提供了知识可视化功能。通过类层次图、关系图和数据图等多种可视化工具, 用户可以直观地理解和分析本体的结构。同时, Protege 还支持与外部可视化工具的集成, 如 Graphviz 和 OntoGraf,

为用户提供更丰富的可视化表达形式。

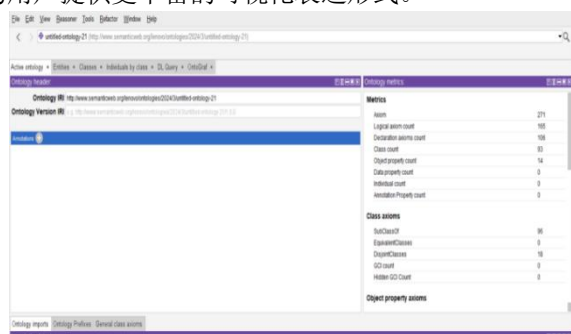


图 1 protege 平台

3.2 系统构架

要构建一个众包知识融合系统，其核心架构由众包机制以及知识融合功能共同支撑。其中，众包模式作为数据获取的关键手段，为系统提供了丰富的数据源；而知识融合则是该系统所承载的核心功能。

在架构中，参与人作为组织的核心成员，扮演着多重角色。他们不仅是众包任务的执行者，还是知识的积极贡献者和交互的主动发起者。参与人可以通过分享自身的知识来换取收益，同时，也可以通过支付报酬来获取所需的知识。在系统中，参与人能够访问组织内部的链下知识、链上共享的知识以及他们自身的信用额度，但知识融合的具体流程则保持透明化。

组织作为连接区块链和参与人的关键桥梁，由分布式应用、区块链对等节点（peer node）和链下存储等模块构成。组织通过分布式应用来管理参与人的身份，并控制他们与系统交互的权限，确保对参与人的交互请求做出及时响应。

区块链则是架构的基石，它由链上存储模块和智能合约两大核心组件组成。链上存储模块承担着储存和管理组织间共享知识的重任，通过先进的密码学技术确保知识的可溯源性和不可篡改性。

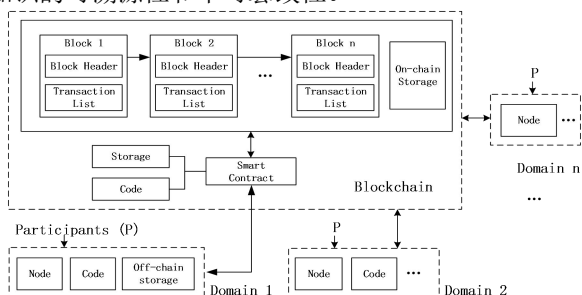


图 2 众包平台构架

3.3 本体模型

3.3.1 本体模型构建

类定义：

Worker：表示工人类。

Judge：表示裁判类。

Domain：表示领域类。

IncompatibleAssignment：表示不兼容的任务分配类。

属性定义：

hasWorker：对象属性，表示领域包含的工人。

hasJudge：对象属性，表示领域包含的裁判。

3.3.2 推理规则定义

定义 SWRL 规则 Judge_Cannot_Evaluate_Worker_Domain，用于确保裁判不能评判工人所属的领域，以下是部分代码：

```
<swrl:Rule>
  <swrl:head>
    <swrl:ClassAtom>
      <swrl:IndividualVariable name="
judge"/>
      <swrl:Class IRI="#IncompatibleA
ssignment"/>
    </swrl:ClassAtom>
  </swrl:head>
  <swrl:body>
    <swrl:ClassAtom>
      <swrl:IndividualVariable name="
worker"/>
      <swrl:Class IRI="#Worker"/>
    </swrl:ClassAtom>
    <swrl:ClassAtom>
      <swrl:IndividualVariable name="
domain1"/>
      <swrl:Class IRI="#Domain"/>
    </swrl:ClassAtom>
    <swrl:ObjectPropertyAtom>
      <swrl:IndividualVariable name="
domain1"/>
      <swrl:ObjectProperty IRI="#hasW
orker"/>
      <swrl:IndividualVariable name="
worker"/>
    </swrl:ObjectPropertyAtom>
  </swrl:body>
</swrl:Rule>
```

3.3.3 规则解释

前提条件:

autogen0:Worker(?worker): 表示 ?worker 是一个 Worker。

autogen0:Domain(?domain1): 表示 ?domain1 是一个 Domain。

autogen0:hasWorker(?domain1, ?worker): 表示 ?domain1 有一个 Worker。

autogen0:Judge(?judge): 表示 ?judge 是一个 Judge。

autogen0:hasJudge(?domain1, ?judge): 表示 ?domain1 有一个 Judge。

结论:

autogen0:IncompatibleAssignment(?judge): 表示将 ?judge 添加到

IncompatibleAssignment 类中。

Name
Judge_Cannot_Evaluate_Worker_Domain
Comment
此规则确保裁判不能评判工人所属的领域
Status
Ok

autogen0:Worker(?worker) ^ autogen0:Domain(?domain1) ^ autogen0:hasWorker(?domain1, ?worker) ^ autogen0:Judge(?judge) ^ autogen0:hasJudge(?domain1, ?judge) -> autogen0:IncompatibleAssignment(?judge)

3.3.4 实验结果

为了验证裁判不能评判工人所属领域的推理过程, 我们设计了以下实验数据:

工人实例:

Worker_SensorExpert01: 属于 Domain2。

裁判实例:

Judge_Domain2: 属于 Domain2。

领域实例:

Domain2: 包含 Worker_SensorExpert01 和 Judge_Domain2。



构建本体模型: 使用 Protege 构建 OWL 本体模型, 定义类、属性和推理规则。

设置实例: 在本体中添加工人实例、裁判实例和领域实例, 设置 hasWorker 和 hasJudge 属性。

运行推理器: 使用 Protege 的 Hermit 推理器进

行推理。

验证结果: 检查推理结果, 验证裁判是否被标记为不兼容的任务分配。

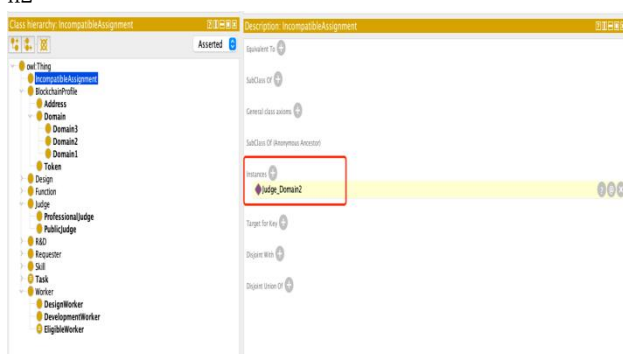
准确率: 正确标记的不兼容任务分配数量占总任务分配数量的比例。

召回率: 正确标记的不兼容任务分配数量占实际不兼容任务分配数量的比例。

F1 值: 准确率和召回率的调和平均值。

实验结果:

IncompatibleAssignment 类的实例: Judge_Domain2



参考文献

- [1] 黎继子, 刘清华, 陆芬等. 区块链背景下的众包供应链决策行为分析研究[J]. 物流工程与管理, 2023, 45(1): 41-47.
- [2] 陈瑞翔, 焦健, 王若华. 基于本体推理的智能合约漏洞检测系统[J]. 计算机科学, 2023, 50(10): 336-342.
- [3] 吉原, 蒋凌云. 基于区块链的众包系统研究综述[J]. 软件工程, 2023, 26(12): 1-5. 10.19644/j.cnki.issn2096-1472.2023.012.001.
- [4] 朱向荣, 吴鸿祜, 胡伟. FactChain: 一个基于区块链的众包知识融合系统[J]. 软件学报, 2022, 33(10): 3546-3564. 10.13328/j.cnki.jos.006627.

作者简介: 章茜 (2003-), 女, 浙江万里学院信息与智能工程学院学生, 主要研究方向为供应链本体
邵鹏飞 (1978-), 男, 教授, 主要研究方向为数字化供应链

基金项目: 浙江省自然科学基金资助项目 (LY23G020007); 2023 年浙江省大学生科技创新活动计划暨新苗人才计划 (2023R419013)

通信作者: 吴耀辉 (1979-), 男, 副教授, 主要研究方向为供应链管理与知识管理,