(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104217146 A
(43)申请公布日 2014.12.17

(21)申请号 201410447978.2

(22)申请日 2014.09.04

(71)申请人 浪潮通用软件有限公司
地址 250101 山东省济南市高新区舜雅路1036号

(72)发明人 刘建华

(74)专利代理机构 济南信达专利事务所有限公司 37100
代理人 姜明

(51)Int.Cl.
G06F 21/31(2013.01)
G06F 21/62(2013.01)

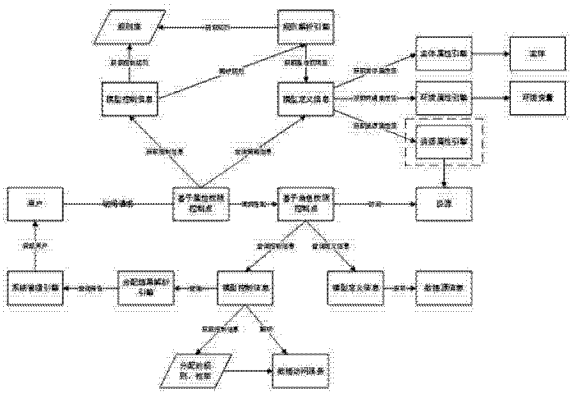
权利要求书2页 说明书6页 附图2页

(54)发明名称

一种基于ABAC和RBAC的权限控制方法

(57)摘要

本发明公开了一种基于ABAC和RBAC的权限控制方法，本方法基于ABAC和RBAC的权限控制，能够实现权限的动态控制和不同粒度的访问控制，尤其适用于业务较复杂的系统中对于资源的权限控制。该控制方法不仅能够更好的利用ABAC和RBAC的自身的优势，而且能够将两者结合起来，按照定义的控制信息，灵活的对资源进行不同粒度的控制，以满足用户对资源的不同粒度的权限控制。尤其适用于比较复杂的业务系统中对资源的访问控制，能够比较圆满的满足系统的权限控制需求。



CN 104217146 A

1. 一种基于 ABAC 和 RBAC 的权限控制方法，其特征在于包括以下步骤和内容：

步骤一：权限模型的实体类定义，实体类的定义的信息包括基本信息、控制项信息、分配结果表信息和维度信息；

步骤二：权限模型控制触发点设置，基于步骤一定义的实体类，将该实体类与需要控制业务资源对象关联起来，即设置权限模型控制的触发点；

步骤三：权限模型控制条件设置，根据步骤一定义的模型实体类，获取定义的控制项信息，然后对定义的各个控制项，设置对应控制的条件，并将设置的条件持久化，以便后面步骤调用；

步骤四：权限模型访问控制的调用，当用户访问某个业务资源时，如果该业务资源对象经过步骤二的设置，则会根据步骤二设置的关联关系，获取定义的实体类信息，然后根据实体类信息，获取步骤三定义的模型控制的条件信息；

步骤五：权限模型控制条件的解析，将步骤四获取出来的控制条件，按照类型不同，调用不同的条件解析器进行解析，然后将结果返回；

步骤六：根据返回结果，访问资源；

根据步骤五解析的模型控制条件结果，对资源进行控制访问。

2. 根据权利要求 1 所述基于 ABAC 和 RBAC 的权限控制方法，其特征在于：步骤一，定义权限模型的实体类，除了包括定义的基本信息之外，还需定义控制项信息，对于基于属性权限对象来说，则需定义属性控制项集合 AttributeItemCollection，里面包括了定义的属性的项名称以及该项对应的属性资源的类型，包括主体属性和环境变量；对于基于角色的权限对象来说，则需定义对应的数据源信息 DataSourceCollection，里面包括数据源的编号、类型以及分配时的设置信息。

3. 根据权利要求 1 所述基于 ABAC 和 RBAC 的权限控制方法，其特征在于：步骤二，设置权限模型控制触发点，选取待控制的业务资源对象，并将定义的模型与业务资源对象关联起来，对于基于属性权限对象，主要是将定义的模型信息与业务资源对象关联 ResAttributeRelation；对于基于角色的权限对象，除了定义的模型信息与业务资源对象关联信息 ResourceRelation 之外，还需要定义相关对应的关联字段信息 ElementMappingCollection。

4. 根据权利要求 1 所述基于 ABAC 和 RBAC 的权限控制方法，其特征在于：步骤三，设置权限模型控制条件，对于基于属性的权限对象，主要根据定义的属性项信息，设置该属性项对应的条件表达式；对于基于角色的权限对象，则根据定义的控制信息，定义模型控制的条件信息，包括规则或者枚举。

5. 根据权利要求 1 所述基于 ABAC 和 RBAC 的权限控制方法，其特征在于：步骤四到步骤六所述的权限模型访问控制的调用、条件解析，其主要的步骤包括以下几步：

(1) 用户请求对资源进行访问，触发了权限模型的控制；

(2) 获取基于属性权限对象的控制信息 ResAttributeRelation，如果该资源不控制属性权限，则直接跳过属性权限的控制；如果控制属性权限，则继续执行以下步骤；

(3) 获取属性权限的定义信息 AttributePermission，属性权限的定义信息，包括定义的属性项类型和属性项名称；

(4) 获取属性权限的控制信息 AttributeAssignment，首先获取上一步定义的属性项信

息，然后根据获取的属性项，获取每个属性项设置的控制信息；

（5）解析属相权限的控制信息，根据上一步获取的控制信息，调用属性权限解析引擎，将控制信息进行解析，将结果返回；根据返回的布尔结果，如果是 FALSE，则拒绝访问资源对象；如果是 TRUE，则继续执行后面的步骤；

（6）获取基于角色权限对象控制信息 ResourceRelation，如果不存在控制信息，则直接访问资源对象；如果存在控制信息，则继续下面的步骤；

（7）获取基于角色权限对象的定义信息 PermissionObject，定义的信息主要包括承载的数据源以及分配结果表设置的信息；

（8）获取基于角色权限对象的控制信息 DataAssignment，根据前面步骤定义的信息，获取对应的控制信息；

（9）解析基于角色权限对象的控制信息，交给基于角色权限模型解析引擎和数据访问引擎，对获取的数据进行条件过滤，然后根据过滤之后的结果，对资源进行访问。

一种基于 ABAC 和 RBAC 的权限控制方法

技术领域

[0001] 本发明公开了一种基于 ABAC 和 RBAC 的权限控制方法，涉及计算机资源访问控制领域，具体涉及复杂业务系统中对资源的访问相关控制。

背景技术

[0002] 访问控制作为一种重要的安全措施在系统安全中得到广泛的用用，在针对访问控制的研究中产生了各种不同的访问控制模型，这些模型的目标是禁止未授权用户访问资源。访问控制核心是访问控制策略和基于策略的授权判定，访问控制策略描述了系统的安全需求，访问控制模型主要研究的是访问控制策略的表示，而访问控制策略自身是否安全以及其是否能够真实、及时的反应实际安全需求则直接影响整个系统的安全性和用户对系统的满意度，为了满足复杂系统对访问控制模型的需求。

[0003] 传统的访问控制模型，如自主访问控制 DAC(Discretionary Access Control)、强制访问控制 MAC(Mandatory Access Control)、基于角色的访问控制 RBAC(Role Based Access Control) 和基于属性的访问控制模型 ABAC(Attribute Based Access Control) 等，并不能完全适用现今对资源的访问控制的要求，他们不能表示复杂的场景对资源的访问控制策略，对主体和资源的描述都比较片面，而且往往无法满足对访问资源控制的需求。

[0004] 本文针对以上问题提出了一种有效的解决方法。

发明内容

[0005] 本发明的目的是提供一种基于 ABAC 和 RBAC 的权限控制方法。

[0006] 本发明的目的是按以下方式实现的，包括以下步骤和内容：

步骤一：权限模型的实体类定义，实体类的定义的信息包括基本信息、控制项信息、分配结果表信息和维度信息；

步骤二：权限模型控制触发点设置，基于步骤一定义的实体类，将该实体类与需要控制业务资源对象关联起来，即设置权限模型控制的触发点；

步骤三：权限模型控制条件设置，根据步骤一定义的模型实体类，获取定义的控制项信息，然后对定义的各个控制项，设置对应控制的条件，并将设置的条件持久化，以便后面步骤调用；

步骤四：权限模型访问控制的调用，当用户访问某个业务资源时，如果该业务资源对象经过步骤二的设置，则会根据步骤二设置的关联关系，获取定义的实体类信息，然后根据实体类信息，获取步骤三定义的模型控制的条件信息；

步骤五：权限模型控制条件的解析，将步骤四获取出来的控制条件，按照类型不同，调用不同的条件解析器进行解析，然后将结果返回；

步骤六：根据返回结果，访问资源。根据步骤五解析的模型控制条件结果，对资源进行控制访问。

[0007] 所述基于 ABAC 和 RBAC 的权限控制方法，步骤一，定义权限模型的实体类，除了包

括定义的基本信息之外，还需定义控制项信息，对于基于属性权限对象来说，则需定义属性控制项集合 AttributeItemCollection，里面包括了定义的属性的项名称以及该项对应的属性资源的类型，包括主体属性和环境变量；对于基于角色的权限对象来说，则需定义对应的数据源信息 DataSourceCollection，里面包括数据源的编号、类型以及分配时的设置信息。

[0008] 所述基于 ABAC 和 RBAC 的权限控制方法，步骤二，设置权限模型控制触发点，选取待控制的业务资源对象，并将定义的模型与业务资源对象关联起来，对于基于属性权限对象，主要是将定义的模型信息与业务资源对象关联 ResAttributeRelation；对于基于角色的权限对象，除了定义的模型信息与业务资源对象关联信息 ResourceRelation 之外，还需要定义相关对应的关联字段信息 ElementMappingCollection。

[0009] 所述基于 ABAC 和 RBAC 的权限控制方法，步骤三，设置权限模型控制条件，对于基于属性的权限对象，主要根据定义的属性项信息，设置该属性项对应的条件表达式；对于基于角色的权限对象，则根据定义的控制信息，定义模型控制的条件信息，包括规则或者枚举。

[0010] 所述基于 ABAC 和 RBAC 的权限控制方法，步骤四到步骤六所述的权限模型访问控制的调用、条件解析，其主要的步骤包括以下几步：

1) 用户请求对资源进行访问，触发了权限模型的控制；

2) 获取基于属性权限对象的控制信息 ResAttributeRelation，如果该资源不控制属性权限，则直接跳过属性权限的控制；如果控制属性权限，则继续执行以下步骤；

3) 获取属性权限的定义信息 AttributePermission，属性权限的定义信息，包括定义的属性项类型和属性项名称；

4) 获取属性权限的控制信息 AttributeAssignment，首先获取上一步定义的属性项信息，然后根据获取的属性项，获取每个属性项设置的控制信息；

5) 解析属相权限的控制信息，根据上一步获取的控制信息，调用属性权限解析引擎，将控制信息进行解析，将结果返回；根据返回的布尔结果，如果是 FALSE，则拒绝访问资源对象；如果是 TRUE，则继续执行后面的步骤；

6) 获取基于角色权限对象控制信息 ResourceRelation，如果不存在控制信息，则直接访问资源对象；如果存在控制信息，则继续下面的步骤；

7) 获取基于角色权限对象的定义信息 PermissionObject，定义的信息主要包括承载的数据源以及分配结果表设置的信息；

8) 获取基于角色权限对象的控制信息 DataAssignment，根据前面步骤定义的信息，获取对应的控制信息；

9) 解析基于角色权限对象的控制信息，交给基于角色权限模型解析引擎和数据访问引擎，对获取的数据进行条件过滤，然后根据过滤之后的结果，对资源进行访问。

[0011] 本发明的优异效果：本发明公开了一种基于 ABAC 和 RBAC 的权限控制方法，能够实现权限的动态控制和不同粒度的访问控制，尤其适用于业务较复杂的系统中对于资源的权限控制。该控制方法不仅能够更好的利用 ABAC 和 RBAC 的自身的优势，而且能够将两者结合起来，按照定义的控制信息，灵活的对资源进行不同粒度的控制，以满足用户对资源的不同粒度的权限控制。

附图说明

[0012] 图 1 是 权限控制模型总体框架图；

图 2 是 权限模型访问控制总体流程图；

图 3 是 权限模型控制流程图。

具体实施方式

[0013] 参照说明书附图对本发明的一种基于 ABAC 和 RBAC 的权限控制方法作以下详细地说明。

[0014] 本发明的目的在于弥补现有的单个权限控制模型自身存在的缺憾，以满足复杂业务系统中对业务资源的控制。ABAC 模型考虑主体、资源和访问所处环境的属性来描述策略，相比于传统的访问控制策略，能够解决复杂信息系统中的粗粒度访问控制，但是无法满足用户对于细微粒度权限的访问控制。而 RBAC 通过“角色”建立起用户和资源的关系，定义了细微粒度的权限控制信息，但是无法定义粗粒度的访问控制信息，存在着一定的局限性。

[0015] 为此，本发明公开了一种基于 ABAC 和 RBAC 的权限控制方法，该方法主要由权限模型定义和权限模型控制两部分组成。所述基于 ABAC 和 RBAC 的权限控制方法的步骤如下：

步骤一：权限模型的实体类定义。实体类的定义的信息包括基本信息、控制项信息、分配结果表的信息和维度信息；

步骤二：权限模型控制触发点设置。基于步骤一定义的实体类，将该实体类与需要控制业务资源对象关联起来，即设置权限模型控制的触发点；

步骤三：权限模型控制条件设置。根据步骤一定义的模型实体类，获取定义的控制项信息，然后对定义的各个控制项，设置对应控制的条件，并将设置的条件持久化，以便后面步骤调用；

步骤四：权限模型访问控制的调用。当用户访问某个业务资源时，如果该业务资源对象经过步骤二的设置，则会根据步骤二设置的关联关系，获取步骤一定义的实体类信息，然后根据实体类信息，获取步骤三定义的模型控制的条件信息；

步骤五：权限模型控制条件的解析。将步骤四获取出来的控制条件，按照类型不同，调用不同的条件解析器进行解析，然后将结果返回；

步骤六：根据返回结果，访问资源。根据步骤五解析的模型控制条件结果，对资源进行控制访问。

[0016] 优选的是，所述的权限模型的实体类定义，除了包括定义的标识、编号等基本信息之外，还需定义控制项信息。对于基于属性权限对象来说，则需定义属性控制项集合(AttributeItemCollection)，里面包括了定义的属性的项名称以及该项对应的属性资源的类型(包括主体属性和环境变量)；对于基于角色的权限对象来说，则需定义对应的数据源信息 (DataSourceCollection)，里面包括数据源的编号、类型以及分配时的一些相关设置信息。

[0017] 优选的是，所述的权限模型控制触发点设置，选取待控制的业务资源对象，并将定义的模型与业务资源对象关联起来。对于基于属性权限对象，主要是将定义的模型信息与业务资源对象关联 (ResAttributeRelation)；对于基于角色的权限对象，除了定义的模型

信息与业务资源对象关联信息 (ResourceRelation) 之外，还需要定义相关对应的关联字段信息 (ElementMappingCollection)。

[0018] 优选的是，所述的权限模型控制条件设置，对于基于属性的权限对象，主要根据定义的属性项信息，设置该属性项对应的条件表达式；对于基于角色的权限对象，则根据定义的控制信息，定义模型控制的条件信息(规则或者枚举)。

[0019] 优选的是，所述的权限模型访问控制的调用、条件解析，其主要的步骤包括以下几步：

1、用户请求对资源进行访问，触发了权限模型的控制；

2、获取基于属性权限对象的控制信息 (ResAttributeRelation)。如果该资源不控制属性权限，则直接跳过属性权限的控制；如果控制属性权限，则继续执行以下步骤；

3、获取属性权限的定义信息 (AttributePermission)。属性权限的定义信息，包括定义的属性项类型和属性项名称；

4、获取属性权限的控制信息 (AttributeAssignment)。首先获取上一步定义的属性项信息，然后根据获取的属性项，获取每个属性项设置的控制信息；

5、解析属相权限的控制信息。根据上一步获取的控制信息，调用属性权限解析引擎，将控制信息进行解析，将结果返回；根据返回的布尔结果，如果是 FALSE，则拒绝访问资源对象；如果是 TRUE，则继续执行后面的步骤；

6、获取基于角色权限对象控制信息 (ResourceRelation)。如果不存在控制信息，则直接访问资源对象；如果存在控制信息，则继续下面的步骤；

7、获取基于角色权限对象的定义信息 (PermissionObject)。定义的信息主要包括承载的数据源以及分配结果表设置的信息；

8、获取基于角色权限对象的控制信息 (DataAssignment)。根据前面步骤定义的信息，获取对应的控制信息；

9、解析基于角色权限对象的控制信息。根据上一步的控制解析，交给基于角色权限模型解析引擎和数据访问引擎，对获取的数据进行条件过滤，然后根据过滤之后的结果，对资源进行访问。

实施例

[0020] 按照图 1 所示，具体实施步骤如下。

[0021] 步骤一：权限模型的实体类定义。实体类包含基于属性权限实体类和基于角色的实体类。

[0022] (1)基于属性的权限实体类主要属性

```
/// 属性控制项集合
GSPAttributeItemCollection AttributeItemCollection { get; set; }
/// 属性权限分配结果表 ID
string ResultTableID { get; set; }
```

(2) 基于角色的权限实体类主要属性

```
        /// 对应数据源集合。
```

[0023] GSPRowDataSourceCollection DataSourceCollection { get; set; }

/// 与其它数据权限对象之间的关系。

[0024] GSPRowAssociationCollection Associations { get; set; }

/// 权限分配设置。

[0025] GSPRowAssignSetting AssignSetting { get; set; }

/// 权限分配结果设置。

[0026] IPermissionResultSetting ResultSetting { get; set; }

步骤二：权限模型控制触发点设置。将业务资源和第一步定义的权限对象关联起来；

(1) 基于属性的权限对象与业务资源的关联关系主要属性

```
/// 业务资源对象 ID
string BizObjID { get; set; }
/// 业务资源对象类型
GSPRacDataResourceType BizObjType { get; set; }
/// 权限对象 ID[行权限对象 ID 或者列权限对象 ID]
string PermissionObjID { get; set; }
/// 元素之间的映射关系集合[行权限]
GSPResRowElementMappingCollection ElementMappings { get; set; }
```

(2) 基于角色的权限对象与业务资源的关联关系主要属性

```
/// 业务资源对象 ID
string BizObjID { get; set; }
/// 业务资源对象类型
GSPRacDataResourceType BizObjType { get; set; }
/// 属性权限对象 ID
string AttributeObjID { get; set; }
```

步骤三：权限模型控制条件设置。根据步骤一定义的模型实体类，获取定义的控制项信息，然后对定义的各个控制项，设置对应控制的条件，并将设置的条件持久化，以便后面步骤调用；

步骤四：权限模型访问控制的调用。其主要的接口方法如下所示。

[0027] (1)基于属性的访问接口方法

```
/// <summary>
/// 根据业务对象、操作，获取字段权限的结果集合
/// </summary>
/// <param name="objID">业务资源对象 ID</param>
/// <param name="opID">操作 ID</param>
/// <returns>单元格权限分配结果</returns>
SPCellPermissionCollection GetCellPermissions(string objID, string opID);
```

(2) 基于角色的访问接口方法

```
/// <summary>
/// 根据业务对象、操作，获取数据权限结果控制的过滤条件
/// </summary>
```

```
/// <param name="objID">业务资源对象 ID</param>
/// <param name="opID">操作 ID</param>
/// <returns>控制的过滤条件集合 </returns>
GSPRacDataFilterCollection GetDataFilters(string objID, string opID);
/// <summary>
/// 根据业务对象、操作、数据授权对象，获取数据权限结果控制的过滤条件
/// </summary>
/// <param name="objID">业务资源对象 ID</param>
/// <param name="opID">操作 ID</param>
/// <param name="permissionID">数据权限对象 ID</param>
/// <returns>控制的过滤条件 </returns>
IRacDataFilter GetDataFilter(string objID, string opID, string
permissionID);
```

步骤五：权限模型控制条件的解析。将步骤四获取出来的控制条件，按照类型不同，调用不同的条件解析器进行解析，然后将结果返回；

步骤六：根据返回结果，访问资源。根据步骤五解析的模型控制条件结果，对资源进行控制访问。

[0028] 除说明书所述的技术特征外，均为本专业技术人员的已知技术。

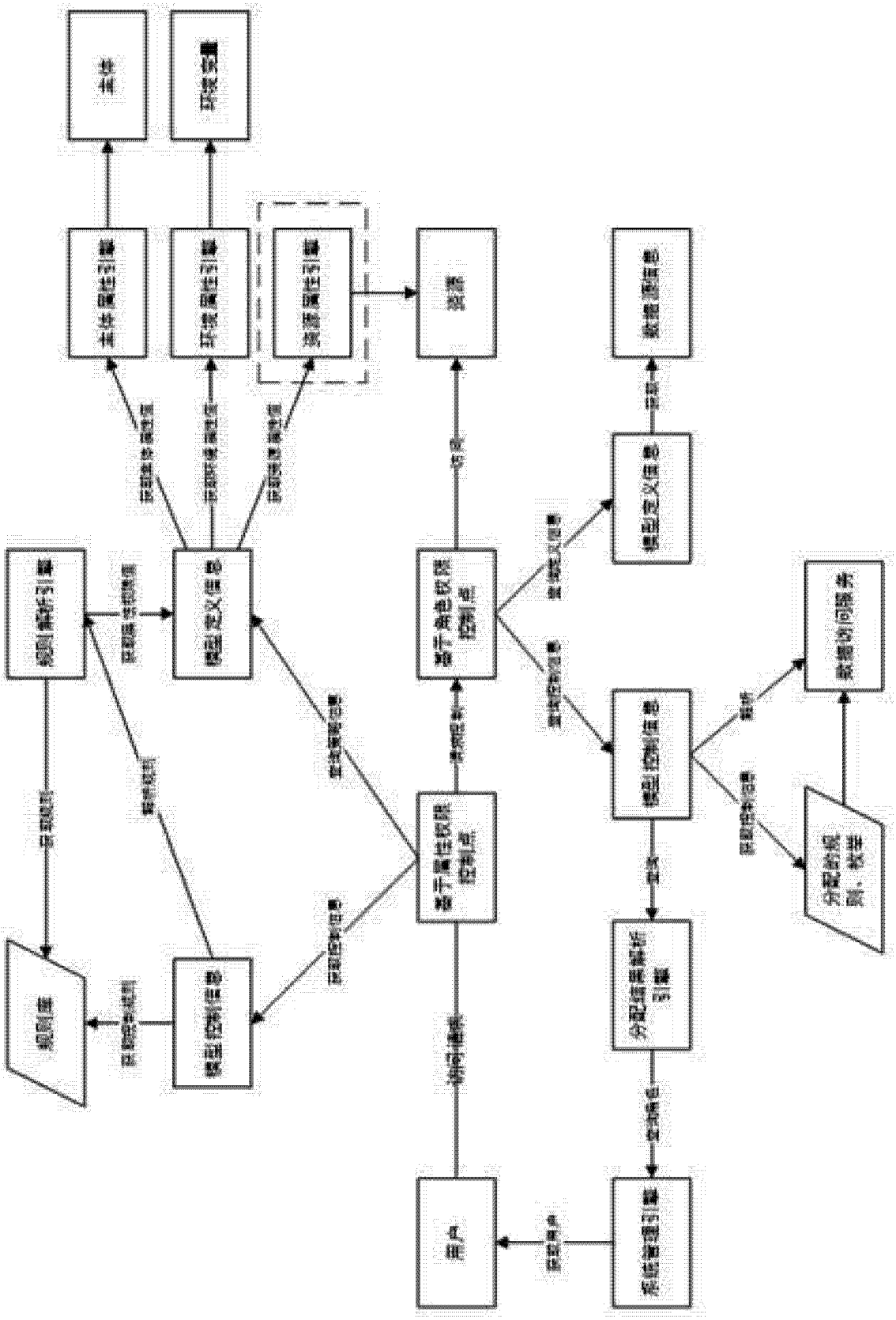


图 1

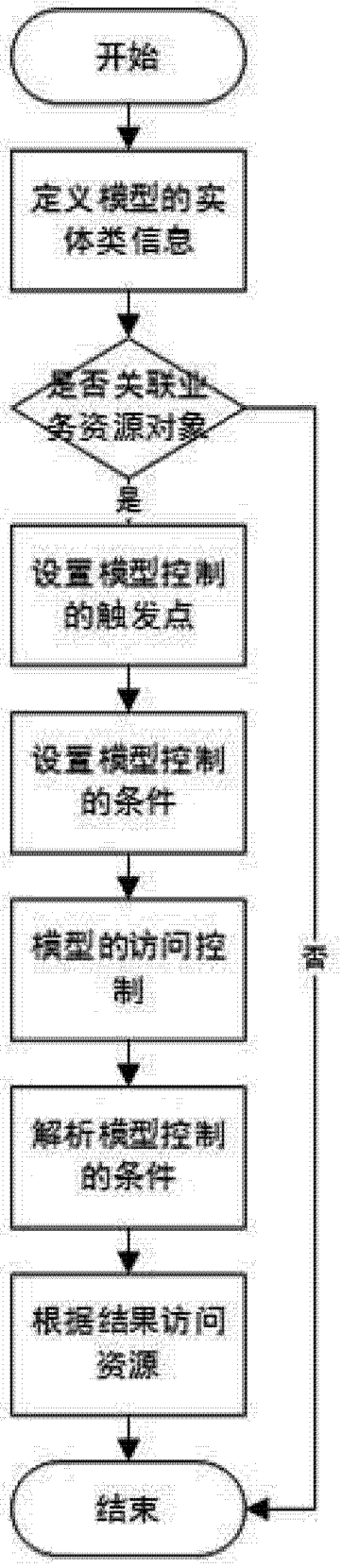

开始
定义模型的实体类信息
是否关联业务资源对象
是
否
设置模型控制的触发点
设置模型控制的条件
模型的访问控制
解析模型控制的条件
根据结果访问资源
结束

图 2

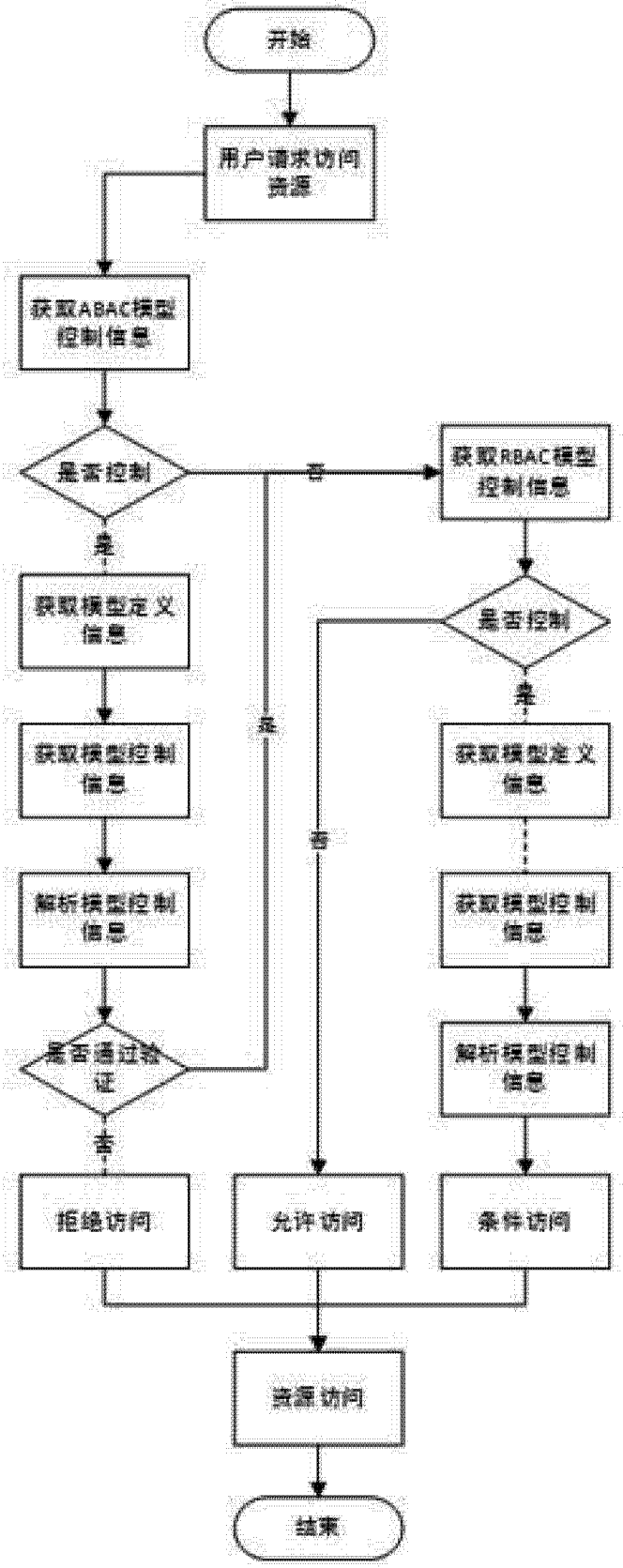

开始
用户请求访问资源
获取ABAC模型控制信息
是否控制
否
是
获取模型定义信息
获取模型控制信息
解析模型控制信息
是否通过验证
否
是
拒绝访问
获取RBAC模型控制信息
是否控制
否
是
获取模型定义信息
获取模型控制信息
解析模型控制信息
允许访问
条件访问
资源访问
结束

图 3