OBACM Documentation

Bincy Veena & Vineetha Shajan
JONKOPING UNIVERSITY | JONKOPING

INTRODUCTION

The Ontology-Based Access Control Module (OBACM) is a semantic framework developed to govern controlled access to Digital Product Passport (DPP) data. By integrating the Open Digital Rights Language (ODRL) ontology for policy definition and the PROV-O ontology for provenance tracking, OBACM enables a transparent, traceable access control mechanism.

Ontology-Based Access Control Module framework

This ontology formalizes key concepts such as users, roles, contextual requirements, policies, assets (DPP modules), and user request actions. Access decisions are derived based on semantic reasoning using SWRL rules, which also allow for provenance metadata to be automatically inferred. The model is intended to enhance data governance, support auditing, and provide machine-interpretable rules for sustainable product lifecycle data sharing.

The main objectives of OBACM are to:

- Establish a semantic access control model that is flexible, interoperable, and policy driven.
- Enable automated access decisions using reasoning over user roles, contextual constraints, and policies.
- Support traceability of data usage through integration with the PROV-O ontology.
- Facilitate compliance with ESPR by ensuring only authorized entities with proper roles and certifications can access specific DPP data modules.

Figure 1 shows the OBACM schema

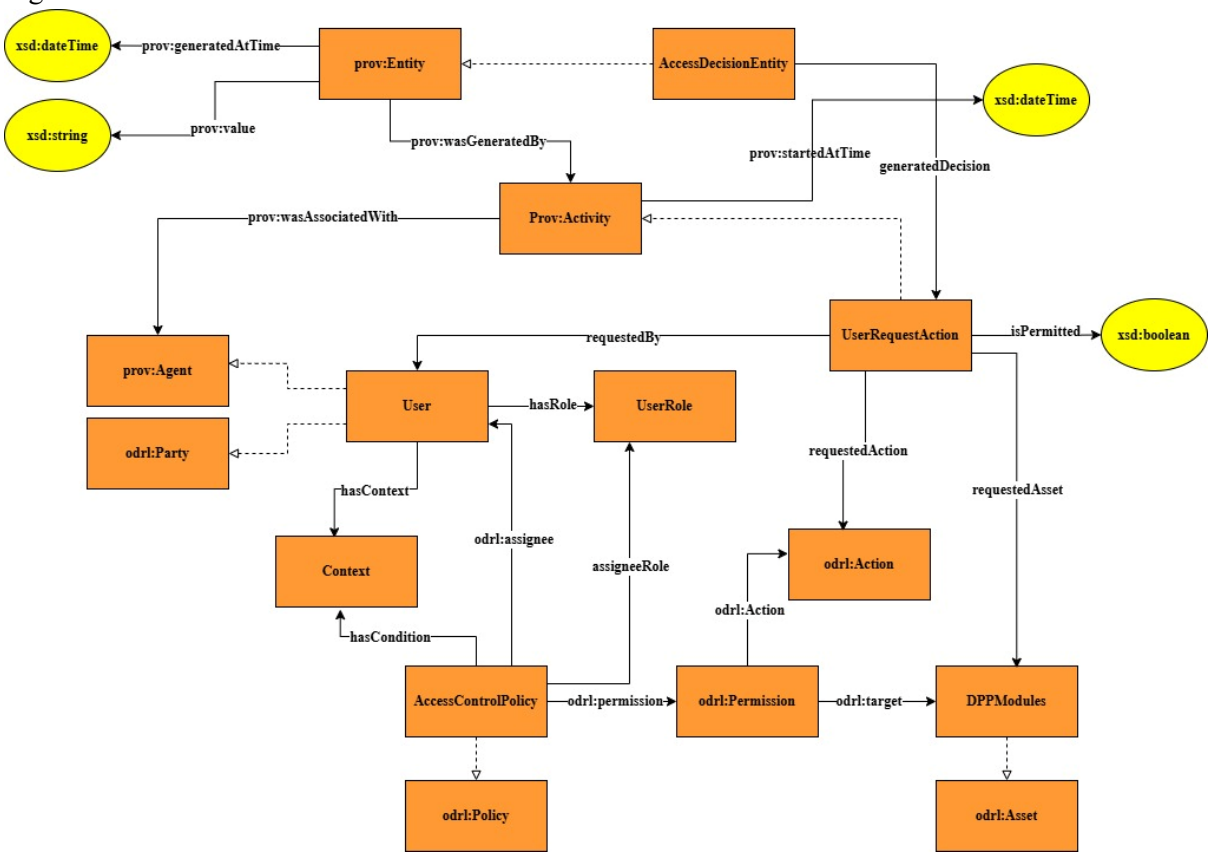

Figure 1:OBACM schema

Class Name: User

URI: http://ju.se/Ontology-BasedAccessControlModule#User

Summary

Represents individuals who request actions on DPP modules. Each user can hold a role and contextual information relevant for access control decisions.

Axiomatization in DL and Explanation

User ⊑ odrl:Party ⊓ prov:Agent

- A user is both an ODRL Party and a PROV Agent.

User ⊑ ∃ hasRole.UserRole ⊔ ∃ hasContext.Context

- Each user must have at least one role, or one context, or both.

Class Name: UserRole

URI: http://ju.se/Ontology-BasedAccessControlModule#UserRole

Summary

Defines the roles that can be assigned to users, such as Manufacturer, Importer, or Customer, to control access levels in the OBACM system.

Axiomatization in DL and Explanation

UserRole ⊑ Thing

- UserRole is a concept within the ontology (basic class).

User ⊑ ∃ hasRole.UserRole

- Every user has an assigned role.

UserRole ⊓ odrl:Party ⊑ ⊥

- UserRole and odrl:Party are disjoint (roles are not parties).

Class Name: Context

URI: http://ju.se/Ontology-BasedAccessControlModule#Context

Summary

Encapsulates metadata about users relevant for context-aware access control, such as certifications, affiliations, or organizational roles.

Axiomatization in DL and Explanation

Context ⊑ owl:Thing

- General concept without a strict superclass.

∃ hasContext.Context ⊑ User

- Used to annotate users with relevant metadata that can influence policy applicability.

Class Name: DPPModules

URI: http://ju.se/Ontology-BasedAccessControlModule#DPPModules

Summary

Represents digital product passport modules that provide structured lifecycle information, including composition, origin, and maintenance history of a product.

Axiomatization in DL and Explanation

DPPModules ⊑ odrl:Asset

- All DPPModules are subclasses of ODRL Assets.

∃ OBACM:requestedAsset.DPPModules ⊑ UserRequestAction

- Any request that targets a DPPModule is considered a UserRequestAction.

Class Name: AccessControlPolicy

URI: http://ju.se/Ontology-BasedAccessControlModule#AccessControlPolicy

Summary

SubClassOf ODRL policies tailored for regulating access to digital product passport (DPP) modules based on user roles and context and defined permissions.

Axiomatization in DL and Explanation

AccessControlPolicy ⊑ odrl:Policy

- AccessControlPolicy is a SubClassOf odrl:Policy

AccessControlPolicy ⊑ ∃ assigneeRole.UserRole

- AccessControlPolicy is assigned to a specific user role.

AccessControlPolicy ⊑ ∃ hasCondition.Context

- AccessControlPolicy is associated with context-based access conditions.

AccessControlPolicy ⊑ ∃ odrl:permission.odrl:Permission

- Every policy includes atleast one permission

Class Name: UserRequestAction

URI: http://ju.se/Ontology-BasedAccessControlModule#UserRequestAction

Summary

Captures an individual user's request to perform a specific action on a DPP module.

Axiomatization in DL and Explanation

UserRequestAction ⊑ Activity

- It is a type of provenance Activity.

UserRequestAction ⊑ ∃ requestedBy.User

- Each request must be made by a user.

UserRequestAction ⊑ ∃ requestedAction.Action

- Each request must involve one requested action.

UserRequestAction ⊑ ∃ requestedAsset.DPPModules

- Each request must specify a target asset.

UserRequestAction ⊑ ∃ isPermitted.{true, false}

- The result of the request must be a boolean permission outcome.

Class Name: AccessDecisionEntity

URI: http://ju.se/Ontology-BasedAccessControlModule#AccessDecisionEntity

Summary

Represents the result of evaluating a user request against applicable access control policies. It records whether access was granted or denied.

Axiomatization in DL and Explanation

AccessDecisionEntity ⊑ prov:Entity

- Considered a provenance outcome entity.

AccessDecisionEntity ⊑ ∃ generatedDecision.UserRequestAction

- Must reference the request it evaluates.

AccessDecisionEntity ⊑ ∃ prov:value.{"Access granted", "Access denied"}

- Expresses the decision in textual form.

Class Name: Permission

URI: http://www.w3.org/ns/odrl/2/Permission

Summary

A Permission defines an authorized action on a specific asset under a given policy. In the OBACM context, permissions are granted via access control policies to allow certain users or roles to act on DPP modules.

Axiomatization in Description Logic (DL) and Explanation

Permission ⊑ Thing

- A Permission is a basic OWL individual (default axiom via ODRL).

Permission ⊑ ∃ action.Action

- Each Permission must define at least one Action that it permits.

Permission ⊑ ∃ target.Asset

- A Permission must identify the Asset (e.g., DPP module) to which it applies.

Class Name: Action

URI: http://www.w3.org/ns/odrl/2/Action

Summary

An Action specifies an operation that can be performed on an asset (e.g., read, modify, write). These are linked to Permission instances and are used in evaluating access control decisions.

Axiomatization in Description Logic (DL) and Explanation

Action ⊑ Thing

- The class Action is a generic concept representing possible operations.

Individuals such as odrl:read, odrl:modify, odrl:write. Odrl:writeTo are instances of the class Action.

Competency Questions for OBACM

Following are the list of competency questions associated with OBACM.

1. What role is assigned to a user?
2. What context is associated with a user?
3. What is the role and context assigned policies?
4. Which access control policies are assigned to a specific asset?
5. What permissions does a policy have?
6. What actions does a permission allow?
7. Which role is allowed to do what action to which asset?
8. What types of actions are modelled within OBACM?
9. Who is the assignee of a specific policy?
10. When does a request start?
11. Which user is responsible for a user request action?
12. Which roles are associated with policy permissions?
13. What assets are being requested in a user action?
14. Which users are affected by a given policy?
15. What was the outcome of a user request?
16. Which action and asset were part of the request?
17. Which user request led to this decision?
18. Which asset is targeted by a permission?
19. What types of actions was requested by a user?
20. Which requests involve what action?
21. What contextual credentials does a user possess?