**Definition (Entity regulation).** *An entity regulation identifies the entities (i.e., users or roles) whose behavior is being regulated by a given policy. Its input parameters are one policy and the set of relationships defined between entities and contexts that are represented in the domain ontology.*

*Entity regulation = Policy × Rel → SR*

**Definition (Object regulation)**. *An object regulation identifies the objects used by the behavior being regulated by a given policy. Its input parameters are one policy and the relationships defined views and objects that are represented in the domain ontology.*

*Object regulation = Policy × Rel → O*

# **Correctness of Policy Propagation**

The aim of this sectio*n* is to demonstrate that, given a set of policies, the policy propagation restrictions specified in the paper preserves conflicts in the given set of policies, i.e., *they do not include or remove any conflict*. Essentially, the propagation restrictions only require a given set of policies to make explicit regulations that were otherwise implicit as they are defined at the level of composite elements such as contexts or entities. In other words, *no new regulation is created or removed by propagation*.

**Lemma 1** (Object composition correctness). *Object composition propagation preserves regulation.*

*Proof.* Let *P* be a set of policies with *p*∈*P*,   
*p* = *〈kp, org, sr, aa, v, ac, dc〉*, that is, *p* is a policy that states a vision. Therefore, by applying *objects regulation function* (see Definition above), *p* applies to all objects linked by the object composition relationship to *v*. The propagation of *p* requires policies *po* = *〈kp, org, sr, aa, o, ac, dc〉*, where *objectComposition* (*v*,*o,org*), also to be in *P*. The propagation of *p* does not imply any new regulation in *P* that was not already specified by *p*. The objects implicitly identified in *p* are exactly the same objects explicitly identified by the set of *po*. In addition, since *po* is propagated based on *p*, its other parameters have not been changed.

**Lemma 2** (Hierarchy propagation correctness). *Hierarchy propagation preserves regulation.*

*Proof.* Let *P* be a set of policies with *p*∈*P*, *p*= *〈kp, org,\_, aa, ov, acc, dac〉*, that is, *p* is a policy defined solely in the context of an organization *org*. Therefore, by applying *entity regulation function* (see Definition above), *p* governs the behavior of all suborganizations of *org*. The propagation of *p* requires policies *ps* = *(〈kp, org,suborg, aa, ov, acc, dac〉*, where *hierarchy*(*org*, *suborg*), also to be in *P*. The propagation of *p* does not imply any new regulation in *P* that was not already specified by *p*. The suborganizations whose behaviors were regulated by *p* are exactly the same suborganizations whose behaviors are being regulated by the set of *ps*. In addition, since *ps* is propagated based on *p*, its other parameters have not been changed.

**Lemma 3** (Play propagation correctness). *Play propagation preserves regulation.*

*Proof.* Let *P* be a set of policies with *p*∈*P*,   
*p*= *〈kp, org, r, aa, ov, ac, dc〉,* that is, *p* is a policy defined in the context of an organization *o* and applied to role *r*. Therefore, by applying *entity regulation function* (see Definition above), *p* governs the behavior of all users playing role *r* in *o*. The propagation of *p* requires policies *pu* = *〈kp, org, s, aa, ov, ac, dc〉*, where *play* (*org*,*s*,*r*), also to be in *P*. The propagation of *p* does not imply any new regulation in *P* that was not already specified by *p*. The users whose behaviors were regulated by *p* are exactly the same users whose behaviors are being regulated by the set of *pe*. In addition, since *pe* is propagated based on *p*, its other parameters have not been changed.

**Lemma 4** (Ownership propagation correctness). *Ownership propagation preserves regulation.*

*Proof.* Let *P* be a set of policies with *p*∈*P*,   
*p*= *〈kp, org,\_,aa, ov, ac, dc〉*, that is, *p* is a policy defined in the context of an organization *org*. Therefore, by applying *entity regulation function* (see Definition above), *p* governs the behavior of all role *r* played in *org*. The propagation of *p* requires policies *pr* = *〈kp,org,r,aa,ov,ac,dc〉,* where *ownership*(*org*,*r*), also to be in *P*. The propagation of *p* does not imply any new regulation in *P* that was not already specified by *p*. The roles governed by *p* are exactly the same roles governed by the set of *pr*. In addition, since *pr* is propagated based on *p*, its other parameters have not been changed.

**Theorem 1** (Policy propagation correctness). *Policy propagation does not add or remove regulation of a given set of policies.*

Proof. By lemmata 1 to 4.