

# An Ontology Design Pattern for Role-Dependent Names

Rushrukh Rayan<sup>1,\*</sup>, Cogan Shimizu<sup>2</sup> and Pascal Hitzler<sup>1</sup>

<sup>1</sup>Kansas State University, Manhattan, KS, USA 66506

<sup>2</sup>Wright State University, Dayton, OH, USA 45435

## Abstract

We present an ontology design pattern for modeling Names as part of Roles, to capture scenarios where an Agent performs different Roles using different Names associated with the different Roles. Examples of an Agent performing a Role using different Names are rather ubiquitous, e.g., authors who write under different pseudonyms, or different legal names for citizens of more than one country. The proposed pattern is a modified merger of a standard Agent Role and a standard Name pattern stub.

## Keywords

Ontology Design Pattern, Agent Role, Modular Ontology Modeling

## 1. Introduction

We present an ontology design pattern for role-dependent names of agents that appears to be of rather ubiquitous importance but has not been described explicitly yet, to the best of our knowledge. The pattern is a relatively straightforward reification that modifies previously published patterns, as we discuss below. However we believe that even straightforward patterns such as this should be modeled carefully and provided to the public in the spirit of easy reuse and development of tool support, in the spirit of, e.g., the Modular Ontology Modeling (MOMo) methodology [1], the Modular Ontology Design Library (MODL) [2], and the COMoIDE software framework [3].

In the MODL library we find an AgentRole pattern as depicted in Figure 1 (top), as well as a Name Stub pattern depicted in Figure 1 (bottom).<sup>1</sup> Typical uses of the former would be for roles of agents, such as being an employee in a company or author of a publication, where it is desirable to adorn the Role with additional context information like dates of employment or placement in the author sequence.

A schema diagram resulting from a naïve combination of these two patterns is depicted in Figure 2 (with some mild renaming – we are using patterns as templates as argued in [4, 1], rather than verbatim). Of course this pattern does not account for dependency of a name on the

---

WOP '23: 14th Workshop on Ontology Design and Patterns, Nov 06–10, 2023, Athens, Greece

\*Corresponding author.

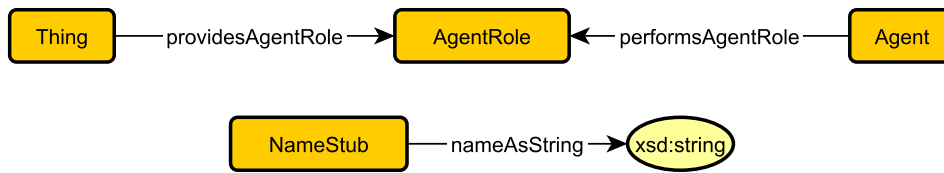
✉ rushrukh@ksu.edu (R. Rayan); cogan.shimizu@wright.edu (C. Shimizu); hitzler@k-state.edu (P. Hitzler)

🌐 <https://cogan-shimizu.github.io> (C. Shimizu); <http://pascal-hitzler.de> (P. Hitzler)

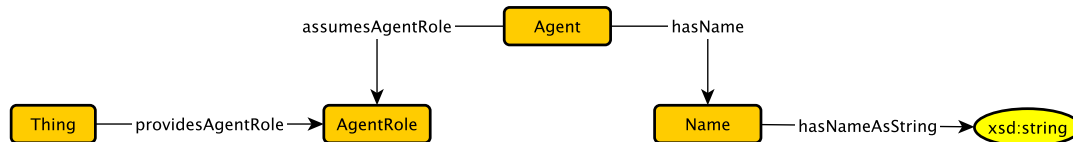
© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

<sup>1</sup>Following MOMo [1], we mostly discuss patterns by means of their schema diagrams, and only dive into axiomatization where needed, or at the very end when presenting the final pattern.



**Figure 1:** Diagrams for Agent Role pattern (top) and Name Stub pattern (bottom), as per [2]



**Figure 2:** Diagram for naively joined AgentRole and NameStub patterns

```

:spiritInBondage :providesAgentRole :sibAuthorRole .
:griefObserved  :providesAgentRole :goAuthorRole .
:csLewis        :assumesAgentRole  :sibAuthorRole ,
                :goAuthorRole ;
                :hasName           :csLewisNameCV ,
                :csLewisNameNWC ,
                :csLewisNameCSL .
:csLewisNameNWC :hasNameAsString  "N. W. Clerk"^^xsd:string .
:csLewisNameCV  :hasNameAsString  "Clive Hamilton"^^xsd:string .
:csLewisNameCSL :hasNameAsString  "C. S. Lewis"^^xsd:string .

```

**Figure 3:** Example triples conforming to the diagram in Figure 2, for the example case of *C. S. Lewis*, who published a collection of poems, *Spirit in Bondage*, under the pseudonym *Clive Hamilton*. On the other hand, his book entitled *Grief Observed* was published under the pseudonym *N. W. Clerk*

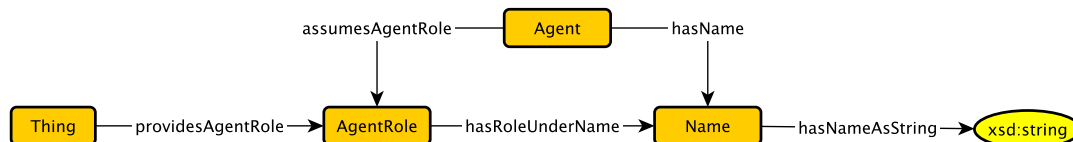
role: as an example for this type of dependency, consider the case of *C. S. Lewis*, who published a collection of poems, *Spirit in Bondage*, under the pseudonym *Clive Hamilton*. On the other hand, his book entitled *Grief Observed* was published under the pseudonym *N. W. Clerk*. Using the diagram in Figure 2 to naively encode this information would result in the triples in Figure 3 which, of course, do not convey which of the two (if in fact any) pseudonym was used for publication of which of the two books. Other example scenarios with essentially the same issue occur if persons or organizations have different legal names in different jurisdictions, may have their name changed at some stage, may use any type of occupational pseudonyms, or in the context of identity falsification using fake names.

The remainder of the paper is organized as follows. In Section 2 we present our pattern diagrammatically. In Section 3 we provide its axiomatic formalization. This is followed by

conclusions in Section 4. The pattern is available on Github. <sup>2</sup>

## 2. Overview of the Role-Dependent Names Pattern

The difficulty posed by the diagram in Figure 2 is, of course, easily addressed by making use of the fact that both *AgentRole* and *Name* are already reifications. The resulting diagram is depicted in Figure 4. We will refer to this pattern as the Role-Dependent Names (in short, RDN) pattern.



**Figure 4:** Schema Diagram for the Role-Dependent Names pattern

Following this diagram, the example triples in Figure 5 address the previously discussed *C. S. Lewis* example.

```
:sibAuthorRole :hasRoleUnderName :csLewisNameCV .  
:goAuthorRole :hasRoleUnderName :csLewisNameNWC .
```

**Figure 5:** Additional triples, completing those in 3, for the example case of *C. S. Lewis*, who published a collection of poems, *Spirit in Bondage*, under the pseudonym *Clive Hamilton*. On the other hand, his book entitled *Grief Observed* was published under the pseudonym *N. W. Clerk*. The combined set of triples conforms with our Role-Dependent Name pattern, the schema diagram of which is depicted in Figure 4.

## 3. Pattern Axiomatization

Following the MOMo methodology [1], we give a full set of axioms that we deem appropriate for the RDN pattern. The RDN pattern is driven by the interplay between the three core concepts: *Agent*, *AgentRole*, and *Name*. The OWL<sub>Ax</sub> approach [5, 6], which we follow here, suggests to first look at each node-edge-node ensemble in the schema diagram, and then at disjointness and additional axioms. Axiomatization is partially derived from the MODL library [2].

*Agent* in the pattern generally refers to a person or an organization, i.e., inanimate objects would not usually fall under the scope of the *Agent* class. For instance, if a *Table* is used for dining, the *Table* cannot be thought of as an *Agent* that assumes the *Role* of a Dining Table in this scenario: it would still be a *role*, but not an *agentrole*.

With regard to axiomatization, it is natural to say that every agent must have a name (1). A structural tautology is used to convey that an *Agent* *may* (but does not necessarily) also assume

<sup>2</sup>Pattern available at: <https://github.com/odpa/patterns-repository/tree/master/RoleDependentNames>

an AgentRole (2). To specify the domain of things that can assume an AgentRole, we make use of a Scoped Domain axiom to say that if something assumes an AgentRole, then it must be an Agent (3).

Additionally, we use Inverse Functionality to restrict the number of agents that can assume a specific AgentRole (4). Concretely, axiom (4) says that an AgentRole can be assumed by at most one Agent. Although this could be done differently, this choice constrains the shape of the RDF graph that complies with the pattern, and thus disambiguates the usage of the pattern.

$$\text{Agent} \sqsubseteq \exists \text{hasName.Name} \quad (1)$$

$$\text{Agent} \sqsubseteq \geq 0 \text{assumesAgentRole.AgentRole} \quad (2)$$

$$\exists \text{assumesAgentRole.AgentRole} \sqsubseteq \text{Agent} \quad (3)$$

$$\text{AgentRole} \sqsubseteq \leq 1 \text{assumesAgentRole}^{-} . \text{Agent} \quad (4)$$

AgentRoles mean the various roles an Agent can assume. We make use of a Scoped Range axiom to say that if an Agent assumes a Role, then it must be an AgentRole (5). Further, and centrally to this paper, an AgentRole may be assumed under a specific name, and we indicate this using a Structural Tautology axiom (6). The hasRoleUnderName property can furthermore be safely declared to have global range Name, and providesAgentRole can likewise be declared to have range AgentRole.

$$\text{Agent} \sqsubseteq \forall \text{assumesAgentRole.AgentRole} \quad (5)$$

$$\text{AgentRole} \sqsubseteq \geq 0 \text{hasRoleUnderName.Name} \quad (6)$$

$$\top \sqsubseteq \forall \text{hasRoleUnderName.Name} \quad (7)$$

$$\top \sqsubseteq \forall \text{providesAgentRole.AgentRole} \quad (8)$$

For Name, Inverse Functionality is used to express that a Name can be the name of at most one Agent (9). This is done with a similar motivation as axiom (4) above, i.e. to constrain the possible RDF graphs conforming with the pattern, i.e., to disambiguate use of the pattern. Axioms (10) and (11) declare global range for hasName and global domain for hasNameAsString, respectively.

$$\text{Name} \sqsubseteq \leq 1 \text{hasName}^{-} . \text{Agent} \quad (9)$$

$$\top \sqsubseteq \forall \text{hasName.Name} \quad (10)$$

$$\exists \text{hasNameAsString} . \top \sqsubseteq \text{Name} \quad (11)$$

We add the obvious disjointness axioms (12–14), and then also two role chains axioms, (15) and (16). (15) formalizes that if an agent assumes a role under a Name, then the Agent must

have the same name. Similarly, (16) formalizes that if an agent has a name and a role is assumed under that name, then the agent must assume the same Role.

$$\text{AgentRole} \sqcap \text{Agent} \sqsubseteq \perp \quad (12)$$

$$\text{Agent} \sqcap \text{Name} \sqsubseteq \perp \quad (13)$$

$$\text{Name} \sqcap \text{AgentRole} \sqsubseteq \perp \quad (14)$$

$$\text{assumesAgentRole} \circ \text{hasRoleUnderName} \sqsubseteq \text{hasName} \quad (15)$$

$$\text{hasName} \circ \text{hasRoleUnderName}^{-1} \sqsubseteq \text{assumesAgentRole} \quad (16)$$

## 4. Conclusion

The Role-Dependent Name pattern is aimed towards situations where there is an association between the Name and the Role as Agents can assume different roles under different names. As usual, the pattern is not meant to be rigid, in the sense that part of the pattern can be omitted when making use of it, or it can be extended as needed. E.g., AgentRoles may carry additional information such as spatio-temporal extents, and names may have a rich structure.

## Acknowledgments

The authors acknowledge funding under the National Science Foundation grants 2119753 "RII Track-2 FEC: BioWRAP (Bioplastics With Regenerative Agricultural Properties): Spray-on bioplastics with growth synchronous decomposition and water, nutrient, and agrochemical management" and 2033521: "A1: KnowWhereGraph: Enriching and Linking Cross-Domain Knowledge Graphs using Spatially-Explicit AI Technologies."

## References

- [1] C. Shimizu, K. Hammar, P. Hitzler, Modular ontology modeling, *Semantic Web* 14 (2023) 459–489. URL: <https://doi.org/10.3233/SW-222886>. doi:10.3233/SW-222886.
- [2] C. Shimizu, Q. Hirt, P. Hitzler, MODL: A modular ontology design library, in: K. Janowicz, A. A. Krisnadhi, M. Poveda-Villalón, K. Hammar, C. Shimizu (Eds.), *Proceedings of the 10th Workshop on Ontology Design and Patterns (WOP 2019) co-located with 18th International Semantic Web Conference (ISWC 2019)*, Auckland, New Zealand, October 27, 2019, volume 2459 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 47–58. URL: <https://ceur-ws.org/Vol-2459/paper4.pdf>.
- [3] C. Shimizu, K. Hammar, P. Hitzler, Modular graphical ontology engineering evaluated, in: A. Harth, S. Kirrane, A. N. Ngomo, H. Paulheim, A. Rula, A. L. Gentile, P. Haase, M. Cochez (Eds.), *The Semantic Web – 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31-June 4, 2020*, *Proceedings*, volume 12123 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 20–35. URL: [https://doi.org/10.1007/978-3-030-49461-2\\_2](https://doi.org/10.1007/978-3-030-49461-2_2). doi:10.1007/978-3-030-49461-2\_2.

- [4] K. Hammar, V. Presutti, Template-based content ODP instantiation, in: K. Hammar, P. Hitzler, A. Krisnadhi, A. Lawrynowicz, A. G. Nuzzolese, M. Solanki (Eds.), *Advances in Ontology Design and Patterns* [revised and extended versions of the papers presented at the 7th edition of the Workshop on Ontology and Semantic Web Patterns, WOP@ISWC 2016, Kobe, Japan, 18th October 2016], volume 32 of *Studies on the Semantic Web*, IOS Press, 2016, pp. 1–13. URL: <https://doi.org/10.3233/978-1-61499-826-6-1>. doi:10.3233/978-1-61499-826-6-1.
- [5] M. K. Sarker, A. A. Krisnadhi, P. Hitzler, OWLax: A protege plugin to support ontology axiomatization through diagramming, in: T. Kawamura, H. Paulheim (Eds.), *Proceedings of the ISWC 2016 Posters & Demonstrations Track co-located with 15th International Semantic Web Conference (ISWC 2016)*, Kobe, Japan, October 19, 2016, volume 1690 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2016. URL: <https://ceur-ws.org/Vol-1690/paper83.pdf>.
- [6] A. Eberhart, C. Shimizu, S. Chowdhury, M. K. Sarker, P. Hitzler, Expressibility of OWL axioms with patterns, in: R. Verborgh, K. Hose, H. Paulheim, P. Champin, M. Maleshkova, Ó. Corcho, P. Ristoski, M. Alam (Eds.), *The Semantic Web – 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings*, volume 12731 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 230–245. URL: [https://doi.org/10.1007/978-3-030-77385-4\\_14](https://doi.org/10.1007/978-3-030-77385-4_14). doi:10.1007/978-3-030-77385-4\_14.