An Ontology Design Pattern for Dynamic Relative Relationships

Holly Ferguson[1], Adila A. Krisnadhi[2,3], and Charles F. Vardeman II[1]

[1] University of Notre Dame
{hfergus2,cvardema}@nd.edu
[2] Wright State University
[3] University of Indonesia
{krisnadhi}@gmail.com

Abstract. This research describes an ontology design pattern for dynamically conceptualizing, establishing, tracking, and updating relative relationships and dependencies between entities (real or representational) of a physical, temporal, and/or importance scope. We present a Relative Relationship (RR) Pattern, associated axioms, an implementation of current geometric scale translation research, a detailed example, and suggestions for other potential use cases. It provides data hooks that allow dynamic updating of linked data as changes occur in preference systems, scaling systems, or time expiration parameters; additionally, it separates the false notion that level of detail is always synonymous with scope. Furthermore, we discuss how this design pattern potentially acts as an intermediate step to assist the transition between open linked-data and decision support frameworks that need to readily update changes within the accurate data over modern, distributed data access points.

1 Introduction

Development of applications using Semantic Web[1] principles has the potential to enable integration of highly diverse and heterogeneous domains by creating data that is as intelligent as possible. This is significant as applications need increasing amounts of interdisciplinary data (such as resilience systems or lifetime cost analysis) there is a need for computational and data abstractions that can interconnect concepts and relate them across a variety of professional fields. Additionally, there is a need for effective methods of integrating dynamic, discoverable, intelligent data sets[2] into tools as they become available, preferably in at least a semi-automated manner [4] so as to only enhance the capabilities of the over-arching Decision Support Systems (DSS) [7] and their methods [1], [13]. This work proposes a methodology to address an aspect of these concerns by creating a specification of the *relative relationship* between conceptual and data elements published using linked data principles.

[1] Semantic Web: www.w3.org/standards/semanticweb/
[2] GeoKnow: geoknow.eu/Welcome.html

Development of the Green Scale[3] (GS) Tool [5], a Multi-Criteria Analysis Tool [7] designed to assist architects in making more informed sustainability decisions based on contrasting and varied design metrics, has required labor intensive, manual integration of heterogeneous data. Calculations utilizing computational models such as thermal flow and embodied energy life cycle inventories require data from sources all over the globe encompassing areas ranging from the mining process to the manufacturing, transportation and overall lifespan of the project. In developing this tool, we found that the necessary data is often drastically different between sources and even missing or incomplete if it exists at all. Even if it were all available, open, and ready to computationally process, there is still the matter of bridging between varied types of units, measuring systems, precision, naming semantics, and so on not only in the data but in the structure of the building models themselves such as gbXML[4], BIM/IFC[5], and CityGML [6]. For example, sustainability metrics are calculated using computational models that originate from different applications and modeling schema. There are semantic, geometric, and data sourcing differences that need to be resolved or aligned before any computations in sustainability can be trusted. Translation between two geometric models requires tracking the differences and relationships between geometric "scale" and coordinates with respect to each other via world and body centered coordinates in the frame of physical objects represented in the schema. To maintain the desired compatibility to translate between building schema, relationships such as these have to be tracked, and this is only a single type of relationship example between entities that needs to be tracked. If scale itself can be abstracted to a type of relationship between two entities (in this case two types of geometry systems) then the goal became to expand the usefulness of the scale sub-pattern into more general terms (i.e. the relative relationship of one entity to another-in this case for geometry systems).

Even with well-structured data published using Linked Open Data[6] and DSS, there can be conceptual gaps between the two systems, especially when decision criteria systems and application consumers need to function over a distributed service oriented architecture. These applications typically have requirements to track relationships across time, limitations, and other dependencies between conceptual entities. During the course of our research we have found a need to answer to these and other questions such as:

- What is the relationship between two entities within the known world?
- What dependencies, if any, exist between any two defined entities?
- How can I automate learning about relationship(s) and define comparisons?

A larger goal that is currently being implemented as an extension to the GS work is to make linked data more portable between decision frameworks by automatically integrating sets of data from a variety of ontological structures as

[3] The Green Scale: www.greenscale.org
[4] GBXML: www.gbxml.org
[5] BIM Extended Markup Language (BIMXML): www.bimxml.org
[6] Linked Open Data: www.linkeddata.org

a basis for switching between lower level data formats and schema. There is a need to compare one "entity" to another and calculate some relative relationship between them; this is the basis for the Relative Relationship (RR) Pattern[7] and presented in this paper. The RR pattern can track relationships and thus is solving some of the issues of mapping between different geometry systems, ontologies, and other standards. The remainder of the paper describes pattern axioms, example pattern applications, and a discussion of potential implications.

2 Relative Relationship Pattern

The RR pattern relates entities [Table 1] to each other instead describing them relative to a global "world frame" of reference. Applications are free, and encouraged, to specify a world frame relative to an entity but this specification would still be considered part of the body frame of that pattern instance for that entity. Previous notions of scale, such as those mentioned in the related work section, use a world frame to situate scaled objects. In both cases scale of some entity is still part of some body frame or another and thus can be compared-or related-to another if that translation could be tracked and would allow comparisons between otherwise incomparable entities. This process of abstraction of situational descriptions can be extended to other notions beyond physical existence. The relationship of one entity relative to another explains how we perceive scale, resolution, and conceptual analysis methods [3] and these notions may be captured with the RR pattern. The idea of relationships corresponds similarly to these notions but RR resides at a higher level of abstraction and can be used to quantify scale and relate to level of detail, etc. There are a number of other design patterns that perform comparisons for specific types of applications such as the PartOf, Material-Transformation, Agent-Role, Activity Reasoning, and Control-Flow patterns[8], for example, but the RR Pattern aims to capture additional aspects of data across time/dependency structures and act in conjunction with or as a super class of other patterns, especially as research progresses into multi-application and dynamic multi-DSS integration.

How can tracking such different concepts (physical, temporal, and informational data) be compared in any logical way? If the Relative Relationship (RR) pattern is considered momentarily as a "Meta-Pattern", then there of course would have to be categorical mechanisms in place so that coherent comparisons are maintained within different instances of the pattern-or at least this mechanism should inform the pattern as to the comparison types. To ease the conversation, a set of definitions describing this pattern is included in Table 1. Additionally, *Pattern Sequences* are discussed which refer to collections of RR pattern instances working together for a larger purpose-they provide the fuller data perspectives necessary to feed/rank information into a DSS, for example. *Dependency* structures are also frequently mentioned, and this vocabulary indi-

[7] `http://ontologydesignpatterns.org/wiki/Submissions:RelativeRelationship`
[8] ODP: http://ontologydesignpatterns.org

Table 1. Vocabulary Explanations of the RR Pattern and Other Associated Terms

Term	Definition in Context
Entity	A physical object, object representation, piece of information, slice of time, or other "thing" that can be described by human perception or understanding that defines some existence.
Relative Existence	This is the instantiation of the pattern representation of an original Entity for a which a comparison is going to be made; it can be one of many attributes of the original Entity.
Relationship Description	This describes the relationships being established between two entity representations via the Relative Existence property.
Domain	Description tag to determine the field of study or data pattern of the instance-for additional interdisciplinary functionality.
Usage	An optional reference to distinguish between the multiple uses of RR instances and to organize pattern application intentions.
Property	Data, characterization, quality of, or description about an established Relative Existence or Entity.
Attribute	Description or quality of an established relationship type.
Scope	Set of scaled entities that are being considered together as a group/type or that are the extents of the known instances from the perspective of the pattern extents, data, or applications.
Level of Detail	Selection out of all possible entities that could be considered in the group or added/removed as needed. This concept is not necessarily synonymous with or proportional to Scope or generality of information-it is application and instance dependent.

cates a single type of Relationship Description used to track the reliance of one Relative Existence state against another, across pattern instances or sequences.

Within the building professions example, this also brings up the challenge of vocabulary and how scale is traditionally perceived. In fact, scale of an object can only be assigned in the presence of a relative comparison between itself and some other entity. For instance, an architectural drawing with scale x is only assigned said physical scale as a representation compared to the physical realization of itself. We consider *Scale* to be the definition of the impact or perception of one entity relative to another; this can be a specialized "relative relationship." A variation of the RR pattern can capture relationships relative to an individual or system setting, or even two ideas, concepts, or values that may or may not be of equal importance. Similarly, various levels of intelligence need to effectively cooperate together within their respective decision support systems to run full analyses; they also need to be capable of adjusting for new data sets as they became available or desirable for use. To integrate several types of ontologies or ontology patterns, it can be observed that again there is a comparison happening and decision being made between two or more entities and often their relative relationship needs determining for translations to be possible.

3 Formal Description: Dynamic RR Pattern

3.1 Model of the Relative Relationship Pattern

The Dynamic Relative Relationship Pattern [Figure 1] is intended to model not only physical data, but also temporal data, importance information, dependencies, and other data/entity relationship combinations. Required data for this purpose should at least have a base of four parts including: 1) an identifier for the respective entities, 2) each type of existence being recorded (i.e. how are we identifying this existence and representing each entity), 3) the relationship description being captured, and 4) which existence type is the target versus origin representations; additional information is based on the specific needs the pattern is meeting. The format of the derived data is somewhat use-case-dependent, although now with the common underlying RR pattern structure [Table 1].

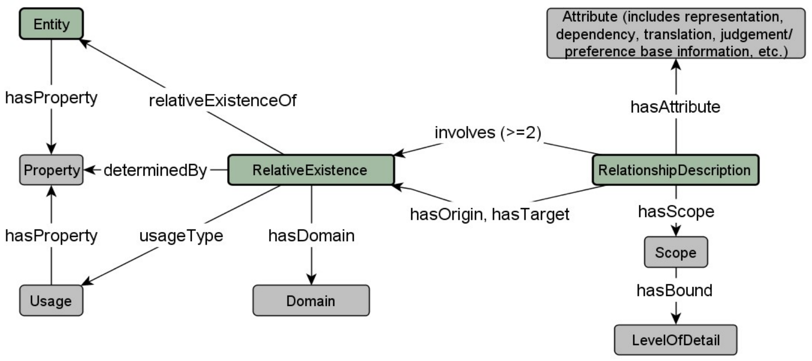

Fig. 1. Dynamic Relative Relationship Pattern Model.

3.2 Pattern Axioms and Explanations

The signature of the Dynamic Relative Relationship Pattern consists of the set $\mathbf{N_C}$ of classes (Eq. (1)), and $\mathbf{N_R}$ of properties (Eq. (2)).

$$\begin{aligned}\mathbf{N_C} =\{&Entity, RelativeExistence, RelationshipDescription,\\ &Property, Usage, Domain, Attribute, Scope, LevelOfDetail\} \quad (1)\\ \mathbf{N_R} =\{&relativeExistenceOf, hasProperty, determinedBy, hasUsage,\\ &hasDomain, involves, hasOrigin,\\ &hasTarget, hasAttribute, hasScope, hasBound\} \quad (2)\end{aligned}$$

Surface Semantics

We assert pairwise disjointness between any two classes in N_C except for the pair of *Scope* and $LevelOfDetail$. In particular, RelativeExistence is disjoint with Entity because a RelativeExistence is only an instantiation of an Entity in the context of some RelationshipDescription – it has no meaning outside the RelationshipDescription it is involved in. Scope and LevelOfDetail are not disjoint due to the fact that a Scope may not be contained to a single LevelOfDetail but a LevelOfDetail can refer to specific Scope.

Surface and Deep Semantics

A RelationshipDescription involves at least two instances of RelativeExistence, including one considered as origin and one as target. Moreover, a RelativeExistence is a relative existence of exactly one Entity.

$$\mathit{RelationshipDescription} \sqsubseteq (\geq 2\ \mathit{involves}.\mathit{RelativeExistence}) \tag{3}$$

$$\mathit{RelationshipDescription} \sqsubseteq \exists hasOrigin.RelativeExistence \tag{4}$$

$$\mathit{RelationshipDescription} \sqsubseteq \exists hasTarget.RelativeExistence \tag{5}$$

$$hasOrigin \sqsubseteq involves, \qquad hasTarget \sqsubseteq involves \tag{6}$$

$$RelativeExistence \sqsubseteq (=1\ relativeExistenceOf.Entity) \tag{7}$$

The pattern demands that the origin of a RelationshipDescription must be different from its target – given by the rule in (8), which cannot be expressed in OWL. This holds even when both RelativeExistences are associated with the same Entity. In other words, at least one aspect must be different between the origin and target, whether it is Domain, Property, or Usage. Applications may compare the same object and entity but instantiate the pattern at different states or points in time. For example, Features and types of each RelativeExistence can be in the form of a Property, Usage, or Domain, and can consist of none, one, or all of these properties. Properties can include physical descriptions (locations, geometry data, data, etc.), temporal features (timestamps, age, lifespan, decomposition rate, state limits, etc.), informational data (preferences, proportions, ranking factors, persistence limits, confidence intervals, etc.), and others. RelationshipDescription can also consist of none, one, or many attributes such as dependency structures, representational versus real world states, as well as scaling and transformation indicators, to name only a few.

$$\mathit{RelationshipDescription}(x) \wedge hasOrigin(x,y) \wedge\ hasTarget(x,z) \rightarrow y \neq z \tag{8}$$

The above rule can be expressed as OWL axioms below (though they will be beyond OWL 2 DL) where R_{RD} BS R_{OT} are fresh properties introduced specifically for the purpose of simulating the rule.

$$hasOrigin^{-} \circ R_{RD} \circ hasTarget \sqsubseteq R_{OT} \tag{9}$$

$$\mathsf{Irreflexive}(R_{OT}) \tag{10}$$

$$RelationshipDescription \equiv \exists R_{RD}.\mathsf{Self} \tag{11}$$

When populating this pattern, the RelationshipDescription class *must* be populated. That is, we do not allow, e.g., the Entity or RelativeExistence class to be populated without having RelationshipDescription populated too. This is not a typical requirement one would find in a pattern description, however, this is considered necessary for the usability of this pattern in the application domain. This requirement can be expressed using the *top property*, i.e., `owl:topProperty`, denoted with U in this paper. The following axioms assert the existence of an instance of RelationshipDescription whenever the other classes are non-empty.

$$\begin{aligned}&X \sqsubseteq \exists U.RelationshipDescription\\ &\quad\text{where } X \in \{Entity, RelativeExistence, Property, Usage, Domain,\\ &\qquad\qquad Attribute, Scope, LevelOfDetail\}\end{aligned} \tag{12}$$

A RelationshipDescription has at least one scope, and each scope has a bound of at least one level of detail.

$$RelationshipDescription \sqsubseteq \exists hasScope.Scope \tag{13}$$

$$Scope \sqsubseteq \exists hasBound.LevelOfDetail \tag{14}$$

A RelativeExistence may consist of one or more of Domain (hasDomainType), Usage (usageType), and/or Property (determinedBy).

$$\begin{aligned}RelativeExistence \sqsubseteq\ &\exists hasDomain.Domain \sqcup \exists hasUsage.Usage\\ &\sqcup \exists determinedBy.Property\end{aligned} \tag{15}$$

Domain and Range Restrictions

We assert guarded domain and range restrictions to all properties. Due to lack of space, we present how we can express these restrictions for the relativeExistenceOf property:

$$\exists relativeExistenceOf.Entity \sqsubseteq RelativeExistenceOf \tag{16}$$

$$RelativeExistenceOf \sqsubseteq \forall relativeExistenceOf.Entity \tag{17}$$

Specific for hasProperty property, we have the following guarded domain and range restrictions:

$$\exists hasProperty.Property \sqsubseteq Entity \sqcup Usage \tag{18}$$

$$Entity \sqsubseteq \forall hasProperty.Property \tag{19}$$

$$Usage \sqsubseteq \forall hasProperty.Property \tag{20}$$

An entity that is the target of a RelativeExistence should also relate to RelativeExistence by having descriptions of itself in the same form as RelativeExistence, i.e. RelativeExistence would be a link to another instance of the Relative Relationship Pattern depicting the RelativeExistence of the referenced entity as well so that entities can be linked across detail levels, scale levels, and similar scopes, time, and importance changes (i.e. pattern instance sequences).

4 Relative Relationship Pattern Usage and Examples

4.1 RR Meta-Pattern as a Bridge Between LD and DSS

Facilitation of data integration within decision frameworks is only one application example of using the RR Pattern as a Meta-Pattern to structure heterogeneous data and relationships. It can be used to track changes and dependencies, in conjunction with other patterns, or as an additional information layer for other patterns. Additionally, size and quality of instantiated patterns from many application domains can be established, and thus the data therein gains structured contextualization. All of these otherwise unconnected notions can in fact be compared and ranked together through a set of properties to create a viable pattern structure to handle Dynamic Relative Relationships.

To facilitate interoperability between all the components needed for better decision support, especially in an era of varied data types, formats, and locations, a dynamic RR meta-pattern is a useful layer to map the changes and relationships between varied data that is intended to be used seamlessly within applications. The larger scope of functionality that the RR pattern enables is a facilitation methodology for incorporating and maintaining linked data with decision support frameworks in a dynamic manner since aggregations of pattern instances create a tractable record of the relevant changes over time.

4.2 Example Scenario with Competency Questions

Structuring, formulating, and answering such questions as, *"What sections of my house should I renovate for the most positive environmental impacts and lowest cost?"*, requires the cooperation of several infrastructural components including structured data, knowledge bases, reasoners, decisions support frameworks, etc. For simplicity, assume that budget only allows the user to decide between replacing brick or interior gypsum finishes, and that the respective data is accessible [Figure 2]. To know that the RR pattern can be successful, competency questions such as the following for the building industry domain (which already requires complex, multifaceted sets of data, processing procedures, and tools) should be able to be answered using the RR pattern:

- Can I find out more than just basic material replacement costs?
- What will be effected in the sequences if I select certain RR instances for replacement; i.e. what materials are structurally dependent on others, etc.?
- Per this example, is an appropriate material found for replacement and what scope and range of materials are considered in that answer?

```
@prefix : <http://purl.org/net/rr/RelativeRelationshipPattern#>.
@prefix d: <http://rr-pattern.example/data#>.
@prefix v: <http://rr-pattern.example/extvocab#>.
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
v:CostToRepair rdfs:subClassOf :Property .
v:Dependency rdfs:subClassOf :RelationshipDescription .
d:brick rdf:type :Entity ; rdfs:label "Brick" .
d:studs rdf:type :Entity ; rdfs:label "Studs" .
d:insulation rdf:type :Entity ; rdfs:label "Insulation" .
d:gypsum rdf:type :Entity ; rdfs:label "Gypsum" .

d:brickrel1 rdf:type :RelativeExistence ; :relativeExistenceOf d:brick ;
    :determinedBy d:brickctp .
d:brickctp a v:CostToRepair ; v:hasValue "3000"; v:hasUnit iso4217:USD .
d:studsrel1 rdf:type :RelativeExistence; :relativeExistenceOf d:studs ;
    :determinedBy d:studsctp .
d:studsctp a v:CostToRepair ; v:hasValue "1000"; v:hasUnit iso4217:USD .
d:insulrel1 a :RelativeExistence; :relativeExistenceOf d:insulation ;
    :determinedBy d:insulctp .
d:insulctp a v:CostToRepair ; v:hasValue "2000"; v:hasUnit iso4217:USD .
d:gypsrel1 rdf:type :RelativeExisttence; :relativeExistenceOf d:gypsum ;
    :determinedBy d:gypsctp .
d:gypsctp a v:CostToRepair ; v:hasValue "2000"; v:hasUnit iso4217:USD .
d:dep1 a v:Dependency; :hasOrigin d:studsrel1 ;
      :hasTarget d:insulre1, d:gypsrel1 .
d:dep2 a v:Dependency; :hasOrigin d:gypsrel1;  :hasTarget d:insulrel1.
```

Fig. 2. Data for queries in Fig. 4 and Fig. 5

4.3 Example Implementation and Solution Using the RR Pattern

To structure this query, assume wall assemblies are made of brick, studs, insulation, and gypsum. This question can potentially be an attribute or potential component of a preference model, but currently this remains partially speculation. Suppose RR Pattern instances represent the building model geometric relationships via IFC or gbXML, but now with component relationship information. Simplistically, the data knows that the brick has stud dependencies (brick hold studs in place), the studs have dependencies of gypsum and insulation, etc. [Figure 3], and depending on the open/closed world assumptions and query style, the reverse information indicating that no dependency exists is potentially also available. Depending on data, the RR pattern might also include material costs, ages, lifespans, and decomposition rates as entity relationship properties. Regardless, a query for minimum replacement cost may yield that gypsum alone is cheaper to replace than brick [Figure 4]; but, the RR Pattern can use its dependency structure to determine that this initially cheaper choice will mean exposing insulation (perhaps asbestos) which will then also require immediate

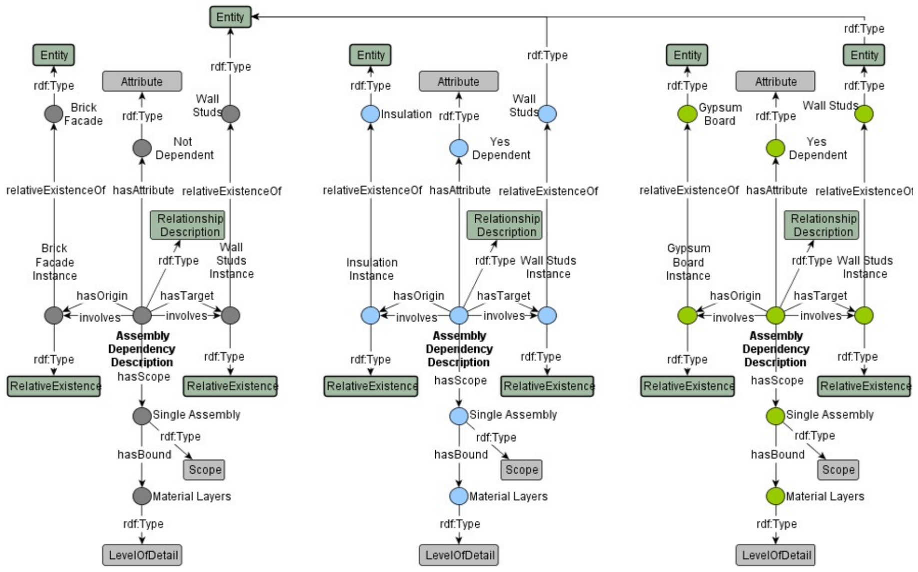

Fig. 3. Instantiation of RR Dependency Modeling for a Building Assembly.

replacement, making the originally cheaper option cost more when gypsum and insulation costs are aggregated together [Figure 5]. Returning to the original competency questions, these pattern instances do indeed give better insight into the user query than a list of base material costs; the dependency structure allows tabulations to be found for all materials that will contribute to building costs as a result of changing the original. The string of dependencies themselves, among other data, are also discoverable individually, leading to possibilities for even more accurate visualizations, decision impact, and overall understanding. Based on [Figure 5], a reasonable answer is found (brick) which is drastically different than a similar query not using the beneficial structures of the RR Pattern.

```
SELECT ?name ?cost
WHERE {
  VALUES ?name { "Brick" "Gypsum" }
  [] :relativeExistenceOf [ a :Entity; rdfs:label ?name ] ;
     :determinedBy [a v:costToRepair ; v:hasValue ?cost ] .
} ORDER BY ?cost
```

?name	?cost
"Gypsum"	"2000"
"Brick"	"3000"

Fig. 4. Query for the repair cost of materials of Brick and Gypsum and order them from the cheapest to most expensive via replacement cost, without considering dependencies.

```
SELECT ?basename ( SUM(?cost) AS ?totalcost )
      ( GROUP_CONCAT(?name; separator=",") AS ?reqs )
WHERE {
  { SELECT ?basere ?basename WHERE {
      VALUES ?basename { "Brick" "Gypsum" }
      ?basere :relativeExistenceOf [ a :Entity; rdfs:label ?basename ] .
    }
  }
  ?basere (^hasOrigin/hasTarget)* ?re .
  ?re :relativeExisenceOf [ a :Entity; rdfs:label ?name ];
      :determinedBy [a v:costToRepair; v:hasValue ?cost] .
} GROUP BY ?basename
```

?name	?totalcost	?reqs
"Brick"	"3000"	"Brick"
"Gypsum"	"4000"	"Gypsum, Insulation"

Fig. 5. Similar to the query in Fig. 4, but with dependencies considered.

5 Related Work

Useful background work exists such as representing geographical scale compatibility over space and time [11]. Many application-specific scaling solutions can be found that apply to both physical scale and temporal changes. For example, changing spatial scale has been captured relative to landscape patterns [14], generalizations of time-scale information and data for geological map services have been aggregated [10], and even ontology analysis of multi-scale modeling has been done for types of geographical features [15]. Visual processing technology also makes use of scale-space and spatio-temporal scale-space theory that assist in advancing applications in computer vision [9]. Furthermore, the specific and traditional concept of "scale" is well-thought through. For example, the integration of multiple domains and scales has been a specialized workshop topic due to needing to be reconcilable across many fields[9], and Spatial Information Theory [8] also documents some of these ideas quite well. For example, certain axioms are strongly related to those from cartographic map scaling [2], or a paper about size and grain being relative [12], or works such as a conceptual analysis of resolution [3]. These works determine scale and time via one or some applications of the concepts, instead of broadening this perspective to an extensible, multi-domain mechanism such as the RR pattern.

6 Conclusion

A dynamic data relationship pattern was presented in the context of a current research use case in the building industry domain, as well as suggested application integration possibilities. A dependency scenario was also provided for

[9] Integrating Multiple Domains and Scales: https://goo.gl/SaVFRi

dynamically determining, establishing, and tracking relative relationships. The Relative Relationship Pattern establishes hierarchies between different entities, idea conceptualizations, and data sets to efficiently track changes and dependencies within physical, temporal, and informational relationships. This means that application queries can be formulated to return data about individual "entities", but more importantly how two entities relate to each other and how those relationships-real or representational-can be tracked over time and through changes in state conditions. It can also potentially facilitate connections between knowledge graphs and decision frameworks that consume the data; this happens by creating an "upper" conceptual layer of information which can potentially enable more powerful inferencing and more accurate predictions.

Acknowledgments. Thank you to the Center for Research Computing (CRC) at the University of Notre Dame, and Jarek Nabrzyski, CRC Director, and also the Green Scale Project. Adila Krisnadhi acknowledges the support of the National Science Foundation under the award 1440202 "EarthCube Building Blocks: Collaborative Proposal: GeoLink – Leveraging Semantics and Linked Data for Data Sharing and Discovery in the Geosciences."

References

1. Blomqvist, E.: The use of semantic web technologies for decision support - a survey
2. Carral, D., et al.: An ontology design pattern for cartographic map scaling. In: The Semantic Web: Semantics and Big Data. No. 7882 in LNCS, Springer Berlin
3. Degbelo, A., Kuhn, W.: A conceptual analysis of resolution. In: Proceedings XIII GEOINFO. pp. 2012, 11–22
4. El Idrissi, B., Baina, S., Baina, K.: Automatic generation of ontology from data models: A practical evaluation of existing approaches pp. 1–12
5. Ferguson, H., et al.: Capturing an architectural knowledge base utilizing rules engine integration for energy and environmental simulations. SimAUD 2015: Proceedings of the Symposium on Simulation for Architecture and Urban Design pp.17-24
6. Ihab Hijazi, M.E.: Initial investigations for modeling interior utilities within 3d geo context: Transforming IFC-interior utility to CityGML/UtilityNetworkADE pp. Springer Berlin Heidelberg, 95–113, 2001
7. Ishizaka, A., Nemery, P.: Multi-criteria Decision Analysis: Methods and Software. John Wiley & Sons, Ltd., 1 edition edn.
8. Ittingen, Kartause: Spatial Information Theory : Foundations of Geographic Information Science. LNCS ; 2825, Springer: International Conference, COSIT 2003
9. Lindeberg, T.: Generalized axiomatic scale-space theory 178, 1–96
10. Ma, X.: Ontology-aided annotation, visualization, and generalization of geological time-scale information from online geological map services 40, 107–119
11. Pereira, G.M.: A typology of spatial and temporal scale relations 34(1), 21–33
12. Reitsma, F., Bittner, T.: Scale in object and process ontologies 2825, 13–27
13. Said Hamani, M.: Ontology-driven method for ranking unexpected rules. CIIA'09
14. Turner, M.G., O'Neill, R., Gardner, R.H., Milne, B.: Effects of changing spatial scale on the analysis of landscape pattern 3(3), 153–162
15. Yanhui, W., Xiaojuan, L., Huili, G.: Ontology-based analysis of multi-scale modeling of geographical features 49, 121–131, WOS:000244756700015