SEMANTIC AUGMENTATION OF GEOSPATIAL CONCEPTS: THE MULTI-VIEW AUGMENTED CONCEPT TO IMPROVE SEMANTIC INTEROPERABILITY BETWEEN MULTIPLES GEOSPATIAL DATABASES

M. Bakillah[1]*, M.A. Mostafavi[1], J. Brodeur[2]

[1]Centre de Recherche en Géomatique, Université Laval, Québec, Canada, G1K 7P4
[2]Centre d'information topographique de Sherbrooke, 2144 King, Canada
Mir-Abolfazl.Mostafavi@scg.ulaval.ca
Jean.Brodeur@RNCan-NRCan.gc.ca

Commission II, WG II/IV

KEY WORDS: Geospatial concept, View paradigm, Rule mining, Knowledge representation, Semantic Interoperability

ABSTRACT:

Semantic interoperability is a key issue for the meaningful sharing of geospatial data between multiples geospatial databases. It requires the establishment of semantic mappings between concepts databases' ontologies. Semantic mappings can be discovered only when semantics is explicit. However, existing concepts' definitions are not always sufficient to represent the semantic richness of geospatial concepts. In addition, semantics may be implicit, refraining from using it during semantic mapping process. This paper, proposes a new representation for geospatial concepts, called Multi-View Augmented Concept (MVAC), which takes into account these drawbacks. We propose a method to generate a MVAC, based on: (1) extraction of the different views of a concept that are valid in different contexts, and (2) augmentation of a concept with implicit dependencies between its features based on rule mining theory. We believe that the proposed approach will play an important role to improve the quality of the semantic interoperability between multiple geospatial databases since it takes into account the implicit semantic relations between different concepts.

1. INTRODUCTION

Semantic interoperability is a major research topic for ensuring data sharing and reuse among heterogeneous systems [Bian and Hu 2007]. It is the knowledge-level interoperability that provides cooperating databases with the ability to resolve differences in meanings of concepts [Park and Ram 2004]. Resolving those differences requires that meaning is available to machines into an explicit representation so it can automatically be processed during semantic mapping, that is, the discovering of semantic relations between concepts of different ontologies. However, current semantic mapping approaches rely on poor concepts' definitions that are not suitable for representing all the richness of geospatial concepts. For example, not considering explicitly the semantics of spatial and temporal properties of a concept reduces its expressivity. In addition, it may contain implicit knowledge that can be inferred from existing knowledge. The structure of the concepts is also important. Considering a concept as a bag of features is not sufficient. To address these problems, we propose a new representation of geospatial concepts, the Multi-View Augmented Concept Model (MVAC) (presented in section 3), and a method to generate MVAC representation (presented in section 4). In this method, we add two additional layers to the definition of the concept. First, we extract the different views it can have in different contexts, and then, we augment it with dependencies between its features. The contribution of the MVAC model is to improve semantic interoperability with a concept that has richer semantics, and a structure that allow discovering semantic relations between concepts of different ontologies that were hard to discover with traditional, lexical-based semantic mapping approaches. This paper is organized as follows. In section 2, we review related work on definition of concepts. In section 3, we propose the MVAC model. In section 4, we propose the MVAC generation method. In section 5, we discuss with a case study how the MVAC can help to improve semantic interoperability. In section 6, we conclude this paper.

2. RELATED WORK ON THE DEFINITION AND REPRESENTATION OF GEOSPATIAL CONCEPTS

Knowledge representation is the problem of encoding the knowledge that human have about the reality, in such a way that it supports reasoning [Kavouras and Kokla 2008]. A knowledge representation is not a complete and perfect picture of the reality; but an abstraction of a portion of reality that is relevant in a domain. Knowledge representation is a fundamental issue for improving semantic interoperability because it supports knowledge sharing (between humans and between machines). The theoretical basis of knowledge representations depends on the different theories of the concept. Cognitively, concepts are mental representations of a category [Medin and Rips 2005], and a category denotes a set of real world entities that have similar properties [Kavouras and Kokla 2008]. It is very difficult to give a framework that would guide the assignment of properties to concepts in a universal way, even if such attempts were made [Bennett 2005]. The choice of a concept's properties depends on the intended purpose [Tomai and Kavouras 2004]. In the geospatial domain, proposed definitions of the concept aim at identifying the special

* Corresponding author: mohamed.bakillah.1@ulaval.ca

properties of geospatial objects. Kavouras and Kokla [2008] define a concept with a term, a set of semantic elements (properties and relations) and their values. This is similar to the definition of the concept in Schwering and Raubal [2005] where concepts are defined by properties (represented as dimensions in a conceptual space) and property values (represented as values of those dimensions). Kavouras and Kokla have identified features such as: *purpose*, *agent*, *shape*, *size*, *location*; *frequency*, *duration*, *is-a*, *part-of* relations; *relative position relations* (upward, downward, behind, etc.); *proximity*, *direction* and *topological relations* (adjacency, connectivity, overlap, etc.). Rodriguez and Egenhofer [2003] have classified features as attributes, functions (representing what is done to or with an object) and parts (structural component of an object). This classification aims at facilitating the separate manipulation of each type of properties in the context of semantic similarity assessment. Brodeur and Bédard [2001] give another set-based definition of concepts. They proposed a definition based on the four-intersection model of Egenhofer [1993]. A concept has an interior, defined by its intrinsic properties (e.g. identification, attributes, attribute values, geometries, temporalities), and a boundary, defined by its extrinsic properties (e.g. relationships and behaviours). The whole set of intrinsic and extrinsic properties forms the *context*. Keßler et al. [2007] argue that the context has two components: the internal context specifies the domain of application and the external context is a set of rules that allows to modify the concept in different circumstances. Bennett [2005] has attempted to provide a generic definition of the concept. He proposes that properties of an object may be classified as physical (including geometry and material properties); historical (how the object came into existence; the events it has undergone, etc.); functional, including static and dynamic functions; conventional properties (related to the *fiat* nature of objects). Bennett mentions "objects that exhibit one property, will very often also exhibit another property", but he does not explicit further those types of dependencies between properties. A first problem with the above approaches is that they define the concept as unstructured set of features. However, features are related through dependencies. For example, position of a moving object depends on time, the value of an object's temperature depend on the value of its altitude, etc. However, if those dependencies are not stated in the concept's definition, it may be possible to discover implicit dependencies by looking in the instances of the concept. A second problem is that in most of the definitions, spatial and temporal properties are not explicit but merged into other classes of properties. This makes the separate manipulation of spatial or temporal properties difficult. Most approaches define properties with their name and range of values, for example, "geometry of house" is a "polygon". This is not sufficient to understand the exact semantics of this spatial property. The polygon may represent "roof of house" or "foundation of house". Spatial and temporal properties have to be described in a more explicit manner. Finally, there are different ways to define a concept depending on the context [Parent *et al*. 2006]. Several researchers have investigated the multi-view paradigm for concepts and propose modelling views in geospatial databases [Bédard and Bernier 2002; Parent et al. 2006] and in ontologies [Bhatt et al. 2006; Wouters et al. 2008]. Beside the strict representation issues, multiples views of a same concept can also provide multiple ways to achieve semantic interoperability. However, existing representation of geospatial concepts tend not to include this paradigm explicitly, nor to demonstate its usefullness in semantic interoperability.

3. THE MULTI-VIEW AUGMENTED CONCEPT (MVAC) MODEL

The new definition of the concept we propose is intended to address the drawbacks of concept definitions identified above, an its contribution is a more rich and structured definition as a basis for improved semantic interoperability. The MVAC adds two additional layers to the original definition of a concept: a set of views valid in different contexts, and a set of dependencies between features of the concept (Fig. 1).

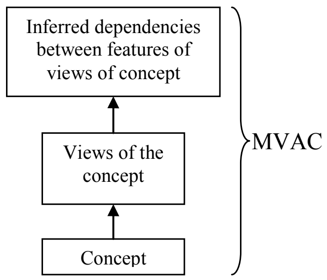

Figure 1. MVAC Model

At the first level, a concept, denoted by c, is defined as: c = <n(c), {p(rp)}, {r(rr)}, {spatial_d(rsd)}, {temporal_d(rtd)}>, where:

- n(c) is the name of the concept;
- {p(rp)} is the set of properties of the concept. The set of possible values of a property, called the range and denoted rp, is given in brackets.
- {r(rr)} is the set of relation that c has with other concepts. rr represents the range of the relation r, that is, the set of concepts c is linked with through relation r.
- {spatial_d(rsd)} is a set of properties, called spatial descriptors, which role is to describe the spatiality of the concept. For example, the concept watercourse could have the spatial descriptor geo-entity (axis of watercourse), meaning that the line geometry representing the watercourse corresponds to the axis of the watercourse. The range od spatial descriptor is denoted rsd.
- {temporal_d(rtd)} is a set of properties, called temporal descriptors, which role is to describe the temporality of the concept. The range of temporal descriptors is denoted rtd. For example, the concept watrecourse may have temporal descriptor waterlogged period (average flooded period) which means that the waterlogged period correspond to the average time the watercourse is flooded overs years.

We provide an example for the concept “watercourse”:

c = <watercourse, {flooding, tourism, transport}, {water level(low, medium, high), category(intermittent, stable), spatial extent(polygon, moving polygon), function(navigation, skating, evacuation area), state(frozen, unfrozen)}, {Connect(Waterbody)}, {geo-entity(bed of watercourse, flooded area, frozen area)}{waterlogged period(average flooding period)}>

This concept may represent different realities in different contexts. For each context, we want to create a view that can be used in that context. In a previous work (Bakillah et al. 2009) we have stated that the view paradigm support ontology reuse, by selecting only parts of a concept that are relevant in a given context. We have defined views as the result of inference over logic rules. We precise that views are inferred from rules on context. A view of a concept is a selection of its features that are valid in a given context. The context represents a given real world situation, for example, a disaster. A view is defined as:

View(c): Context(Name of context) → $<\{p(rp_v)\}, \{r(rr_v)\}, \{spatial\_d(rsd_v)\}, \{temporal\_d(rtd_v)\}>$

This expression means that in the named context, the concept c takes its value for a property, a relation or a descriptor in a restricted range rp_v, rr_v and rsd_v, rtd_v respectively. For example, two possible views of the concept watercourse are:

Context(flooding) → function (watercourse, evacuation area)
Context(tourism) → function(watercourse, [navigable, skating])

Meaning that in the context of a flooding, the watercourse has the function of evacuation area to allow boats rescuing people. A view is a spatial view when the condition is imposed on a spatial property, a spatial relation (topology, proximity, orientation) or a spatial descriptor:

Spatial View: Context(Name of context) → spatial property (concept, value of spatial property)
Spatial View: Context(Name of context) → spatial relation (concept, range of spatial relation)
Spatial View: Context(Name of context) → spatial descriptor (concept, value of spatial descriptor)

A view is a temporal view when the condition is imposed on a temporal property, a temporal relation or a temporal descriptor:

Temporal View: Context(Name of context) → temporal property (concept, value of temporal property)
Temporal View: Context(Name of context) → temporal relation (concept, range of temporal relation)
Temporal View: Context(Name of context) → temporal descriptor (concept, value of temporal descriptor)

Besides views, dependencies between features can be inferred to semantically augment a concept. Dependencies express that a first feature's values are related to a second feature's values. For example, property "temperature" depends on property "altitude". We formalize dependencies with rules head → body. The body in the rule is a consequence of the head. Here are examples of thematic, spatial and temporal rules respectively:

Altitude(land, low)→ FloodingRisk(land, high)
Width(watercourse, larger than 7m)→ Geometry(surface)
Flooding frequency(land, more than twice a year)→ Status(land, periodically waterlogged).

Dependencies are rarely represented. However, they may be implicit in the concept's instances; for example, cities with similar values of average temperature have similar values of altitude. The concept, the views and the augmented dependencies form the MVAC:

$c^{MVA} = <n(c), \{p(c)\}, \{r(c)\}, \{spatial\_d(c)\}, \{temporal\_d(c)\}, \{v(c)\}, \{ctx\}, \{dep(c)\}>$

where {v(c)} is the set of views, {ctx} is a set of different contexts for the concept, and {dep(c)} is the set of augmented dependencies. The methodology that will augment a concept to a MVAC is composed of two main methods, a view extraction method, and a method to discover dependencies.

4. MVAC GENERATION METHOD

We have developed this method to transform a concept into a MVAC. The method integrates view extraction paradigm, mining rules techniques and ontology reasoning principles. Fig. 2 shows the MVAC generation method. It consists of two phases: 1) the *view extraction* phase, 2) the *augmentation* phase. The method takes as input an ontology with original concepts as defined in section 3. The first step involves the user in specifying the context extraction rules.

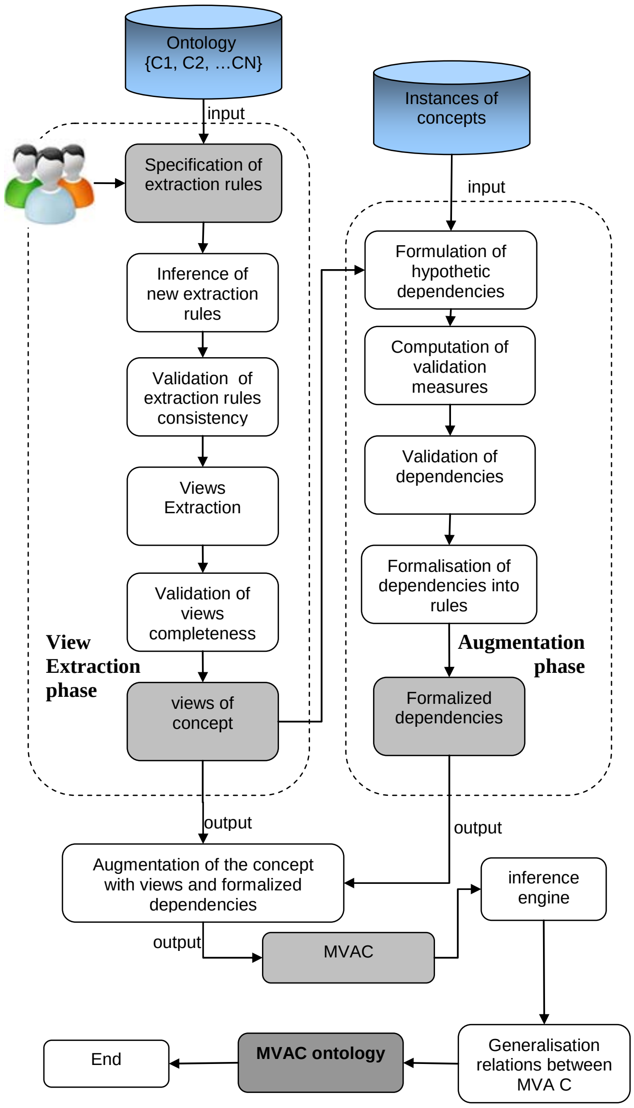

Figure 2. MVAC and Ontology Generation Method

Step 1. Specification of Context Extraction rules. This step requires interaction between users and the view extraction algorithm. The users specify with context rules the values of the properties, relations and descriptors of a concept that are valid in a context. For example, considering the concept "watercourse" with properties "depth" and "category of watercourse", the user specifies their possible values in the context of dryness:

Context(dryness) → water level(watercourse, low) (rule 1)
Context(dryness)→ category of watercourse (watercourse, intermittent) (rule 2)

The contexts of a concept are inferred from those rules.

Step 2. Inference of new Extraction rules. Having a set of extraction rules on contexts of the concept, we verify if new extraction rules can be inferred by combining them. We also use other existing rules that are part of the ontology, and which represent the knowledge of domain experts. This is a way of reusing the existing knowledge to produce new one. The inference of new extraction rules (1) takes as input the extraction rules specified in step 1, plus the rules that are part of the ontology, (2) send them to an inference mechanism, (3) produces new inferred rules, and (4) restart the cycle from (1) to (3) until no new rules are inferred. The inference mechanism determine that if the body of a rule implies the head of a second rule, then the head of the first rule implies the body of the second rule. For example, consider a rule saying that intermittent watercourse are represented with moving polygon: Category of watercourse(watercourse, intermittent) → geometry(watercourse, moving polygon). From this rule and the ones that were specified by the user in step 1, we can infer the following new rule: Context(dryness) → geometry(watercourse, moving polygon) (rule 3). New inferred rules are added to the set of rules that will be used to extract views of the concept.

Step 3. Validation of extraction rule consistency. Before using those rules to extract the views of a concept, we verify if the inferred rules are correct, that is, if they are consistent with the reality. In this case, the reality corresponds to the instances of the concept, which are representation of real world objects stored in the database. To verify is the rules are correct, we assess the consistency between the rules and the instances. Consistency can be defined as the degree of consistency of the data with respect to its specifications (Mostafavi et al. 2004). In our context, data corresponds to instances whereas specifications correspond to rules (since rules define the semantic). Therefore, a rule is consistent if the instances of the concept verify this rule. For example, if we have a rule Context(dryness) → water level(watercourse, low), we verify if instances of the concept "watercourse" which have the context "dryness", also have "low water level". To determine whether an extraction rule is consistent enough, we propose a ratio that will compare the number of instances that respect the rule (denoted with |verifying instances|) with the total number of instances which have for context the one indicated in the rule (denoted with |targeted instances|). Only those rules that have a sufficient degree of consistency are used for view extraction:

$$\text{Degree of consistency } = \frac{|\text{verifying instances}|}{|\text{targeted instances}|} \tag{1}$$

Step 4. View Extraction. View extraction, as we have defined in (Bakillah et al. 2009), includes two main steps, the extraction of partial views and the merging of partial views. First, in the extraction of partial views, each extraction rule is applied to the concept to create the subconcept that will always respect this rule. For example, for the concept watercourse defined in section 3, applying rule 1 gives the following partial view:

Partial view: Context(dryness) → <watercourse, {water level(low), category(intermittent, stable), spatial extent(polygon, moving polygon), function(navigation, skating, evacuation area), state(frozen, unfrozen)}, {Connect(Waterbody)}, {geo-entity(bed of watercourse, flooded area, frozen area)}{waterlogged period(average flooding period)}>

This partial view imposes a restriction only on the values of property "water level". In the second step of the view extraction, all partial views that pertains to a same context and that are non contradicting are merged into a single view. This is the partial view merging process. For example, merging partial views generated by rule 1 to 3 would lead:

view: Context(dryness) → <watercourse, {water level(low), category(intermittent), spatial extent(moving polygon), function(navigation, skating, evacuation area), state(frozen, unfrozen)}, {Connect(Waterbody)}, {geo-entity(bed of watercourse, flooded area, frozen area)}{waterlogged period(average flooding period)}>

During the view extraction, relations between views of a concept and other concepts of the ontology are inherited from the definition of the concept when it applies; for example, the above view is linked to the concept "waterbody" with the spatial relation "connect".

Step 5. Validation of view completeness. When all views of a concept are created, we verify if they are complete, that is, the union of all views of the concept result in the concept itself. The restricted range of a property p_i (or relation R_i, descriptor d_i) in a view v_j is r_{ij}. The view completeness can be validated if the following generic expression is verified: $c = < n(c), \{p_1(r_{11} \cup r_{12} \cup r_{13} ...), ... p_n(r_{n1} \cup r_{n2} \cup r_{n3} ...) \}, \{R_1(r_{11} \cup r_{12} \cup r_{13} ...), ... R_n(r_{n1} \cup r_{n2} \cup r_{n3} ...) \}, \{d_1(r_{11} \cup r_{12} \cup r_{13} ...), ... d_n(r_{n1} \cup r_{n2} \cup r_{n3} ...) \}>$, that is, by taking, for all features of the concept, the union operator on the restricted ranges of all views of the concept. The next steps are about augmenting the concept (with its views) with implicit dependencies.

Step 6. Formulation of possible dependencies. Possible dependencies are dependencies that have to be verified against data. For every view of a concept, our method formulates dependencies that express relations between each pair of their features (properties, relations or descriptors). Those dependencies are expressed as rules. For example, for a concept "watercourse" with properties "state (frozen, unfrozen)" and "function (skating, navigable)", we can have:

"If *state* of watercourse = frozen, then *function* = skating"
"If *state* of watercourse = frozen, then *function* = navigable"
"If *state* of watercourse= unfrozen, then *function* = skating"
"If *state* of watercourse= unfrozen, then *function* = navigable"

"If *function* of watercourse = skating then *state* = frozen"
"If *function* of watercourse = skating then *state* = unfrozen"
"If *function* of watercourse = navigable then *state* = frozen"
"If *function* of watercourse = navigable then *state* = unfrozen"

Because the number of possible dependencies may be high, they can be classified (the first series being classified as "function depends on state" rules, and the second as "state depends on function" rules) so that the user can reject the ones that seems non-verifiable. Once we have formulated a set of possible dependencies, we have to validate which ones are true among instances of a view.

Step 7. Computation of rule validation measures. For each rule expressing a possible dependency, we determine the values of two measures that will help to determine if we can retain it as a valid dependency. Those measures, which are *support* and *confidence*, are adapted from the rule-mining domain, which aims at finding correlations between items in datasets (Ceglar and Roddick, 2006). The support measure how many instances respects either the head (Ihead) or the body(Ibody) of a rule, with respect to the total set of instances (Itotal), and the confidence measures how many instances respect the body of the rule among those that respect the head of the rule:

$$\text{Support} = \frac{|\text{Ihead} \cup \text{Ibody}|}{|\text{Itotal}\ |} \tag{2}$$

$$\text{Confidence} = \frac{|\text{Ibody}|}{|\text{Ihead}|} \tag{3}$$

Step 8. Validation of dependencies. For the validation of dependencies, we choose those dependencies for which support and confidence are satisfying. Those measures complete each other since a high confidence but a low support means while this rule is usually respected, it is not frequent in the instance set, so it may be less interesting.

Step 9. Formulation of dependencies into rules. If the rule checked in step 4 is determined to be true, then it is added to the definition of the view in a form: Feature 1(concept, value of feature 1) → Feature 2(concept, value of feature 2).
Now that views and dependencies are extracted, the concept's definition is rewritten with those new elements. However, relations between views and augmented concepts need to be re-computed to form the MVA ontology.

Step 10. The inference of Relations. Views needs to be linked together by generalisation/specialisation relations to create the MVA ontology. Those links are established between the different views of a same concept, and between views of different concepts. Generalisation is when the instances of a first view /concept include all instances of a second view/concept. To perform this task, we can, for example, express MVACs with OWL-DL language and use subsumption-reasoning mechanism provided by reasonign engines. For example, if we have the following view:

View1: Context(dryness) → <watercourse, {water level(low), category(intermittent), spatial extent(moving polygon), function(non navigable, skating), state(frozen, unfrozen)}, {Connect(Waterbody)}, {geo-entity(bed of watercourse, frozen area)}{waterlogged period(average flooding period)}>,

it would generalise the following view:

view2: Context(dryness in summer) → <watercourse, {water level(low), category(intermittent), spatial extent(moving polygon), function(non navigable), state(unfrozen)}, {Connect(Waterbody)}, {geo-entity(bed of watercourse,)}{waterlogged period(average flooding period)}>

which represents a smaller number or real world objects. Therefore, views can be categorised within the MVA ontology.

5. CASE STUDY

Having defined the MVA model and a method to generate it from an existing concept, we aim to show with the following examples that the MVAC can help to improve semantic interoperability. Consider the user of a geospatial database which ontology contains the following concept "watercourse":

C1: <**watercourse**, {water level(low, high), spatial extent(polygon, moving polygon), function(navigable, non navigable}, {Connect(Waterbody)}, {geo-entity(bed of watercourse, waterlogged area)}>

Suppose that this user search a network of geospatial databases for "watercourses" in the context of "dryness".

Consider the concept "stream" which is included in the ontology of another database of the network.

C2: <**stream**, {depth(low, high), spatial extent(surface, moving surface), role(navigable, non navigable)}, {Meet(Lake)}, {geo-entity(bed of watercourse, waterlogged area)}>

First, with no views being defined, and therefore no contexts being specified, we are unable to find if "stream" and "watercourse" can be in a similar context of "dryness". With a lexical matching approach, we would however find pairs of synonyms: "watercourse" ↔ "stream", "polygon"↔ "surface", "connect" ↔ "meet", "waterbody" ↔ "lake", "function" ↔ "role". With semantic mapping rules such as those that were presented in (Bakillah et al. 2009), we would find that "watercourse" overlap "stream", but note that we would be unable to identify that water level corresponds to depth since those properties are not lexically related. Now consider that we employ the MVA generation method we have developed and we build MVACs for "watercourse" and "stream". Suppose we have extracted two views for the concept watercourse, corresponding to contexts dryness, and flooding:

MVAC1: Watercourse
View1(watercourse): Context(**dryness**) → {water level(low), spatial extent(polygon), function(non navigable}}, {Connect(Waterbody)}, {geo-entity(bed of watercourse)}>
View2(watercourse): Context(**flooding**) → <watercourse, {water level(high), spatial extent(moving polygon), function(navigable)}, {Connect(Waterbody)}, {geo-entity(waterlogged area)}>.
In addition to the following dependencies being extracted for "watercourse":
{(d1:water level(watercourse, low)→ function(watercourse, not navigable), (d2:water level(watercourse, high)→ function(watercourse, navigable)}

For the concept "stream", we have for example extracted:

MVAC2: Stream
View1(stream): Context(**lack of rain**) → <**stream**, {depth(low), spatial extent(surface), role(non navigable)}, {Meet(Lake)}, {geo-entity(bed of watercourse)}>
View2(stream): Context(**rain season**) → <**stream**, {water level(high), spatial extent(moving surface), role(navigable)}, {Meet(Lake)}, {geo-entity(waterlogged area)}>
And the following dependencies:
{d3:(depth(stream, low)→ role(stream, not navigable), (d4:depth(stream, high)→ function(stream, navigable)}.

We show how the MVAC will enable to improve answering to the user query by detecting implicit matches using the structure of the MVAC. After having deduced the lexical matches indicated above, comparing the different dependencies of C1 and C2, we find that d1 has the same structure as d3, and d2 the same structure as d4, which allow proposing the following match: Water level↔Depth. We were able to find this match only because we augment the concept with dependencies that brings a richer structure. Comparing the contexts of the different views of “watercourse” and “stream” from a lexical-based approach does not allow finding that “lack of rain” corresponds to “dryness”. However, if we compare the definitions of **View1**(stream) and **View1**(watercourse), knowing the previous matches, we find that **View1**(stream) is equivalent to **View1**(watercourse), which allow to propose the following match: Context(**lack of rain**) ↔ Context(**dryness**). This allows the user finally to retrieve “stream” as a concept similar to “watercourse” in the context of dryness. This example shows that augmenting the concept with new structures (views and dependencies) can help to match concepts, contexts or features of concepts that seems dissimilar, and supports improving semantic interoperability between geospatial databases.

6. CONCLUSIONS

In this paper, we have argued that for improving semantic interoperability approaches, one main problem is the poor definition of concepts. This is especially true regarding the geospatial domain where concepts are defined by spatial and temporal features, in addition to multiple contexts and implicit dependencies between features. To address this issue, we have proposed the Multi-View Augmented Concept Model (MVAC), and a MVAC generation approach that includes a view extraction and semantic augmentation methods. We have shown that with the MVAC, we can improve semantic interoperability because we can discover more semantic relations between concepts of different ontologies. Therefore, the MVAC can play an important role in a global semantic interoperability approach designed for ad hoc networks where ontologies of databases are very heterogenous, such as in disaster management and in environmental and health domains. The future research will consider the MVAC as a basis for such an approach, with the goal of developing a semantic interoperability approach that is adapted to the MVAC model, since the quality of semantic interoperability depends on the ability of the semantic mapping approach to consider all the characteristics of the input concepts (Bakillah et al. 2008).

References

Bakillah, M., Mostafavi, M.A., Brodeur J., Bédard, Y., 2009. Sim-Net: a View-Based Semantic Similarity Model for ad hoc Network of Geospatial Databases. *Transactions in GIS*, 13(5-6), 417-447.

Bakillah, M., Mostafavi, M. A., Brodeur, J., and Bédard, Y., 2008. *Elements of Semantic Mapping Quality*. Boca Raton, FL, CRC Press, 37–45.

Bhatt, M., Flahive, A., Wouters, C., Rahayu, W., Taniar, D., 2006. MOVE: A distributed Framework for Materialized Ontology Views Extraction. *Algorithmica*, 45(3), 457-481.

Bédard, Y., E. Bernier & R. Devillers, 2002. La métastructure VUEL et la gestion des représentations multiples. *Généralisation et Représentation multiple*, Hermès, 150-162.

Bennett, B., 2005. Modes of concept definition and varieties of vagueness. *Applied Ontology*, 1(1), 17–26.

Bian, L., Hu, S., 2007. Identifying Components for Interoperable Process Models using Concept Lattice and Semantic Reference System. *International Journal of Geographical Information Science*, 21(9), 1009-1032.

Brodeur, J., Bédard, Y., 2001. Geosemantic Proximity, a Component of Spatial Data Interoperability. International Workshop on "Semantics of Enterprise Integration" ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, Tampa Bay, Florida.

Ceglar, A. and Roddick, J. F., 2006. Association mining. *ACM Computing Surveys*, 38(2), 5.

Egenhofer, M., 1993. A Model for Detailed Binary Topological Relationships. *Geomatica*, 47, 261-273.

Kavouras, M., Kokla, M., 2008. *Theories of geographic concepts*. CRC Press, Taylor & Francis Group.

Keßler, C., Raubal, M., Janowicz, K., 2007. The Effect of Context on Semantic Similarity Measurement. *On the Move to Meaningful Internet Systems 2007,* 1274-1284.

Medin, D.L., Rips, L.J., 2005. Concepts and Categories: Memory, Meaning and Metaphysics, Concepts and Categorization. *The Cambridge Handbook of Thinking and Reasoning*. Ed. K.J. Holyoak and R.G. Morrisson, 37-72. Cambridge, UK: Cambridge University Press.

Mostafavi, MA., Edwards G., Jeansoulin, R., 2004. Ontology-based Method for Quality Assessment of Spatial Data Bases. In: *Proceedings of ISSDQ'04*, GeoInfo Series, Bruck/Leitha, Austria, 49-66.

Parent, C., Spaccapietra, S., Zimanyi, E., 2006. The MurMur Project: Modeling and Querying Multi-Representation Spatiotemporal Databases. *Information Systems*, 31(8) 733-769.

Park, J., Ram, S., 2004. Information systems interoperability: what lies beneath? *ACM Transactions on Information Systems*, 22(4), 595-632.

Rodriguez, A., Egenhofer, M., 2003. Determining Semantic Similarity Among Entity Classes from Different Ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15 (2), 442–456.

Schwering, A., and Raubal M., 2005. Measuring Semantic Similarity Between Geospatial Conceptual Regions. Proceedings of the First International Conference on Geospatial Semantics, Mexico City, Mexico.

Tomai, E., Kavouras, M., 2004. From "Onto-GoeNoesis" to "Onto-Genesis": The Design of Geographic Ontologies. *GeoInformatica*, 8(3), 285-302.

Wouters, C., Dillon, T.S., Rahayu, W., Meersman, R., Chang, E., 2008. Extraction Process Specification for Materialized Ontology Views. T.S. Dillon et al. (Eds.): *Advances in Web Semantics I*, LNCS 4891, 130–175.

Acknowledgements

This research supported by an operating grant from Natural Sciences and Engineering Research Council of Canada (NSERC).