A COGNITIVE APPROACH FOR LANDSYSTEM IDENTIFICATION USING A GRAPH DATABASE - TOWARDS THE IDENTIFICATION OF LANDFORMS IN CONTEXT

Hariniaina Ramiaramanana[a,*], Eric Guilbert[b], Bernard Moulin[c].

a Dept. of Geomatics Sciences, Laval University, Québec, G1V 0A6 (QC) Canada - hariniaina.ramiaramanana.1@ulaval.ca
b Dept. of Geomatics Sciences, Laval University, Québec, G1V 0A6 (QC) Canada - eric.guilbert@scg.ulaval.ca
c Professor Emeritus, Dept. of Computer Science and Software Engineering, Laval University, Québec, G1V 0A6 (QC) Canada - bernard.moulin.1@ulaval.ca

Commission IV, WG IV/2

KEY WORDS: landsystem, landform identification, knowledge modelling, graph database, knowledge-based system.

ABSTRACT:

A landform is any physical feature of the earth's surface having a characteristic, recognizable shape. Most landform identification methods rely on OBIA (Object-Based Image Analysis) techniques to segment the terrain data and classify segments into objects that are assumed to compose the landform. However, geomorphologists can visually recognize any landform, considering the characteristics of the surrounding environment that plays the role of context. This notion of context was not considered in previous landform identification methods. We propose to model it using the notion of landsystem. Landsystems are geomorphologic elements that result from a set of natural geomorphological processes. They are also easily recognized by geomorphologists. In this paper, we present a new knowledge-based method to automatically identify landsystems as the context for landform identification. We first present a conceptual model as a core ontology of geomorphologic elements including landsystems and landforms, capturing relevant geomorphologists' knowledge. Then, we present how this model is extended to create a domain ontology for a chosen domain in geomorphology. We illustrate such an extension for the case of mountainous glacial valleys. We used the graph database engine Neo4J to implement the domain ontology and to develop a knowledge-based system (a framework) to automatically identify landsystems from spatial datasets. We present the architecture of our framework and discuss how it is used to support: 1) the knowledge acquisition tasks; 2) the spatial data preparation task; 3) the processing of the user's request seeking landsystems in a chosen study area.

1. INTRODUCTION

A landform is any physical feature of the earth's surface having a characteristic, recognizable shape (Schoeneberger and Wysocki, 2017) as for example mountains, valleys or canyons. Geomorphologists easily recognize landforms on the earth surface and can associate them with natural processes that shaped the landscape over geological times. Landforms can be characterized by typical attributes (soil characteristics, morphometry, etc.), called 'structural elements' that specialists can recognize in any region where these processes occurred.

Another way that specialists define landforms is through the identification of 'landform elements' such as hill-top, shoulder and backslope which are parts of a landform. Most current (semi) automated landform identification methods rely on image analysis to identify landform parts. The OBIA approach involves two successive steps: 1) segmentation of the terrain data (raster images) into clusters of pixels (called segments) corresponding to landform elements; and 2) classification of segments into objects that are assumed to compose the sought landform (Blaschke, 2010). Such methods are usually designed for particular landform categories and scales; they also require that the practitioner possess specific skills in image processing (Eisank et al., 2011; MacMillan and Shary, 2009). Indeed, the results of the segmentation process depend on a proper selection of features and rely on the choice of threshold values for feature classification. Hence, segment classification is often subjective and based on trial and error.

Indeed, these methods do not reflect the way geomorphologists identify landforms in practice. While observing the geographical environment, these experts are trained to recognize geomorphologic elements, salient features and typical patterns that characterize landforms and their context of appearance. They often raise hypotheses about the natural processes that shaped the observed landscape over geological times. Hence, they identify a landform by considering the geographical and geomorphological context. Since OBIA tools are based on pixel clustering, they cannot carry out the geomorphologist's contextual identification of landforms. Eisank et al. (2011) emphasized that, to enhance landform classification approaches, it is essential to develop and apply structured knowledge models that capture domain experts' knowledge in a formalized way. In this paper we propose a new approach to formalize geomorphologists' contextual identification of landforms using the notion of landsystem.

A landsystem is defined as a "sub-division of a region into areas sharing common physical attributes (e.g. geomorphic elements, geological characteristics, soils, vegetation) which are different from those of adjacent areas" (International Association of Geomorphologists, 2014). It can also be defined as a repeated landform pattern that results from a set of natural processes (Evans, 2012). For example, a mountainous region is a landsystem in which a geomorphologist can identify various landforms such as mountains and valleys and their component parts such as summits and talwegs, called landform elements. Moreover, a mountain range is considered as a landsystem contained in the more global *mountainous region landsystem*. As for landform recognition, a geomorphologist can recognize a landsystem by considering a set of particular characteristics, called structural elements, which are typical of the geological and geomorphologic processes that created or modified the observed area (identified as a particular landsystem) and the embedded landforms (Cooke and Doornkamp, 1990). Indeed, a landsystem

---

[*] Corresponding author

https://doi.org/10.5194/isprs-annals-V-4-2022-17-2022

provides the spatial and geomorphologic contexts in which landforms are observed. Depending on the granularity of the observation, a landsystem may contain other embedded landsystems and/or landforms: this can be represented as a hierarchy of embedded landsystems, a landform being considered as an elementary landsystem.

The first contribution of this paper is the proposal of a new knowledge-based method based on the construction of a conceptual model that: 1) depicts how landsystems are organized from structural elements observed by geomorphologists; 2) describes the various relationships associating landsystems, landforms and other relevant geomorphologic elements.

The second contribution is a knowledge-based system (that we call a framework) that can be used to automatically identify landsystems from a spatial dataset. The knowledge base is implemented using the graph database Neo4J that is also used to interface various spatial data sources managed in a GIS (Geographical Information System). The proposed conceptual model is generic and can be adapted to different application domains. The knowledge base can also be easily extended by adding new concepts to describe new categories of landsystems.

Section 2 sets the background of this research and presents an overview of the proposed method. Section 3 presents the first phase (generic knowledge acquisition) of the method and the resulting conceptual model. Section 4 presents the architecture of the software framework which supports our approach. The representation phase which translates the concepts into a graph database is presented in Section 5. Section 6 presents the operational phase where landsystem instances are identified from the data. Section 7 illustrates the application of the method to the case of mountainous glacial valleys. In Section 8, we discuss our proposal and conclude the paper with some perspectives.

2. OVERVIEW OF THE PROPOSED METHOD

2.1 Background

Earth surface features, also called geomorphologic element (Drăguţ and Blaschke 2006) or geomorphologic features, result from the action of a set of geomorphologic processes through long periods of time (Davis, 1909). Measurement and analysis of land surface and its features are studied in geomorphometry and mostly relate to landforms. Most methods developed for landform detection in geomorphometry rely on image analysis and use OBIA techniques to segment the terrain in landform elements which are then assembled to create so-called objects that lead to the identification of landforms. One practical limit of these quantitative methods is the need to identify significant features to describe landforms and to set prescriptive parameter thresholds to control the image analysis. A new generation of OBIA systems use machine learning (ML) and/or deep learning (DL) techniques to detect landforms (Ma et al., 2019). These techniques are data-driven (Arvor et al., 2019) and the systems need to be trained on large data sets to identify specific types of landforms. Indeed, these systems must be adjusted (if not reprogrammed) and trained on other large data sets to detect new kinds of landforms. Since landforms are usually vague objects with an unclear delineation, DL approaches may yield mitigated results (Arundel et al. 2020) which do not fit with what users would perceive as landforms (Arundel and Sinha, 2019).

Moreover, these techniques do not explicitly take into account the geomorphological contexts which are familiar to geomorphologists. In (Arvor et al., 2019) it is stated that "in current approaches the expert knowledge mobilized by environmental scientists […] tends to be somehow discarded from the data-driven image analysis". Therefore, these approaches are rarely transferable. These authors consider that knowledge-driven approaches should remain one of the most important directions of research in OBIA. They therefore advocate for the integration of expert knowledge and point out several benefits of using a knowledge base: the association of high-level concepts with low-level data; knowledge sharing through the adoption of common concepts; and the inference of new knowledge from explicit descriptions. Previously, (Dehn et al. 2001) emphasized that landforms are described mainly in two different ways: either *based on their geometry* as in quantitative approaches, or *based on their semantics* used to express and capture geomorphologists' mental models. Moreover, knowledge-based approaches are based on the recognition of geomorphologic elements from semantics, which refers to the user's perception of the real world (Rishe, 1992).

2.2 Overview

Geomorphologists recognize landforms by identifying their typical characteristics and spatial relations, considering their context of appearance. Deng (2007) stressed the need to consider the topographic context when identifying terrain units. In our approach we model such a context using the notion of landsystem for the following reasons. Since landsystems are defined as homogeneous regions differing from adjacent landsystems, they provide a natural decomposition of the terrain in which the effects of different processes and the resulting landforms can be observed. Consequently, a landsystem can be thought of as a terrain unit defining a uniform context for all elements contained in it. These elements have been shaped by similar geological and geomorphologic processes; and geomorphologists recognize this evolution by identifying a common set of structural elements (Cooke and Doornkamp, 1990).

Because processes occur in regions and alter the features within these regions, landforms do not exist in isolation: they form some patterns that result from these processes. *Hence, a landform is identified as a component of a homogeneous region that is recognized as a landsystem*. Therefore, studying landsystems allows geomorphologists to understand the processes occurring in them and the evolution of each constituent landform, leading to their identification. Moreover, thanks to qualitative studies of the relief, geomorphologists can associate to each landsystem category a list of potential landforms that can be found in them. For all these reasons, we chose to explicitly introduce the notion of landsystem in the conceptual model that is the foundation of our knowledge-based approach towards the contextual identification of landforms.

Knowledge-based approaches proceed first by acquiring relevant domain knowledge from experts; then by formalizing it in a conceptual model (or an ontology) which provides an unambiguous definition of the concepts (Guilbert, and Moulin, 2017) as well as their semantic relations. The conceptual model is then transformed into data structures that are the foundations of the software to be implemented. In accordance with these principles, our knowledge-based method aims to transform geomorphologists' knowledge into models and algorithms that can identify landsystems from data sets stored in a GIS. Moreover, our approach can be used to enhance OBIA methods by providing the notion of context that is not present in such methods. The method is composed of three phases.

· The *Conceptual Phase* (presented in Section 3) is the initial knowledge acquisition step carried out with expert geomorphologists with the aim to create a generic conceptual model for landsystem identification.

· The *Representation Phase* (presented in Section 5) is supported by the knowledge-based framework described in Section 4. The generic conceptual model is transformed into a knowledgebase (stored as a graph database) including the definitions of landsystem and of structural elements.

. The *Operational Phase* (presented in Section 6) enables a user to use the framework to identify landsystems in areas she has selected on a map. The results are presented as a vectorized map.

3. GENERIC CONCEPTUAL MODEL

The ***Conceptual Phase*** aims at collecting geomorphologists' knowledge for landsystem recognition in order to set up a generic conceptual model that can be used to recognize any kind of landsystem for landform identification. Acting as knowledge engineers, we carried out this knowledge acquisition stage (Milton, 2007) in collaboration with expert geomorphologists. For generality sake, we worked on different geomorphological domains (i.e. glacial valleys, deltas, canyons) and exploited the relevant scientific literature in geomorphology. The resulting conceptual model (Figure 1) can be thought of as a core ontology (Scherp et al., 2011) for the characterization of landsystems and landforms: it provides the ontological foundation for the construction of the knowledge-based system presented in Section 5. The conceptual model (Figure 1) structures geomorphologists' knowledge about landsystems and the associated spatial and geomorphologic concepts as well as the relations between them.

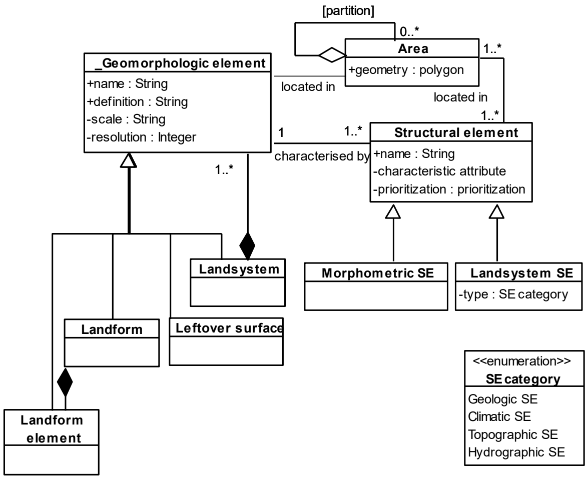


Figure 1: Generic conceptual model

Below we discuss three main principles underpinning our model. (1) A **geomorphologic element** is an element of the terrain *characterized by* a set of **structural elements** (SE) that allow geomorphologists to recognize it. Depending on the type of structural elements and the level at which geomorphologists observe them, geomorphologic elements can be of four kinds: **landsystems**, **landforms, landform elements** and **leftover surfaces.** Landsystems mainly refer to 'sub-divisions of a region into areas characterized by common physical attributes (e.g. geology, soils, vegetation, etc.) which are different from those of adjacent areas' (International Association of Geomorphologists, 2014). **Structural elements** (SE) are important in our model since they are the observable marks that geomorphologists rely on when performing terrain analysis. **Landsystem SE** are indicators that geomorphologists look for to recognize landsystems. **Landform elements** (MacMillan and Shary, 2009) are regions of homogenous morphometry (e.g. flat or regular slope or curvature). Hence, they are characterized by **Morphometric SE.** A landform is both a feature that is defined with regards to its form and to its origin, cause or history: it can be described as: 1) a part of a landsystem, whose SEs describe this history; 2) an assemblage of landform elements whose SEs describe the shape (MacMillan and Shary 2009). Landforms do not have to cover the whole land surface. There are "bits left between landforms" that are often forgotten in classifications (Evans, 2012): the leftover surfaces. We consider a **leftover surface** as a geomorphologic element and define it as 'an area that is contained within a landsystem area and that is located between delimited landforms'.

The relations between the different types of geomorphologic elements are represented in the UML structure of our conceptual model (Figure 1) by an inheritance relation.

(2) **Hierarchy** is a fundamental property of landsystems (MacMillan and Shary, 2009). The hierarchical relation associating landsystems of different levels is the *containment relation*. All geomorphologic elements (landsystem, landform, landform element, leftover surface) are *contained in* a landsystem at some level in the hierarchy of landsystems. This relation allows for the definition of landsystems at different levels of detail (granularity): a landsystem contained in a larger landsystem inherits the structural elements of its parents. This definition is systematically used in the model to distinguish specific landsystems from more generic ones. This relation also applies to landforms that are contained inside a landsystem and to landform elements that are contained in landforms. The relation is shown by a black diamond in Figure 1.

(3) **Understanding space** is a prerequisite to understanding geomorphologic elements. Landform analysis approaches should aim at partitioning the spatial continuum into units related to the geomorphologic elements (Dikau, 1999). Each geomorphologic element is *located in* a specific *area*. Areas can be *included in* other areas. The areas of all geomorphologic elements contained in a landsystem form a *partition*. Notably, the areas (associated with landsystems) partition the space in which geomorphologic elements (embedded landsystems and landforms) are located. Thanks to the introduction of leftover surfaces, landforms and leftover surfaces form a true partition of the geographical space. For example, a mountainous glacial valley landsystem is defined by a linear depression in a mountainous and glacial region. Indeed, geomorphologists recognize them by identifying a set of characteristics such as the linear depression, remnants of glaciations and the mountainous topography.

To solve interoperability issues between GIS systems used by geomorphologists, Löwner (2013) proposed a conceptual model in which are represented: 1) 'geomorphic objects' (akin to landforms) as 3D objects that evolve in time; 2) interactions between geomorphic objects and geomorphological processes; 3) 'geomorphic systems' as aggregations of geomorphic objects. In contrast, our approach aims at identifying landsystems from 2D maps. Our notion of landsystem is different from 'geomorphic systems' and more in line with the way geomorphologists identify geomorphologic elements in the landscape, both on maps and on the ground. Furthermore, the hierarchical nature of landsystems was not captured in Löwner's model.

4. OUR KNOWLEDGE-BASED FRAMEWORK

To support the representation and operational phases of our method, we developed a knowledge-based framework. As a software foundation for our framework, we chose the Neo4j graph database system (https://neo4j.com/docs/) to store and manipulate knowledge and data. Graph structures are composed of nodes, edges and properties (Medhi and Baruah, 2017). Unlike relational databases, graph databases offer a flexible data model on which reasoning can be carried out (Robinson et al., 2015). Moreover, the *node-edge* structure of a graph database allows for representing the ontology of landsystems and structural elements taking into account the different relations elicited in the conceptual model. Thanks to its extension *Neo4j Spatial,* this platform also allows for the integration of spatial data in the graph structure and the manipulation of data obtained from a GIS. The architecture of our software framework is presented in Figure 2. On the top of the figure, the rectangle named *Neo4J engine* represents the graph database system including its plug-in *Neo4J Spatial*. It is linked by a double arrow to a store box named *Spatial Data Store*, representing files containing the spatial data needed by *Neo4J Spatial*.

The big grey round-cornered rectangle below the *Neo4J engine* rectangle includes the main components of our framework that we developed using the *Neo4j engine* and its programming language *Cypher*, enhanced with the *Neo4J Spatial* plug-in. In this grey rectangle are 3 big dashed rectangles that represent the main components of our three-tier architecture. The top left dashed rectangle represents the *Knowledge/Data Package* which contains the three main parts of the graph database that we developed. The *GraphDB (ontology)* contains the generic graph structure for landsystem identification, including nodes and relations that implement the domain ontology (Section 5). The *GraphDB (user space)* contains the graph structures corresponding to the user's data. The *Knowledge/Data Package* contains the *Knowledge Acquisition Module* used by the knowledge engineer to specify the different elements of the *GraphDB (ontology)*. This module is written in Java with embedded Cypher commands for the creation of nodes, edges and properties of the graph DB. *Neo4j Spatial* permits the creation of algorithms to perform spatial operations such as spatial indexing and spatial querying on the spatial data, in parallel with the reasoning and data manipulation on the graph structures. These capabilities enabled us to develop algorithms that can exploit the hierarchical structure of our conceptual model to identify landsystems in relation to observed structural elements.

Our knowledge-based system enables users (who may not be experts in geomorphology) to identify landsystems in a study area of their choice, using available data sets (Section 6). The *Presentation Package* (bottom dashed rectangle) contains two modules. The *User Input Module* allows the user to choose: 1) the categories of landsystems that she wants to identify; 2) the study area as a geographical region on a map that is displayed using background maps. These maps are contained in the *Spatial Data Store* and obtained through the *Control Module*. The *Result Visualization Module* displays to the user the results in a map format after all the processing are carried out by the framework.

Using *GIS tools* (at the bottom of Figure 2) the GIS specialist prepares the *Spatial Data Store* (at the top of Figure 2) which contains the geographical data characterizing the areas associated with the structural elements of the domain ontology (*GraphDB (ontology)*). The GIS specialist uses the *Spatial Data Acquisition Module* to access, through the *Control Module*, the required information contained in the *GraphDB (ontology)*). The *Control Module* (within the *Control Package* in the top-left dashed rectangle) receives the user's request and retrieves relevant knowledge in the Neo4j graph database. This module contains all the programs written in Java with Cypher commands (including Neo4J Spatial functions) and with GIS commands (using the *Geotools* and *JTS* libraries for Java). It coordinates all the operations to manipulate graph data bases and spatial data, the creation of new knowledge and data in graph layers. Neo4J also allows for modularizing a knowledge base by creating so-called *graph layers* that contain different portions of a graph base. In our system, layers are used to store spatial data and to relate them to nodes in the *GraphDB (ontology)*. The *Spatial Data Acquisition* module is used by the GIS specialist to monitor, via the *Control Module*, the transfer of the data contained in the *Spatial Data Sets* within the appropriate graph layers.

The *Control Module* performs several operations (coded in Java with Cypher commands), including searches for the user's requested landsystem, paths (Section 6.1) in graphs of the *GraphDB (ontology)* to retrieve all the landsystems related to the user's requested landsystem, as well as the relevant structural elements and associated spatial data. When the user has chosen the study area and the landsystem category she is interested in, the *Control Module* uses the *Graph DB(ontology)* to identify the structural elements and landsystems of interest to the user. It then creates a graph database for the user, *Graph DB (user space)*, by cloning the appropriate portion of the *DB(ontology)*, including the layers associated with the cloned nodes. More about this in Section 5.

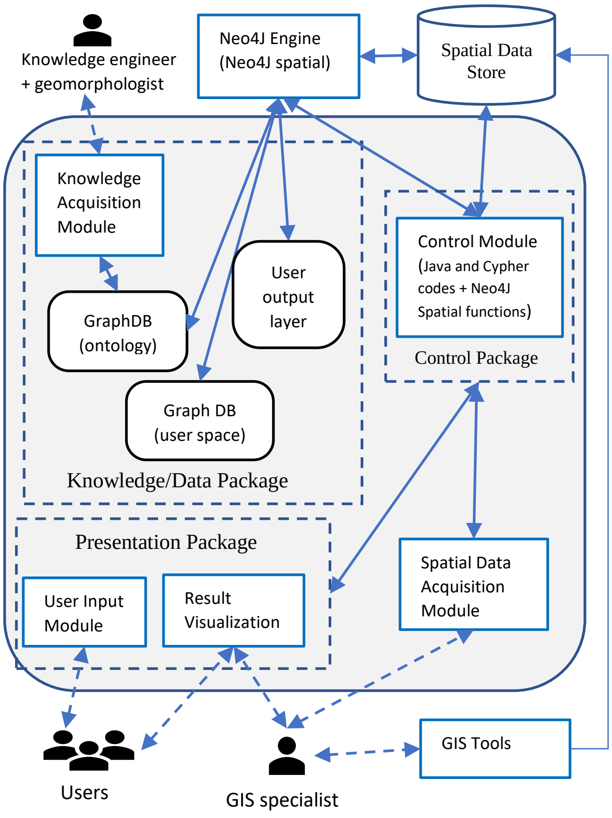


Figure 2: System Architecture

The *Control Module* performs several operations (coded in Java with Cypher commands), including searches for the user's requested landsystem, paths (Section 6.1) in graphs of the *GraphDB (ontology)* to retrieve all the landsystems related to the user's requested landsystem, as well as the relevant structural elements and associated spatial data. When the user has chosen the study area and the landsystem category she is interested in, the *Control Module* uses the *Graph DB(ontology)* to identify the structural elements and landsystems of interest to the user. It then creates a graph database for the user, *Graph DB (user space)*, by cloning the appropriate portion of the *DB(ontology)*, including the layers associated with the cloned nodes. More about this in Section 5.
Then, the *Control Module* triggers all necessary spatial operations that are performed on the areas associated with the chosen study area and the involved landsystems and structural elements, using the *Spatial Data Store* (Section 6). Finally, the system creates a new graph layer called *User Output Layer* in the *Knowledge/data Package* of the framework (Figure 3). This layer contains the resulting data. Its content is presented as a vectorized map to the user.

5. THE REPRESENTATION PHASE

The *Representation Phase* aims to enable a team composed of a knowledge engineer, expert geomorphologists and a GIS specialist to create the knowledge base (a domain ontology) to identify typical landsystems in a chosen domain of geomorphology. The domain ontology extends the core ontology represented by the conceptual model (Figure 1). The case study used in this paper is the *detection of glacial valleys in mountainous areas*. This phase is composed of several steps described in the sub-sections below.

5.1 Knowledge Acquisition and Modelling

During this step, another knowledge acquisition activity takes place to structure the domain knowledge deemed relevant by expert geomorphologists. This step requires several iterations, including the study of a number of practical examples and references found in the scientific literature. Geomorphologists need to identify the different kinds of landsystems and structural elements that are of interest when they look for the targeted landforms in the chosen domain. In our case study we consider the mountainous glacial valleys. The notion of landsystem is easily grasped by expert geomorphologists. The knowledge engineer proposes examples of upper level landsystems (*LS* for short) that may be useful to organize landsystems on the planet surface: *Planetary LS, Glacial LS, non-Glacial LS, Terresterial LS, Marine LS*. These landsystems are discussed with geomorphologists and added to the ontology as specializations of the *Landsystem* class of the conceptual model (Figure 1). In our case study, the focus was then put on *Glacial LS* which has been sub-categorized as *Glacial Marine LS* and *Glacial Terrestrial LS*; the latter contains the *Mountainous Glacial LS*. After discussions with the experts, the landsystem that contains mountainous glacial valleys, the targeted landforms, was called *Mountainous Glacial Valley LS* (MGVLS for short). MGVLS was defined as '*broad valleys in mountainous areas that are/were subject to glaciation processes*'. From this definition it was determined that the immediate parent landsystems of MGVLS are *Mountainous Glacial LS*, and *Valley LS*. *Valley LS* and *Mountainous Glacial LS* have *Terrestrial LS* and *Mountainous LS* as immediate parent respectively. The ontology was revised accordingly.

Each of these landsystems is recognizable by geomorphologists who can determine the corresponding structural elements (SE) and define them. For example, *Terrestrial LS* is characterized by *Terrestrial Topography SE* that defines the topography above sea level; *Mountainous LS* is characterized by *Mountainous Topography SE* corresponding to high altitude rugged topography (Karagulle et al., 2017); *Glacial LS* is characterized by *Glaciation SE* corresponding to areas that were exposed to glaciation; *Valley LS* is characterized by *Valleyness SE* corresponding to a linear depression. Figure 3 describes part of the domain ontology (concepts and relationships) including the MGVLS. The concepts extend the classes *Structural Element* and *Landsystem* of the conceptual model (Figure 1). *Blue concepts* correspond to structural elements, *brown concepts* correspond to landsystems. The brown arrows depict *inheritance* relations. Hence, we see on the left-hand side of the figure the hierarchy of structural elements (in blue) and on the right-hand side the hierarchy of landsystems (in brown). Green arrows depict the relationship *'characterised by'* between structural elements and landsystems. Dashed arrows depict the *containment* relations. Let us emphasize that starting from the *Planetary LS* all child landsystems are related to their parents by a *containment relation*. This shows that the spatial extent (area) of a landsystem is contained in the intersection of the spatial extents of its parent landsystems. Indeed, the *containment relation* allows for the definition of a hierarchy of landsystems that denotes the various levels of context that need to be considered when looking for a type of landform. Moreover, a given landsystem inherits all the structural elements characterizing its parent landsystems.

Structural elements and geomorphologic elements are located in areas (i.e. relation *Located-in* in Figure 1). Geomorphologists use different kinds of data sources and formats to identify these areas. To capture this information, we introduce another ***datatype*** class (not displayed in Figure 3) in the domain ontology in order to record the properties and formats of the data sources that the system will need to load the user data (Section 6.1).

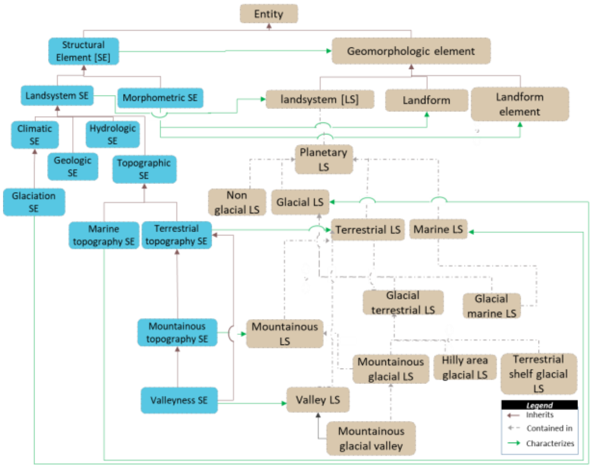


Figure 3: Part of the domain ontology

For example, the *Terrestrial Topography SE* can be obtained by extracting the areas with the attribute "positive elevation" from a digital terrain model (DTM). The *Valleyness SE* is identified by extracting areas based on their *valley bottom flatness* from the DTM (Gallant and Dowling, 2003). The areas associated with the *Mountainous Topography SE* are provided by polygons of rugged mountainous areas that we vectorized from raster maps provided by (Karagulle et al., 2017). Indeed, datatype properties include the characteristics of needed data, the type of file that contains the data as well as how to extract the data. Datatypes are linked to *Structural Elements* by an "*identified by*" relationship.

A given structural element can be identified by different datatypes. For example, the *Terrestrial Topography* SE (blue node in Figure 4) can be identified either by a DTM or directly by a vector representing a Landmass (green nodes in Figure 4).

5.2 Generation of the domain ontology graph database

The concepts and relations of the domain ontology (Figure 3) are stored in the graph database as nodes and edges. The geomorphologist and the knowledge engineer do not need to specify manually the graphs in Neo4J. They are generated automatically by the *Knowledge Acquisition Module* developed by our team (Figure 2). The domain ontology is specified using a number of structured text files (such as CSV) that are easily editable, hence facilitating the maintenance of the knowledge base. These structured files are used by the *Knowledge Acquisition Module* to generate the *Graph DB (ontology)* automatically (Figure 2). Then, the graph database can be inspected using Neo4J's developer interface as shown in Figure 4. This figure displays a small portion of the *Graph DB (ontology)* which shows the three main types of nodes: *Geomorphologic Element* nodes (in brown), *Structural Element* nodes (in blue) and *Datatype* nodes (in green). For example, in Figure 4 *Terrestrial LS* (a brown node, since it is a child of the *Geomorphologic Element*) is *characterized by* a *Terrestrial Topography* (blue *Structural Element* node) which can be *identified by* either *Landmass Shapefile* or a *Terrestrial DEM* (green *Datatype* nodes). Figure 5 displays the list of properties of these 3 types of nodes. We see that there are properties to identify and name nodes (such as *_id* and *_name*). In addition, *GeomorphologicElement* mainly contains information useful to the geomorphologist: *domain* (its application domain), *ge_definition* (definition written in natural language), *ge_properties* (translations of definitions in semantic properties that are transformable into symbolic definitions), *ge_existing_study* (relevant references in scientific literature). For *StructuralElement*, in addition to *names* and *id*, *se_datatype_number* records the number of datatypes that can identify the structural element.

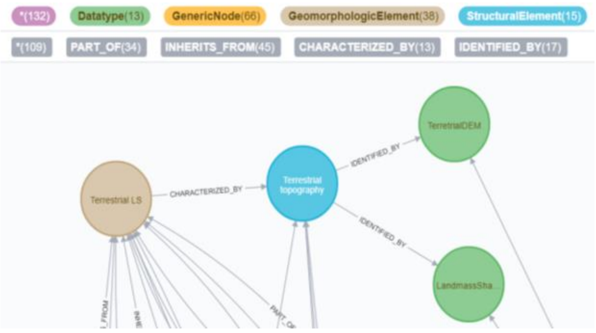


Figure 4: Portion of graph DB ontology

Properties of the *Datatype* node (Figure 5) also describe how to process the spatial data that is imported from GIS files and manipulated in the graph database. In addition to *names* and *ids*, some properties are: data resolution (*dtp_resolution*) and the type of data (*dtp_layer_type*) needed, and properties of the Neo4J layer (*dtp_layer_type*, *dtp_output_layer)* in which data are recorded in the user space (*GraphDB (user space)*). The properties (*dtp_operation*, *dtp_formal-formula*) are used by the *Control Module* to call the algorithms to process the user's data (Section 6), when needed. The *download* property is used if there exists an external data repository for the current datatype.

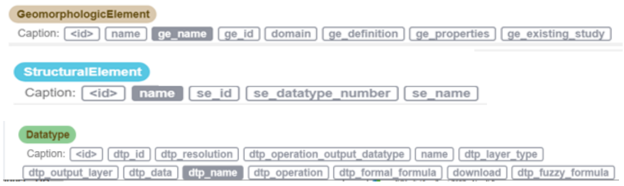


Figure 5: Properties of Geomorphologic Element, Structural Element and Datatype in the graph database

5.3 Spatial Data Acquisition

Since the landsystems of the current domain ontology are generic it is now possible to select geographical data sets that will be available to all users interested in these landsystems. This is a way to 'populate' the knowledge base with geographical data available to all users in certain regions of the earth. This spatial data acquisition task is carried out by the GIS specialist who uses GIS tools to prepare the *Spatial Data Store* which contains the geographical data (vector data and/or images) that correspond to the areas associated with the landsystems and structural elements of the domain ontology *GraphDB (ontology)*.

The properties introduced in datatype nodes (*dtp_resolution*, *dtp_operation_output_data*) are displayed through the *Spatial Data Acquisition Module* (Figure 2) and provide the information needed by the GIS specialist. She can prepare the required data sets and store them in the *Spatial Data Store* (Figure 2). For our case study, we chose to import all spatial data as polygons. Hence, the GIS specialist processed some datasets such as the rugged mountainous areas that were vectorized from the raster maps provided by (Karagulle et al. 2017). When data are ready, the GIS specialist notifies the *Control Module* via the *Spatial Data Acquisition Module*: data are ready to be loaded in the spatial layers in the *GraphDB (instances)* (as described in Section 6.1). Imported GIS data can be either *usable data* or *raw data*. In the first case, data can be directly used. In the second case, the *Control Module* accesses the datatype properties (*dtp_operation*, *dtp_formal-formula*) to retrieve the algorithm needed to acquire the corresponding data. For example, the *Terrestrial Topography SE* (Figure 4) is linked to two datatypes *LandmassShapefile* and *TerrestrialDEM* that offer two possibilities for loading GIS data about terrestrial topography. *LandmassShapefile* is an example of directly usable data: the *Control Module* loads the shapefile polygons data into the graph layer. *TerrestrialDEM* is an example of raw data. In this case the *Control Module* needs to access the appropriate algorithm in the *dtp_operation* property of the datatype: 'select all the regions with an elevation superior to a chosen value x'. Now, the knowledge and databases are ready for use as we show in the next section. The GIS specialist can prepare data sets for any region of the earth. In our case, data sets were prepared for the French Alps.

6. OPERATIONAL PHASE

This phase has three steps and enables a user to exploit the framework to identify landsystems in a chosen geographical area.

6.1 User graphDB instantiation

Using the *User Input Module* (Figure 2) the user specifies her study area. The system displays the list of landsystem categories currently available in the *GraphDB (ontology)* and the user chooses the kinds of landsystems she is looking for. Let us mention that if the user is interested in a kind of landsystems that is not currently available in the system, she needs to present her request to the knowledge engineer and the expert geomorphologist. If feasible, the *Representation Phase* (Section 5) is resumed to integrate in the knowledge base the requested landsystem (and possibly other needed intermediate landsystems) and associated structural elements, as well as the corresponding spatial data. When the study zone and the requested landsystem are chosen by the user, landsystem identification is performed automatically. The *Control Module* first activates the *Neo4J engine* that creates the user's graph database: the *GraphDB (user space)*. Considering the selected landsystem, the *Control Module* instantiates this graph database by cloning the relevant sub-graphs of the *GraphDB (ontology)* between the root node (*Entity* in Figure 3) and the node of the requested landsystem. By performing this cloning operation, the system preserves the integrity of the domain ontology. Then, all graph manipulations are performed in the *GraphDB (user space)*, called the *user's graph DB* in the following paragraphs.

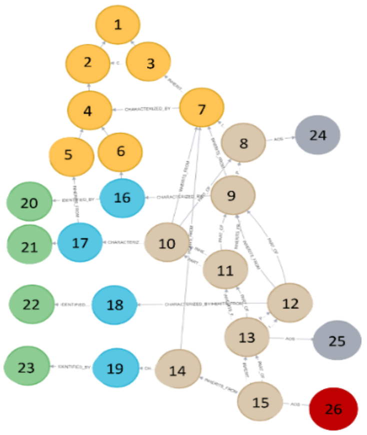
Figure 6 displays an example of the *user's graph DB*. The orange nodes are clones of the upper part of the *GraphDB (ontology)*. The brown nodes starting with the *Planetary LS* (node 8 in Figure 6) correspond to all the landsystems that are parents of the *Mountainous Glacial Valley LS* (node 15) requested by the user. The *Control Module* also creates the node representing the study area linked to *the Planetary LS* (node 24).

The algorithm takes advantage of Cypher's *Path* command to extract sub-graphs. For example, in our program *command C1*:

(C1)
```
Match path1= (n:GeomorphologicElement:GenericNode
    {ge_name:'Mountainous glacial valley'})-[*]-> (o:Geomor
    phologicElement:GenericNode)-[:CHARACTERIZED_BY]-
```

(l:StructuralElement:GenericNode)
gets all the parent nodes of *Mountainous Glacial Valley LS* and their associated structural elements.

In Figure 6, we distinguish two other kinds of cloned nodes: The structural elements (in blue) and the datatypes (in green). For each datatype, the *Control Module* does: 1) create a 'spatial layer' and links it to the corresponding datatype node; 2) use the specifications in the datatype (Figure 5) to import in each spatial layer the sets of polygons retrieved from the Spatial Data *Store*. For simplicity, the layer nodes are not displayed in Figure 6.



ID	Name
1	Entity
2	Structural element (SE)
3	Geomorphologic element
4	Landsystem SE
5	Climatic SE
6	Topographic SE
7	Landsystem (LS)
8	Planetary LS
9	Terrestrial LS
10	Glacial LS
11	Glacial terrestrial LS
12	Mountainous LS
13	Mountainous glacial LS
14	Valley LS
15	Mountainous glacial valley LS
16	Terrestrial topography SE
17	Glaciation SE
18	Moutnainous topography SE
19	Valleyness SE
20	LandmassShapefile
21	GlaciationShapefile DTP
22	MountainousShapefile DTP
23	ValleyShapefile DTP
24	Study area SpEx
25	Mountain glacial SpEx
26	User output SpEx

Figure 6: Portion of the User's Graph DB

6.2 Determination of the requested landsystem

We present here the general idea of the relatively complex Java program (embedding Cypher and JTS commands) that we developed to compute the spatial extent of the requested landsystem (*ReqLS*) over the study area. To simplify the text, we use the term 'PolArea' (i.e. polygonal area) instead of 'spatial extent'. The PolArea of a landsystem is determined by intersecting the PolAreas of all the structural elements (proper and inherited) that characterize this landsystem. To optimize the intersections, the list is ordered starting with the most abstract LS. Using a working area *W-Area* to compute intermediate intersections, the intersection procedure is performed as follows:

1) Create an ordered list *userLSList* of all the parent landsystems of the requested landsystem and the list *userSEList* of associated structural elements (see command *C1* - Section 6.1);

2) Retrieve the PolArea of the study area to initialize *W-Area*

3) For the PolArea of the structural element SE_1 of the first LS_1 in *userLSList* (LS_1 is the most abstract LS of *ReqLS* in *userLSList*),
Perform *W-Area* = *Intersection* (*W-Area*, PolArea(SE_1))

4) Iterate for all SE_i in *userSEList* and all LS_j in *userLSList*
Perform *W-Area* = *Intersection* (*W-Area*, PolArea(SE_i))

5) IF the user requested results for an intermediate LS_w,
Create an output node $OutNode_w$ and attach to it a layer with the current content of *W-Area*; $OutNode_w$ is linked to LS_w node

6) At the end of the iteration, *W-Area* contains the result of all the intersections. The output node $OutNode_{req}$ is created and attached to the node of *ReqLS*. The current content of *W-Area* is saved in a layer attached to $OutNode_{req}$.

All the *OutNodes* and their associated spatial layers are saved in the *User Output Layer* (Figure 2) to be displayed to the user in a map format (Section 6.3). Going back to our example (Figure 6) let us illustrate this procedure step by step.

Step1: The ordered *userLSList* contains nodes (9, 10, 11, 12, 13, 14); the parent LSs of MGVLS (node 15). The *userSEList* contains nodes (16,17,18,19).

Step2: *W-Area* is initialized with polygons associated with the study area (node 24).

Step3: Intersection of *W-Area* and the PolArea of datatype Node 20 linked to SEnode 16 associated with the first LS (LSnode 9) of *userLSList*.

Step4: The iteration goes on by considering the LSs in *userLSList*. Since only nodes 10, 12 and 14 are associated with a SE, the intersections of *W-Area* that effectively take place are those with the PolArea of datatype Nodes 21, 22, and 23.

Step5: The user requested the intermediate result for *Mountainous Glacial LS* (node 13) that is recorded in the spatial layer associated with node 25.

Step6: The final content of *W-Area* is recorded in the spatial layer associated with node 26.

6.3 Displaying results

Figure 7 presents the final map provided by our system. The red polygon delineates the study area associated with node 24 in Figure 6. The green areas represent the extent of the *Mountainous Glacial LS*. It is the spatial layer associated with node 25. The purple areas represent the MGVLS requested by the user. This result is contained in the spatial layer associated with node 26. The background map represents the terrestrial topography in grey shades. The user can inspect this map and the expert can study the different contexts in which they are seeking landforms.

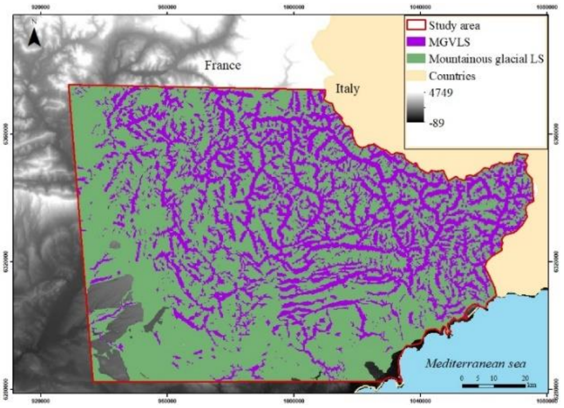


Figure 7: Output Vectorized Map

Using such a vector map, the user can zoom in and out in any area to explore landsystems' geographical characteristics. This illustrates the practical interest of our approach and tool to explore at any scale the geographical contexts deemed important for the user's task. Our system can notably assist the user in identifying any kind of landsystem at a high level of detail over large areas, provided that she (and the GIS specialist) can provide data sets for the structural elements she wants to consider in the study area. Moreover, the exploration of the result map may lead the user to consider new structural elements. In our example, she may consider that the definition of MGVLS is not precise enough and add for example: 1) a new hydrologic SE describing the hydrographic network to eliminate hanging and other tributary valleys; 2) a topographic SE to keep larger valleys only. She may also be interested in introducing a SE to identify glaciers in relation to glacial valleys using appropriate data (Graf, 1970). Hence, our approach and system provide the user with means to expand her analysis and research in geomorphology.

7. DISCUSSION AND PERSPECTIVES

We proposed a new approach which takes advantage of the notion of landsystem to formalize how geomorphologists contextually identify landforms. This paper brought three main contributions: 1) a conceptual model that depicts how landsystems (LS) are organized from structural elements (SE) observed by geomorphologists, and structure the context in which geomorphologic elements (landsystems, landforms, etc.), are sought; 2) a framework for the operationalisation of the model, relying on the Neo4J graph database; 3) a demonstration of the interest of making contextual data available in an exporable map format to support studies and research in geomorphology.

A landsystem may contain other landsystems and/or landforms. The identification of these embedded elements depends on the granularity of the observation and on the geomorphologist's interest. Moreover, our model offers a hierarchical representation of embedded LS. This amounts to considering the hierarchy of contexts in which the sought landform can be found. The results presented in Section 6 are LS that combine a number of structural elements delineating favorable conditions for the identification of a glacial valley. While the system does not detect landforms per se, it identifies landsystems where the landforms of interest can be found.

Our approach using a graph database is particularly appropriate for such knowledge processing since both concepts and data can be added easily. New LS and SE can be added to the knowledge base and, once the data defining the spatial extent of the SE are added to the graph DB, landsystem identification and the vectorized map display are performed automatically. This can save significant time to the user. Indeed, the results obtained using our method and system could be used as input to OBIA approaches. When dealing with very large datasets, our approach and system to extract landsystems provides the context (Deng 2007) that is missing in OBIA techniques and tools.

Arvor et al. (2019) observed that expert knowledge has been discarded from recent approaches of landform detection. Our approach can thus contribute to bring back such knowledge in a well structured and operational way. Our conceptual analysis can be used to better identify features that are relevant to OBIA approaches and to domains where OBIA can apply.

On the short term, we aim to process more complex datasets and to increase their reusability. Broadening the approach means that the user may look for landsystems in different areas; that spatial data may come from different data stores; and that they have to be merged over the area of interest.

Our longer-term objective is to introduce landform detection techniques in our framework. Since our conceptual model includes the definition of landforms, they can be included in the graph database. The *Control Module* can already perform spatial analysis to extract structural elements. More spatial operations can be added in the module to perform landform detection.

ACKNOWLEDGMENTS

Many thanks to Dr Patrick Lajeunesse for his contribution to the project as a geomorphologist expert, and to the Canadian research council, NSERC, for its support to this study (grant 2016-05129).

REFERENCES

Arundel S. and G. Sinha, 2019. 'Validating the use of object-based image analysis to map commonly recognized landform features in the United States'. *Cartography and Geographic Information Science* 46(5): 441-455.

Arundel S, Li W, Wang S, and W. Sizhe, 2020. 'GeoNat v1.0: A dataset for natural feature mapping with artificial intelligence and supervised learning'. *Transactions in GIS* 24:556–572.

Arvor, D., Belgiu, M., Falomir, Z., Mougenot, I., and L. Durieux, 2019. 'Ontologies to interpret remote sensing images: Why do we need them?'. *GIScience & Remote Sensing*, *56*(6), 911–939.

Blaschke, T. 2010. 'Object based image analysis for remote sensing'. *ISPRS J. Photogramm. Remote Sens.*, *65*(1), 2–16.

Cooke, R.U. and J.C. Doornkamp, 1990. '*Geomorphology in Environmental Management: an introduction*'. 2nd edition Clarendon Press, Oxford.

Davis, W. M. 1909. 'The Systematic Description of Land Forms'. *The Geographical Journal*, *34*(3), 300.

Dehn, M., Gärtner, H., and Dikau, R. 2001. 'Principles of semantic modeling of landform structures'. *Computers & Geosciences*, *27*(8), 1005–1010.

Deng, Y. 2007. 'New trends in digital terrain analysis: Landform definition, representation and classification'. *Progress in Physical Geography*, 31(4), 405-419.

Dikau, R. 1999. 'The need for field evidence in modelling landform evolution'. In: Hergarten, S. & H. Neugebauer (1999): Process modelling and landform evolution, Lecture Notes in Earth Sciences 78: 3-12.

Drăguț, L. and T. Blaschke, 2006. Automated classification of landform elements using object-based image analysis'. *Geomorphology*, *81*(3–4), 330–344.

Eisank, C., Dragut, L. and Blaschke T., 2011. 'A generic procedure for semantics-oriented landform classification using object-based image analysis', In T. Hengl, et al., eds, *Geomorphometry*, 125-128.

Evans, I. S. 2012. 'Geomorphometry and landform mapping: What is a landform?'. *Geomorphology*, *137*(1), 94–106.

Gallant, J.C. and Dowling, T.I., 2003. 'A multiresolution index of valley bottom flatness for mapping depositional areas', Water Resources Research, 39/12:1347-1359.

Graf, William L. 1970. 'The Geomorphology of the Glacial Valley Cross Section'. *Arctic and Alpine Research* 2 (4): 303.

Guilbert, E. and B. Moulin, 2017. 'Towards a Common Framework for the Identification of Landforms on Terrain Models'. *ISPRS Int. Journ. of Geo-Information* 6 (1): 12.

International Association of Geomorphologists, 2014 (July). *IAG2014GlossaryOfGeomorphology.pdf*.

Karagulle, D., C. Frye, R. Sayre, S. Breyer, P. Aniello, R. Vaughan, and D. Wright, 2017. 'Modeling global Hammond landform regions from 250-m elevation data'. *Transactions in GIS*, 21 (5): 1040–60.

Löwner, M.O. 2013. 'On problems and benefits of 3D topology on under-specified geometries in Geomorphology'. In *Progress and New Trends in 3D Geoinformation Sciences*, edited by J. Pouliot et al., Springer Verlag.

Ma, L., Yu L., Xueliang Z., Yuanxin Y., Gaofei Y., and B. A. Johnson, 2019. 'Deep Learning in Remote Sensing Applications: A Meta-Analysis and Review'. *ISPRS J. Photogramm. Remote Sens.* 152 (June): 166–77.

MacMillan, R. A. and P. A. Shary, 2009. Chapter 9. 'Landforms and Landform Elements in Geomorphometry'. In *Developments in Soil Science* vol. 33, pp. 227–254. Elsevier.

Medhi, S., and H. Baruah., 2017. 'Relational Database and Graph Database: A Comparative Analysis'. *Journal of Process Management. New Technologies* 5 (2): 1–9.

Milton, N. R. 2007. '*Knowledge Acquisition in Practice*', Springer Verlag.

Rishe, N. 1992. '*Database Design: the Semantic Modeling Approach*', McGraw-Hill.

Robinson, I., Webber, J. and E. Eifrem, 2015. *Graph Databases: New Opportunities for Connected Data*. 2nd edition. CA: O'Reilly.

Scherp, A., Saathoff, A., Saathoff, C. and S. Staab, 2011. 'Designing Core Ontologies', *Applied Ontology* 6(3) 177–221.

Schoeneberger, P. J., and D. A. Wysocki, 2017. '*Geomorphic Description System, Version 5.0*', Natural Res. Conservation Service, National Soil Survey Center, Lincoln, NE.