International Journal of
Geo-Information

MDPI

Article

Coupling Knowledge with GIS Operations: The Benefits of Extended Operation Descriptions

Barbara Hofer [1,*], Emmanuel Papadakis [1] and Stephan Mäs [2]

[1] Interfaculty Department of Geoinformatics—Z_GIS, University of Salzburg, 5020 Salzburg, Austria; emmanouil.papadakis@sbg.ac.at

[2] Professorship Geoinformatics, Technische Universität Dresden, 01069 Dresden, Germany; stephan.maes@tu-dresden.de

* Correspondence: barbara.hofer@sbg.ac.at; Tel.: +43-662-8044-7507

Academic Editors: Ozgun Akcay and Wolfgang Kainz
Received: 13 October 2016; Accepted: 29 January 2017; Published: 8 February 2017

Abstract: The automated development of spatial analysis workflows is one of the envisioned benefits of Web services that provide geoprocessing functionality. Automated workflow development requires the means to translate a user objective into a series of geographic information system (GIS) operations and to evaluate the match between data and operations. Even though full automation is yet out of reach, users benefit from formalized knowledge about operations that is available during workflow development. This article presents user support during workflow development based on a recent approach to extended operation descriptions. User support thereby focuses on the discovery of operations across GIS tools and the validation of chains of spatial analysis operations. The required knowledge about operations is stored in a knowledge base, which builds on an approach called geooperators and extends the geooperator approach with a data-type ontology for describing the interfaces of geooperators and for expressing constraints of geooperator inputs. The advantages of the knowledge base are demonstrated for the construction of a multi-criteria decision making workflow. This workflow contains a set of pre-processing tasks for the input datasets and eventually the calculation of a cost distance raster. A critical discussion of the complexity of the knowledge base and a comparison with existing approaches complement this contribution.

Keywords: geoprocessing; geooperators; spatial analysis workflows; service metadata

1. Introduction

Volume, velocity, variety, and value of data [1] are challenges for today's spatial data and service infrastructures and for the generation of knowledge across themes and domains [2,3]. Data and geoprocessing web services are promoted as approaches to automating the development of spatial analysis workflows in order to derive information from data [4–6]. The vision of full automation is yet out of reach because the translation of user objectives into a series of spatial analysis operations requires knowledge about spatial data and geoprocessing operations that is not accessible to machines to date [7,8].

Recently, there has been progress on the provision of user support during workflow development. User support in this context can address the translation of spatial questions into workflows, the discovery of suitable data and operations, and the validation of chains of operations. The proposed approaches put different components of the workflow development process at their centre; core concepts of spatial information [9,10], data [7,11], operations [12,13], domain knowledge [6], and workflow verification [14].

Approaches working on the representation of knowledge about data, operations or domain knowledge need to formalize this knowledge. It can be expected that eventually the approaches will have to be combined to provide support for all facets of workflow development. In order to facilitate the integration of approaches later on, we claim that such formalizations should be generic and cover a broad range of data, operations, or domain concepts. In addition, standard tools should be used for representing the knowledge, and the ways to generate the formalization should be documented to support contributions from the community [15,16].

Under consideration of these claims, we analysed the requirements of pieces of information about geoprocessing operations during the workflow development process and conceptualized a knowledge base [17]. The knowledge base contains extended operation descriptions for the tasks of discovery and composition of geoprocessing operations.

In order to be able to test the knowledge included in the knowledge base, we translated the conceptualized knowledge base into an ontology expressed in the web ontology language (OWL). We also implemented a demonstrator tool that serves as the intermediate layer between the user and the knowledge base. The demonstrator tool is implemented in Java and can translate user input into SPARQL queries (SPARQL Protocol And RDF Query Language, [18]), which retrieve information from the ontology. These two developments are presented in this paper; they allow us to test the benefits of the proposed operation descriptions as well as the chosen representation of the knowledge in standard tools using OWL and SPARQL.

The benefits of the extended operation descriptions are demonstrated for the specific use case of the development of a multi-criteria decision making process. We derive concrete discovery queries and the required evaluations of chains of tools that the developer of the workflow faces in the context of the use case. The use case also shows an example of an automated discovery of a geoprocessing operation; automated discovery refers to the construction of a discovery query based on results from the evaluation of matching data types in the provided and required input of operations.

The paper is structured as follows; in the following section, Section 2, we review related work. In Section 3, we introduce the knowledge base and its elements for discovery and composition. Section 4 then presents the demonstrator tool, and Section 5 presents the use case and the application of the demonstrator tool. In Section 6 we critically discuss the presented approach before we conclude the article.

2. Approaches to Automated Workflow Development

This section reviews related work in the field of approaches for improved workflow development. After a general overview on the different starting points of various approaches, we go into depth in the area of extended operation descriptions for support during workflow development.

2.1. Improved Workflow Development

ArcGIS ModelBuilder and GRASS GIS Graphical Modeler are tools that support the development of workflows in established GIS. They check, for example, whether the data formats and geometrical properties of the provided input are appropriate and provide error messages in case the execution fails. The information about the operations required for this feedback is available within the workflow tools; users can consult an extensive textual documentation for details about the interface and constraints of operations. The discovery of tools is mainly keyword-based or follows the organization of operations of the tool at hand; e.g., in the case of ArcGIS, tools are organized in toolboxes and toolsets.

The framework processing for QGIS [19] addresses the seamless integration of tools from GRASS GIS, R, and other libraries in workflows, which is a distinctive feature in the above mentioned workflow tools. None of the mentioned tools provides an interface for the discovery of operations with tool-independent annotations of the geoprocessing operations or a machine-accessible formalization of constraints of operations. This restricts the development of recommendation functionality for geoprocessing operations and the ability to search operations across established GIS tools.

A transfer of the kind of support provided by workflow tools from established GIS into service-oriented tools has not been achieved yet. There are tools like GeoBrain [20,21], GeoPW [22,23], the CyberGIS tool [24], or the RichWPS environment [25] that provide service-oriented environments for developing workflows. They generally integrate Web Processing Services (WPS), which follow a standardized interface specification [26], or more general REST services. Service interfaces per se provide syntactic information about the input and output of operations, which is used in the various tools for the chaining of services. However, syntactic information alone leads to limitations concerning discovery, feedback during composition, and the validation of service chains [12,17,27].

The improvement of the workflow development process has been researched from various perspectives; including the translation of spatial questions into operations [28], semantic descriptions of spatial data in order to define which operations can be sensibly applied to the data [7,11], ontologies of domain knowledge that support the automated translation of a user task into operations [6], extended descriptions of geoprocessing operations [12,13,29], and verification of workflows before execution [14,30]. These approaches are discussed in the following section; extended operation descriptions are discussed in more detail in Section 2.2.

KuhnandBallatore [9] developed a language for spatial computing that uses the core concepts of spatial information [10] for translating a spatial problem into an analysis workflow. The motivation of their work is an abstraction from GIS terminology and concepts that do not relate to the spatial problem that users actually try to solve. The language for spatial computing shall thereby be placed as an intermediate layer between the users and existing GIS tools. A description of GIS operations from different legacy GIS tools, as presented in our work, could be beneficial in establishing the link between the language components and the GIS operations for execution.

The work by Stasch et al. [7] addresses meaningful spatial analysis for the cases of interpolation and aggregation. Based on a sophisticated ontological framework, the objective of their work is to infer whether the given input data allow the application of an interpolation operation, for example. The focus on the suitability of data sets for given tasks has been continued in the work of Scheider et al. [11] and ScheiderandTomko [31]. In both articles the objective relates to the generation history of data sets and the usability of data sets for analysis tasks.

Al-Areqi et al. [6] developed an approach to automatic geospatial service composition based on domain modelling. The domain model includes domain specific services, their descriptions, and a taxonomy of services. This taxonomy contains knowledge about the relations between services and allows the provision of different solutions for combinations of services given a task at hand. The approach is demonstrated for a use case of a rise in sea-levels. The final choice of operations is influenced by constraints that are set by the user. The approach is considerably different from previous approaches. Open questions concern how the domain models can be generated given the variety of domains and tasks and whether the constraints for the workflows depend on specific workflows or are similar in nature across workflows.

The composition of workflows and their verification has been worked on using various techniques from computer science, like artificial intelligence planning [32] or semantic service descriptions based on OWL-S [33]. Lutz et al. [30] expressed the functionality of services as rules and used backward-chaining [34] to generate the inputs required for satisfying the rules. These previous works contribute a set of approaches for using extended operation descriptions. This article is based on a requirement analysis of operation descriptions specific for the spatial analysis workflow development process and aims at the development of a generic and extensible formalization of geoprocessing operations.

2.2. Extended Operation Descriptions

OGC WPS provide syntactic service descriptions. A process has a title, description, identifier, and optional metadata, as well as input and output parameters. Inputs and outputs can be complex data or literal data, which again have title, abstract, identifier, and metadata. The data themselves are specified in the wps:Format tag with mimeType (e.g., 'text/xml'), encoding (e.g., 'UTF-8'), and schema (e.g., 'http://schemas.opengis.net/gml/3.2.1/feature.xsd'). It is further possible to specify data types of literal data as well as allowed values. Feedback that can be provided during the execution of WPS includes; too many inputs, too many outputs, size of data exceeded, input data type not conforming to process requirements, and the like. The feedback focuses on the syntactic elements of the interface. A task like service discovery is not strongly supported with syntactic interface descriptions, as different operations can have the same input elements or similar operations can have different inputs and outputs [12]. The generic nature of descriptions of OGC WPS has been criticised, as users and also software clients have difficulties understanding the requirements of input data and interpreting the service output [35,36].

In the current WPS 2.0 interface specification, a further development of WPS profiles has been proposed to improve the documentation of services [26]. These profiles are designed to link specific implementations of functionality to concepts in order to overcome implementation uncertainty [37]. WPS profiles may provide documentation of an implementation, but they are not (yet) directly linked to a repository that would be a central access point for discovering geoprocessing services.

An alternative to WPS profiles are semantic or extended operation descriptions [12,13,29], which are the focus of this work. The work on extended operation descriptions is generally independent of a particular service specification and aims at using ontologies for describing geoprocessing operations and data in general [12,13,29]. Various approaches to extended operation descriptions have been reviewed in [17]. Here, we focus on the work of Fitzner et al. [13] as the latest approach to improved operation discovery based on extended operation descriptions.

Fitzner et al. [13] developed a methodology for evaluating discovery queries in sketched operation ontologies using the logic programming language Datalog. Their approach considers the syntactic elements of operation interfaces together with constraints related to the input and output of operations, which are specified as pre- and postconditions. The operation itself is represented with a keyword; in their example the keyword is 'overlay'. In the example presented by Fitzner et al. [13], a user looks for an overlay operation on polygons that assures that the coordinate reference systems (CRS) of the two input polygons are identical with the CRS of the output polygon of the overlay operation. This user query is evaluated against a service description that specifies the inputs as in the user request but does not assure that the resulting polygon has the same CRS as the inputs. In this example, the query is not meant to be successful as the request and the advertisement do not correspond concerning the CRS of the output.

Fitzner et al. [13] demonstrated that extended operation descriptions support discovery under consideration of the constraints on input and output. However, they focused on the development of a query mechanism of operation and data type ontologies. To the best of the authors' knowledge, they did not expand the operation ontologies to a larger number of operations nor work on an approach to generate user requests in the required formalization. We expand their work in a series of ways; rather than focusing on discovery only, we take the whole workflow development process into consideration, and we provide a knowledge base that already covers a considerable set of geoprocessing operations.

3. A Knowledge Base Providing Extended Operation Descriptions

In this section, we present a knowledge base that contains extended operation descriptions that support the workflow development process. The knowledge base has been conceptualized in previous work [17]. The contributions made in this article are a realization of the proposed knowledge base in Protégé OWL and the implementation of a demonstrator (Section 4), which together allow the testing of the operation descriptions (Section 5). In the following, we briefly introduce the foundation of the knowledge base and the knowledge it holds about geoprocessing operations.

The knowledge base has been designed to support discovery through the user, to evaluate chains of operations, and to anticipate automated discovery of operations. Discovery requires the annotation of geoprocessing operations with various pieces of information of relevance to users. The evaluation of combinations of operations requires knowledge about interfaces and constraints of operations. Automated discovery refers to the possibility of using the results from evaluations of combinations of operations to issue queries for specific operations like data format transformations or coordinate transformations. To realize these capabilities, the knowledge base provides (Figure 1):

- Concepts of geoprocessing operations,
- Categories of geoprocessing operations,

- Relations between operations, and
- A data type ontology for specifying the interfaces of operations including constraints.

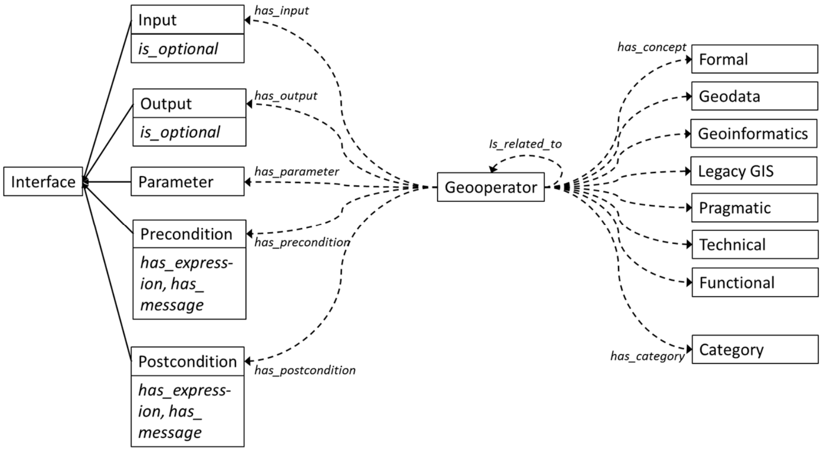


Figure 1. Elements of the knowledge base for annotating geoprocessing operations, i.e., geooperators.

The concepts, categories and relations between geooperators are used for annotating operations for their discovery; this part of the knowledge base is based on work by Brauner [15]. Brauner [15] developed a geooperator thesaurus in the Simple Knowledge Organization System SKOS [38] following an extensive analysis of possible perspectives on geoprocessing operations. The geooperator thesaurus includes about 40 geooperators from ArcGIS and GRASS GIS and is implemented in the geooperator browser (http://purl.org/net/geooperators (last accessed on 03 February 2017)), which supports faceted browsing of operations across GIS tools. Faceted browsing thereby allows the combination of concepts and categories for goal-directed searches. A special feature of the approach is the inclusion of related matches to operations from the same GIS or operations from different legacy GIS tools.

The SKOS thesaurus provided by Brauner [15] has been translated into an OWL ontology using Protégé OWL to increase the expressiveness of the geooperator approach and to facilitate the integration of required extensions. Also, the about 40 geooperators included in the thesaurus were translated into OWL. We included an additional concept in our OWL ontology; the functional concept shown in Table 1. This concept shall support the search for specific GIS functionality and is based on the universal GIS operations identified by Albrecht [39]. To realize evaluations during composition of operations, the initial geooperator approach has been extended with a spatial data type ontology and interface descriptions of geooperators, which are described in the following sections.

Table 1. Functional concept added to the concepts for annotating geooperators.

Geooperator Concept	Sub-Concepts (If Available)	Individuals	Explanation
Functional concept	Data management	CRS-conversion, format conversion, raster statistics, etc.	The functional concepts are based on the groups of GIS operations identified by Albrecht [39]. As data management is not considered in Albrecht's work, this category has been added.
	Distribution/neigh-borhood	Nearest neighbor, proximity, cost-diffusion-spread	
	Location analysis	Buffer, overlay, Voronoi, etc.	
	Search	Thematic search, spatial search, etc.	
	Reclassification	Interpolation, reclassification	
	Spatial analysis	Pattern-dispersion, multivariate analysis, etc.	
	Terrain analysis	Slope, aspect, viewshed analysis	
	Measurements	Adjacency, distance, height, etc.	

3.1. Data Type Ontology

In [17] a spatial data type specification that provides the vocabulary for specifying the interfaces of the geooperators and for expressing their constraints was introduced. It is based on the schemas of the International Standard Organisation (ISO) for vector data [40] and coverages [41]. This spatial data type specification has been transformed into a data type ontology and linked with the geooperator ontology. Figure 2 shows a simplified image of the data type ontology. The main class of the data type ontology is the class 'Spatial Object', which contains the subclasses spatial collection and raster. Each spatial object is linked with object properties to the classes 'Attribute' and 'Geometry', which, in turn, represent the thematic and the spatial information that is attached to the spatial object. Attributes can be complemented with additional properties such as description, scale of measurement, unit of measurement, and data type. The geometry class, which contains different vector geometry classes, a raster geometry class, and an envelope or bounding box class, is linked to the coordinate reference system. A class for non-spatial parameters of operations completes the data type ontology. Non-spatial parameters specify settings of geooperators like the geographic transformation parameter of the project operation; they can be optional and can have a data type assigned. The descriptions of attributes, geometries, and parameters allow the detailed specification of input and output data of geooperators and their preconditions and postconditions (Section 3.2).

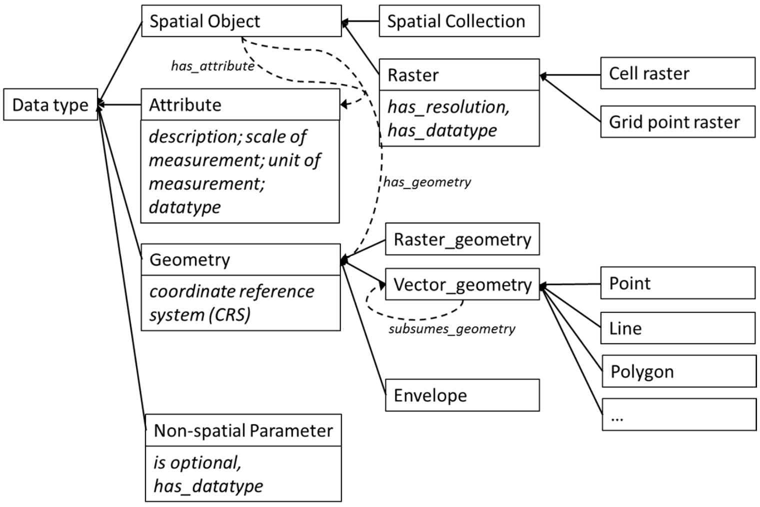


Figure 2. Data type specification built into a data type ontology.

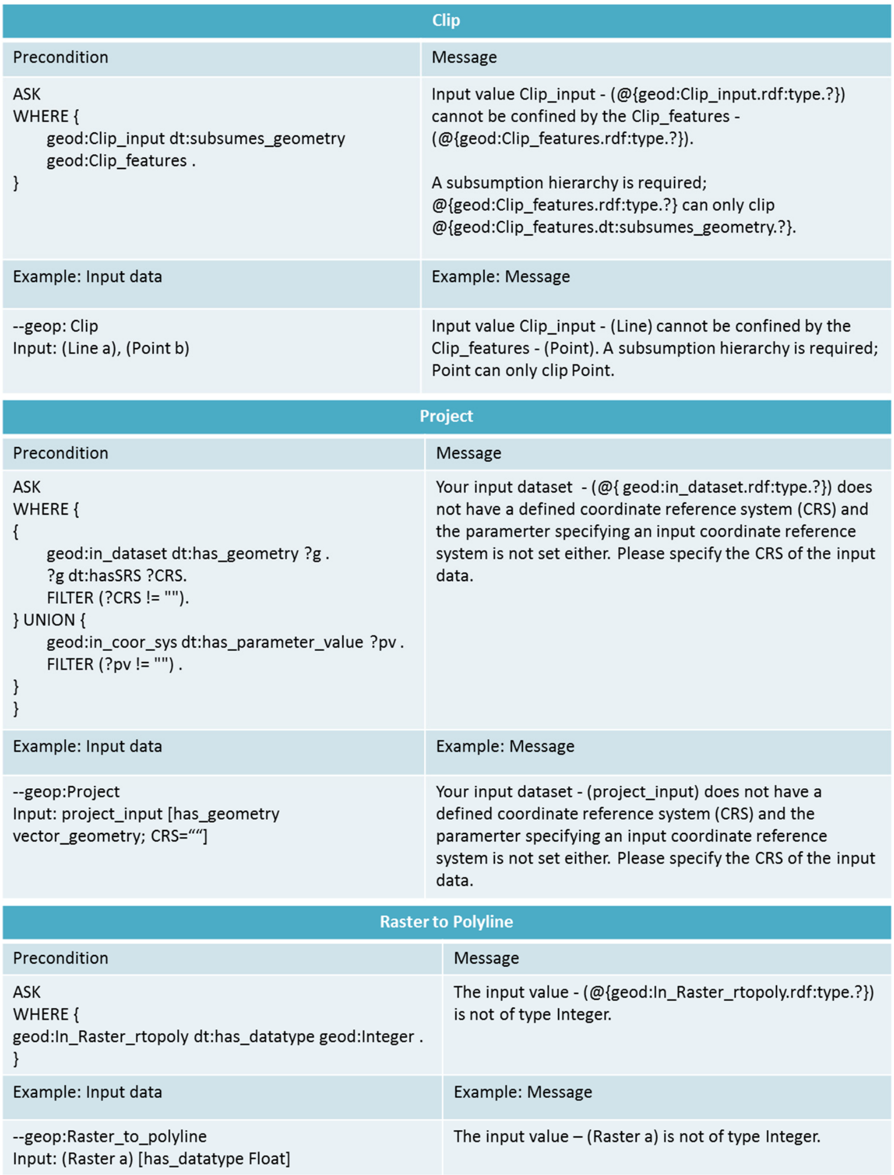
3.2. Interfaces of Geooperators

The data type ontology is used to specify the interfaces of geooperators. The class 'Interface' consists of the subclasses Input, Output, Parameter, Precondition, and Postcondition (Figure 1). Pre- and postconditions refer to the constraints of geooperators that need to be considered in addition to the specification of the inputs. Preconditions assure the correct functioning of the operation during execution. These constraints generally concern the thematic and geometric properties of input and the parameters of operations. Postconditions refer to the expected type of output, which is relevant information when operations are chained.

The expression of constraints is done with SPARQL queries of the ASK form to check whether or not a query pattern has a solution making use of the expressive power of OWL. Consequently, they provide the means to test whether the input provided to a geooperator actually fulfils the requirements of the operations. The testing of the constraints requires that specific inputs are matched with the interface of the operations, which takes place in the demonstrator tool presented in Section 4. The SPARQL statements for the required tests are provided in two datatype properties of preconditions and postconditions, which can be read by the demonstrator tool:

- *Has_expression*: this property contains SPARQL expressions of the constraints;
- *Has_message*: this property contains a message, which is a mix of text and SPARQL expressions that can be evaluated in the demonstrator tool.

In the following, we show three exemplary formulations of preconditions in the ontology together with an example of the evaluation of these constraints and the message provided to the user (Figure 3). In the example of the clip operation, the geometries of the clip_inputs need to subsume the geometries of the clip_features; i.e., points can be clipped with points, lines, and polygons, but polygons can only be clipped with polygons. The project operation can only work in case the CRS of the input dataset is known or a parameter with the input CRS is provided. Again, this constraint is formulated as a SPARQL query, which can be evaluated against the input provided to the operation. In the third example, the constraint assures that the raster dataset provided to the raster to polyline operations has the datatype integer.



Clip

Precondition	Message

ASK
WHERE {
geod:Clip_input dt:subsumes_geometry
geod:Clip_features .
}
	Input value Clip_input - (@{geod:Clip_input.rdf:type.?}) cannot be confined by the Clip_features - (@{geod:Clip_features.rdf:type.?}).
	A subsumption hierarchy is required; @{geod:Clip_features.rdf:type.?} can only clip @{geod:Clip_features.dt:subsumes_geometry.?}.

Example: Input data	Example: Message
--geop: Clip Input: (Line a), (Point b)	Input value Clip_input - (Line) cannot be confined by the Clip_features - (Point). A subsumption hierarchy is required; Point can only clip Point.

Project

Precondition	Message

ASK
WHERE {
{
geod:in_dataset dt:has_geometry ?g .
?g dt:hasSRS ?CRS.
FILTER (?CRS != "").
} UNION {
geod:in_coor_sys dt:has_parameter_value ?pv .
FILTER (?pv != "") .
}
}
	Your input dataset - (@{ geod:in_dataset.rdf:type.?}) does not have a defined coordinate reference system (CRS) and the paramerter specifying an input coordinate reference system is not set either. Please specify the CRS of the input data.

Example: Input data	Example: Message
--geop:Project Input: project_input [has_geometry vector_geometry; CRS=“”]	Your input dataset - (project_input) does not have a defined coordinate reference system (CRS) and the paramerter specifying an input coordinate reference system is not set either. Please specify the CRS of the input data.

Raster to Polyline

Precondition	Message

ASK
WHERE {
geod:In_Raster_rtopoly dt:has_datatype geod:Integer .
}
	The input value - (@{geod:In_Raster_rtopoly.rdf:type.?}) is not of type Integer.

Example: Input data	Example: Message
--geop:Raster_to_polyline Input: (Raster a) [has_datatype Float]	The input value – (Raster a) is not of type Integer.

Figure 3. SPARQL expressions of preconditions of the clip, project, and raster to polyline operations.

The postconditions of the operations are expressed in a corresponding form to the preconditions. For example, the postcondition of the clip operation states that the clip_output has the same geometry type and CRS as the clip_input. The preconditions and the postconditions are an essential part of the knowledge about operations stored in the knowledge base, which allows the evaluation of possible chains of operations. The mechanism of the evaluation is presented in Section 4.

4. A Demonstrator for the Validation of the Knowledge Base

The interaction between the user and the knowledge base for the discovery and composition of operations is managed through a tool, which is currently available in the form of a demonstrator. The user interacts with the user interface of the tool to describe his/her request, which is subsequently translated into SPARQL queries (Figure 4). The tool is implemented in Java 8, using the apache Jena framework for RDF management. The tool comes with a framework that supports the main functions of accessing the knowledge base, issuing queries, and other generic operations such as deciphering the precondition messages. In addition, query pattern mechanisms are provided that allow users with no SPARQL expertise to form new queries using simple Java functions. Finally the tool offers customization using configuration files, which in principle allows the use of other knowledge bases with the condition of following the same schema as the proposed knowledge base.

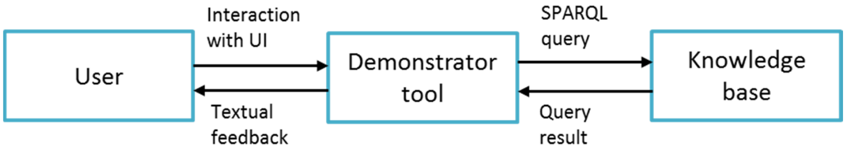


Figure 4. Interaction between user, demonstrator tool, and knowledge base.

The first set of actions supported by the tool is the discovery and selection of geooperators. The interface for this user task is intended to resemble the facet-browsing approach of the original geooperator browser [15]. Figure 5 illustrates the sequence of steps that realize the discovery process using flowchart notations. Particularly, the users are presented with a set of filters, including geooperator concepts, categories, and input/output constraints. Along with the filters, more sophisticated search mechanisms are provided, such as keyword matching or a search for geooperators related to a specific operator. The users can flexibly select the relevant options for their search, and the tool generates the appropriate query to be issued. In addition to the SPARQL query that is generated, a Java function tests string similarity between the provided keywords and the geooperators that have been identified through the SPARQL query (example provided in Figure 10).

According to the results of the query, either a list of geooperators is shown, which satisfies the user-specified filters, or the users are prompt with a warning message that suggests that the search criteria be re-specified.

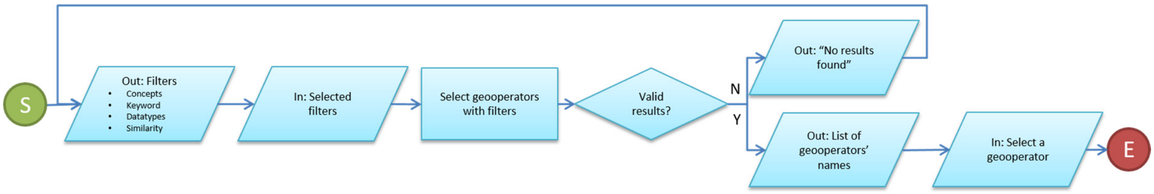


Figure 5. Flowchart for the manual discovery and selection of geooperators.

The second set of actions concerns the combination of geooperators with input data as part of the composition of operations. This functionality includes a step-wise evaluation of whether the given input fulfils the requirements of a specific geooperator interface. As it is depicted in Figure 6, the composition of a workflow comprises three phases; setup, evaluation, and potential connection. First, the user sets up a geooperator; using the discovery process, the user selects a geooperator and provides the required inputs and parameters based on the interface of the selected operation. Afterwards, the given inputs and parameters are evaluated following a three level check, which includes number of inputs, datatypes, and preconditions. As mentioned, a SPARQL expression of the preconditions is contained in the ontology via the data property *has_expression*. In such a way, the knowledge base contains all knowledge required to test whether a geooperator can function correctly given some input data. If a mismatch between the input and the interface occurs or a violation of some precondition is detected, the user receives a message. This message includes details about the failed test and is generated on the fly by the program. Messages related to preconditions may require more detailed formulations than mismatches related to the input data. For that reason, the precondition can be linked to a data property named *has_message*, which provides the blueprint of the message that is adapted to the particular case.

If the evaluation phase fails, the user can either re-specify the given input or start over with the set up phase, or the tool can perform an automated discovery of operations. In the case of a successful evaluation, the output of the current geooperator is generated. It is worth noting that if a geooperator documentation specifies postconditions using the *has_expression* property, the output is formatted accordingly.

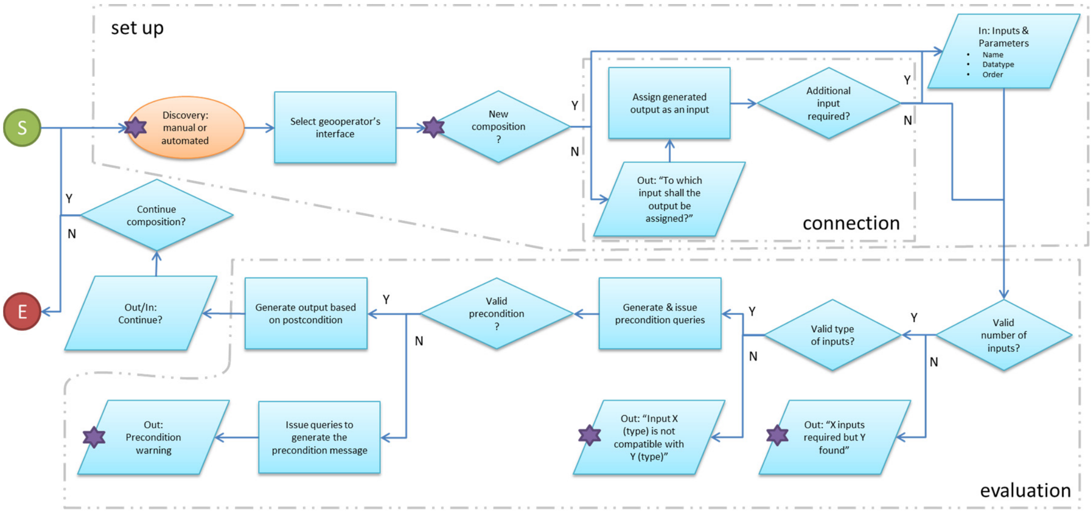

set up
S
Discovery: manual or automated
Select geooperator's interface
New composition ?
Y
N
Assign generated output as an input
Additional input required?
Y
N
In: Inputs & Parameters
Name
Datatype
Order
Out: "To which input shall the output be assigned?"
connection
Y
Continue composition?
N
E
Out/In: Continue?
Generate output based on postcondition
Y
Valid precondition ?
N
Generate & issue precondition queries
Y
Valid type of inputs?
N
Y
Valid number of inputs?
N
Out: Precondition warning
Issue queries to generate the precondition message
Out: "Input X (type) is not compatible with Y (type)"
Out: "X inputs required but Y found"
evaluation

Figure 6. Flowchart for workflow composition.

In case the evaluation phase fails at the point where the types of inputs are evaluated, the tool could issue an automated discovery of operations (Figure 7). Automated discovery is not implemented in the demonstrator tool yet, but the conception of this functionality is available. In the case of automated discovery, the information about input required by an operation and the input provided by the user is already available to the tool and can be translated into a discovery query. This discovery query will search for operations that change the data types as required and satisfy the functional concept format conversion. The resulting operations are provided as recommendations to the user, who can decide to include one of the proposed operations in the workflow. The main difference between automated and manual discovery is that, in case of automated discovery, the system selects the required filters for the functional concept and the input and output of operations for the query. The focus on type conversions is a first step for recommending operations to workflow developers, which can be extended for queries that consider the result of precondition evaluations in the future.

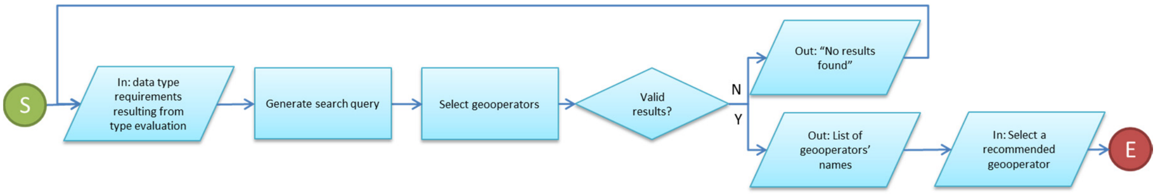

S
In: data type requirements resulting from type evaluation
Generate search query
Select geooperators
Valid results?
N
Y
Out: "No results found"
Out: List of geooperators' names
In: Select a recommended geooperator
E

Figure 7. Flowchart for automated discovery and selection of geooperators.

After the evaluation phase, the user can either finish the composition process or continue by appending the sequence with additional geooperators. Having selected the next operation, the tool stores internally the generated output in order to use it as an input to the subsequent operation. At this point, the user has to select to which of the available inputs of the new geooperator the generated output shall be assigned. Note that the user will be asked to provide any additional inputs or parameters if there is an arity mismatch between the input of the new operation and the output of the preceding one. As a result, the tool passes on the knowledge it gained about one operation to the following in the workflow context.

5. Use Case and Application

Section 5 introduces a use case of a workflow development process and demonstrates the benefits of extended operation descriptions for tasks during workflow development. The use case we look at is a multi-criteria decision making process in the context of transport route planning for a highway in Poland, which was developed by Keshkamat et al. [42]. An implementation of this workflow in ArcGIS has been provided by Brauner [15], which is used as reference for the workflow here.

5.1. Use Case: Multi-Criteria Decision Making Process

The objective of the transport route planning workflow under investigation is to determine a possible route for a highway through North-East Poland. The selection of the proposed route should be optimized based on criteria considering transport efficiency, ecology, social impact, safety, and economic costs [42]. In the decision making process, experts weight the information layers with costs. The weighted layers are then overlaid in order to determine the route with the least cost between the town of Budzisk and Warsaw [15].

The development of a spatial analysis workflow through a GIS expert consists of discovery, composition, and execution of spatial analysis operations [4]. We assume that the GIS expert initially has a workflow concept in mind, which defines the objectives and conditions of the analysis and directs the selection of operations [17]. Looking at the use case of the multi-criteria decision making process for transport route planning, the workflow concept may resemble the representation in Figure 8; the required data need to be pre-processed, reclassified, and combined in a weighted raster before the least cost path can be calculated, which results in a proposal for a route of a highway through Poland.

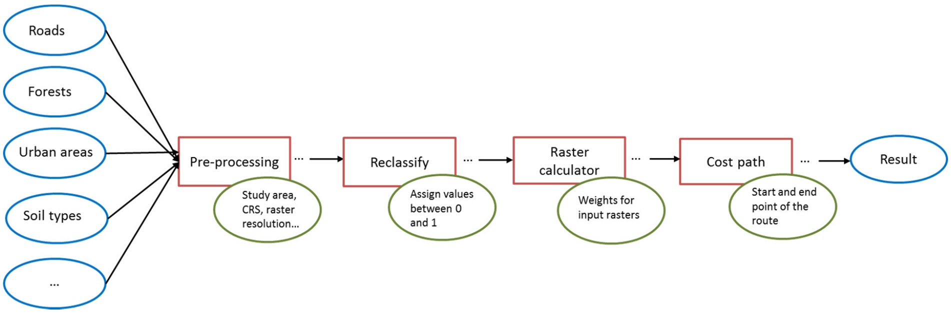

Roads
Forests
Urban areas
Soil types
...
Pre-processing
...
Study area, CRS, raster resolution...
Reclassify
...
Assign values between 0 and 1
Raster calculator
...
Weights for input rasters
Cost path
...
Start and end point of the route
Result

Figure 8. Workflow concept of the multi-criteria decision making process; data are represented in blue, operations in red, and requirements and conditions in green.

The pre-processing of input data involves a series of operations, which are chosen by the workflow developer based on the properties of the input data. The workflow developer also sets the sequence of operations based on his/her expertise. The pre-processing steps of the vector data set roads and the raster data set forests are shown as examples in Figure 9. Overall, the following pre-processing operations are applied in the workflow:

- Project: project data into the coordinate reference system used in the workflow,
- Clip: clip data to the study area at hand,
- Resample: resample raster data to the required resolution,
- Feature to raster: transform vector data input to raster.

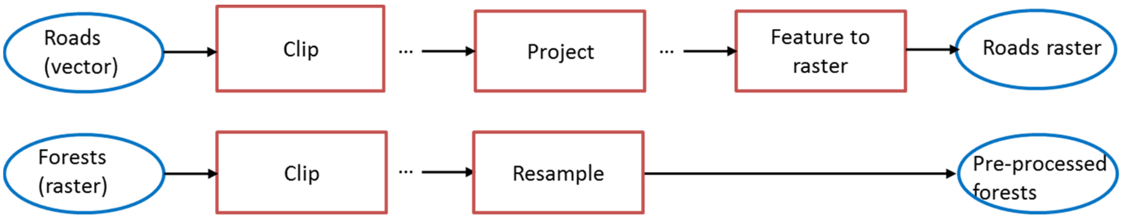


Figure 9. Pre-processing of the vector dataset roads and raster dataset forests.

During the workflow development process, the expert discovers operations for the tasks at hand in an iterative manner and chains the operations according to his/her experience and the context of the analysis. Thereby, the workflow concept is concretized and supplemented with additional operations. Important for our work are the questions the developer of the workflow has as he/she moves towards the concrete workflow and the feedback the system can provide to support his/her tasks. Specific discovery queries of the workflow developer can be:

- Discover ArcGIS operations like project, resample, reclassify etc.;
- Discover clipping operations for raster and vector data provided by ArcGIS;
- Discover the operation required to generate the input for the cost path operation, i.e., the cost distance operation.

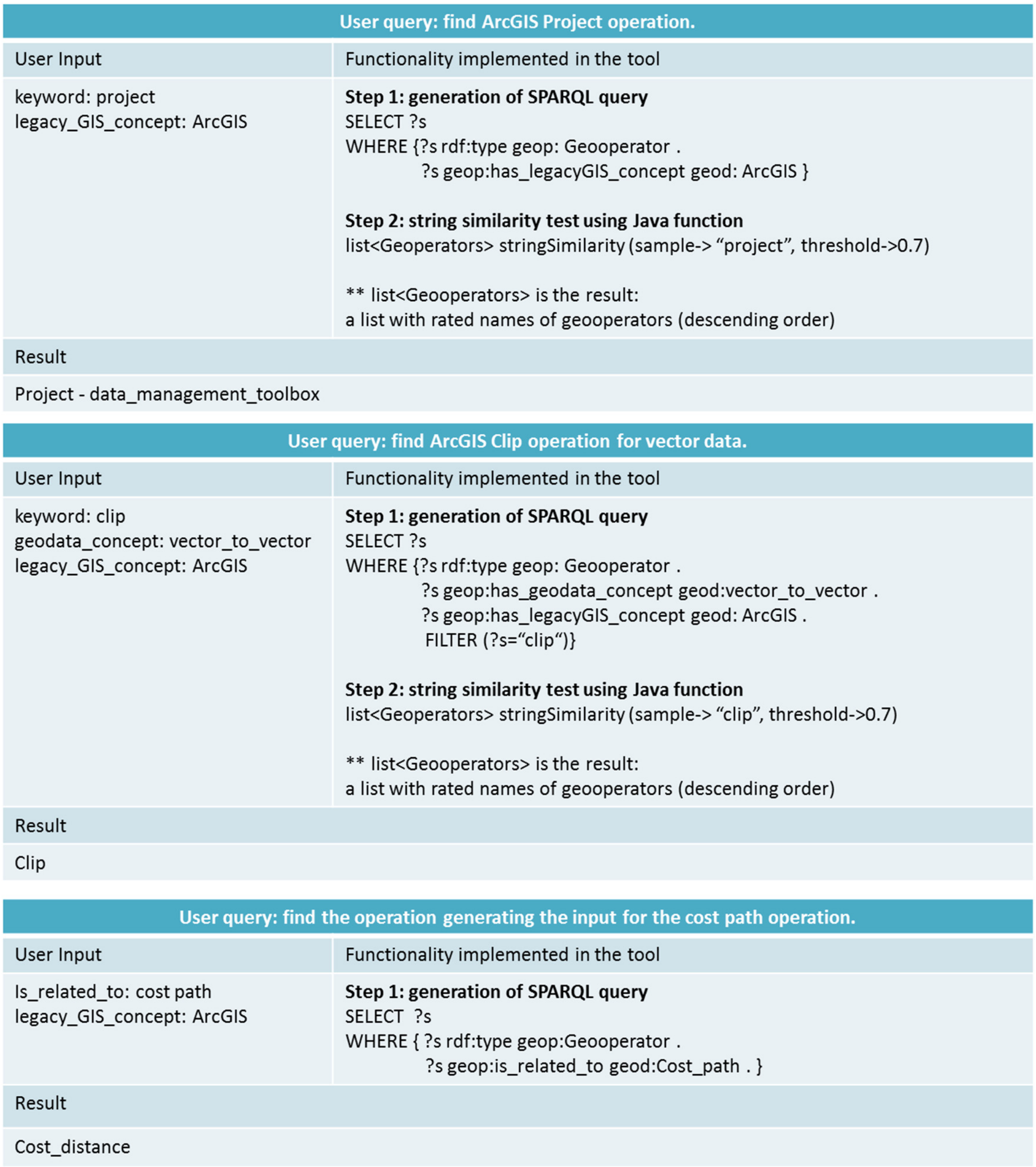
In addition to discovery, the chaining of operations needs to be accomplished. A system having knowledge about operations can evaluate whether the number and type of inputs provided to an operation is appropriate and whether the preconditions of operations hold. In addition, operations can be recommended given a mismatch of types between provided input and required input of operations. Examples derived from the use case are:

- Feedback as to whether the inputs for the clip operation satisfy the precondition concerning geometries of inputs (e.g., polygons can only be clipped with polygons);
- Feedback as to whether the output of the clipped roads can directly be used as input for the project operation;
- Feedback stating that the output of the project operation cannot be directly linked to the reclassify operation;
- Recommendation of the feature_to_raster tool in between the steps project and the reclassification of roads.

5.2. Benefits of Knowledge about Operations during Workflow Development

Taking the specific examples from the use case in Section 5.1, we here demonstrate the benefits of the knowledge about operations stored in the knowledge base. We describe the input that the workflow developer provides to the demonstrator tool presented in Section 4, which translates the queries to SPARQL and evaluates them in the knowledge base.

Discovery queries aim at finding operations that provide some defined functionality, are implemented in the chosen GIS, work on specified data formats, and are related to specific tools. All these pieces of information about GIS operations can be set as filters in our demonstrator tool. The filters provided are concepts, keywords, datatypes, and similarity between operations. The specific examples related to the use case refer to the discovery of the project, clip, and related operations of the cost path operation. Figure 10 shows the input provided by the user in the user interface of the demonstrator tool and the processing of this input through the tool. In a two-step procedure, first the SPARQL query is generated based on the selected filters, and then a string similarity function compares the keyword provided to the user with the list of geooperators resulting from the query.



User query: find ArcGIS Project operation.	
User Input	Functionality implemented in the tool
keyword: project legacy_GIS_concept: ArcGIS	**Step 1: generation of SPARQL query** SELECT ?s WHERE {?s rdf:type geop: Geooperator . ?s geop:has_legacyGIS_concept geod: ArcGIS } **Step 2: string similarity test using Java function** list<Geoperators> stringSimilarity (sample-> "project", threshold->0.7) ** list<Geooperators> is the result: a list with rated names of geooperators (descending order)
Result	
Project - data_management_toolbox	

User query: find ArcGIS Clip operation for vector data.	
User Input	Functionality implemented in the tool
keyword: clip geodata_concept: vector_to_vector legacy_GIS_concept: ArcGIS	**Step 1: generation of SPARQL query** SELECT ?s WHERE {?s rdf:type geop: Geooperator . ?s geop:has_geodata_concept geod:vector_to_vector . ?s geop:has_legacyGIS_concept geod: ArcGIS . FILTER (?s="clip")} **Step 2: string similarity test using Java function** list<Geoperators> stringSimilarity (sample-> "clip", threshold->0.7) ** list<Geooperators> is the result: a list with rated names of geooperators (descending order)
Result	
Clip	

User query: find the operation generating the input for the cost path operation.	
User Input	Functionality implemented in the tool

Is_related_to: cost path
legacy_GIS_concept: ArcGIS
	Step 1: generation of SPARQL query
	SELECT ?s
	WHERE { ?s rdf:type geop:Geooperator .
	?s geop:is_related_to geod:Cost_path . }

Result	
Cost_distance	

Figure 10. Discovery queries formulated by the user and their SPARQL equivalents.

After discovery, operations can be chained together, whereby the output of one operation will become the input of the next operation. As described in Section 4, a series of evaluation steps are preformed upon the chaining of operations. The following examples show the feedback that can be provided to the user in the context of the use case. The example in Figure 11 focuses on the pre-processing of the roads dataset. The workflow developer starts with clipping the dataset and projecting it to the required projection. As a subsequent step, he/she chooses the reclassify operation, which leads to a type mismatch between the provided vector inputs and the required raster inputs. Given the result from the evaluation, the tool triggers an automated discovery and recommends the user to add the feature_to_raster operation to the workflow (Figure 12). The feature_to_raster operation transforms a vector input into a raster output, which is required as intermediate operation in the workflow. This operation completes the pre-processing part of the roads dataset.

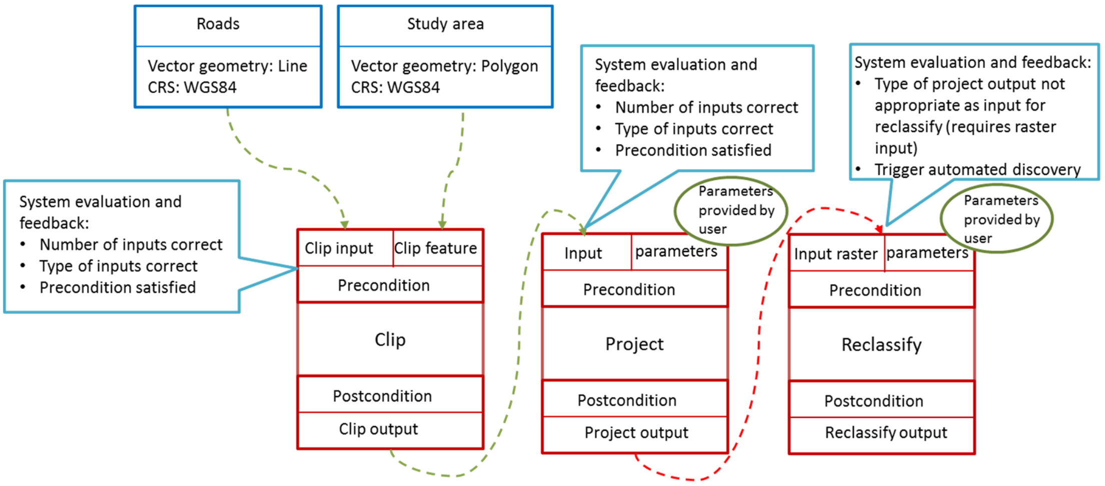


Figure 11. Feedback provided to the workflow developer working on the chain of operations for pre-processing of the roads dataset.

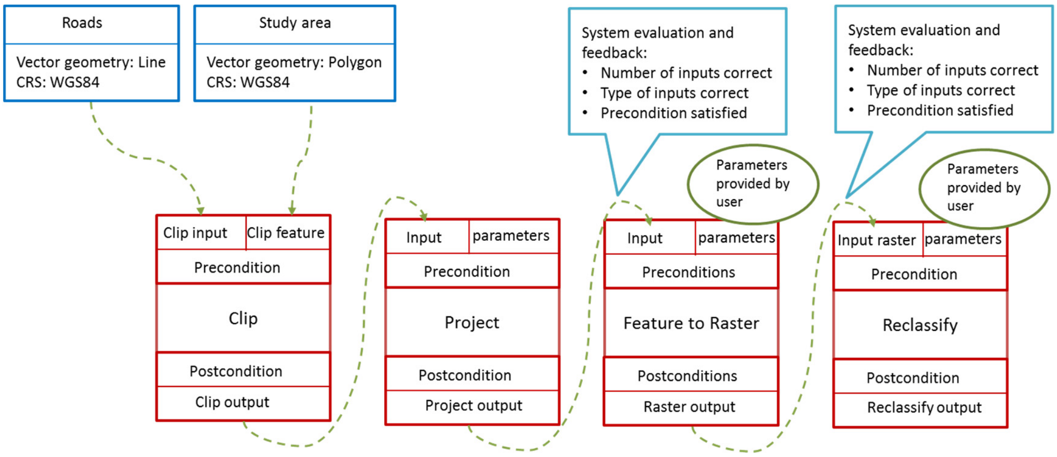


Figure 12. Finalized pre-processing of the roads dataset including the recommended feature_to_raster operation.

The examples show that the demonstrator tool supports the formulation of queries for discovery and the evaluation of constraints of operations. Given the knowledge in the knowledge base, feedback can be provided to the workflow developer, and in specific cases automated discovery of geoprocessing operations can be triggered.

6. Conclusions and Future Work

Extended operation descriptions are a means to provide information about geoprocessing operations that exceed the information available through syntactic interface descriptions [12] and also the functionality provided in established model builders. The objective of this paper was to show what support the information stored in the knowledge base provides for discovery and composition of operations. In summary, the knowledge base provides support for:

- The structured search for geooperators based on concepts and their combinations,
- The discovery of geooperators from different GIS tools,
- The exploitation of relations between geooperators within or across tools,
- The syntactically correct chaining of geoprocessing operations through feedback in case of violations of the constraints of operations,
- The automated discovery of geooperators in case type mismatches are identified between operations.

Complex discovery queries that have been introduced in previous work [13] can be reproduced with the knowledge base through combining the knowledge about geooperators, provided by concepts and categories, with their interface descriptions. Our knowledge base extends previous work in the following points; the number of geooperators exceeds the number of operations documented in previous work [12,13], further pieces of information like relations between operations are available, and the feedback resulting from the evaluation of constraints can be provided. The SPARQL formalization of the preconditions and postconditions is generic and not restricted to specific operations.

We also demonstrated the potential of the knowledge base to support the automated discovery of geooperators. Automated discovery refers to the automated generation of SPARQL queries when a mismatch between the provided input and an operation at hand has been found. In particular, we focused on cases in which type mismatches occur and transformation operations are required (e.g., a transformation from vector to raster data). The full implementation and integration of automated discovery and the recommendation of operations to users are still under elaboration.

The current version of the knowledge base in OWL needs to be extended in a series of respects; the description of operation constraints needs to be analyzed in more detail and a level of detail needs to be specified. This includes an evaluation of the data type ontology and its suitability for capturing the constraints of operations. Potentially, the integration with related work on data type formalizations could be beneficial. Depending on the context in which the knowledge is used, the constraints of operations could also be complemented with information messages. For example, in the project raster operation, a resampling takes place in the course of projecting the raster data. It could be desirable to include information messages rather than warnings about such implicit operations in application cases related to software training.

The knowledge base needs to be extended for further geooperators and linked to a workflow development tool. Ideally, the geooperator descriptions could be extracted at least automatically from available documentations. Brauner [15] demonstrated how the geooperator descriptions can be injected in service description files. In addition to the technical feasibility of the extension of existing service descriptions, the documentation of additional geooperators and the maintenance of the knowledge base require community involvement [16].

Once the knowledge base is integrated in a workflow development tool, its use could lead to a documentation of a developed workflow on a conceptual level. In this context, the contribution of the knowledge base regarding the topics of reproducibility and replicability (e.g., [43]) and its linkage with the PROV-O ontology [44] could be explored. A conceptual workflow documentation could also allow the translation of a workflow from one GIS tool into another, as relations between operations are included in the knowledge base.

Acknowledgments: We thank the anonymous reviewers for their suggestions that helped to improve the paper. We acknowledge financial support by the Open Access Publication Fund of the University of Salzburg.

Author Contributions: Barbara Hofer conceived the work and designed the demonstration of the knowledge base. She filled the knowledge base in OWL with support of Emmanuel Papadakis. Emmanuel Papadakis wrote the Java program for generating SPARQL queries. Barbara Hofer wrote most of the paper; Emmanuel Papadakis contributed to Section 4. Stephan Mäs helped shaping early ideas of this work, provided feedback on the article and on some reviewers' requests.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Janowicz, K.; Hitzler, P. The Digital Earth as knowledge engine. *Semant. Web IOS Press* **2012**, *3*, 213–221.
2. Gore, A. *The Digital Earth: Understanding Our Planet in the 21st Century*; Speech at the California Science Center; California Science Center: Los Angeles, CA, USA, 1998.
3. Craglia, M.; de Bie, K.; Jackson, D.; Pesaresi, M.; Remetey-Fülöpp, G.; Wang, C.; Annoni, A.; Bian, L.; Campbell, F.; Ehlers, M.; et al. Digital Earth 2020: Towards the vision for the next decade. *Int. J. Digit. Earth* **2011**, *5*, 4–21. [CrossRef]
4. Yue, P.; Baumann, P.; Bugbee, K.; Jiang, L. Towards intelligent GIServices. *Earth Sci. Inform.* **2015**, *8*, 463–481. [CrossRef]
5. Kiehle, C.; Greve, K.; Heier, C. Requirements for next generation spatial data infrastructures-standardized web based geoprocessing and web service orchestration. *Trans. GIS* **2007**, *11*, 819–834. [CrossRef]
6. Al-Areqi, S.; Lamprecht, A.-L.; Margaria, T. Constraints-driven automatic geospatial service composition: Workflows for the analysis of sea-level rise impacts. In Proceedings of the International Conference on Computational Science and its Applications, Beijing, China, 4–7 July 2016.
7. Stasch, C.; Scheider, S.; Pebesma, E.; Kuhn, W. Meaningful spatial prediction and aggregation. *Enviorn. Model. Softw.* **2014**, *51*, 149–165. [CrossRef]
8. Laniak, G.F.; Olchin, G.; Goodall, J.; Voinov, A.; Hill, M.; Glynn, P.; Whelan, G.; Geller, G.; Quinn, N.; Blind, M.; et al. Integrated environmental modeling: A vision and roadmap for the future. *Environ. Model. Softw.* **2013**, *39*, 3–23. [CrossRef]
9. Kuhn, W.; Ballatore, A. Designing a language for spatial computing. In *AGILE 2015*; Bacao, F., Santos, M.Y., Painho, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2015; pp. 309–326.
10. Kuhn, W. Core concepts of spatial information for transdisciplinary research. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 2267–2276. [CrossRef]
11. Scheider, S.; Gräler, B.; Pebesma, E.; Stasch, C. Modeling spatiotemporal information generation. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 1980–2008. [CrossRef]
12. Lutz, M. Ontology-Based descriptions for semantic discovery and composition of geoprocessing services. *Geoinformatica* **2007**, *11*, 1–36. [CrossRef]
13. Fitzner, D.; Hoffmann, J.; Klien, E. Functional description of geoprocessing services as conjunctive datalog queries. *Geoinformatica* **2011**, *15*, 191–221. [CrossRef]
14. Qi, K.; Gui, Z.; Li, Z.; Guo, W.; Wu, H.; Gong, J. An extension mechanism to verify, constrain and enhance geoprocessing workflows invocation. *Trans. GIS* **2015**. [CrossRef]
15. Brauner, J. Formalizations for Geooperators—Geoprocessing in Spatial Data Infrastructures. Ph.D. Thesis, Technische Universität Dresden, Dresden, Germany, 2015.
16. Hofer, B.; Brauner, J.; Jackson, M.; Granell, C.; Rodrigues, A.; Nüst, D.; Wiemann, S. Descriptions of spatial operations—Recent approaches and community feedback. *Int. J. Spat. Data Infrastruct. Res.* **2015**, *10*, 124–137.
17. Hofer, B.; Mäs, S.; Brauner, J.; Bernard, L. Towards a knowledge base to support geoprocessing workflow development. *Int. J. Geogr. Inf. Sci.* **2016**, *31*, 694–716. [CrossRef]
18. World Wide Web Consortium. *SPARQL 1.1 Recommendation*; World Wide Web Consortium: Cambridge, MA, USA, 2013.
19. Graser, A.; Olaya, V. Processing: A python framework for the seamless integration of geoprocessing tools in QGIS. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 2219–2245. [CrossRef]
20. Di, L. GeoBrain—A web Services based geospatial knowledge building system. In Proceedings of the NASA Earth Science Technology Conference, Palo Alto, CA, USA, 22–24 June 2004.
21. Qiu, F.; Ni, F.; Chastain, B.; Huang, H.; Zhao, P.; Han, W.; Di, L. GWASS: GRASS web application software system based on the GeoBrain web service. *Comput. Geosci.* **2012**, *47*, 143–150. [CrossRef]
22. Yue, P.; Gong, J.; Di, L.; Yuan, J.; Sun, L.; Wang, Q. GeoPW: Towards the geospatial processing Web. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5886, pp. 25–38.
23. Sun, Z.; Yue, P.; Di, L. GeoPWTManager: A task-oriented web geoprocessing system. *Comput. Geosci.* **2012**, *47*, 34–45. [CrossRef]
24. Wang, J.; Pierce, M.; Ma, Y.; Fox, G.; Donnellan, A.; Parker, J.; Glasscoe, M. Using service-based GIS to support earthquake research and disaster response. *Comput. Sci. Eng.* **2012**, *14*, 21–30. [CrossRef]

25. Bensmann, F.; Alcacer-Labrador, D.; Ziegenhagen, D.; Roosmann, R. The RichWPS Environment for Orchestration. *ISPRS Int. J. Geo-Inf.* **2014**, *3*, 1334–1351. [CrossRef]
26. Open Geospatial Consortium. *OGC WPS 2.0 Interface Standard*; OGC Document 14-065; Open Geospatial Consortium: Wayland, MA, USA, 2015.
27. Open Geospatial Consortium. *Semantic Annotations in OGC Standards*; OGC Best Practices; Open Geospatial Consortium: Wayland, MA, USA, 2012; p. 59.
28. Vahedi, B.; Kuhn, W.; Ballatore, A. Question-based spatial computing—A case study. In *Geospatial Data in a Changing World*; Sarjakoski, T., Santos, M.Y., Sarjakoski, L.T., Eds.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 37–50.
29. Lemmens, R.; Granell, C.; Wytzisk, A.; de By, R.; Gould, M.; van Oosterom, P. Integrating semantic and syntactic descriptions to chain geographic services. *IEEE Internet Comput.* **2006**, *10*, 42–52. [CrossRef]
30. Lutz, M.; Lucchi, R.; Friis-Christensen, A.; Ostländer, N. A rule-based description framework for the composition of geographic information services. In Proceedings of the 2nd International Conference on GeoSpatial Semantics, Mexico City, Mexico, 29–30 November 2007; pp. 114–127.
31. Scheider, S.; Tomko, M. Knowing whether spatio-temporal analysis procedures are applicable to datasets. In Proceedings of the 9th International Conference on Formal Ontology in Information Systems (FOIS 2016), Annency, France, 6–9 July 2016; pp. 67–80.
32. Farnaghi, M.; Mansourian, A. Disaster planning using automated composition of semantic OGC web services: A case study in sheltering. *Comput. Environ. Urban Syst.* **2013**, *41*, 204–218. [CrossRef]
33. Yue, P.; Di, L.; Yang, W.; Yu, G.; Zhao, P. Semantics-based automatic composition of geospatial Web service chains. *Comput. Geosci.* **2007**, *33*, 649–665. [CrossRef]
34. Küster, U.; Stern, M.; König-Ries, B. A classification of issues and approaches in automatic service composition. In Proceedings of the First International Workshop on Engineering Service Compositions (WESC05), Amsterdam, The Netherlands, 1 December 2005.
35. Jones, R.; Cornford, D.; Bastin, L. UncertWeb processing service: Making models easier to access on the web. *Trans. GIS* **2012**, *16*, 921–939. [CrossRef]
36. Lopez-Pellicer, F.J.; Rentería-Agualimpia, W.; Béjar, R.; Muro-Medrano, P.R.; Zarazaga-Soria, F.J. Availability of the OGC geoprocessing standard: March 2011 reality check. *Comput. Geosci.* **2012**, *47*, 13–19. [CrossRef]
37. Müller, M. Hierarchical profiling of geoprocessing services. *Comput. Geosci.* **2015**, *82*, 68–77. [CrossRef]
38. Miles, A.; Bechhofer, S. *SKOS Simple Knowledge Organization System Reference. Recommendation*; World Wide Web Consortium: Cambridge, MA, USA, 2009.
39. Albrecht, J. *Universal GIS-Operations—A Task-Oriented Systematization of Data Structure-Independent GIS Functionality Leading Towards a Geographic Modelling Language*; ISPA—Mitteilungen: Vechta, Germany, 1996; Volume 23, pp. 1–94.
40. The International Organization for Standardization. *ISO 19107: Geographic Information—Spatial Schema*; The International Organization for Standardization: Geneva, Switzerland, 2003.
41. The International Organization for Standardization. *ISO 19123: Geographic Information—Schema for Coverage Geometry and Functions*; The International Organization for Standardization: Geneva, Switzerland, 2005.
42. Keshkamat, S.S.; Looijen, J.M.; Zuidgeest, M.H.P. The formulation and evaluation of transport route planning alternatives: A spatial decision support system for the Via Baltica project, Poland. *J. Transp. Geogr.* **2009**, *17*, 54–64. [CrossRef]
43. Ostermann, F.O.; Granell, C. Advancing science with VGI: Reproducibility and replicability of recent studies using VGI. *Trans. GIS* **2015**. [CrossRef]
44. World Wide Web Consortium. *PROV-O: The PROV Ontology Recommendation*; World Wide Web Consortium: Cambridge, MA, USA, 2013.