

Article

A Task-Oriented Knowledge Base for Geospatial Problem-Solving

Can Zhuang ¹, Zhong Xie ^{1,2,*}, Kai Ma ¹, Mingqiang Guo ¹ and Liang Wu ^{1,2}

¹ School of Information Engineering, China University of Geosciences, Wuhan 430074, China; zhuangcan@cug.edu.cn (C.Z.); makai@cug.edu.cn (K.M.); guomingqiang@mapgis.com (M.G.); wuliang@cug.edu.cn (L.W.)

² National Engineering Research Center for GIS, Wuhan 430074, China

* Correspondence: xiezhong@cug.edu.cn; Tel.: +86-133-8750-5800

Received: 5 September 2018; Accepted: 27 October 2018; Published: 31 October 2018



Abstract: In recent years, the rapid development of cloud computing and web technologies has led to a significant advancement to chain geospatial information services (GI services) in order to solve complex geospatial problems. However, the construction of a problem-solving workflow requires considerable expertise for end-users. Currently, few studies design a knowledge base to capture and share geospatial problem-solving knowledge. This paper abstracts a geospatial problem as a task that can be further decomposed into multiple subtasks. The task distinguishes three distinct granularities: Geooperator, Atomic Task, and Composite Task. A task model is presented to define the outline of problem solution at a conceptual level that closely reflects the processes for problem-solving. A task-oriented knowledge base that leverages an ontology-based approach is built to capture and share task knowledge. This knowledge base provides the potential for reusing task knowledge when faced with a similar problem. Conclusively, the details of implementation are described through using a meteorological early-warning analysis as an example.

Keywords: task; workflow; geospatial problem-solving; knowledge base

1. Introduction

In recent years, with the rapid development of cloud computing and web technologies, an increasing number of geospatial information resources (GIRs), e.g., geospatial data, geospatial analysis functions, models, applications, etc., have been encapsulated into a wide variety of geographic information services (GIServices) [1] which are accessible to general public users over the web [2,3]. For example, a web service toolkit, named GeoPW [4], provides a set of geoprocessing services, which are used to fulfill data processing and spatial analysis tasks over distributed information infrastructures [5]. In the geospatial community, The Open Geospatial Consortium (OGC) established a series of standard interface specifications, such as Web Feature Service (WFS), Web Map Service (WMS), Web Coverage Service (WCS), and Web Processing Service (WPS), which further improve the interoperability and web-based sharing of GIServices [5–7].

In the geospatial application domain, geospatial problems usually relate to heterogeneous data and several computational processes [8]. The capabilities of a single GIService are limited and cannot be effectively conducted because of the complexity of geospatial problems [4]. In the last decade, workflow-based approaches have evolved to a major way to address complex geospatial problems [9]. Currently, with the assistance of standard interface specifications, GIServices published by different organizations can be chained as a geoprocessing workflow that can describe the execution order of problem-solving steps and enhance the power of atomic GIServices to fulfill complex geoprocessing tasks [10–13]. In general, geospatial problems require comparatively deep expertise, which therefore

need experts to contribute their problem-solving knowledge by means of a conceptual workflow. In previous studies, there are already some investigations in the formalization of workflow [7,14,15] and semantic interoperability for GIServices [16–18]. Additionally, a number of studies have employed the task concept to facilitate the expression of user requirements at a semantic level [14,19,20]. In fact, many geospatial problems can share similar conceptual workflows. Therefore, the conceptual workflows can be formalized into a knowledge base, which can facilitate future users to solve the similar problems.

In this paper, we focus on using ontologies in association with a task-oriented approach to construct a knowledge base to enhance geospatial problem-solving. It is generally believed that ontology is the foundation and a significant part of the semantic web. Ontology provides unified terms to improve the semantic interoperability of domain knowledge [21]. A task is introduced as a reusable component to model the sequence of inference steps involved in the process of solving certain kinds of geospatial problems at a conceptual level. The knowledge base can store conceptual workflows that are considered to be a priori knowledge accumulated from past experience of domain experts [22], which can enable problem-solving knowledge reusable [23]. A geospatial problem is abstracted as a task, and the knowledge for the task is considered as a problem solution. Under many circumstances, tasks need to decompose into simpler tasks, each of which can be solved by one or a set of functions [24]. As the smaller task is simpler than the overall task, the complexity of the task is reduced significantly [22]. Hence, we further divide the task into three distinct granularities: (1) a geoperator, which is basic processing functionality; (2) an atomic task, which is indecomposable; and (3) a composite task, which is decomposed into multiple subtasks.

The main work of this paper includes the following: (1) Concepts: the task concept is introduced as a reusable component for geospatial problem-solving and is used to reflect users' requirements; (2) Model: a task model is proposed to simulate problem-solving processes; (3) Knowledge base: an ontological knowledge base is designed, that comprises several interoperable ontologies to capture and share problem-solving knowledge; and (4) Implementation: taking the meteorological early-warning (MEW) analysis, for example, we describe the details of the implementation conclusively. We focus on geospatial problem solution as a task that is composed of conceptual geoprocessing operations not in connection with any concrete services. The instantiation and execution of a task, the low-level interaction with operations (such as accessing input data), and the validation of the processing chain are not in the scope of this article.

The remainder of this paper is organized as follows. Section 2 offers related work on the task-based approach and geospatial problem-solving. Section 3 proposes the task concept and task model to describe problem-solving processes. Section 4 presents a task-oriented knowledge base, some core ontologies are described in detail. Section 5 introduces the detailed information of implementation. Finally, conclusions and future work are given in Section 6.

2. Related Work

2.1. The Task-Based Approach

The notion of the task was proposed by Albrecht in the field of geographic information system as early as in the 1990s [25] and has been used in many studies. However, there is still no unified definition of a task [19]. In general, the task concept reflects user requirements and describes all actions or operations to solve a specific problem. Some studies have been performed using a task-based approach. We summarize and classify them as follows:

1. The task-based language. A task ontology language based on the OWL (Web Ontology Language), named OWL-T, has been proposed to define task templates to formalize user demands and business processes at a high-level abstraction, which is used for the task of a trip plan [26]. Hu et al. [19] extended the task-oriented approach to the OGC Sensor Web domain. A Task

Model Language, called TaskML, is a language for modeling tasks. The significant features of TaskML are Task Trigger, Task Priority, and Task QoS.

2. The task ontology approach. Sun et al. [27] proposed a task ontology-driven approach for the geospatial domain to realize live geoprocessing in a service-oriented environment, which includes three steps: task model generation, process model instantiation, and workflow execution. A case study of flood analysis is used to illustrate the effect and role of the task. Liu [28] proposed a task ontology model for domain-independent dialogue management and created a dialogue manager that is task-independent. Park et al. [29] presented a task ontology based on travelers' perspectives using tasks, activities, relations, and properties. A prototype system was developed using task-oriented menus.
3. A task-based approach for geospatial data acquisition. Wiegand and García [21] proposed a task-based approach to advance geospatial data source retrieval. More concretely, they designed a conceptual model that combines ontologies of tasks, data sources, metadata, and places and uses the Jess rule engine and Protégé tool to provide automatic processing for data retrieval. Qiu et al. [30] proposed a task-oriented approach for efficient disaster data management that performed mapping from emergency tasks to data sources and calculated the correlation between the data set and a generic task. A flood emergency example illustrates the use of this approach.

2.2. Geospatial Problem-Solving

Currently, geoprocessing service technology is widely employed to solve specific geospatial problems in distributed information infrastructures. Much research has been devoted to utilizing or facilitating geoprocessing services to support problem solving. Mikita [31] published a geoprocessing service for forest owners to optimize clearcut size and shape during the process of forest recovery. Müller [18] proposed a hierarchical framework to identify the semantic and syntactic properties of geoprocessing services with four levels of granularity, which is conducive to service retrieval, service comparison, service invocation.

In most cases, a single geoprocessing service is not enough to solve the complex geospatial problem. Therefore, geoprocessing workflow technology provides a solution. The integration of geoprocessing services has become a popular research topic, and a series of tools and architectures were developed to support geoprocessing service chaining. For example, an open source geoprocessing workflow tool, named GeoJModelBuilder, is able to integrate interoperable geoprocessing services and compose them into a workflow [6,7]. A RichWPS orchestration engine in combination with a DSL (Domain Specific Language) is used to orchestrate WPS processes and publish the composition as a WPS process for further composition [32]. In addition, there are many popular workflow management systems to facilitate the integration of geoprocessing services, such as Taverna [33,34], Triana [35], Kepler [36], jABC [9,37]. However, they only simplify the workflow construction process at the syntactical level, and building a workflow composed of services for geospatial problem-solving is still challenging for end-users.

Recently, more studies have focused on semantic and automatic workflow composition for geospatial problem-solving. Farnaghi and Mansourian [12] proposed an automatic composition solution using the AI (Artificial Intelligence) planning algorithm and SAWSDL (Semantic Annotations for Web Service Description Language) to improve the disaster management process. Al-Areqi et al. [10] applied a constraints-driven synthesis method to implement the semiautomatic composition of a workflow for analysis of the impacts of sea-level rise. Samadzadegan et al. [38] designed a framework for an automatic workflow for fire detection early warning based on OGC services. Arul and Prakash presented a unified composition algorithm that adds a new phase called Validation and Optimization to automatic web service composition and generated a scalable composition process according to the dynamic change of user requirements [39].

3. Task as a Reusable Problem-Solving Component

3.1. An Application Scenario

In this section, we demonstrate an example that uses a workflow composed of distributed data and various geoprocessing services. This example is used throughout the remainder of the paper to help understand the concept of a geospatial task. Assuming an end-user is a staff of a meteorological disaster monitoring department, he needs to predict the probability of the occurrence of geological disasters in a certain region in the next day. The ideal result is a thematic map of the early-warning region that uses different colors to represent different early-warning levels.

To achieve the early-warning results, the most common approach is to formulate a geographic processing workflow that can generate an early-warning result map. As shown in Figure 1, the elliptical shape represents a data node, and the rectangular shape represents a data processing node. First, it uses geological hazard point data, influence factor data and early-warning unit data as input data to calculate the potential degree index of early-warning units respectively. Similarly, it can obtain effective rainfall data. Then, the potential distribution map and effective rainfall data from the previous step with forecast rainfall data go through early-warning analysis calculations to achieve the early-warning result map.

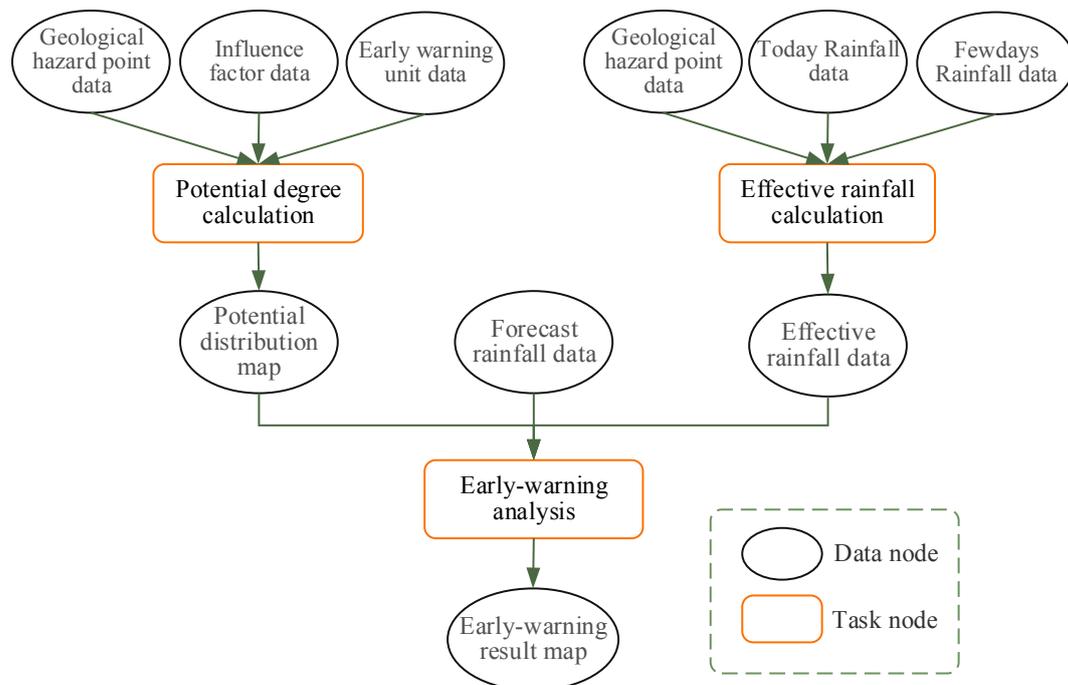


Figure 1. Sequences of the meteorological early-warning (MEW) process.

For the aforementioned application, the entire workflow can be considered a task. GIS domain experts with professional knowledge are able to analyze the technological procedures and abstract them in the form of conceptualization, which are then used to describe the skeleton knowledge of the problem-solving process. The MEW task, which was previously performed manually and had the requirements of GIS skills and knowledge of business processes, can now be executed automatically.

3.2. Task and Task Model

In this paper, the task concept is proposed to reflect user requirements, which can be accomplished by one or more geoprocessing services. A geospatial problem is abstracted as a task that denotes a high-level business goal, and users execute a sequence of processes to achieve the goal. Tasks are

different from operations or services, as tasks focus on what users want to solve, while operations or services mainly focus on the implementation of geoprocessing computations.

A complex problem can consist of multiple problem-solving processes with different requirements, which makes it difficult to define the solution as a single task [22]. Hence, a complex task can be decomposed into several smaller tasks, each of which can be solved in a relatively independent way by one or more geoprocessing services and then combined together into a complete solution [24]. The granularity of the task plays an important role during the problem-solving process. As shown in Figure 2, there are three distinct granularities: (1) a geoperator as elementary functionality for an atomic task, (2) an atomic task as a building block for a composite task, and (3) a composite task as a building block for a complex geospatial task. Consequently, the task is a reusable component for construction of the problem-solving workflow.

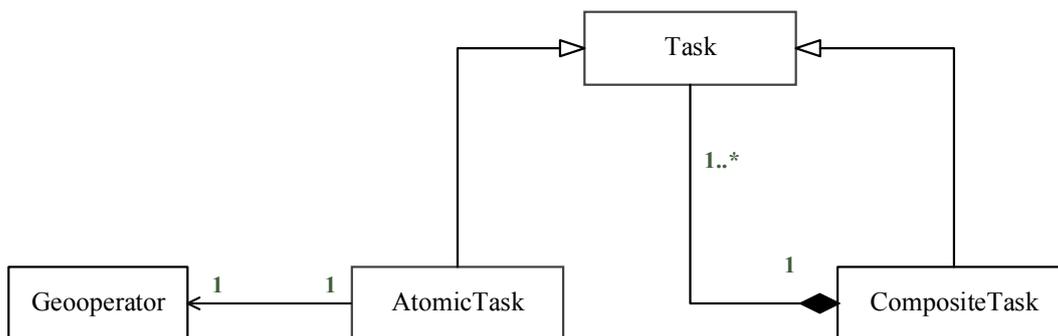


Figure 2. Relationships between Task, AtomicTask, CompositeTask, and Geooperator.

The process property of the geospatial task is expressed by a task process graph (TPG), which is used to capture the execution order of problem-solving steps and closely describe how a task should be achieved. Each TPG contains a set of edges that compose an acyclic directed graph structure. An edge denotes a workflow of two tasks. The directions of edges decide the dependencies between the tasks. The combination of TPG and task composes a task model that provides an approach to allow users to specify complex geospatial problems at an abstract level.

3.3. Geoperator

Geospatial problem-solving knowledge is represented at a conceptualized level that requires categorization and formalization of geoprocessing services. Geoperators are developed mostly for improving the discoverability and exchangeability of geoprocessing functionality and providing an approach to formalize well-defined geoprocessing functionality [40]. In Brauner's work, geoperators are categorized in terms of multiple different perspectives, such as geodata, legacy GIS, pragmatic, formal or technical perspectives [41]. An overview of perspectives and top-level categories identified by Brauner is shown in Figure 3a, and elements described by the geoperator are given in Figure 3b, which can facilitate our work. The former is used to define the subclasses of Geooperator class in the GIS operation ontology without further modification; the latter is partially transformed into data properties and object properties of the Geooperator class.

In this paper, geoperators are introduced to provide a conceptualization for geoprocessing services (such as a geospatial analysis or transformation service) that are encapsulated as standard web services (e.g., WPS) for providing geoprocessing functionalities on the web. From an object-oriented perspective, geoperators act as wrappers for existing geoprocessing services and subsequently serve as building blocks for elementary geoprocessing tasks.

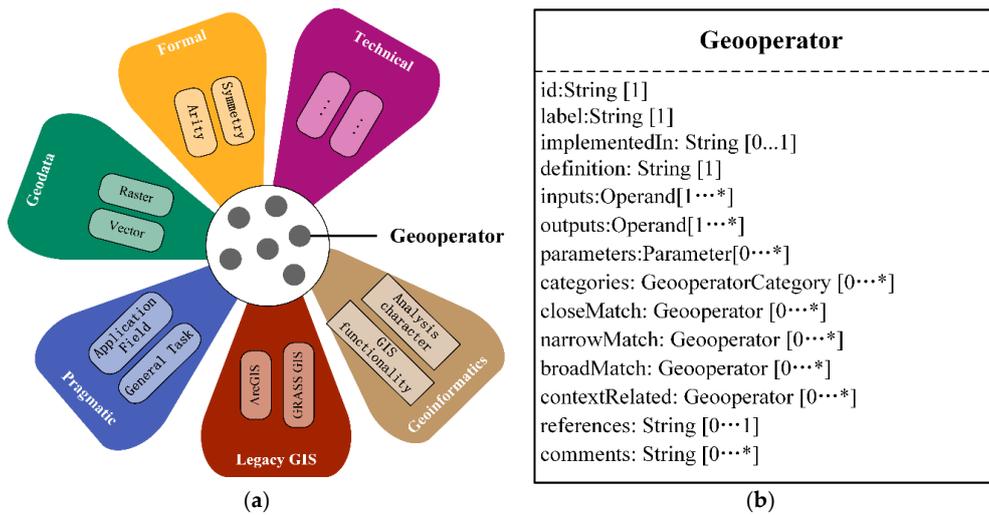


Figure 3. (a) Different perspectives on Geooperator [41] (b) Description elements of Geooperator [17].

3.4. Formal Definition

Definition 1 (Task). A task can be defined as a quadruple:

$$T = (PT, OP, PA, C), \tag{1}$$

where *PT* specifies the type of task, *OP* is spatial inputs and outputs (e.g., spatial datasets), *PA* is a set of non-spatial parameters of a task and *C* consists of the precondition and result that generally constrains the thematic and geometric attributes of input or output data for geoprocessing tasks [42].

Definition 2 (Task Process Graph). A task process graph defines the basic structure of task decomposition [43], which is an acyclic directed graph defined as follows:

$$TPG = (V, E), \tag{2}$$

where *V* is a finite set of *n* vertices $\{v_1, v_2, v_3, \dots, v_n\}$, and each node $v \in V$ represents a task t_v . *E* is a finite set of directed edges $\{e_{v_i, v_j}\}$. Each edge $e_{v_i, v_j} \in E$ can be characterized by a tuple (p_{v_i, v_j}, c_{ij}) . $p_{v_i, v_j} = \langle v_i, v_j \rangle$ is an ordered pair that represents execution precedence between task t_{v_i} and task t_{v_j} ; in other words, t_{v_i} is ahead of t_{v_j} in the sequence of task decomposition that can also be denoted by $v_i \leq v_j$. c_{ij} represents the control flow connector between two tasks, which includes sequence, branching, loop, and so forth.

Definition 3 (Task Model). A task model is defined by a 2-tuple as follows:

$$TModel = (t, tpg), \tag{3}$$

where $t \in T$ is a task instance, and *tpg* denotes a task process graph associated with *t* that defines the decomposition structure. If *tpg* only contains a geooperator, we consider this task to be an atomic task; otherwise, it is a composite task.

Definition 4 (Task Decomposition). Following the definition of the task model, we can further accomplish the task decomposition. Given a task process graph $tpg = (V, E)$, assuming $v \in V, t_v \in T, v$ associates with t_v . If t_v has a corresponding model $tmodel_v = (t_v, tpg_v)$, then the decomposition of the task can be defined by

$$tpg' = Decompose(v, tpg, tmodel_v), \tag{4}$$

where *tpg'* is a new task process graph obtained replacing node *v* with *tpg_v* in *tmodel_v*.

Taking the workflow mentioned in Section 3.1 as an example, Figure 4 depicts the task decomposition procedure. The node “Early-warning analysis” is replaced by a task process graph, which is defined in a task model, where the edges previously connected with this node are revised.

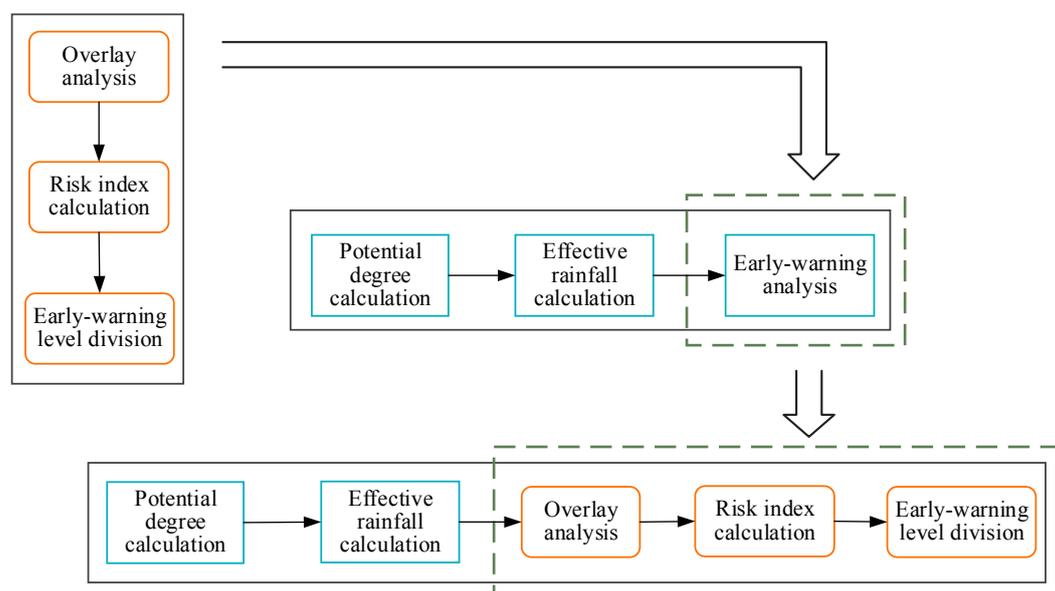


Figure 4. An example of task decomposition.

4. A Task-Oriented Knowledge Base

This section presents the knowledge base that adapts the ontology-based approach and provides comprehensive knowledge to support the geoprocessing task. To build the knowledge base, a set of ontologies are needed to capture knowledge related to the problem-solving solution. The use of ontologies makes the semantic meaning of problem-solving procedures explicit and further facilitates users to obtain the problem solution [44]. Formalizing the knowledge base will assist both GIS non-specialist users and specialists in automating problem solving, allowing reuse and sharing of solutions [21]. Accordingly, we deem that the knowledge base is valuable.

4.1. Background on Ontologies

It is widely known that ontology provides a formal language to standardize and share the semantics of various kinds of domain knowledge. The word ontology was first used as a philosophical concept and address the nature of existence, and it was subsequently introduced into the information domain by researchers. Currently, one of the most prevalent definitions of ontology is “Ontology is an explicit specification of a conceptualization”, which was proposed by Gruber in 1993 [45]. Based on this definition, ontology is essentially a taxonomy of the objective world and a knowledge representation model. Meanwhile, ontology also supports non-taxonomic relations.

According to Perez [46], knowledge in ontologies is formalized by five kinds of modeling primitives: concepts, relations, functions, axioms, and instances. From a mathematical point of view, ontology can be formally expressed by an Equation as follows:

$$O = \{C, R, F, A, I\}, \quad (5)$$

where C is a set whose elements are called concepts; R is a set of relations between concepts, $R \subseteq C \times C$; F is a special relation in which the former $n - 1$ elements can uniquely determine the n -th element, and it can be defined as follows: $F: C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$; A represents a geographic axiom, that is, a collection of assertions in a logical form that are always true; and I stands for instances of concepts.

In the process of building ontology, instances represent objects that can be anything in a domain, and concepts are a set of objects that are mapped to classes. The relations between concepts are realized by properties that are classified into two types: an object property and a data property [47]. An object property specifies the relations between two classes, and it connects two individuals from different classes. A data property defines the relations between individuals and data values, which is similar to an inherent attribute of an object.

4.2. Ontologies at the Heart of the Knowledge Base

To realize the capability to represent the knowledge of the problem-solving process, the knowledge base provides a set of ontologies as follows: Task Ontology, Process Ontology, GIS Operation Ontology, Interface Ontology, Data Type Ontology, GIS Data Ontology, and GIService Type Ontology. These ontologies are combined to provide support for all facets of problem-solving, each of which plays a key role in building a rich, dynamic and flexible task-oriented knowledge base. Figure 5 shows the delineations of the definitions of ontologies and how they relate to each other. Several important ontologies are discussed in detail in the following section.

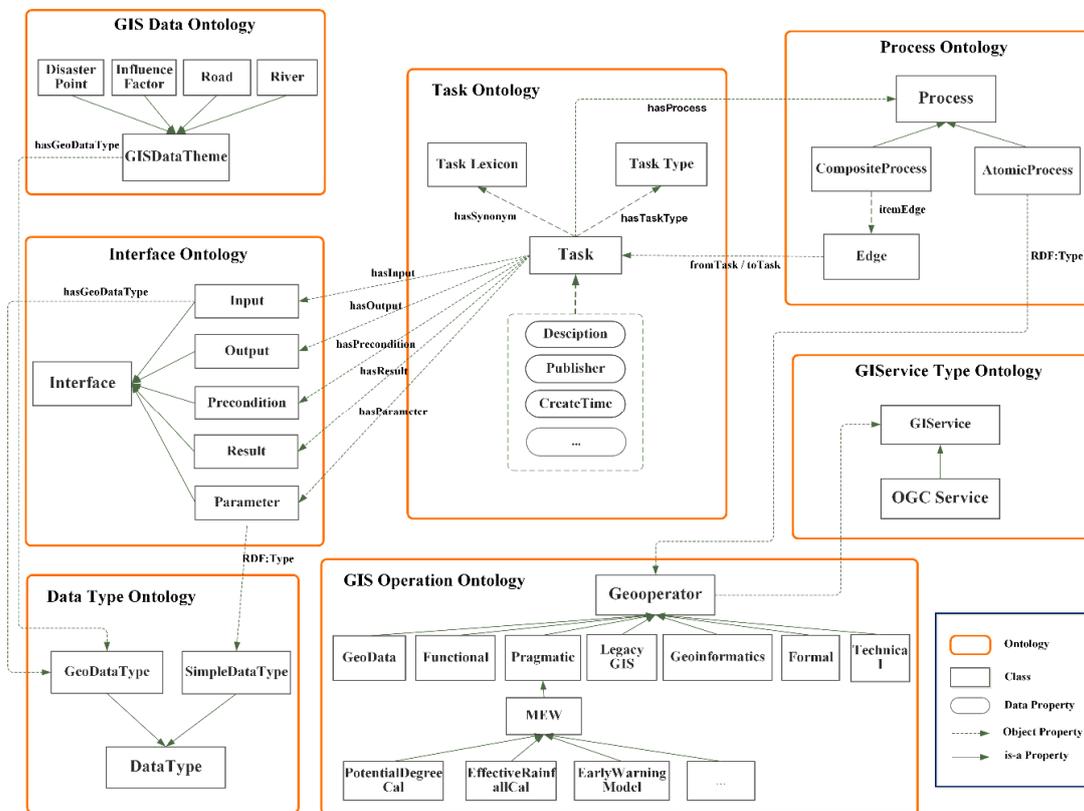


Figure 5. The relationships of ontologies in the knowledge base.

4.2.1. Task Ontology

Task Ontology is the core for supporting problem solving, that defines the Task class to represent a geospatial problem. Its property relations are composed of object properties and data properties. The data properties mainly describe the metadata information of task instances, such as Description, Publisher, Create Time, and so on. The object properties include: hasSynonym, hasTaskType, hasProcess, hasInput, hasOutput, etc.

The Task class refers to the Task Lexicon class through the hasSynonym property for semantic annotations of tasks to provide the words and phrases describing tasks, on the basis of which end users externalize their own expression of the target problem in natural language. This can broaden the scope

of keyword queries and dispose synonyms to support natural language retrieval. The Task Type class describes the categorization of tasks on the basis of functionalities that tasks can implement. The MEW analysis in the example mentioned above is a sort of geospatial task. The Task Type class is linked to the Task class for semantic reference to state the type of task individuals through the predefined hasTaskType property. Each individual of the Task class has at least one conceptual solution which is denoted in the Process ontology. The interfaces of the Task class are defined in the Interface Ontology, which will be described in detail in the following section.

4.2.2. Process Ontology

Process Ontology is used to define problem-solving processes at a conceptual level for a certain type of task, that is not associated with any concrete services. The AtomicProcess and CompositeProcess classes are created as subclasses of the Process class to classify the process individuals according to the number of processes involved. The atomic process directly refers to the Geooperator class in the GIS Operation Ontology using the RDF:Type property; however, the composite process is an edge set that contains some edges. Each edge denotes the sequence of two task nodes that are semantically annotated to the Task class using the fromTask and toTask properties. A series of edges form a directed graph that is called a task process graph that describes how the task works. In this paper, we only consider the linear sequence between two tasks; other control flow logics will be included in future work.

4.2.3. Data Type Ontology

Data Type Ontology is defined to describe the data types that are divided into two categories: SimpleDataType and GeoDataType, as illustrated in Figure 6. SimpleDataType includes some primitive data types in some programming language or description language such as xml:string and xml:float in XML. GeoDataType is an abstract representation of geospatial data, which has some data properties shared by any type of geospatial data, including attribute, data format, and coordinate reference system (CRS). Based on abstract specifications of the International Standard Organization (ISO) for vector [48] and raster data [49], GeoDataType is differentiated as VectorDataType and RasterDataType, each of which has unique characteristics. In vector data, each geospatial feature must identify a geometric type, such as point, polyline, and polygon following OGC Simple Feature Specification [50]. The resolution and band number must be identified in raster data.

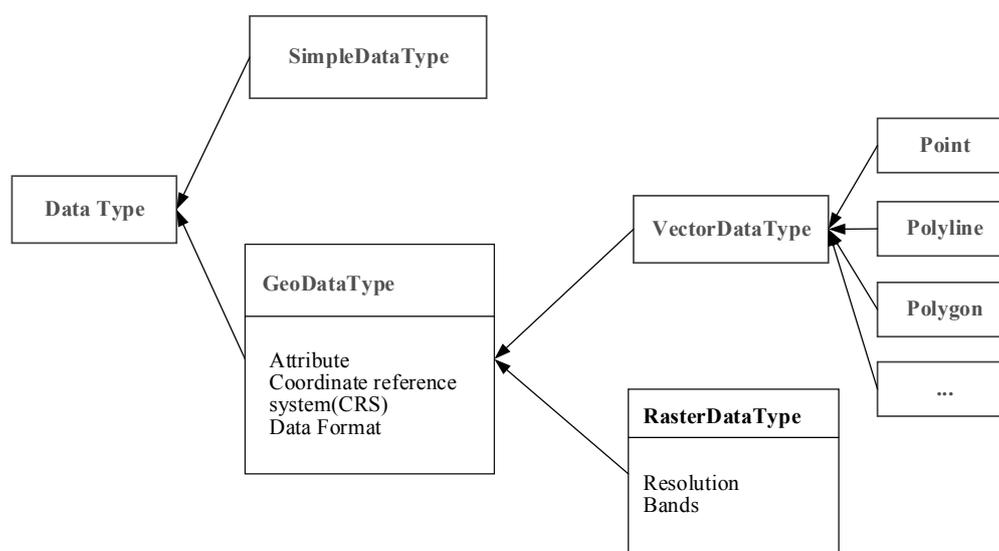


Figure 6. Data type specifications.

4.2.4. GIS Operation Ontology

In GIS Operation Ontology, the Geoperator class is employed to conceptualize geoprocessing functionalities. The notion of Geoperator has been introduced in the previous section. The geoperators are used as building blocks for the conceptual workflow of geospatial problem-solving. This ontology of the knowledge base is based on work by Hofer [42] who translated the SKOS (Simple Knowledge Organization System) thesaurus provided by Brauner [41] into an OWL ontology and included an additional concept that is known as a functional concept. The SKOS thesaurus contains 40 geoperators. This ontology can be extended by extra categories, if necessary. The categories of the Pragmatic perspective originate from the general task, and are task-oriented categories. Users can further integrate new categories based on practical application. Therefore, in this paper, an additional category named MEW is integrated into the Pragmatic perspective of the geoperator, and subcategories or geoperators can be created for a further description of geoprocessing operations. Based on this classification, geoprocessing services that perform geospatial functionalities are thought of as individuals of the Geoperator class.

4.2.5. Interface Ontology

As introduced in the previous section, tasks are used as reusable components to accomplish the composition of problem-solving processes. The composition requires an evaluation of the correspondence of interfaces. The knowledge base needs to include sufficient information of interfaces to satisfy the needs of the composition. An interface requires the description of operands that contain inputs and outputs, constraints that contain a precondition and result, and non-spatial parameters. Consequently, as illustrated in Figure 7, the Interface class consists of the subclasses Input, Output, Parameter, Precondition, and Result. GeoDataType in Data Type Ontology is used to specify operands of interfaces, whereas non-spatial parameters can refer to SimpleDataType which includes conventional data types. The Precondition class focuses on the thematic and geometric properties of the input to ensure the correct function of the operation [42]. The Postcondition class defines the expected result of the output.

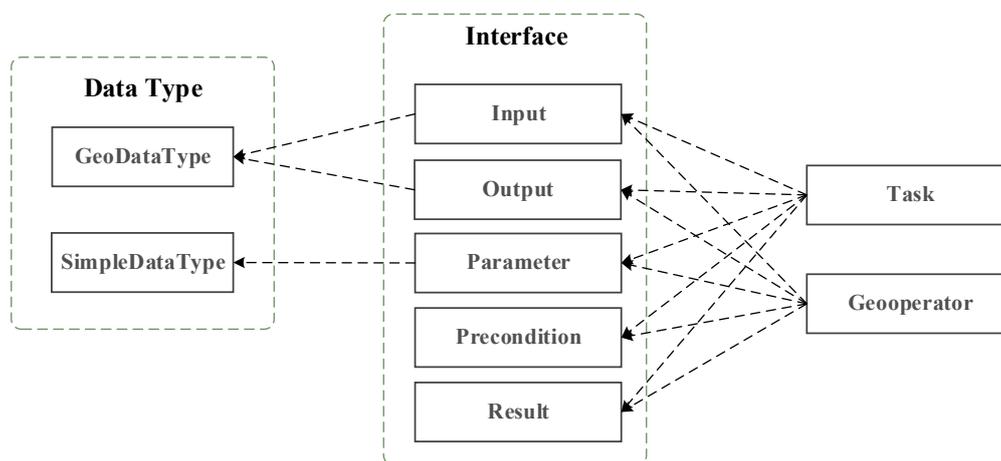


Figure 7. Interface for annotating Task and Geoperator, and Data Type for specifying Interface.

Similarly, we extend the interface properties of geoperators using the Interface Ontology which presently does not involve the related interface specifications.

5. Implementation

In Section 3, we introduce an application scenario that is a geospatial problem-solving process in the context of MEW. We take this example to demonstrate the benefits of the ontology-based knowledge

base for tasks during the process of geospatial problem-solving. The implementation includes three parts: creation of ontologies, representation of knowledge, and task instances.

5.1. Creation of Ontologies

Based on the proposed architecture of the task-oriented knowledgebase described in Section 4.2, we build different abstract ontologies to represent the hierarchy and relationships of the concepts using Protégé 5.2.0 which is an OWL ontology development platform that allows creation and query of ontologies [21]. In general, an ontology is composed of the following components: concepts and properties of each concept, relationships or constraints between concepts, and instances of concepts [28]. Figure 8a presents all concepts or classes defined in the ontological knowledge base. All object properties that represent the relationships between classes are shown in Figure 8b; they include `hasTaskType`, `hasSynonym`, `hasProcess`, etc. The abstract ontologies can be instantiated for specific tasks. In this paper, the task instances for meteorological early-warning are implemented, which are detailed in the next section.

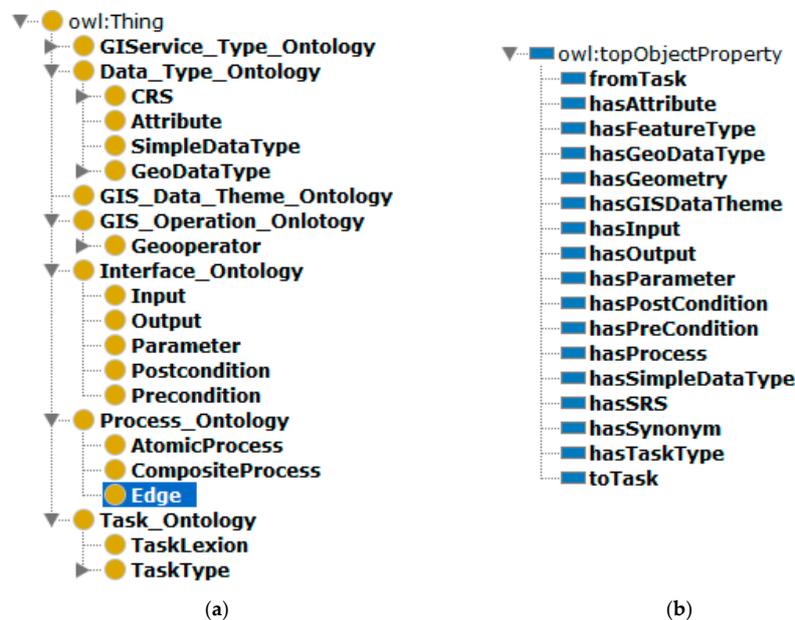


Figure 8. An excerpt of ontologies where (a) depicts the classes of ontologies, and (b) illustrates the object properties between classes.

5.2. Representation of Ontology Knowledge

Once the components of an ontology are developed, the ontology can be represented by ontology description language, such as Resource Description Framework (RDF) and Web Ontology Language (OWL). RDF is built upon XML, which uses triples of object, property, and value to describe resources. OWL is a W3C-recommended standard semantic markup language being developed by the Semantic Web community, which is an extension of RDF [15,21]. In this paper, we use OWL as a standard and machine-readable language to represent the knowledge of ontologies, which is presented as an OWL file.

Meanwhile, we use property restrictions including `hasValue` and quantifier restrictions to limit associations between different classes [15]. The `hasValue` restriction specifies that the individuals of a class have a given value. Nevertheless, the quantifier restriction limits the individuals of a class using an existential restriction (\exists , `owl:someValuesFrom`) or a universal restriction (\forall , `owl:allValuesFrom`). The former states that values for the restricted property have at least one instance of class, which is defined by existential restriction; however, the latter states that all values for the restricted relationship must be a type of instance. For example, an MEW analysis task only needs effective rainfall data,

forecast rainfall data, and potential degree data that can be restricted with the following formal statement: $\forall \text{hasInput} (\text{Effective_Rainfall_Data} \cup \text{Forecast_Rainfall_Data} \cup \text{Potential_Degree_Data})$. This statement defines a universal restriction on the “hasInput” property between the Task class and the Input class (Figure 5). The OWL notation using the “owl:allValuesFrom” restriction is shown in Figure 9.

```

<owl:Class rdf:ID = "Meterological Early-warning Task" >
  <rdfs:subClassof>
    <owl:Restriction>
      <owl:onProperty rdf:resource = "hasInput" />
      <owl:allValuesFrom rdf:resource = "#Effective_Rainfall_Data" />
      <owl:allValuesFrom rdf:resource = "#Forecast_Rainfall_Data" />
      <owl:allValuesFrom rdf:resource = "#Potential_Degree_Data" />
    </owl:Restriction>
  </rdfs:subClassof>
</owl:Class>

```

Figure 9. Snippets of owl notation using a universal restriction.

5.3. Task Instances

The specific task instances can be represented using classes and properties defined in the ontologies. Using the meteorological early-warning mentioned in Section 3.1 as an example, the tasks involved in the MEW example are listed in Table 1, in which there are two composite tasks (e.g., EWATask) and six atomic tasks (e.g., ERCTask, and FQTask). We use ERCTask as an example of an atomic task instance, which is used to calculate the effective rainfall. Figure 10 shows the individuals and properties involved in the ERCTask instance. The process of an atomic task is an individual of AtomicProcess, while those of composite tasks are not. We list the namespace declaration of ontologies and the syntax of class, subclass, and property definitions using OWL, as shown below Figure 10.

Table 1. Tasks involved in the MEW example

Task Type	Abbreviation	SubTask	Description
PotentialDegreeCalTask	PDCTask	FQTask FWCTask PDITask	Calculate potential degree index from multiple influence factor data
EffectiveRainfallCalTask	ERCTask		Calculate effective rainfall
EarlyWarningAnalysisTask	EWATask	OATask HRITask EWLTask	Generate a forecast map according to an early warning model
FactorQuantificationTask	FQTask		Quantify the factor data according to a certainty factor model
FactorWeightCalTask	FWCTask		Calculate factor weight
PotentialDegreeIndexCalTask	PDITask		Calculate potential degree index
OverlayAnalysisTask	OATask		Overlay the input data.
HazardRiskIndexCalTask	HRITask		Calculate the hazard risk index
EarlyWarningLevelTask	EWLTask		Divide early-warning level according to the risk index

Differing from the atomic task, the process of the composite task is composed of multiple edge individuals, each of which describes the data flow between two task instances. A set of edges compose a process graph that denotes how the task works. For example, Figure 11 shows the task instance of a composite task called EWATask. The process individual “process:EWAProcess” contains two edge individuals: “process:EWAEdge1” and “process:EWAEdge2”. The former connects the

two task instances: “task:QATask” and “task:HRITask”, and the latter connects “task:HRITask” and “task:EWLTask”. These edge individuals are linked to process individuals with the itemEdge property.

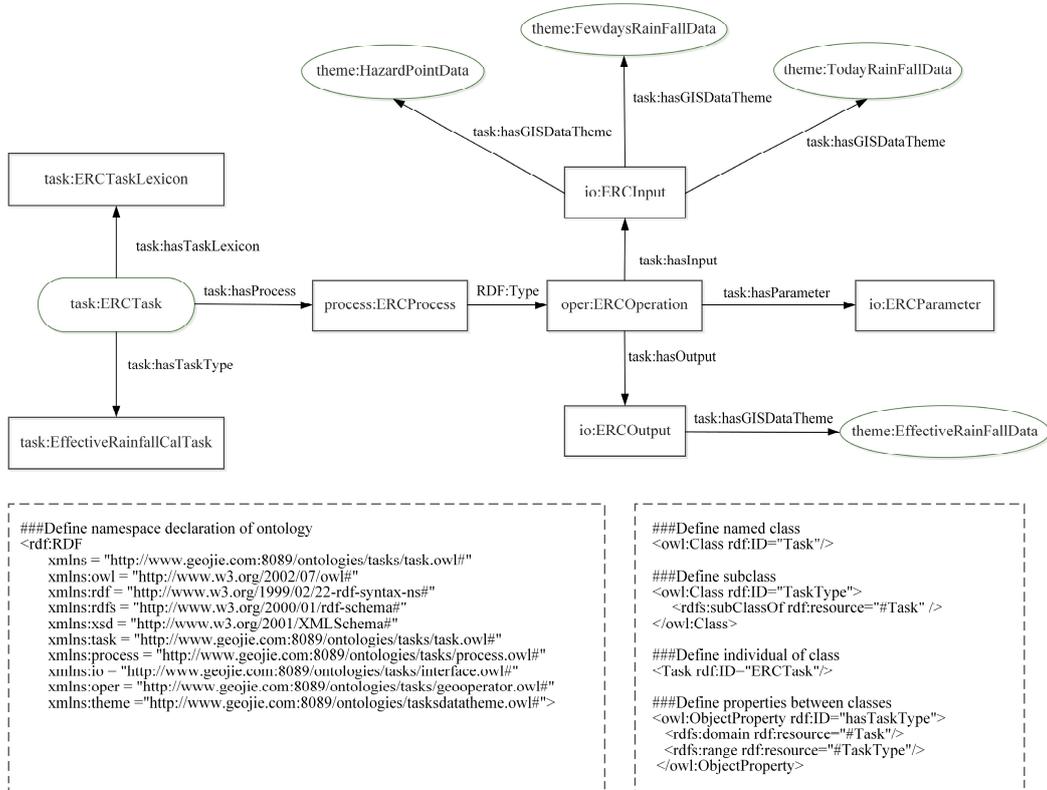


Figure 10. The task instance of an atomic task (EffectiveRainfallCalTask).

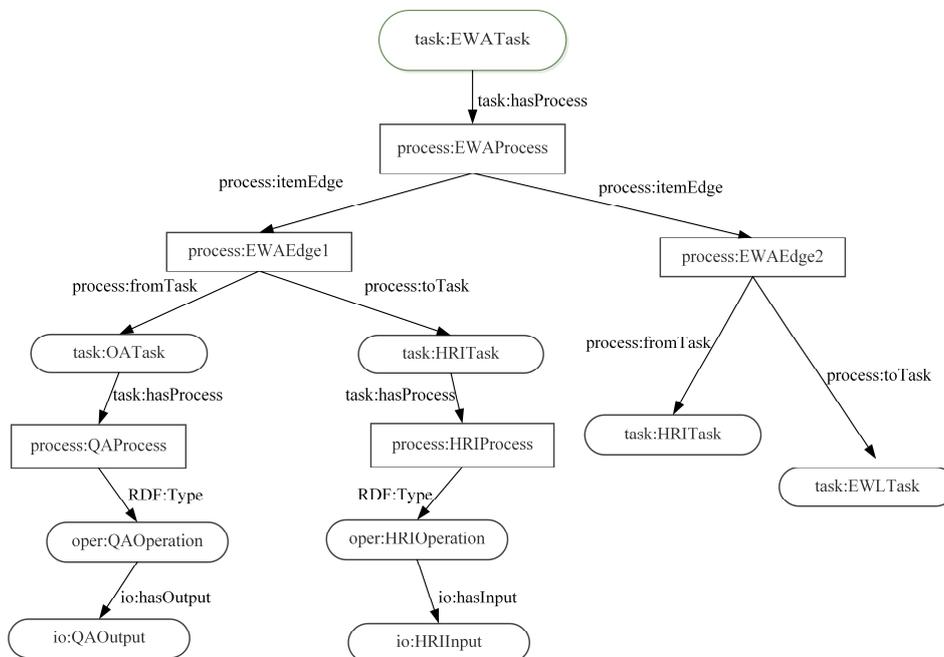


Figure 11. The task instance of a composite task (EarlyWarningAnalysisTask).

5.4. Prototype

A prototype system based on the realized ontology representation and formalized task instances was developed to facilitate users to solve complex geospatial problems. The implemented prototype,

leverages a number of web techniques, such as Ajax, XML, JSON, EasyUI, GoJS, OpenLayers, Apache Axis2, and so on. Ajax is used for asynchronous data exchange between the client and server sides. XML and JSON are data exchange formats. EasyUI and GoJS are client UI frameworks, and GoJS is employed to draw a flowchart. OpenLayers is a JavaScript class library package for WebGIS client development, which is used to achieve map data access. Apache Axis2 is used to provide the Web Service interface. The Java API package Apache Jena [51], a Semantic Web framework for Java, is used to parse the ontology file, access ontology definitions, and infer knowledge [52]. The Apache Tomcat server was employed as a web container. The prototype system can be accessed using Microsoft IE or a Google browser in a Windows operating system

The MEW analysis in Henan, China is used as an example to utilize the knowledge base to support geospatial problem solving. First, we define formal semantics in the ontology-based knowledge base by creating task instances using an ontology editor. The task instance is named EWATask (Figure 11) which can decompose into three subtasks (OATask, HRITask, and EWLTask), The ontology files are generated using OWL format language which is mentioned in the previous section.

Second, the web services, including three data access services (Potential_Degree_Data, Effective_Rainfall_Data, and Forecast_Rainfall_Data) and three geoprocessing services (wps_overlay, wps_riskIndex, and wps_ewLevel), are published with the support of MapGIS IGServer [53]. The details of data access services are shown in Table 2, the geoprocessing services follow the WPS specification, and the workflow model for EWATask is shown in Figure 12.

Table 2. Data access services involved in EWATask.

Data Name	Service Type	SRS	Geometry
Potential_Degree_Data	WFS	Xi'an 80	Polygon
Effective_Rainfall_Data	WFS	Xi'an 80	Polygon
Forecast_Rainfall_Data	WFS	Xi'an 80	Polygon

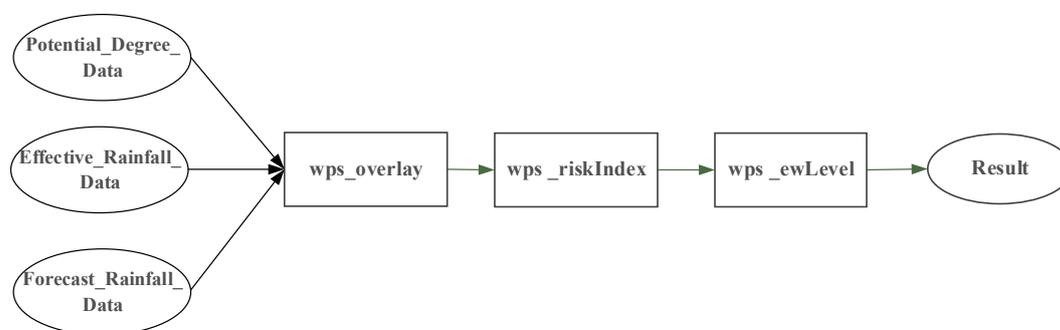


Figure 12. The workflow model of EWATask. Data access services are represented in elliptical shapes, and geoprocessing services are represented in rectangular shapes.

Finally, the prototype system provides an intuitive and easy-to-use graphical user interface (GUI). The end-users can access the GUI of the prototype system using a web browser. As shown in Figure 13, in the left panel, there is a tree structure showing the task lists that are parsed from the knowledge base. The user selects and clicks on a tasknode; then, the process model of the task will be displayed in the form of a flowchart in the right panel (step 1). Next, right-click on a process node and select the menu “service binding” (step 2). A service binding window pops up and allows end-users to bind the appropriate service and input the related parameters manually (step 3). Repeat this step for each process node. Finally, execute the task and get the result map (step 4). For instance, Mr. Wang took Henan province in China as a forecast area for the risk analysis and forecasted the possibility of the occurrence of geological hazards in the next 24 h. He clicked the EarlyWarnAnalysisTask node in the prototype system, the process graph of which was shown in the right panel (Figure 13). Following this workflow, he bound the appropriate geoprocessing

services (wps_overlay, wps_riskIndex, and wps_ewLevel) that were invoked with a linear sequence (wps_overlay → wps_riskIndex → wps_ewLevel). According to the forecast results, an early-warning result map as shown in the lower right of Figure 13, was obtained, which uses different colors to represent different early-warning levels.

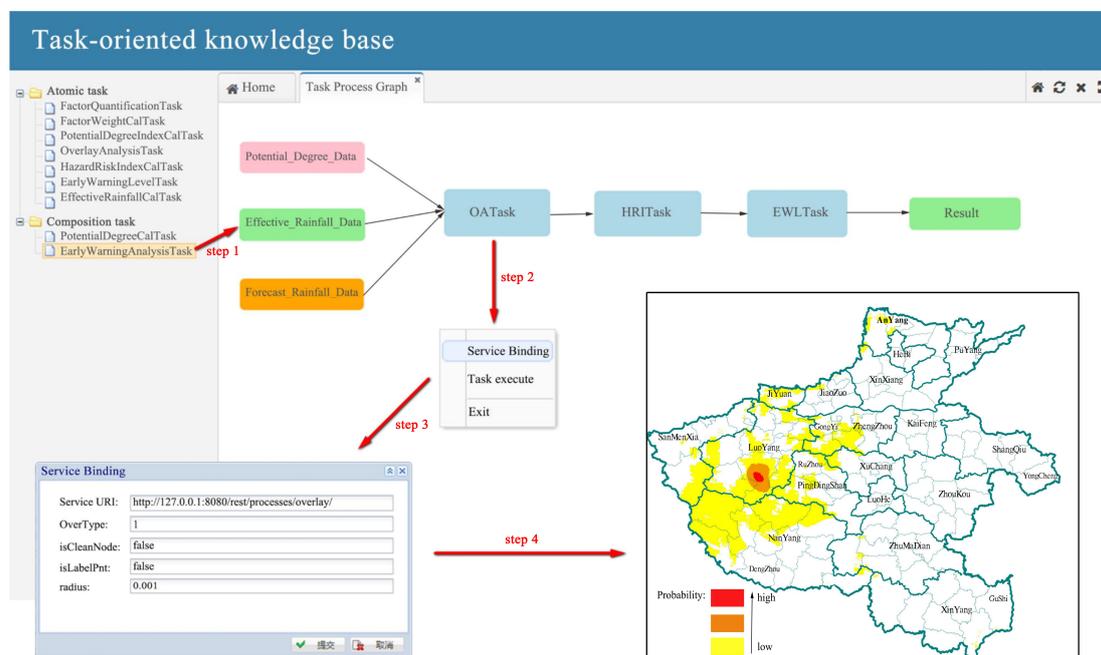


Figure 13. The graphical user interface of the prototype system.

6. Conclusions and Future Work

This paper proposes a task model and abstracts a geospatial problem as a task that can be used as a reusable component for problem-solving. A task-oriented knowledge base is built to capture sharable and reusable geospatial problem-solving knowledge. In the knowledge base, we combine multiple ontologies (e.g., Task Ontology, Process Ontology, and GIS Operation Ontology) to provide assistance for all facets of problem-solving. This knowledge base is not tightly-coupled with any specific workflow language. The required knowledge about problem-solving is stored in the knowledge base which employs ontology and task-oriented approach to achieve the formalization and reusability of tasks.

This knowledge base is tailored for domain experts to create and share their professional geospatial problem-solving knowledge. For the end-users, a user-friendly interface is needed to submit a geospatial problem and query the problem solution. An approach that has the capabilities of parsing natural language input will be developed in future work. This approach would allow users to input free-text to submit problem requirements.

In this paper, we only concentrate on using ontologies to describe a conceptual workflow that is composed of a linear sequence of GIS functionalities. We do not present an algorithm to instantiate into a concrete service chain and execute this workflow. The approach of knowledge transformation, instantiation and execution of a workflow will be implemented in future work.

Author Contributions: C.Z. designed the knowledge base, implemented the prototype system, and wrote the paper. K.M. and M.G. deployed and performed the prototype system. L.W. contributed the materials and tools. Z.X. conceived the early ideas of this work, reviewed the paper, and provided some suggestions and feedback.

Funding: This work was funded by the National Key Research and Development Program of China (Grant Nos. 2017YFB0503600, 2018YFB0505500, 2017YFC0602204), National Natural Science Foundation of China (Grant Nos. 41671400, 41701446), and Hubei Province Natural Science Foundation of China (Grant No. 2017CFB277).

Acknowledgments: We acknowledge the anonymous reviewers for their valuable comments and suggestions to improve this paper.

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

References

1. Yue, P.; Baumann, P.; Bugbee, K.; Jiang, L. Towards intelligent giservices. *Earth Sci. Inform.* **2015**, *8*, 463–481. [[CrossRef](#)]
2. Li, X.; Song, J.; Huang, B. A scientific workflow management system architecture and its scheduling based on cloud service platform for manufacturing big data analytics. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 119–131. [[CrossRef](#)]
3. Yang, Z.L.; Cao, J.; Hu, K.; Gui, Z.P.; Wu, H.Y.; You, L. Developing a cloud-based online geospatial information sharing and geoprocessing platform to facilitate collaborative education and research. In Proceedings of the ISPRS—International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016 XXIII ISPRS Congress, Prague, Czech Republic, 12–19 July 2016; Volume XLI-B6, pp. 3–7.
4. Peng, Y.; Gong, J.; Di, L.; Jie, Y.; Sun, L.; Sun, Z.; Qian, W. GeoPW: Laying blocks for the geospatial processing web. *Trans. GIS* **2010**, *14*, 755–772.
5. 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* **2016**, *20*, 240–258. [[CrossRef](#)]
6. Zhang, M.; Bu, X.; Yue, P. Geojmodelbuilder: An open source geoprocessing workflow tool. *Open Geospat. Data Softw. Stand.* **2017**, *2*, 8. [[CrossRef](#)]
7. Yue, P.; Zhang, M.; Tan, Z. A geoprocessing workflow system for environmental monitoring and integrated modelling. *Environ. Model. Softw.* **2015**, *69*, 128–140. [[CrossRef](#)]
8. 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](#)]
9. Lamprecht, A.L.; Steffen, B.; Margaria, T. Scientific workflows with the jabc framework. *Int. J. Softw. Tools Technol. Transf.* **2016**, *18*, 629–651. [[CrossRef](#)]
10. 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; pp. 134–150.
11. Kliment, T.; Bordogna, G.; Frigerio, L.; Crema, A.; Boschetti, M.; Brivio, P.A.; Sterlacchini, S. Image data and metadata workflows automation in geospatial data infrastructure deployed for agricultural sector. In Proceedings of the Geoscience and Remote Sensing Symposium, Milan, Italy, 26–31 July 2015; pp. 146–149.
12. 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](#)]
13. Al-Areqi, S.; Lamprecht, A.-L.; Margaria, T. Automatic workflow composition in the geospatial domain: An application on sea-level rise impacts analysis. In Proceedings of the 19th AGILE International Conference on Geographic Information Science, Helsinki, Finland, 14–17 June 2016.
14. Sun, Z.; Yue, P.; Di, L. Geopwtmanager: A task-oriented web geoprocessing system. *Comput. Geosci.* **2012**, *47*, 34–45. [[CrossRef](#)]
15. Jung, C.T.; Sun, C.H.; Yuan, M. An ontology-enabled framework for a geospatial problem-solving environment. *Comput. Environ. Urban Syst.* **2013**, *38*, 45–57. [[CrossRef](#)]
16. Lutz, M. Ontology-based descriptions for semantic discovery and composition of geoprocessing services. *Geoinformatica* **2007**, *11*, 1–36. [[CrossRef](#)]
17. Hofer, B.; Mäs, S.; Brauner, J.; Bernard, L. Towards a knowledge base to support geoprocessing workflow development. *Int. J. Geogr. Inf. Syst.* **2016**, *31*, 694–716. [[CrossRef](#)]
18. Müller, M. Hierarchical profiling of geoprocessing services. *Comput. Geosci.* **2015**, *82*, 68–77. [[CrossRef](#)]
19. Hu, L.; Yue, P.; Zhang, M.; Gong, J.; Jiang, L.; Zhang, X. Task-oriented sensor web data processing for environmental monitoring. *Earth Sci. Inform.* **2015**, *8*, 511–525. [[CrossRef](#)]
20. Gorton, S.; Reiff-Marganiec, S. Towards a task-oriented, policy-driven business requirements specification for web services. In Proceedings of the International Conference on Business Process Management, Vienna, Austria, 5–7 September 2006; pp. 465–470.
21. Wiegand, N.; García, C. A task-based ontology approach to automate geospatial data retrieval. *Trans. GIS* **2007**, *11*, 355–376. [[CrossRef](#)]

22. Luo, J. The Semantic Geospatial Problem Solving Environment: An Enabling Technology for Geographical Problem Solving under Open, Heterogeneous Environments. Ph.D. Thesis, The Pennsylvania State University, Pennsylvania, PA, USA, 2007.
23. Jung, C.T.; Sun, C.H. Ontology-driven problem solving framework for spatial decision support systems. *Tetsu- to-Hagane*. **2010**, *47*, 512–515.
24. Vahedi, B.; Kuhn, W.; Ballatore, A. Question-based spatial computing—A case study. In *Geospatial Data in a Changing World*; Springer International Publishing: Cham, Switzerland, 2016; pp. 37–50.
25. Albrecht, J. Universal elementary GIS tasks-beyond low-level commands. In Proceedings of the Sixth International Symposium on Spatial Data Handling, Edinburgh, UK, 3–7 August 1994; pp. 209–222.
26. Tran, V.X.; Tsuji, H. Owl-t: An ontology-based task template language for modeling business processes. In Proceedings of the Acis International Conference on Software Engineering Research, Management & Applications, Busan, Korea, 20–22 August 2007; pp. 101–108.
27. Sun, Z.; Yue, P.; Lu, X.; Zhai, X.; Hu, L. A task ontology driven approach for live geoprocessing in a service-oriented environment. *Trans. GIS* **2012**, *16*, 867–884. [[CrossRef](#)]
28. Yuan, X.; Liu, G. A task ontology model for domain independent dialogue management. In Proceedings of the IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems, Tianjin, China, 2–4 July 2012; pp. 148–153.
29. Park, H.; Yoon, A.; Kwon, H.C. Task model and task ontology for intelligent tourist information service. *Int. J. U- E-Serv. Sci. Technol.* **2012**, *5*, 43–58.
30. Linyao, Q.; Zhiqiang, D.; Qing, Z. A task-oriented disaster information correlation method. In Proceedings of the 2015 International Workshop on Spatiotemporal Computing, Fairfax, VA, USA, 13–15 July 2015; Volume II-4/W2, pp. 169–176.
31. Mikita, T.; Balogh, P. Usage of geoprocessing services in precision forestry for wood volume calculation and wind risk assessment. *Acta Univ. Agric. Silvic. Mendel. Brun.* **2015**, *63*, 793–801. [[CrossRef](#)]
32. Bensmann, F.; Alcacelrabador, D.; Ziegenhagen, D.; Roosmann, R. The richwps environment for orchestration. *ISPRS Int. J. Geo-Inf.* **2014**, *3*, 1334–1351. [[CrossRef](#)]
33. Hull, D.; Wolstencroft, K.; Stevens, R.; Goble, C.; Pocock, M.R.; Li, P.; Oinn, T. Taverna: A tool for building and running workflows of services. *Nucleic Acids Res.* **2006**, *34*, 729–732. [[CrossRef](#)] [[PubMed](#)]
34. Wolstencroft, K.; Haines, R.; Fellows, D.; Williams, A.; Withers, D.; Owen, S.; Soilandreyes, S.; Dunlop, I.; Nenadic, A.; Fisher, P. The taverna workflow suite: Designing and executing workflows of web services on the desktop, web or in the cloud. *Nucleic Acids Res.* **2013**, *41*, 557–561. [[CrossRef](#)] [[PubMed](#)]
35. Taylor, I.; Shields, M.; Wang, I.; Harrison, A. The triana workflow environment: Architecture and applications. In *Workflows e-Science*; Springer: London, UK, 2007; pp. 320–339.
36. Altintas, I.; Berkley, C.; Jaeger, E.; Jones, M. Kepler: An extensible system for design and execution of scientific workflows. In Proceedings of the International Conference on Scientific and Statistical Database Management, Santorini Island, Greece, 23 June 2004; pp. 423–424.
37. Lamprecht, A.L.; Margaria, T.; Steffen, B. *Modeling and Execution of Scientific Workflows with the jABC Framework*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 14–29.
38. Samadzadegan, F.; Saber, M.; Zahmatkesh, H.; Joze Ghazi Khanlou, H. An architecture for automated fire detection early warning system based on geoprocessing service composition. In Proceedings of the SMPR 2013, Tehran, Iran, 5–8 October 2013; pp. 351–355.
39. Arul, U.; Prakash, S. A unified algorithm to automatic semantic composition using multilevel workflow orchestration. In *Cluster Computing*; Springer: New York, NY, USA, 2018; pp. 1–22.
40. 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.
41. Brauner, J. Formalizations for Geooperators-Geoprocessing in Spatial Data Infrastructures. Ph.D. Thesis, Technische Universität Dresden, Dresden, Germany, 2015.
42. Hofer, B.; Papadakis, E.; Mäs, S. Coupling knowledge with GIS operations: The benefits of extended operation descriptions. *Int. J. Geo-Inf.* **2017**, *6*, 40. [[CrossRef](#)]
43. Crubézy, M.; Musen, M.A. *Ontologies in Support of Problem Solving*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 321–341.
44. Zhao, P.; Di, L.; Yu, G.; Yue, P.; Wei, Y.; Yang, W. Semantic web-based geospatial knowledge transformation. *Comput. Geosci.* **2009**, *35*, 798–808. [[CrossRef](#)]

45. Gruber, T.R. A translational approach to portable ontologies. *Knowl. Acquis.* **1993**, *5*, 199–220. [[CrossRef](#)]
46. Perez, A.G.; Benjamins, V.R. Overview of knowledge sharing and reuse components: Ontologies and problem-solving methods. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99), Stockholm, Sweden, 31 July–6 August 1999.
47. Zhong, S.; Fang, Z.; Zhu, M.; Huang, Q. A geo-ontology-based approach to decision-making in emergency management of meteorological disasters. *Nat. Hazards* **2017**, *89*, 531–554. [[CrossRef](#)]
48. The International Organization for Standardization (ISO). *ISO 19107: Geographic Information—Spatial Schema*; The International Organization for Standardization: Geneva, Switzerland, 2003.
49. ISO. *ISO 19123: Geographic Information—Schema for Coverage Geometry and Functions*; The International Organization for Standardization: Geneva, Switzerland, 2005.
50. OGC. *OGC Abstract Specifications: Topic 5—Features*; Open Geospatial Consortium: Wayland, MA, USA, 2009; pp. 8–126.
51. Apache Jena. Available online: <http://jena.apache.org/> (accessed on 29 August 2018).
52. Zhang, C.; Zhao, T.; Li, W. Automatic search of geospatial features for disaster and emergency management. *Int. J. Appl. Earth Obs. Geoinf.* **2010**, *12*, 409–418. [[CrossRef](#)]
53. Wu, X.; Liu, X.; Zhou, S. *Principle and Method of MapGIS IGServer*; Publishing House of Electronics Industry: Beijing, China, 2012.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).