

Article

A Spatio-Temporal VGI Model Considering Trust-Related Information

Yijiang Zhao ^{1,2}, Xiaoguang Zhou ^{1,*}, Guangqiang Li ¹ and Hanfa Xing ³

¹ School of Geosciences and Info-physics, Central South University, Changsha 410083, China; zhaoyijiang@163.com (Y.Z.); ligq168@163.com (G.L.)

² Key Lab of Knowledge Processing and Networked Manufacturing, Hunan University of Science and Technology, Xiangtan 411201, China

³ College of Geography and Environment, Shandong Normal University, Jinan 250014, China; xinghanfa@163.com

* Correspondence: zxcgsu@foxmail.com; Tel.: +86-731-8887-7984

Academic Editor: Wolfgang Kainz

Received: 31 May 2015; Accepted: 28 January 2016; Published: 3 February 2016

Abstract: Over the past several years, volunteered geographic information (VGI) has expanded rapidly. VGI collection has been proven to serve as a highly successful means of acquiring timely and detailed global spatial data. However, VGI includes several special properties. For example, the contributor's reputation affects the quality of objects edited, and a geographic object may have multiple versions. The existing spatio-temporal data model cannot describe the unique properties of VGI. Therefore, a spatio-temporal VGI model considering trust-related information is presented in this paper. In this model, central elements of the VGI environment, e.g., geographic entity, entity state, state version, contributor, reputation, geographic event, and edit event, and their interaction mechanisms are analysed. Major VGI objects and relations are determined using the object-oriented method and trust-related operations, and their relationships are analysed, and nine linkage rules among trust-related operations are found to maintain the consistency of corresponding data. A prototype system for the spatio-temporal VGI model is presented, and the effectiveness of the model is verified.

Keywords: GIS; VGI; spatio-temporal model; reputation; trust; operation

1. Introduction

Crowdsourcing data, *i.e.*, volunteered geographic information (VGI) [1–3], constitute a promising tool for borderland data acquisition [4]. At present, VGI has served as an alternative source of professional geographic information. As VGI offers broad and timely information and is free, it has been applied in many domains, e.g., transportation, emergency relief, disaster management, navigation, *etc.* [5–10].

Whereas traditional spatio-temporal data are collected and updated according to standard specifications by professionals and data quality is ensured through strict quality control procedures and implementation specifications, one state of an entity has only one version in a traditional geographical database, and the quality of all objects is typically uniform; VGI is spontaneously compiled by citizens, most of whom are not professionals [1,11]. Rather, one state of an entity may be edited several times by several volunteers and can have several versions in a VGI database, and the quality of each version may vary. Thus, new problems arise in VGI environments. Because there are several versions of an entity state, which version is the best one? How can one select the best multi-version objects?

Several researchers have focused on the overall quality evaluation of VGI datasets, e.g., data completeness, positional accuracy, *etc.*, by comparing crowdsourcing data with authoritative data [12].

However, such methods cannot be used to assess the quality of single features in a VGI database [13]. In solving such problems, some researchers have evaluated the quality of VGI based on its trust and provenance [14]. For example, Bishr and Mantelas introduced the concept of “information trust”, which has successfully been applied to evaluate the trust of urban growth data. The trust of a geographic object was computed using a rating mechanism [14]. Keßler *et al.* used trust as a proxy measure for the quality of VGI, and the trust of a geographic object was determined based on its VGI system editing history [13]. The authors concluded that trust-related information is essential in a VGI system.

In VGI environments, the trust of each version may vary and is typically affected by the reputations of contributors involved. The reputation of each contributor may vary as well. A contributor’s reputation is affected by his or her degree of professionalism honesty, attitudes, and *etc.* In a VGI system, therefore, trust-related information, *i.e.*, version, state, version trust, contributor reputation, *etc.*, must be determined.

In existing VGI models, *e.g.*, OpenStreetMap (OSM), Google Map Maker, Wikimapia, whereas some contributor information is represented, trust-related information, *e.g.*, version trust, contributor reputation, *etc.*, is not represented directly. Thus, from existing VGI systems, one cannot determine the best version of an entity state.

Based on the above observations, a spatio-temporal VGI model considering trust-related information is presented in this paper. In this model, geographic objects, object states, and state versions are used as three different element levels. Every object may have several states, and every state may have one or more versions as well. Geographic and edit events are also distinguished. Whereas a geographic event may cause a change in the state of one or several geographic objects, an edit event is the cause of a change in a version in a database. All versions of VGI objects are stored in our model. Contributor reputations and the trust of VGI object version can be computed, stored and accessed. To ensure the consistency of data used during dynamic operations, *e.g.*, object creation, modification, and deletion, *etc.*, a set of trust-related operations and their linkage rules are presented in this paper.

This paper is divided into seven sections. Related work is presented in Section 2. Then, in Section 3, we discuss features of VGI by comparing it with traditional geographic information. A spatio-temporal VGI model based on trust-related information is proposed in Section 4. Trust-related operations and their linkages are discussed in Section 5. The experimental procedure and analysis results are presented in Section 6. Finally, Section 7 provides a summary and concludes the discussion.

2. Related Work

VGI has been a contentious issue in GIS research over the past decade [10]. Several researchers have examined VGI applications [7,15,16], data quality and reliability [11–13,17–19], data processing and cleaning methods [20,21], data organization and management systems [22], *etc.* However, related works of our study mainly include spatio-temporal data models, approaches to VGI object quality evaluation, and VGI data organization methods. We discuss related work on these three issues in this section.

2.1. Spatio-Temporal Data Model

Over the past several decades, dozens of spatio-temporal data models have been created. Existing models can be divided into four categories based on their areas of focus. The first category of models focuses on recording the states of geographic objects and includes the snapshot model [23], base state with amendment model [23], space-time composite data model [24], and space-time cube model [24]. The second category of models focuses on object change processes and causal relationships between events and includes event-state-temporal data models [25,26], event-based versioning models [27], and event-based cause-effect models [28]. The third category of models focuses on the description of spatio-temporal objects and relationships and includes object-oriented spatio-temporal data models [29,30], the three-domain model [31], the history graph model [32],

and the spatio-temporal entity-relationship model [33]. The fourth category of models focuses on modelling moving objects and includes moving objects data models [34,35]. Each of the aforementioned spatio-temporal data models presents unique characteristics. However, these models primarily focus on traditional professional geographic information and do not consider contributor reputation and the trust of each geographic object. Therefore, these models cannot answer the following question: “Which version is the best for an object state with multi-versions?”

2.2. Evaluation of VGI Quality

Many efforts have been made to study the quality of VGI. Methods for evaluating the quality of VGI can be generally divided into three categories. For the first category, the quality of VGI is evaluated by comparing VGI with corresponding professional data. For example, OpenStreetMap (OSM) data have been evaluated by various scholars *via* comparisons with corresponding professional data, e.g., UK [11], Germany [36–38], France [39], Iran [12], *etc.* For the second category, VGI quality is evaluated based on its history rather than through comparisons with reference data. For example, all OSM data and their historical versions are stored in an OSM full history dump [40]. Barron *et al.* proposed a comprehensive framework for assessing the intrinsic quality of OSM data [41]. Bégin *et al.* also proposed a means of assessing the completeness of VGI based on contributor behaviours [42]. For the third category, trust is used as a proxy of VGI quality. For example, Bishr *et al.* proposed a formal trust and reputation model for filtering and extracting high quality information on urban growth behaviours on the part of actors (people) [14]; Keßler *et al.* used trust as a proxy measure for the quality of VGI and measured VGI trustworthiness based on five indicators, *i.e.*, VGI contributors, versions, confirmations, corrections and revisions [13]. D’Antonio *et al.* presented a data trustworthiness and user reputation model whereby trust of VGI was evaluated based on direct, indirect and temporal effects and whereby user reputation could be computed based on the trust value of objects created by a user [43]. However, existing VGI quality evaluation methods cannot cover each version of all the VGI objects.

2.3. VGI Data Organization

As VGI is application driven, only a few researchers have studied VGI organization methods, e.g., Qian *et al.* designed Zoom Quadtree to support multi-representations of VGI data [44]. Few researchers have described the special properties of VGI. On the other hand, VGI systems have developed organizational methods for managing crowd-sourcing data.

OpenStreetMap (OSM) has been one of the most popular and successful VGI projects. It uses PostgreSQL to store vector and attribute data. Three data primitives are used to represent geographic information, *i.e.*, nodes, ways, and relations. Only nodes store latitudes and longitudes. OSM separately store current and historical data. In OSM, map browsing and data updating services are also separate. However, OSM does not process user and object trustworthiness. Data primitives in Google Map Maker include place marks, lines, and polygons. Map Maker does not store topological information, and features added by users must be validated before being used in Google Map applications. When a user makes numerous successful contributions, his/her edits will be monitored less closely and may be published on the map immediately [45]. Wikimapia combines an interactive web map with a geographically referenced wiki system. A user can add new items or can edit items on a Wikimapia layer. When an object is created, the contributor is invited to specify its category, to add a textual description, and to upload relevant photos. Wikimapia stores the editing histories of all users and rates them. High-level users have achieved increased access to editing tools and face fewer restrictions on editing activities [46]. Both Google Map Maker and Wikimapia use the concept of user reputation rather than that of object trustworthiness. However, these systems do not manage trust-related information.

From the above analyses, we conclude that whereas several researchers have reported that trust-related information is essential for VGI quality control, in traditional professional spatio-temporal data models and existing VGI systems, user reputations and object trustworthiness are not monitored.

Therefore, existing methods cannot answer the question presented in Section 1, *i.e.*, “Which version is best for a certain state?” Based on this observation, we propose a spatio-temporal VGI model that considers trust-related information.

3. Characteristics of VGI Data Organization

As numerous volunteers contribute to VGI systems, a large amount of geographic data can be collected over a short period of time [1]. VGI spatio-temporal data present the following main advantages, *i.e.*, real time, rapid transmission, rich information, and low cost [11,47]. However, most citizens are not professionals and are randomly distributed throughout the world [39], causing VGI leading to the following quality issues, *e.g.*, heterogeneous quality, redundancy, and incompleteness [41,47]. To solve such quality issues, aspects of VGI data organization are analysed through comparisons with traditional spatio-temporal data in this section.

3.1. VGI States, Versions and Events

A state is “characterized by the absence of change” [48]. From a spatial perspective, a state refers to the condition (including positioning, geometry, attributes, *etc.*) of a geographic entity in the real world at a certain time [30], and it is related to changes that can occur during the lifespan of an entity [49]. Over the lifespan of an entity, such changes can be made as a result of man-made or natural causes. For example, at time t_1 , the state object A is s_1 . A was changed in the real world (*e.g.*, its position or shape) at time t_2 . Then, the state of A was changed and its state was adapted to state s_2 . According to the Cambridge dictionary, a version is “a particular form of something that is different from other forms of the same thing” [50]. In the information science domain, a version is used to store different states of a given entity in a database [51]. In other words, a version is the database reflection of a state of an entity. Database versioning mechanisms (*e.g.*, relation-level and tuple-level versioning) are used to track the evolution of an object’s state through time [52] and have been typically used to manage the evolution of geo-referenced data in GIS [24]. States and versions of VGI have the same meanings as they do in traditional scenarios. However, their relationships in VGI are somewhat different from those in traditional cases. In a traditional spatio-temporal system, when the state of an involved object has changed, to guarantee the currency of spatial data in the database, the current state of the object must be updated as the current version, and the previous version becomes the historical version [53]. Thus, object updating occurs only when an object’s state has changed. In this case, a version of the database corresponds only to an object state in the real world, *i.e.*, there is a 1:1 relationship between states and versions.

As noted in the first section of this paper, in a VGI context, one state of an entity may be edited several times by several volunteers to improve data quality or to support different cognition perspectives. Figure 1 describes how multiple users can participate in the editing of object A. Object A contains m ($m \geq 1$) states, and each state may have several versions. For example, state s_j has several versions, *i.e.*, v_i, v_{i+1}, \dots, v_k . Contributor c_1 contributes the first version v_1 of state s_1 ; Contributor c_i contributes the first version v_i of state s_j ; User c_{i+1} generates version v_{i+1} by editing version v_i ; User c_k edits the selected version of state s_j , and generates version v_k ; and Contributor c_n edits the selected version of state s_m and generates version v_m . From the analysis presented in Figure 1, we conclude that there are two main reasons for updating objects in VGI: in the first case, a change in an object’s state results from a real geographic event (*e.g.*, an earthquake, floods or development plan). This type of event results in transitions between states [48] and always leads to the generation of a new version of a new state from a previous entity state [49]; a second reason for object updating is to improve the quality of VGI (not caused by a geographic event), and in such cases, the state of a corresponding geographic entity in the real world does not change. This type of event can also be viewed as non-geographic event. In our model, each process of object updating to a GIS database, including the two reasons described above, belongs to an event. To support comparisons with geographic events, we refer to this event as an edit event. An edit event occurs when a database is updated. Geographic events and

non-geographic events spur edit events. For example, when the state of object A is changed due to an earthquake, a geographic event called an earthquake is documented. Then, a new version of A is updated in a GIS database, creating an edit event. If the state of object B is not changed, a user modifies it, generating a new version. In this case, no geographic event is documented, but an edit event is. In Figure 1, a change from state s_1 to state s_j occurs in the first case, *i.e.*, update caused by a geographic event. The real state of versions v_i, v_{i+1}, \dots, v_k corresponds to s_j , and an update is performed by the contributors through edition operations to improve data quality or to contribute another perspective. In this case, state presents a 1:m relationship with the version. Therefore, it is necessary to store both state and version information in the database in this VGI context.

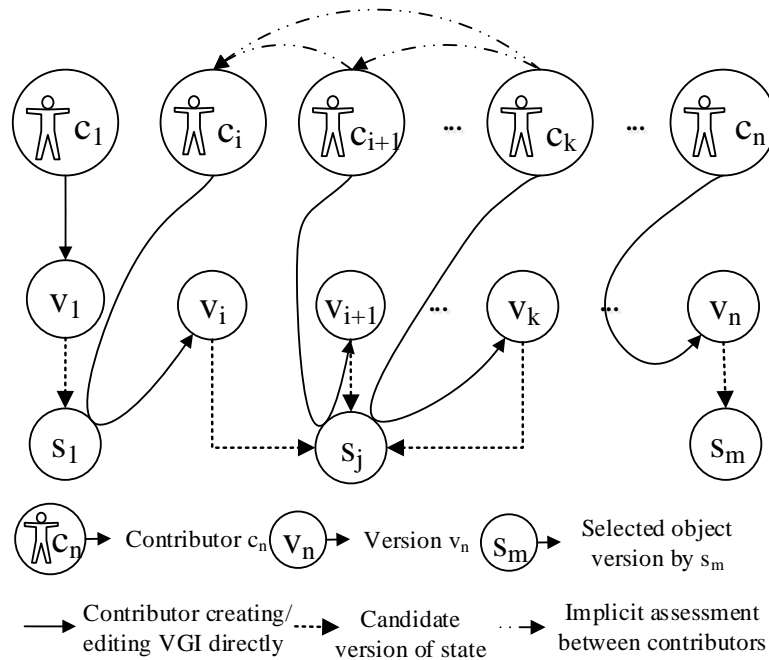


Figure 1. Relationships among contributors, object versions and trust degrees using a simple evolutionary process of a geographic object as an example.

Therefore, in a VGI system, concepts of states, versions, and events are the same as those of traditional systems. However, their relationships change, *i.e.*, unlike one version corresponding to one state in traditional cases, states of an entity in a VGI system may have several versions. Whereas in traditional systems, updates are made only due to geographic events, in VGI systems, updates can be made due to geographic or non-geographic events. Thus, it is necessary to identify special relationships among states, versions and events in VGI systems.

3.2. VGI Trust and Reputation

Several studies have discussed issues of VGI trust and reputation. Mooney *et al.* examined the editing histories of OpenStreetMap objects and concluded that the final version of a VGI object is not always the best version [54]. For example, state s_j in Figure 1 has n versions. The last version v_n may not be the best version of the state. Therefore, the selection of VGI objects is still a contentious issue. However, traditional quality criteria (e.g., lineage, accuracy, consistency and completeness) are based on probability theory. One cannot evaluate the quality of each object version by this criteria as it is not feasible for VGI environments [11,14]. Many scholars believe that the trust degree of an object can be used as a proxy measure for the quality of VGI [13,14]. Trust can be defined as “a bet about the future contingent actions of others” [55]. It can also be regarded as having confidence in a person or thing one is interacting with [56]. Bishr *et al.* have successfully applied this principle to VGI scenarios [14]. Therefore, if we can determine degrees of trust for each version of every object state, the version with

the highest trust degree can be considered the best (or selected) version of the corresponding object state. Bishr *et al.* proposed that trust between a consumer and a VGI information entity “is mediated by interpersonal trust between the VGI originator and the VGI consumer” [57], and a contributor’s reputation is the main factor that affects trustworthiness of a VGI object. Keßler *et al.* assessed the trustworthiness of VGI by tracking VGI editing processes [58]. Based on factors proposed by Bishr and Keßler *et al.*, the trustworthiness of VGI can be evaluated based on contributor reputations and editing processes. Therefore, the trustworthiness of VGI can be defined as follows.

In a VGI scenario, the trustworthiness of an object version is based on a subjective perception of trustworthiness inferred from information on the historical editing process of corresponding objects. Therefore, factors that affect the trustworthiness of an object version mainly include the editing process and the reputation of all contributors involved in the editing process.

Many researchers believe that contributor reputations play an important role in evaluations of the trustworthiness of an object [13,14] and have discussed the various assessment methods. In service-oriented environments, an agent’s reputation is “an aggregation of the recommendations from all of the third-party recommendation agents” [56]. In reference to VGI environments, Keßler *et al.* argued that user reputations can be evaluated by “how the edited features have been treated by the community afterwards (e.g., by looking at revisions), potentially weighted by their respective local knowledge” [13]. Van Exel *et al.* argued that user reputations are related to an individual’s “local knowledge, experience and recognition” [59].

As in online service-oriented environments, VGI contributor reputation is typically affected by initial reputation and by assessment reputation derived from contributions made to a system. An initial reputation typically refers to “the reputation values for newly deployed participants, or newcomers” [60]. When a user is initially registered into a system, he/she has made no contribution to the system; thus, his/her reputation cannot be assessed by what he/she has contributed. Thus, initial reputation is typically determined by the authenticity and integrity of registration information, e.g., identity, e-mail, true name, education, *etc.* Once a user contributes a VGI object to a system and once others evaluate his/her contribution, his/her assessment reputation is yielded. Unlike in the e-business system, users like to evaluate commodities using words or through direct rankings; in VGI systems, users typically modify objects to an acceptable state through edition operations without direct evaluation. In our opinion, this type of edition operation—*i.e.*, edit operations do not occur because the state of an entity has changed but to improve the quality of an object—can be referred to as an implicit assessment. For example, in Figure 1, state s_j of object A has multiple versions. In fact, over the state lifespan of state s_j , the entity state does not change per se, and modifications within a state are instead made to improve the quality of the object. In addition, similarities between the versions denote support for the latter’s contributions to the former version, *i.e.*, a high degree of similarity corresponds to a greater degree of support and vice versa. Thus, in VGI systems, a contributor’s reputation assessments among users can be divided into two categories, *i.e.*, explicit and implicit assessment. Explicit assessment refers to a rating mechanism similar to that used in an e-business system, *i.e.*, a user or product (a geographic object in VGI) is directly rated by others. Implicit assessment refers to the assessment of edition operations.

Therefore, in a VGI scenario, a contributor’s reputation is the synthesis of his/her initial and assessment reputations. For new contributors, initial reputations from registration information play a key role; for other contributors, assessment reputations play an increasingly important role as they make more contributions.

From the above analysis, it can be concluded that geographic objects, object states, and geographic events (reason for object or object state changes) are common between VGI and professional GIS data. Contributors, state versions, contributor reputations, edit events, and trust of object versions are additional trust-related elements found in VGI environments. As these elements must be represented in VGI systems, a model that considers these elements is discussed in the next section.

4. Spatio-Temporal Model Incorporating Trust-Related Information

As noted in Section 3, elements of VGI systems include geographic objects, object states, geographic events, contributors, state versions, contributor reputations, edit events, object version trustworthiness, *etc.* These elements and their relationships must be represented. As Unified Modelling Language (UML) has been an industrial standard in the domain of visualization modelling language and has been widely used in object-oriented programming, it is used here to describe our model. According to the above analysis, our spatio-temporal model that incorporates trust-related features mainly includes the geographic object class (*GeoObject*), object state class (*State*), *Version* class, *Trust* class, *Contributor* class, contributor’s experience class (*UserExpRep*), initial reputation class (*InitRep*), *Reputation* class, geographic event class (*GeoEvent*) and edit event class (*EditEvent*). The simple UML graph for our model is shown in Figure 2.

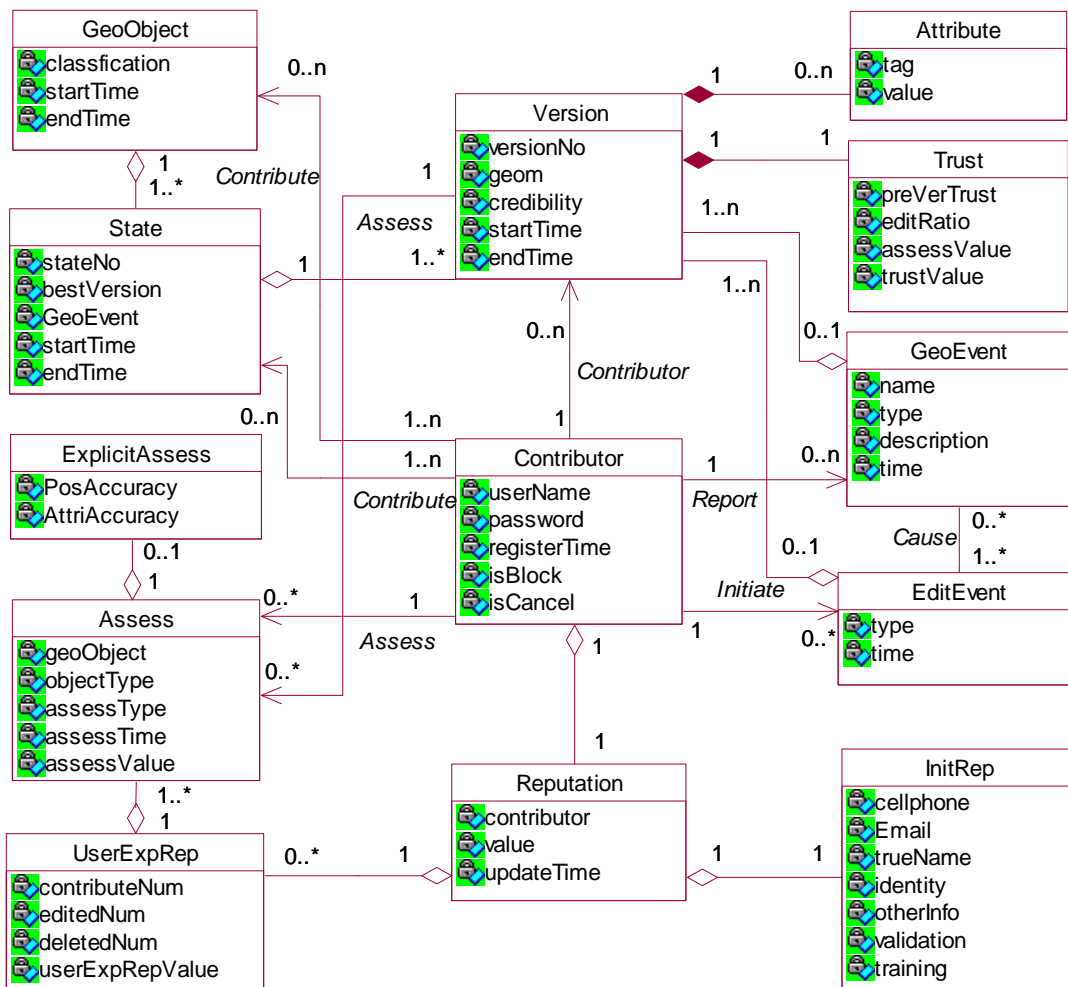


Figure 2. UML (Unified Modelling Language) diagram of the spatio-temporal VGI model that considers trust-related information.

The geographic object class (*GeoObject*) is used to describe geographic objects, and it is the core component of the spatio-temporal model. *GO* objects are the objects of this class. *GO* objects describe their entire lifespan in a VGI system, and their life span is represented by interval time (*ST*, *ET*). A *GO* object must have a classification that refers to a classification code to describe different map layers (e.g., water, transportation, vegetation, residential area, *etc.*). This attribute is conducive to the fusion of crowdsourcing and professional geographic data, and it is denoted by *C*. Therefore, data fields of the *GeoObject* class can be described as tuples (*C*, *ST*, *ET*). A *GO* object is added by a

contributor and owns one or more states. The *GeoObject* class includes three subclasses (point, line and polygon), and its operations mainly include *addGeoObjects* (adding objects), *deleteGeoObjects* (deleting objects), *getGeoObjects* (querying geographic objects), and *moveInventory*. Among these operations, the *deleteGeoObject* function is mainly used to delete an error object. Depending on spatial data use frequency patterns, older data are typically used less frequently and must be moved into an inventory to improve system efficiency (e.g., storage and retrieval), and the *moveInventory* function refers to the movement of data from an existing database to a historical inventory.

The object state class (*State*) refers to a snapshot of a geographic object over a certain time interval. The *STA* is the object of this class. An *STA* object describes its entire lifespan in a VGI system, which is represented by a time interval (*ST*, *ET*). An *STA* object owns one or more *Versions*, it must have a state number (*SN*) and an *SV* attribute, which denotes the best version or a selected version of an object state and which is used to quickly retrieve the corresponding version of a state. Therefore, *State* class data fields can be described from tuples (*SN*, *SV*, *ST*, *ET*). An *STA* object is created by a contributor and has one or more versions, and its operations mainly include the following: *addStates* (adding states), *getStates* (querying states), *deleteStates* (deleting states), and *moveInventory*.

The object version class (*Version*) refers to an editing snapshot of an object at a certain time, and it is added by a volunteer. A *VER* is an object of this class. A *VER* object includes data members as spatial characteristics (*S*), a start time (*ST*), an end time (*ET*), a version number (*VN*), etc. Fields of the *Version* class can be described by tuples (*VN*, *S*, *ST*, *ET*, *C*), where *C* denotes the credibility of a *VER* object, which accounts for trustworthiness, contributor reputations and the traditional quality elements (positional accuracy, attribute accuracy, logical consistency, completeness, and lineage) of an object version. Although traditional quality elements of an object version are beyond the scope of this article, this issue requires further discussion. A *VER* object is added by a contributor and includes attribute characteristics and trust degree, and one or several *VER* objects are owned by a state object (*State*), a geographic event (*GeoEvent*) and an edit event (*EditEvent*). Due to the implicit assessment mechanism, a *VER* object is also associated with an *Assess* class. Operations of the *Version* class mainly include the following: *addVersion/deleteVersion/getVersion* (adding/deleting/querying *VER* objects), *getTrust* (the trustworthiness of a *VER* object), and *moveInventory*.

The *Trust* class refers to trust information and object versions. A *TRU* is an object of this class. A *TRU* object is used to compute and store the trust value of an object version, and it thus forms a part of a *VER* object. A *TRU* object thus includes the following data members: the trust degree (*V*), the assessment value (*AV*), edit proportions between the current version (*VID*) and former version (*EP*), and the trust degree of the former version (*PVT*). Fields of the *Trust* class can be determined from tuples (*PVT*, *EP*, *AV*, *V*). Operations of the *Trust* class mainly include the following: *ComputeEditRatio* (computing edit ratio), *getTrust* (the trustworthiness of a version), *getPreVerTrust* (the trustworthiness of a previous version of an object), *updateTrust* (updating the trustworthiness of a version), *getContributorRep* (the contributor's reputation), *computeTrust* (computing trust), and *moveInventory*.

The *Contributor* class plays an important role in the spatio-temporal VGI model as it is used to describe volunteer account information. A *USER* is an object of this class. A *USER* object is used to describe registration information and includes data members as usernames (*Name*), passwords (*PWD*) and registered times for volunteers (*RT*). A *USER* object has a *Reputation* and is associated with the *Version* and *UserExpRep* objects, and its data fields can be described by the following tuple (*Name*, *PWD*, *RT*, *isB*, *isC*), where *isB* and *isC* are *Boolean* variables that represent whether an account has been temporarily blocked or cancelled, respectively. Operations of the *Contributor* class mainly include the following: *addContributor* (user adding), *cancelContributor* (user cancelling), *blockContributor* (user blocking), and *queryContributor* (user querying).

A contributor's experience class (*UserExpRep*) is used to describe a contributor's assessment reputation. A *UER*, an object of this class, forms part of an *REP* object and includes one or more *Assess* objects. A *UER* object includes the following data members: contributions (*NC*), the number of objects

edited by others (*NE*), the number of objects deleted by others (*ND*), the value of user experience reputations (*V*) and the updating times (*T*). Therefore, fields of the *UserExpRep* class can be described from a tuple (*NC, NE, ND, T, V*). Operations of the *UserExpRep* class mainly include the following: *computeUEP* (computing user assessment reputations), *getUEP* (retrieving user assessment reputations) and *updateUEP* (updating user assessment reputations).

The user assessment class (*Assess*) is used to describe contributor assessment information. It forms part of the *UserExpRep* class. An *AS* is an object of this class. An *AS* object describes data members as geographic objects (*GO*) types of geographic objects (*GT*), assessment types (*AT*), assessment times (*T*) and values (*V*). Therefore, *Assess* class fields can be described from a tuple (*GO, GT, AT, T, V*), where *AT* includes two features: explicit and implicit assessments. Operations of the *Assess* class mainly include the following: *addAssess* (adding assessment), *getAssess* (retrieving assessment), *deleteAssess* (deleting assessment), and *moveInventory*. The *deleteAssess* operation is used to eliminate baleful evaluation effects.

The initial reputation class (*InitRep*) refers to initial reputation-related information on volunteers and owned by *Reputation*. An *IR* is an object of this class. An *IR* object includes data members such as the following: cell phone (*Phone*), e-mail (*Email*), true name (*TNAME*), identity *ID* (*IID*), training status (*TRA*), initial reputation value (*V*), and update time (*T*). Therefore, fields of the *InitRep* class can be described from a tuple (*Email, Phone, TName, IID, TRA, RegInfo, Validate, V, T*), where *RegInfo* represents other user information, such as a contributor's major or education (to simplify the tuple, many user information entries are represented by this field); *Validate* denotes the validation of identity information, e-mail and cell phone contacts, etc. Operations of the *InitRep* class mainly include the following: *addInitRep/updateInitRep/getInitRep* (adding/updating/querying initial reputations), *validateUserInfo* (validating the volunteer's information), *ComputeInitRep* (computing and updating a contributor's initial reputation).

The *Reputation* class is used to describe a contributor's reputation-related information. An *REP* is an object of this class. *REP* objects form part of the *Contributor* class and include *IR* and *UER* objects. They include reputation-related data members, such as reputations (*R*), start times (*ST*) and end times (*ET*). Therefore, fields of the *Reputation* class can be described from a tuple (*R, ST, ET*). Operations of the *Reputation* class mainly include the following: *addReputation* (adding reputations), *updateReputation* (updating reputations), and *getReputation* (querying reputations).

The geographic event class (*GeoEvent*) refers to events occurring in the real world. A geographic event spurs several edit events. A *GE* object must have a *VER* object and is reported by a contributor. It includes data members such as the following: names of geographic events (*Name*), event types (*Type*), event descriptions (*Detail*), and event times (*T*). Therefore, fields of the *GeoEvent* class can be described from a tuple (*Name, Type, Detail, T*). Operations of the *GeoEvent* class mainly include the following: *addEvent/modifyEvent/getEvent* (adding/modifying/querying an event) and *moveInventory*.

The edit event class (*EditEvent*) is used to describe edit event-related information. Such events occur in the information world and not in the real world. An *EE* is an object of this class. An *EE* object must have a *VER* object and is initiated by a contributor. Such objects are also related to *GE* objects and include data members such as the following: event type (*Type*), and event time (*T*). Therefore, *EditEvent* class fields can be described from a tuple (*Type, T*). Operations of the *EditEvent* class mainly include the following: *addEvent/modifyEvent/getEvent* (adding/modifying/querying an event) and *moveInventory*.

5. Trust-Related Operations and Their Linkage Relationships

The trustworthiness of a VGI object and contributor reputations interact. On one hand, a contributor's reputation significantly affects the trust degree of an object that he/she contributes; on the other hand, the modification process of an object affects the contributor's reputation. To keep all trust-related data consistent, we discuss trust-related operations and relationships between these operations in this section, and a set of linkage operation rules are presented.

5.1. Trust-Related Operations

As noted in Section 4, two main trust-related operations are included in our model: user-related reputation operations and object-related trust operations. User-related reputation operations mainly include the following: adding/updating/obtaining initial reputations operations (*addInitRep/updateInitRep/getInitRep*) and validating contributor information operation (*validateUserInfo*) of class *InitRep*; computing/obtaining/updating user experience reputation operations (*computeUEP/getUEP/updateUEP*) of class *UserExpRep*; and adding/updating/obtaining reputation operations (*addReputation/updateReputation/getReputation*) of class *Reputation*. Object-related trust operations mainly include: computing/updating/obtaining trust degree operations (*computeTrust/updateTrust/getTrust*) of class *Trust*. Definitions and descriptions of these operations are shown in Table 1.

Table 1. Trust-related operations.

Trust-Related Operations	Class	Description
<i>add/updateInitRep(initRep)</i>	<i>InitRep</i>	Adding/updating an initial reputation. Parameter <i>initRep</i> refers to an <i>InitRep</i> object.
<i>getInitRep(c)</i>	<i>InitRep</i>	Querying a contributor's initial reputation. Parameter <i>c</i> is a contributor (user id is <i>c</i>).
<i>validateUserInfo(c, email/Phone/..)</i>	<i>InitRep</i>	Validating user (user id is <i>c</i>) information (e.g., email, phone number, etc.) and triggering the operation <i>InitRep.updateInitRep()</i> for updating user initial reputation.
<i>getUEP(c)</i>	<i>UserExpRep</i>	Querying a contributor's assessment reputation (user id is <i>c</i>).
<i>updateUEP(c)</i>	<i>UserExpRep</i>	Updating a contributor's assessment reputation (user id is <i>c</i>).
<i>addReputation(reputation)</i>	<i>Reputation</i>	Adding a user's reputation information. The <i>reputation</i> parameter represents a <i>REP</i> object.
<i>updateReputation(c, Value)</i>	<i>Reputation</i>	Updating a user's reputation value (user id is <i>c</i>) to <i>Value</i> . A <i>Value</i> parameter can be computed based on data obtained from <i>InitRep.getInitRep()</i> and <i>UserExpRep.getAssess()</i> operations.
<i>getReputation(c)</i>	<i>Reputation</i>	Querying a user's reputation value (user id is <i>c</i>).
<i>computeTrust(v_i)</i>	<i>Trust</i>	Computing the trust degree of an object version (<i>v_i</i>). The trust degree can be determined based on a contributor's reputation and the editing process involved.
<i>updateTrust(v_i, Value)</i>	<i>Trust</i>	Updating the trust degree of the object version (<i>v_i</i>) to <i>Value</i> . <i>Value</i> can be determined based on human-computer interactions or can be the computed result of the <i>Trust.computeTrust()</i> function.
<i>getTrust(v_i)</i>	<i>Trust</i>	Querying the trust degree of an object version (<i>v_i</i>).

5.2. Linkage Rules of Trust-Related Operations

In the spatio-temporal VGI model that considers trust-related information, relationships between object trustworthiness, user reputations and editing processes are complex. To maintain the consistency of trust-related data, it is necessary to design corresponding linkage rules. It is assumed that geographic object A has *n* versions (*n* is an integer and is greater than 0), which are referred to as *v₁, v₂, v₃, ..., v_n* based on their generated times, and the contributor to version *v_i* ($1 < i \leq n$) is *c_i*. Linkage operations of a contributor's reputation and the trustworthiness of a version are analysed as follows.

5.2.1. Linkage Operations of Contributor Reputations

A change in a contributor's reputation mainly results from the following four processes:

- (1) When a user is added to a VGI system and validates his/her registration information, an *IR* object and *REP* object should be added.

- (2) When a user's registration information changes, his/her initial reputation and user reputation should be updated.
- (3) When a user is assessed by another user, his/her assessment reputation and user reputation should also be updated.
- (4) If some user assessments are undone, his/her assessment reputation and user reputation should be updated.

These four cases are formally described by four linkage rules (Rules 1 to 4), which are shown in Figure 3.

Rule 1: IF *Contributor.addContributor(c)* THEN *InitRep.addInitRep(c)* AND *Reputation.addReputation(c)*. When contributor *c* enters the VGI system, an *REP* object and an *IR* object should be added for user *c*. This rule is shown in the left-hand section of Figure 3.

Rule 2: IF *Contributor.validateUserInfo(c, userinfo)* THEN *InitRep.updateInitRep(c)* AND *Reputation.updateReputation(c, value)*. If the registration information of contributor *c* is added or validated, then the user's reputation should be updated. This rule is shown in the right-hand section of Figure 3.

Rule 3: IF *Assess.addAssess(A, v_c, v_i/c₂)* THEN *UserExpRep.updateUEP(c)* AND *Reputation.updateReputation(c, value)*. Version *v_c* is a version of object *A* that is added by user *c*; when user *c* receives a single implicit (by *v_i*) or explicit assessment (by contributor *c₂*) from others, his/her assessment reputation and user reputation should be updated. This rule is shown in the right-hand section of Figure 3.

Rule 4: IF *Assess.deleteAssess(assess, c)* THEN *UserExpRep.updateUEP(c)* AND *Reputation.updateReputation(c, value)*. When an assessment (*assess*) of user *c* is undone, his/her assessment reputation and user reputation should be updated. This rule is shown in the right-hand section of Figure 3.

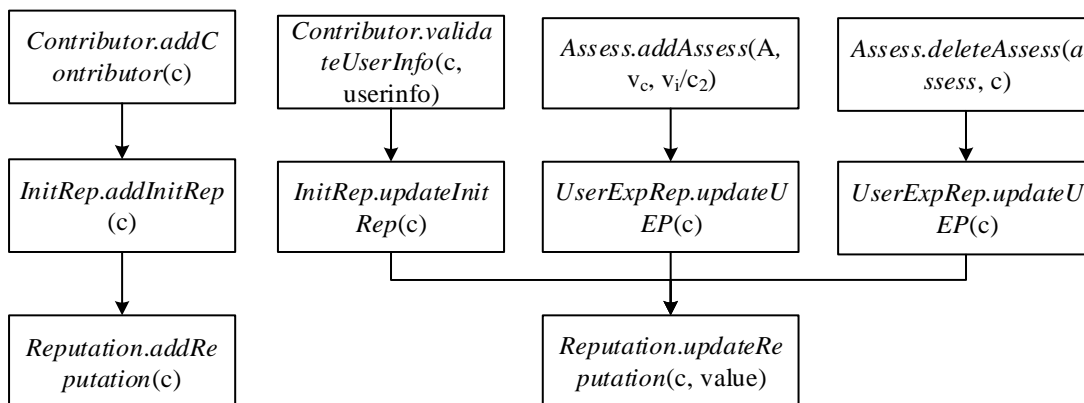


Figure 3. Linkage operations of a contributor's reputation.

5.2.2. Linkage Operations of Adding an Object Version

When version *v_i* of object *A* is added by a user, corresponding event objects (a geographic event or edit event or both) must be added first. If version *v_i* is the first version of object *A*, corresponding *GO* and *STA* objects should be added, and the trust degree of *v_i* should be updated. When a version number is larger than 1 and when the event is a non-geographic event, a *UER* object should be added, and the user's reputation and the trustworthiness of the object version should be updated; otherwise, the *STA* and *TRU* objects should be updated. Linkage operations of adding a version are outlined in Rules 5 to 9, which are shown in Figure 4.

Rule 5: IF *Version.addVersion(v_i)* AND *isGeoEvent = true* THEN *GeoEvent.addEvent()*, *EditEvent.addEvent()* ELSE *GeoEvent.addEvent()*. When user *c_i* adds a version *v_i* of any object *A* and

if the event type is a geographic event, *GE* and *EE* objects should be added; otherwise, a *GE* object should be added.

Rule 6: IF *Version.addVersion*(*A*, v_i) AND $i = 1$ THEN *GeoObject.addGeoObject*(*A*), *State.addState*(s_1), *Reputation.updateReputation*(c_j , value), *Trust.updateTrust*(v_1 , *Reputation.getReputation*(c_i)). If contributor c_i adds a new version v_1 (first version) of a geographic object *A*, linkage operations in the model involve adding a *GO* object, adding a *STA* object, updating the user reputations of all contributors c_j who are implicitly assessed by v_i , and setting the trust degree of the version.

Rule 7: IF *Version.addVersion*(*A*, v_i) AND $i > 1$ AND *isGeoEvent* = true THEN *State.addState*(*Version.getVersionState*(v_i) + 1), *Trust.updateTrust*(v_i , *Reputation.getReputation*(c_j)) ($0 < j < i$). When an object version is modified and the event type involved is a geographic event, a new *STA* object should be added.

Rule 8: IF *Version.addVersion*(*A*, v_i) AND $i > 1$ AND *isGeoEvent* = false THEN *Assess.addAssess*(*A*, v_k , v_i) ($k < i$ AND $s_{v_k} = s_{v_i}$). When an object version is modified and the event involved is a non-geographic event, version v_i should assess all previous versions v_k of the same object state.

Rule 9: IF *UserExpRep.updateUEP*(c) THEN *Reputation.updateReputation*(c) AND *Trust.updateTrust*(v , value). When the assessment reputation of user c is updated, his/her reputation should be updated. The trust degree of all final versions added by user c_i should also be updated.

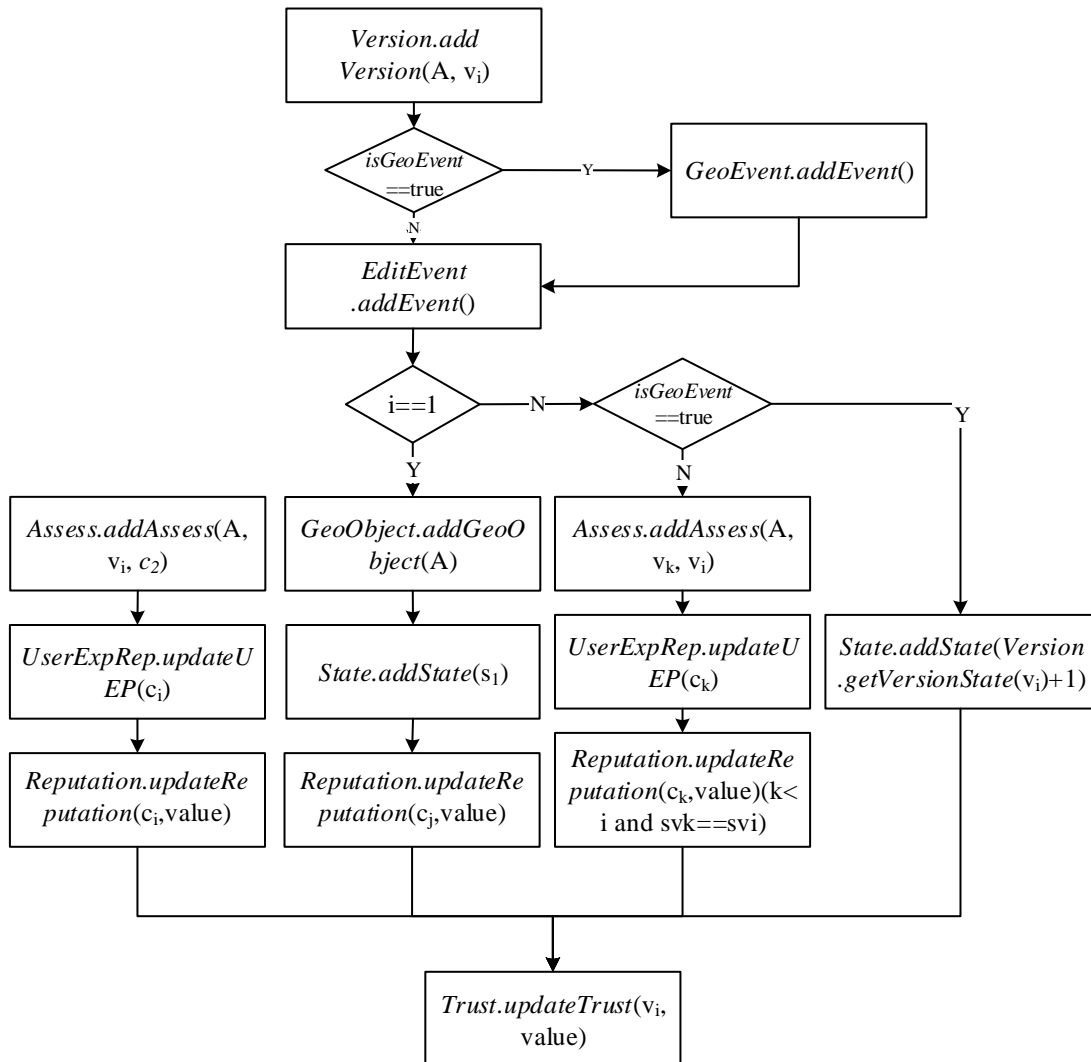


Figure 4. Linkage operations of adding an object version.

6. Experiment and Analysis

From the models and rules described above, we designed and implemented a prototype system to validate the feasibility and effectiveness of our model. The prototype system was developed using Visual studio 2010 and PostgreSQL 9.4. Spatial data were managed using ArcGIS Engine 10.

As collecting large quantities of crowdsourcing data is time-consuming (can take several years), a historical dump dataset drawn from OpenStreetMap (OSM, data are downloadable) was used in our experiments [40]. The dataset includes the editing history of all geographic objects. In our experiments, historical data for Berlin and Pakistan were used. Among them, historical data for Berlin were collected from 28 January 2006 to 15 February 2013. The data include 4,238,401 points, 554,872 lines, and 469,080 polygons, which were added by 4842 volunteers. Most of the volunteers are non-professionals. Historical data for Pakistan were collected from 26 May 2007 to 12 December 2014. The data include 2,005,806 points, 195,765 lines, and 34,326 polygons that were added by 790 volunteers. As the OSM data were in XML format, we first imported the data into our database by programming. By contrast, the data for Berlin are more complex than that for Pakistan, particularly with regards to polygon data. As other researchers have found, developed countries and regions typically have more contributors than developing countries and regions [47]. A city in Europe (Berlin) has more than six times more volunteers than a developing country (Pakistan). For the data on Pakistan, line data are apparently more plentiful than polygon data, suggesting that volunteers in Pakistan tend to contribute basic line features, such as roads and rivers, rather than contributing detailed city features, such as buildings. The data for Berlin have also been modified more than that for Pakistan. Moreover, Berlin includes 431,724 objects, Pakistan includes 118,313 objects (including polygons and polylines), and objects with only one version account for 65.80% and 50.99% of Pakistan's and Berlin's data, respectively. In particular, polygon objects account for 83.88% of all data for Pakistan. Only roughly 16.12% of all Pakistan polygon objects have ever been modified.

In our experiments, a user's reputation was the aggregation of his/her initial reputation and assessment reputation. If the reputation of user c is $R(c)$, the following equation is used.

$$R(c) = R_0(c) \cdot e^{-m/M} + R_e(c) \cdot (1 - e^{-m/M}) \quad (1)$$

where $R_0(c)$ is the initial reputation of user c , which is computed by the completeness of the user's registration information and by his/her authenticity of identification. As registration information on OSM contributors is difficult to acquire, it is not considered in our experiments. e is Euler's constant. m denotes the time of assessment. M is a constant of a natural number that regulates the weight of $R_0(c)$ and $R_e(c)$. $R_e(c)$ is the assessment reputation computed from the weighted average of all or part of the contributor's assessments. An assessment is divided into two categories, *i.e.*, explicit assessment and implicit assessment. Among them, an explicit assessment denotes the explicit value of each assessment and is easy to determine. An implicit assessment is difficult to determine as there is no explicit value for each assessment. In our experiments, it is computed using Equation (2).

$$E(c_i, c_j) = VS(v_i, v_j) \cdot R(c_j)^c \quad (2)$$

where $R(c_j)$ is the reputation of user c_j when he/she assesses the contribution v_i of user c_i . $c \in [0,1]$ is an exponent invariant that is used to determine that degree to which an evaluator's reputation affects the trustworthiness of his/her assessment. $VS(v_i, v_j)$ denotes version similarity between versions v_i and v_j , and it represents the assessed value of versions v_j to v_i . Higher levels of similarity between the two versions correspond to a higher assessment value and vice versa. Version similarity levels can be computed from geometric and attribute similarities between different versions.

If the trustworthiness of v_i of an object is $T(v_i)$ and if version number i takes a value of 1, the trust degree of the object version is the contributor's reputation. Otherwise, if i is larger than 1, the trust degree of object version v_i is determined based on the trust degree of its previous version v_k (k is less

than i), contributor reputation of v_i and modification proportions between versions v_i and v_k . The trust degree of version v_i of an object is computed using Equation (3).

$$T(v_i) = \begin{cases} R(c_i) & i = 1 \\ EP(v_i, v_k) \cdot R(c_i) + (1 - EP(v_i, v_k)) \cdot \text{Max}(T(v_k), R(c_i)) & i > 1 \end{cases} \quad (3)$$

where $R(c_i)$ denotes the reputation of user c_i who created version v_i . $T(v_k)$ represents the trust degree of version v_k . $\text{Max}(T(v_k), R(c_i))$ is the maximum of $T(v_k)$ and $R(c_i)$. $EP(v_i, v_k)$ denotes the degree of modification between v_i and v_k , where $0 \leq EP(v_i, v_k) \leq 1$, $1 - EP(v_i, v_k)$ can be denoted as an unedited component. $EP(v_i, v_k)$ is inversely proportional to version similarity. Lower proportions of modification correspond to higher degrees of version similarity and vice versa.

The data for Berlin and Pakistan were organized based on our spatio-temporal VGI model. In the historical data from the OSM, there is no record of why a contributor has modified an object or of the state VGI object change. Therefore, whether the state of an object has been changed or not is unknown. In our model, if a version and its former version are of the same state, the model will conduct an implicit assessment of the former version; otherwise, the implicit assessment will be ignored. Generally speaking, a narrower time interval between two versions denotes a higher probability that the state of an object has not changed (a modification corresponding to a non-geographic event); otherwise, the state of the object has changed (a modification corresponding to a geographic event). Therefore, in our experiments, when the time interval between two versions was larger than a threshold, the latter version was ascribed to a new state; otherwise, both versions were allocated to the same state. As OSM data also include no explicit assessments, assessments made between users were computed through implicit assessment. Such values were determined based on similarities between versions (Equation (2)). From the generated time of all versions in the OSM, the reputations of all participating contributors and the trust degrees of all object versions were computed. For example, each version of a geographic object (object ID is 73533774) is shown in Figure 5. The object is a building and has three versions. Let us analyse how to determine the trustworthiness of this object and its contributors' reputations during its editing process. First, version v_1 of the object was added by a contributor named "ceyockey", whose reputation took a value of 0.828 at the time. The trust degree of version v_1 is its contributor's reputation (a value of 0.828). Then, version v_2 was added by another user named "MMDawood", whose reputation took a value of 0.856 at the time. The degree of version similarity between v_2 and v_1 was 0.843. An implicit assessment of v_2 to v_1 was conducted using Equation (2) (setting $c = 1$), and a value of 0.820 was obtained. The reputation of user "ceyockey" was updated based on his/her evaluations, and his/her reputation suffered slightly. The trust value of v_2 was found to be 0.856 as computed by Equation (3). Finally, version v_3 was added by user "landfahrer", whose reputation took a value of 0.8735 at the time. Implicit assessments for v_3 to v_2 and for v_3 to v_1 were determined using Equation (2), and values of 0.884 and 0.821, respectively, were found. The reputations of users "ceyockey" and "MMDawood" were updated based on their evaluations. The reputation of "MMDawood" improved after he/she provided a better evaluation. Finally, a trust value of 0.874 for v_3 was computed using Equation (3).

After object trustworthiness and contributor reputations were calculated, information on each object and volunteer was queried and viewed. As shown in Figure 5, trust-related information on each version of object 73,533,774 was retrieved. Geometric patterns of each version, which overlap in a satellite image, can also be seen on the right-hand side of the figure. Statistical information on the contributors of any version can be retrieved, and all contributor information (e.g., name, reputation and number of contributions) for the object is shown in the lower right-hand section of the figure. As shown on the left-hand side of the figure, all states and versions of the object can be freely selected for viewing in the right-hand window. Moreover, we also can determine the best version of a state using a human-computer interface.

The experimental results show that the trustworthiness of an object and contributor reputations can be effectively stored and managed using our VGI model. Unlike the traditional spatio-temporal data model, our model presents and stores trust-related information (e.g., contributors, initial contributor reputations, contributor assessments, and contributor reputations). Among these features, contributor assessments are one of the main factors that require access to additional funds. In our model, all fields of *UER* objects (expressing contributor assessments, including primary and foreign keys) are numerical and require only 40 bytes to store records. That is, 1,000,000 records should occupy approximately 38.15 MB of storage space. For historical OSM data for Berlin, 489,386 records (polygon versions) occupy 196.22 MB in the database. That is, 1,000,000 records should occupy approximately 400.95 MB of storage space. If the ratio between the quantity of object and evaluation records is 1:1, the data ratio between records of assessment and polygon objects will be approximately 9.51%. Given that other information is included in the data model, the incremental data (trust-related information) should be smaller than it. Therefore, when using our model, a larger pool of data is controllable. The spatial data in our model are stored in reference to OSM data to ensure the usability and utility of our model. Regarding the speed of our prototype system, we conducted experiments on a personal computer (Intel Core i5 3.0 Hz, 8 G, Windows 7 professional 64 bit). As trust-related computation involves geometric calculations, polygon or polyline objects with more versions and nodes serve as better experimental objects. Therefore, a geographic object drawn from the OSM (ID: 4317997) was used for the test. The object has 11 versions, and each version has 87 nodes on average. In our experiments, 27 rounds of trust-related calculations and database operations were conducted (each version evaluated the previous three versions (if applicable) in the experiments). Each of the 27 rounds took 1.073 s on average. On average, each round of trust-related calculations and database operations will take 0.0397 s. This processing speed is acceptable for personal computers and is worth improving upon further.

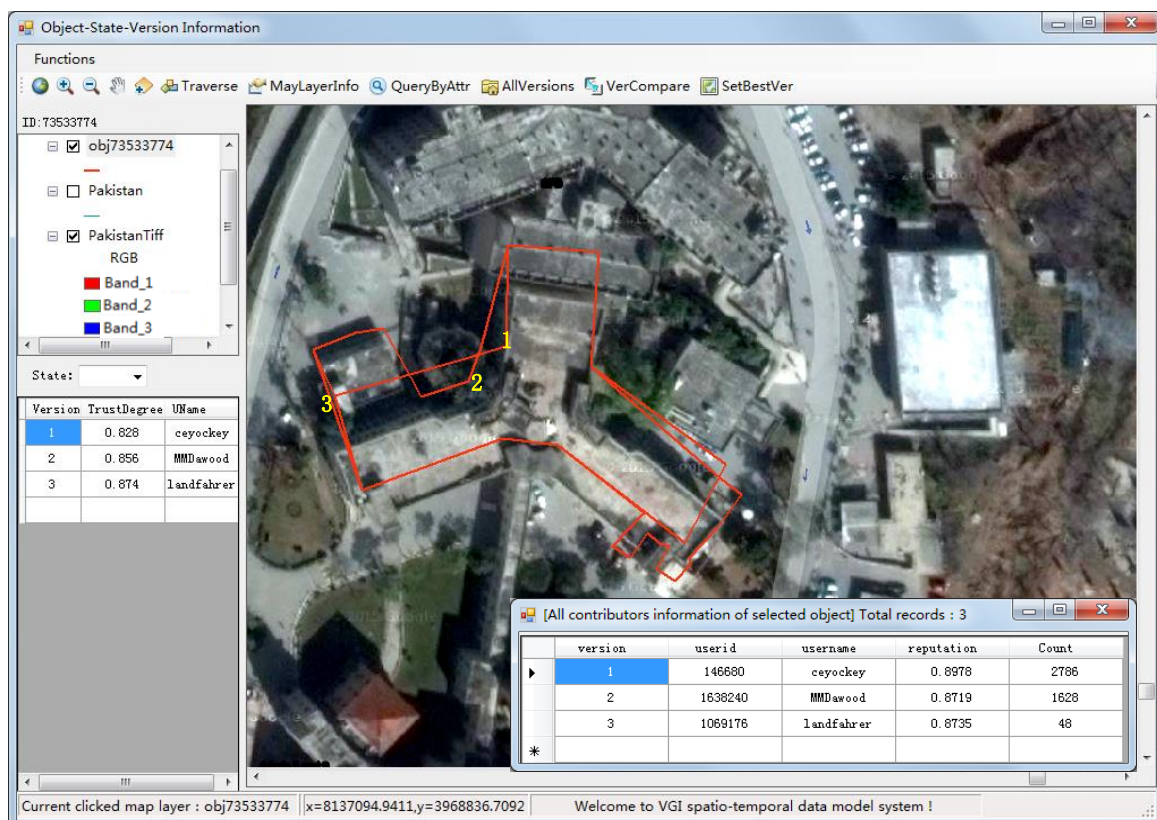


Figure 5. A VGI geographic object and corresponding contributor information.

7. Conclusion and Discussion

VGI is becoming increasingly important as an alternative source of spatial data [4]. However, uncertainties in terms of its quality have limited its use in GIS-related applications [12]. Many scholars agree that contributor reputations significantly affect the quality of VGI [13,61]. However, few studies have attempted to develop a spatio-temporal VGI model considering contributor reputations and object trustworthiness. Meanwhile, in VGI environments, the state of an object may have one or more versions. Thus, ways to quickly select the best versions of states must be formulated. We present a VGI model based on trust-related information that effectively stores and manages corresponding information. In this model, a geographic object is represented by an “Object-State-Version” three-layer mechanism; each state is associated with its best version for the rapid querying of corresponding snapshots. In turn, issues related to the existence of numerous VGI versions can be more effectively managed to some extent. Trust-related VGI data are formed from contribution processes of involved objects; contributor reputation is composed of initial reputation and assessment reputation, and assessments among users may include both explicit assessments and implicit assessments or only one of these. In turn, the reputations of all contributors can be computed. Trust-related operations and their relationships in the model were also analysed, and nine linkage rules among trust-related operations were found to maintain the consistency of trust and reputation data. Finally, a prototype experimental system was implemented to validate the effectiveness of the model presented in this paper.

It is necessary to state that according to the mechanisms of our model, some basic requirements must be met before it can be used in a real VGI system.

- (1) Information on all versions of feature states that volunteers have added to the VGI model must be recorded. Information on a version should include contributor, geometry feature, attributes, generation time, *etc.* Information can be stored explicitly or implicitly.
- (2) Systems should store registration information on volunteers. This information is key for computing a contributor’s initial reputation. When it has not been recorded, initial reputations cannot be calculated, and the accuracy of user reputations is affected (especially when contributors have made few assessments).

In this paper, we focus on the modelling of trust-related information in VGI environments. The collection of such information, e.g., initial reputation and explicit rating information, is essential to the design and implementation of VGI systems using the model presented. In our opinion, multi-perspective information can be used to synthetically evaluate the quality of crowdsourcing objects. Furthermore, such information can be validated against each other. Therefore, the quality of VGI objects can be reasonably controlled by storing and validating registration information, by providing mechanisms that allow individuals to leave ratings of different contributors (or VGI objects), and by allowing contributors to upload quality metadata information.

The different ways of computing trust degrees of VGI objects are still open for further examination. In our experiment, user reputations were defined as the aggregation of initial and assessment reputations. Initial reputations are computed from registration information, and assessment reputations are computed based on similarities between versions of the same state and of the same entity. VGI object trustworthiness is determined based on contributor reputations and proportions for the corresponding contributor. We found that VGI object quality is positively correlated with trust degrees of VGI objects. However, other factors, e.g., remaining version time (*i.e.*, if an object remains the same in an area that has experienced several updates, it must be of high quality and its contributor can achieve a higher level of trustworthiness), topological consistency (*i.e.*, a VGI object without topological conflicts is typically more trustworthy and vice versa), *etc.*, affect the computation of VGI object trustworthiness. Such issues lie outside the scope of the present study.

In conclusion, though we presented a spatio-temporal VGI model considering contributor reputations and object trustworthiness in this paper, it does not mean that keeping the last feature

created by a more trustworthy contributor is always better. The computation of VGI object trustworthiness is still an open issue. Therefore, effective means of computing object trustworthiness and user reputations will be examined in future studies.

Acknowledgments: This study was funded by the National Natural Science Foundation of China (Project No. 41371366, 41501420) and the National Key Technology R&D Program of China (Project No. 2012BAK12B01-2). All of the authors gratefully thank the reviewers and editors.

Author Contributions: Yijiang Zhao and Xiangguang Zhou designed the framework of this paper and drafted and finalized the manuscript. Guangqiang Li and Hanfa Xing participated in discussions on the paper and contributed to the experiments.

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

References

1. Goodchild, M.F. Citizens as sensors: the world of volunteered geography. *GeoJournal* **2007**, *69*, 211–221. [[CrossRef](#)]
2. Elwood, S. Volunteered geographic information: Key questions, concepts and methods to guide emerging research and practice. *GeoJournal* **2008**, *72*, 133–135. [[CrossRef](#)]
3. Coleman, D.J.; Georgiadou, Y.; Labonte, J. Volunteered geographic information: The nature and motivation of producers. *Int. J. Spat. Data Infrastruct. Res.* **2009**, *4*, 332–358.
4. Chen, J.; Li, R.; Dong, W.; Ge, Y.; Liao, H.; Cheng, Y. GIS-based borderlands modeling and understanding: A perspective. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 661–676. [[CrossRef](#)]
5. Mooney, P.; Corcoran, P. Can Volunteered geographic information be a participant in eEnvironment and SDI? In *Environmental Software Systems. Frameworks of eEnvironment*; Springer: Berlin, Germany; Heidelberg, Germany, 2011; pp. 115–122.
6. Zook, M.; Graham, M.; Shelton, T.; Gorman, S. Volunteered geographic information and crowdsourcing disaster relief: A case study of the Haitian earthquake. *World Med. Health Policy* **2010**, *2*, 7–33. [[CrossRef](#)]
7. McDougall, K.; Temple-Watts, P. The use of LiDAR and volunteered geographic information to map flood extents and inundation. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *1*, 251–256. [[CrossRef](#)]
8. Li, Q.; Fan, H.; Luan, X.; Yang, B.; Liu, L. Polygon-based approach for extracting multilane roads from OpenStreetMap urban road networks. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 2200–2219. [[CrossRef](#)]
9. Bakillah, M.; Liang, S.; Mobasher, A.; Jokar Arsanjani, J.; Zipf, A. Fine-resolution population mapping using OpenStreetMap points-of-interest. *Int. J. Geogr. Inf. Sci.* **2014**, 1–24. [[CrossRef](#)]
10. Zhou, X.; Zeng, L.; Jiang, Y.; Zhou, K.; Zhao, Y. Dynamically integrating OSM data into a borderland database. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 1707–1728. [[CrossRef](#)]
11. Haklay, M. How good is volunteered geographical information? A comparative study of OpenStreetMap and ordnance survey datasets. *Environ. Plan. B Plan. Des.* **2010**, *37*, 682–703. [[CrossRef](#)]
12. Forghani, M.; Delavar, M.R. A quality study of the OpenStreetMap dataset for Tehran. *ISPRS Int. J. Geo-Inf.* **2014**, *3*, 750–763. [[CrossRef](#)]
13. Keßler, C.; de Groot, R.T.A. Trust as a proxy measure for the quality of volunteered geographic information in the case of OpenStreetMap. In *Geographic Information Science at the Heart of Europe*; Springer: Berlin, Germany; Heidelberg, Germany, 2013; pp. 21–37.
14. Bishr, M.; Mantelas, L. A trust and reputation model for filtering and classifying knowledge about urban growth. *GeoJournal* **2008**, *72*, 229–237. [[CrossRef](#)]
15. Roche, S.; Propeck-Zimmermann, E.; Mericskay, B. GeoWeb and crisis management: Issues and perspectives of volunteered geographic information. *GeoJournal* **2013**, *78*, 21–40. [[CrossRef](#)]
16. Kumar, C.; Heuten, W.; Boll, S. Visual overlay on OpenStreetMap data to support spatial exploration of urban environments. *ISPRS Int. J. Geo-Inf.* **2015**, *4*, 87–104. [[CrossRef](#)]
17. Neis, P.; Goetz, M.; Zipf, A. Towards automatic vandalism detection in OpenStreetMap. *ISPRS Int. J. Geo-Inf.* **2012**, *1*, 315–332. [[CrossRef](#)]
18. Fairbairn, D.; Al-Bakri, M. Using geometric properties to evaluate possible integration of authoritative and volunteered geographic information. *ISPRS Int. J. Geo-Inf.* **2013**, *2*, 349–370. [[CrossRef](#)]

19. Jackson, S.; Mullen, W.; Agouris, P.; Crooks, A.; Croitoru, A.; Stefanidis, A. Assessing completeness and spatial error of features in volunteered geographic information. *ISPRS Int. J. Geo Inf.* **2013**, *2*, 507–530. [[CrossRef](#)]
20. Mummidi, L.N.; Krumm, J. Discovering points of interest from users' map annotations. *GeoJournal* **2008**, *72*, 215–227. [[CrossRef](#)]
21. Qian, X.; di, L.; Li, D.; Li, P.; Shi, L.; Cai, L. Data cleaning approaches in Web2.0 VGI application. In Proceedings of the 17th IEEE International Conference on Geoinformatics, Fairfax, VA, USA, 12–14 August 2009; pp. 1–4.
22. Rehl, K.; Gröechenig, S.; Hochmair, H.; Leitinger, S.; Steinmann, R.; Wagner, A. A conceptual model for analyzing contribution patterns in the context of VGI. In *Progress in Location-Based Services*; Springer: Berlin, Germany; Heidelberg, Germany, 2013; pp. 373–388.
23. Langran, G.; Chrisman, N.R. A framework for temporal geographic information. *Cartogr. Int. J. Geogr. Inf. Geovis.* **1988**, *25*, 1–14. [[CrossRef](#)]
24. Langran, G. *Time in Geographic Information Systems, Technical Issues in Geographic Information Systems*; Taylor & Francis: Abington, MA, USA, 1992.
25. Peuquet, D.J.; Duan, N. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *Int. J. Geogr. Inf. Syst.* **1995**, *9*, 7–24. [[CrossRef](#)]
26. Worboys, M. Event-oriented approaches to geographic phenomena. *Int. J. Geogr. Inf. Sci.* **2005**, *19*, 1–28. [[CrossRef](#)]
27. Claramunt, C.; Thériault, M. Managing time in GIS an event-oriented approach. In *Recent Advances in Temporal Databases*; Springer: London, UK, 1995; pp. 23–42.
28. Chen, J.; Jiang, J. An event-based approach to spatio-temporal data modeling in land subdivision systems. *GeoInformatica* **2000**, *4*, 387–402. [[CrossRef](#)]
29. Worboys, M.F. Object-oriented approaches to geo-referenced information. *Int. J. Geogr. Inf. Syst.* **1994**, *8*, 385–399. [[CrossRef](#)]
30. Raza, A. Object-Oriented Temporal GIS for Urban Applications. Ph.D. Thesis, University of Twente, Enschede, The Netherlands, 2001.
31. Yuan, M. Use of a Three-Domain representation to enhance GIS support for complex spatiotemporal queries. *Trans. GIS* **1999**, *3*, 137–159. [[CrossRef](#)]
32. Renolen, A. History graphs: Conceptual modelling of spatiotemporal data. In Proceedings of the GIS Frontiers in Business and Science, Brno, Czech Republic, 20–24 April 1996.
33. Tryfona, N. Modeling phenomena in spatiotemporal databases: Desiderata and solutions. In *Database and Expert Systems Applications*; Springer: Berlin, Germany; Heidelberg, Germany, 1998; pp. 155–165.
34. Güting, R.H.; Böhlen, M.H.; Erwig, M.; Jensen, C.S.; Lorentzos, N.A.; Schneider, M.; Vazirgiannis, M. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.* **2000**, *25*, 1–42. [[CrossRef](#)]
35. Hornsby, K.; Egenhofer, M.J. Modeling moving objects over multiple granularities. *Ann. Math. Artif. Intell.* **2002**, *36*, 177–194. [[CrossRef](#)]
36. Zielstra, D.; Zipf, A. A comparative study of proprietary geodata and volunteered geographic information for Germany. In Proceedings of the 13th AGILE International Conference on Geographic Information Science, Guimarães, Portugal, 11–14 May 2010.
37. Neis, P.; Zielstra, D.; Zipf, A. The street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011. *Future Internet* **2011**, *4*, 1–21. [[CrossRef](#)]
38. Fan, H.; Zipf, A.; Fu, Q.; Neis, P. Quality assessment for building footprints data on OpenStreetMap. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 700–719. [[CrossRef](#)]
39. Girres, J.F.; Touya, G. Quality assessment of the French OpenStreetMap dataset. *Trans. GIS* **2010**, *14*, 435–459. [[CrossRef](#)]
40. OpenStreetMap, Full History Dump. Available online: <http://planet.openstreetmap.org/planet/full-history/> (assessed on 1 July 2015).
41. Barron, C.; Neis, P.; Zipf, A. A comprehensive framework for intrinsic OpenStreetMap quality analysis. *Trans. GIS* **2013**, *18*, 877–895. [[CrossRef](#)]

42. Bégin, D.; Devillers, R.; Roche, S. Assessing volunteered geographic information (VGI) quality based on contributors, mapping behaviours. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *XL-2/W1*, 149–154. [[CrossRef](#)]
43. D'Antonio, F.; Fogliaroni, P.; Kauppinen, T. VGI Edit History Reveals Data Trustworthiness and User Reputation. 2014. Available online: http://www.didattica.univaq.it/moodle/pluginfile.php/142568/mod_resource/content/2/VGI%20Trust%20Reputation%20DAntonio-Fogliaroni.pdf (accessed on 29 January 2016).
44. Qian, X.; Zhu, X. Multi-representation geographic data organization method dedicated for vector-based WebGIS. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 815–818.
45. Wikipedia, Google Map Maker. Available online: https://en.wikipedia.org/wiki/Google_Map_Maker (assessed on 2 November 2015).
46. Wikipedia, WikiMapia. Available online: <https://en.wikipedia.org/wiki/WikiMapia> (assessed on 2 November 2015).
47. Goodchild, M.F.; Li, L. Assuring the quality of volunteered geographic information. *Spat. Stat.* **2012**, *1*, 110–120. [[CrossRef](#)]
48. Galton, A. *Qualitative Spatial Change*; Oxford University Press: Oxford, UK, 2000.
49. Wachowicz, M. *Object-Oriented Design for Temporal GIS*; Taylor & Francis: London, UK, 2003.
50. Cambridge Dictionaries. Version. Available online: <http://dictionary.cambridge.org/dictionary/english/version> (assessed on 28 December 2015).
51. Medeiros, C.B.; Jomier, G. Using versions in GIS. In *Database and Expert Systems Applications*; Springer: Berlin, Germany; Heidelberg, Germany, 1994; pp. 465–474.
52. Loomis, M.E. Object versioning. *J. Object-Oriented Program.* **1992**, *4*, 40–43.
53. Zhou, X.; Chen, J.; Jiang, J.; Zhu, J.; Li, Z. Event-based incremental updating of spatio-temporal database. *J. Cent. South Univ. Technol.* **2004**, *11*, 192–198. [[CrossRef](#)]
54. Mooney, P.; Corcoran, P. Characteristics of heavily edited objects in OpenStreetMap. *Future Internet* **2012**, *4*, 285–305. [[CrossRef](#)]
55. Sztompka, P. *Trust: A Sociological Theory*; Cambridge University Press: Cambridge, UK, 1999.
56. Chang, E.; Hussain, F.; Dillon, T. *Trust and Reputation for Service-Oriented Environments: Technologies for Building Business Intelligence and Consumer Confidence*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
57. Bishr, M.; Janowicz, K. Can we trust information?-the case of volunteered geographic information. In Proceedings of the Towards Digital Earth Search Discover and Share Geospatial Data Workshop at Future Internet Symposium, Berlin, Germany, 20 September 2010.
58. Kefler, C.; Trame, J.; Kauppinen, T. Tracking editing processes in volunteered geographic information: The case of OpenStreetMap. In Proceedings of the Workshop on Identifying Objects, Processes and Events in Spatio-Temporally Distributed Data, Belfast, ME, USA, 12–16 September 2011.
59. Van Exel, M.; Dias, E.; Fruijtjer, S. The impact of crowdsourcing on spatial data quality indicators. In Proceedings of the 6th GIScience International Conference on Geographic Information Science, Zurich, Switzerland, 14–17 September 2010.
60. Malik, Z.; Bouguettaya, A. Reputation bootstrapping for trust establishment among web services. *IEEE Internet Comput.* **2009**, *13*, 40–47. [[CrossRef](#)]
61. Grira, J.; Bédard, Y.; Roche, S. Spatial data uncertainty in the VGI world: Going from consumer to producer. *Geomatica* **2010**, *64*, 61–72.

