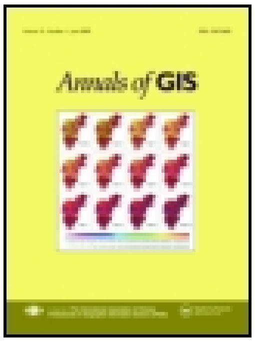

Annals of GIS

ISSN: 1947-5683 (Print) 1947-5691 (Online) Journal homepage: https://www.tandfonline.com/loi/tagi20

Next generation of GIS: must be easy

A-Xing Zhu, Fang-He Zhao, Peng Liang & Cheng-Zhi Qin

To cite this article: A-Xing Zhu, Fang-He Zhao, Peng Liang & Cheng-Zhi Qin (2020): Next generation of GIS: must be easy, Annals of GIS, DOI: 10.1080/19475683.2020.1766563

To link to this article: https://doi.org/10.1080/19475683.2020.1766563

Published online: 18 May 2020.

ANNALS OF GIS
https://doi.org/10.1080/19475683.2020.1766563

Next generation of GIS: must be easy

A-Xing Zhu[a,b,c,d], Fang-He Zhao[b,e], Peng Liang[b,e] and Cheng-Zhi Qin[b,e]

[a]Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, School of Geography, Nanjing Normal University, Nanjing, China; [b]State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, China; [c]Department of Geography, University of Wisconsin-Madison, Madison, WI, USA; [d]Center for Social Sciences, Southern University of Science and Technology, Shenzhen, China; [e]College of Resources and Environment, University of Chinese Academy of Sciences, Beijing, China

ABSTRACT
Existing GIS software mainly target at expert users and do not sufficiently integrate resources for efficient computing. They are difficult for non-experts to use and are often slow in completing the complicated geographic analysis. To address these problems, future generation of GIS software must be 'easy'. By 'easy' we mean 'easy to use' and 'easy to compute'. 'Easy to use' means that software system should be goal-oriented, rather than the currently procedure-oriented doctrine. The goal-oriented will relieve users, particularly novice users, the burden of knowing the exact commands and their sequences to perform for achieving the goal they want to achieve. 'Easy to compute' means that implementation of GIS analytical functionality should be able to utilize the high-performance computing infrastructures for complicated geographic analysis. Two case studies, one in digital soil mapping and the other in digital terrain analysis, are presented to illustrate the meaning of 'easy'. We believe that the future generations of GIS platforms should be goal-driven, intelligent, high-performance computing enabled, easily accessible, and participatory. It allows anyone to participate in geo-computation at anywhere and anytime.

ARTICLE HISTORY
Received 9 December 2019
Accepted 4 May 2020

KEYWORDS
Next generation of GIS; geo-computing platform; must be easy; high-performance computing; intelligent digital soil mapping; efficient digital terrain analysis

1. Introduction

Geospatial analysis has become an important analytical tool for a variety of fields which involve the integrated analysis of spatially distributed geographic information. This integrated analysis (geo-computing) of geographic information is often accomplished through the use of GIS software which is designed to store, retrieve, analyse, and visualize geographic information (Maguire 1991). GIS software can be thought of two general types: (a) general-purpose geo-computation platforms such as ArcGIS (ESRI 2019), Q-GIS (QGIS Development Team 2019), and SuperMap (SuperMap Software Co. 2019); and (b) specialized geo-computation tools, including Landserf (Wood 2009), TauDEM (Tarboton 2005), and SoLIM (Zhu et al. 2018). These forms of GIS software have simplified the management and analysis of geospatial data to some extent, yet users need to be trained on these software sufficiently before they can adequately use these planforms/tools to accomplish the analytical tasks the users need to perform. Sufficiency in these software does not come easily and more than often users need to take specialized courses or even complete a degree to achieve this sufficiency. Therefore, the current forms of GIS software are not easy to use, particularly for non-specialists (novice users). These difficulties can be shown in the following three major aspects: knowledge of operations, efficiency of computation, and software management.

The first challenge is the knowledge required to conduct the management and analysis of geographic information (geospatial data) using these GIS software/platforms. Before spatial analysis can be carried out, the relevant geospatial data must be brought together in compliance with the requirements of the project and stored in a particular GIS software. Geospatial data are highly heterogeneous, that is they exist in various forms and/or formats (Di 2004; Wei, Santhana-Vannan, and Cook 2009). Some volunteered data does not even have schema (Elwood, Goodchild, and Sui 2012). The data need to be 'brought' into the same coordinate systems and into the same format required by the specific software in use. In addition, a specific GIS software has a particular way of storing the geospatial data. Thus, knowledge about converting the exogenous data into system native formats is required for bringing the heterogeneous geospatial data into the particular form required by the GIS software. Users must possess this knowledge before any spatial analysis can be conducted.

CONTACT Fang-He Zhao zhaofh@lreis.ac.cn State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing, China

When coming to performing specific spatial analysis in GIS, a user not only needs to know what he/she wants to achieve but also needs to know exactly how to achieve it because existing GIS software is procedure-oriented. This means that the user needs to know the steps for completing the task, the specific commands for each step, and the parameters for each command. For example, if a user wants to delineate the watershed from gridded DEM, he/she needs to know the complete work-flow (the steps) such as pit filling, flow direction determination, flow accumulation calculation, stream network extraction, and watershed delineation. For each of these steps, the user also needs to know which command to use and the parameters to specify for each command. For example, when coming to pit filling, should the incremental or decremental method be used? For the flow direction calculation, should the D8 algorithm or the multiple flow direction (MFD) algorithm, or even the D-Infinity (D-INF) method be used? Clearly, users who wish to extract watersheds need to know all of the above to complete the watershed extraction for their projects.

The second challenge is the efficiency of computation. On one hand, the spatial extent (scope) for geospatial analyses are becoming larger and larger. It is not unusual to see a geographic analysis to be performed over a regional to continental spatial extent. At the same time, researches have been conducted at much finer spatial details in order to better understand the geographic phenomenon for better decision support. Spatial analyses conducted at higher spatial resolution and over larger spatial extent will certainly make geographic analysis data-intensive and computation-intensive. On the other hand, the computation process is also getting more and more complicated. As our understanding of geographic phenomena grows deeper, geographic algorithms or models are becoming more and more complex (Neitsch et al. 2011; Zhu et al. 2019). The complexity of models or algorithms adds to the computation intensity which already overwhelms the computation power of the existing GIS software platforms due to the increase over spatial extent at a fine spatial detail requested by current geographic analyses. As a result, many of geographic analyses become difficult and even unachievable for existing GIS software to complete.

The third challenge in using these types of software is brought forward by software management. These GIS software systems usually require extensive and complicated setup which includes installation and configuration of the software (Qin et al. 2013; Jiang et al. 2016). For example, the installation of the ArcGIS Desktop is a rather complex process, including checking the prerequisite software, separate installation of the ArcGIS Desktop and licence manager, selection between different installation types, and configuration of the licence. In addition, maintaining the installed software (such as updates, upgrades and licence continuation) can be annoying chores for users of GIS. The process of managing a GIS software can be not only very time-consuming but also frustrating to users who might just need a few functions for what they need to do. Thus, specialists in managing these software are often needed. However, the provision of the specialized personnel is only possible for major institutes and large enterprises. Otherwise, users have to develop these skills before they can carry out the GIS analysis they need. Luckily, the arduous installation process has been alleviated by some other GIS software/platforms. Some portable software require little efforts of installation. But the utilities of these software are usually not as substantial as ArcGIS, and the maintenance of the software will always be necessary. Overall, management of the existing GIS software is an added cost or overburden for GIS application users.

In addition to the difficulties for software users, these systems are also inefficient and even difficult for researchers to share their innovations in algorithm and method development. The common practice for researchers to share inventions through existing GIS software is to integrate their innovations as add-ons to these software which are often controlled by vendors (Sorokine 2007; Boroushaki and Malczewski 2008; Thieler et al. 2009; Miller et al. 2007). This practice often requires the innovators not only to understand the data structure and software engineering of the target software but also to obtain permission or acceptance from the companies or organizations to add to the software. With all these efforts required to share an invention, it is only possible for a very limited number of researchers who have the time and the technical sufficiency. Often times, new innovations are made available only when the software companies or some organizations decide to include them in their new releases of the software. Obviously, this form of sharing is not only inefficient but also prohibitive, which makes GIS software not as updated in frontier algorithms or methods as they should.

Clearly, some of the challenges stated above can be mitigated through learning about GIS software and their uses albeit a steep learning curve. However, some of the difficulties are rooted in the software and cannot be addressed through users' efforts. Furthermore, there is no reason for users to spend a great deal of time and energy to learn a piece of software and specific work-flows of analysis which will often change with a new version of software. This is an inefficient use of users' time and resources. In addition, it is also a waste of time for users to spend hours and even days waiting for the completion of the computation. Users should be able to participate in geo-computation without extensive knowledge and to complete the computation with high efficiency.

Obviously, these GIS software have posed severe digital divides between user needs and software features. The digital divides can generally be grouped into two categories of difficulties: user-divide and computation-divide. The user-divide is the contradiction between users of GIS and target users of GIS software. Current GIS software are mainly oriented for geo-computing experts. They are difficult for non-expert users to use, and even to the level of preventing non-expert users from participating in the geo-computation. But these non-experts, either experts in other fields or the public, indeed have demands for geo-computation, due to the need for interdisciplinary research, decision-making and management, or even simple interests. Yet they have to be trained to become GIS experts first in order to use GIS. The computation-divide is the gap between computation demand of GIS and the capability and efficiency of GIS software. These software mostly utilize single-core desktop to compute, instead of high-performance computing resources such as multiple cores, clusters computing, or cloud computing. This leads to the inefficiency of GIS software in the face of data intensity and computation complexity. It is difficult for current GIS software to process a large amount of data and to complete the complex computation. These two divides need to be carefully tackled to make GIS easier for users to participate in geo-computation.

2. Existing efforts

Researchers have made efforts to address the two digital divides. These efforts can be summarized into three major stages: scripting, visual modelling, and modelling environment.

During the scripting stage, users (experienced GIS analysts) package the commonly used sequence of GIS commands for specific applications or for specific analytical tasks into macro-commands in the form of scripts which allow the analysts to reuse these analyses repeatedly, as well as to share their works with other users (ESRI 1996; Williams et al. 2000; Clerici et al. 2006; Abdella and Alfredsen 2010). For example, the Arc Macro Language (AML) in ArcInfo allows users to write AML commands (macro commands) to invoke the geospatial analysis methods that are supported by the ArcInfo. The parameters are specified in the commands by expert users regarding the task requirements or features. When these commands are organized according to the operation sequence of the workflow in a script, it wraps separated services into a complete workflow. The script tells the ArcInfo what the steps of the workflow are and what the settings for each step are. It saves expert users from the cumbersome step-by-step operations when conducting massive repetitive works. Non-expert users might also be able to reuse the workflow with the scripts shared by expert users. However, to build the workflow with programming language is technical and not intuitive even for expert users. In addition, the reusability of the scripts is very limited to specific analyses. This is because the chosen command for each step and the chosen parameter for each command in the scripts are fixed. This could be inappropriate when the research areas or input data or the type of analyses are changed.

During the visual modelling stage, researchers focus on the construction of the geographic workflow that can be shared and reused using visual modelling tools (Allen 2011; Dobesova 2011; Graser 2013; Magesh, Chandraseka, and Kaliraj 2012; Mericskay 2018). Such visual modelling tools include the *ModelBuilder* in ArcGIS and the *Graphic Modeler* in QGIS. The visual modelling tools visualize the input data and spatial analysis methods as graphics and connect them by arrows. The spatial analysis workflow is shown as flow charts in these tools. They usually allow users to create workflows by the drag-and-drop of the data and methods supported by the geo-computing platforms. Comparing to scripting, the visual modelling makes the building of workflow easier and more visually explicit for users. The visual workflow building process is somewhat easier yet is still arranged by users and requires knowledge on model structure, algorithm and parameter selections. Therefore, the construction of workflows is still manual and constrained to expert users. In addition, the same limitation for reusability with the scripting still exists. Non-expert users can only reuse the workflow built and shared by expert users that has rather limited reusability. Therefore, it is still complicated for experts and hard for non-experts to obtain a workflow that meets their specific geo-computation needs.

The third stage is the modelling environment which was developed to further ease the process of geo-computation (Wang 2010; Lü 2011). A modelling environment is a platform that uses resources from different computation platforms to organize, configure, and run models and/or geospatial analysis. The modelling environment simplifies the process of modelling and/or geographic analysis from four major aspects: the interpretation of the geographic analysis, the automation of workflow building and executing, data provision and management, the improvement of computation efficiency. However, the efforts in this area are quite fragmented.

The interpretation of the geographic analysis aims to translate the geographic question in natural language into machine-understandable geographic analysis tasks with identified methods or tools needed for completing the specific geographic analysis. Natural language process and semantic web (Gao and Goodchild 2013; Scheider, Ballatore, and Lemmens 2019; Yin et al. 2019) are used to relate the question asked in natural language with geographic analysis tools. For some simple questions, the questions in natural language can be directly interpreted into analysis tasks. For more complicated questions, the question might need to be deconstructed into several sub-questions to complete the task. For complicated tasks with several sub-questions, the question-answering approach can be used to interpret these sub-questions into tasks implemented with details (Kuhn and Ballatore 2015; Vahedi, Kuhn, and Ballatore 2016). The question-based approaches allow users to focus only on the question to be answered instead of the implementation details. However, this process does not involve the selection between different tools that can be used to answer the same question. The interpreted solution might not be the most appropriate one.

The automated workflow building consists of three subareas: description and connection of the workflow elements, the selection of appropriate algorithms and parameters, and the quality evaluation of the machine-generated workflow. Prior to building the workflow, researchers try to use the semantic web to describe the elements that can be joined into the workflow (Lemmens 2006; Brodaric 2007; Jiang et al. 2016). The elements include geospatial data and geospatial analysis services, and are described in terms of name, semantic meaning, relationship with other elements, usage, type, and/or other properties (Saquicela, Vilches-Blázquez, and Corcho 2012; Griraa et al. 2015; Hofer, Papadakis, and Mäs 2017; Diallo et al. 2018). The semantic web is recorded in a machine-processable structure like ontology (Lutz and Kolas 2007; Zhao, Foerster, and Yue 2012), which makes it feasible for the machine to connect these elements and build the workflow based on the semantic web (Alameh 2003; Di 2005; Yue et al. 2007; Scheider and Ballatore 2018; Škerjanec et al. 2014). The details of the model such as selecting appropriate algorithms and parameters are furnished by massive data analysis or regulated rules (Qin et al. 2016; Brown, Bennett, and French 2017; Chau 2007; Jiang et al. 2019; Liang et al. 2020). Once completed, the workflow is verified and then sent for execution (Qi et al. 2016). Users are able to execute the workflow simply by clicking a mouse button. ArcGIS Insights is one example that automatically generates workflows that can be used to analyse data easily according to data types and user objectives. (ESRI 2020a).

Modelling environment is also in charge of data provision and management. Data provision is to discover and retrieve geospatial data over the internet for possible model input (Hou et al. 2019). It uses semantic web or metadata that describes properties of data such as information on providers, quality, location, entity and attributes to discover geospatial data for specific analyses (McCarthy and Graniero 2006; Lutz and Kolas 2007; Wiegand and Garcia 2007; Gui et al. 2013; Qiu et al. 2017; Durante and Hardy 2015; Zhu et al. 2017; Zhu and Yang 2019). The modelling environment also provides utilities (simple database management functionalities) to organizes data in terms of data storage and retrieval, data format conversion (Yue et al. 2018; Zhao, Foerster, and Yue 2012; Belete, Voinov, and Morales 2017; Wang et al. 2018). The data provision and management free users from the trivial yet cumbersome work of data searching, preparing, and organizing process.

Another aim of the modelling environment is to improve the computation efficiency of the constructed workflows. Researchers develop different parallelization strategies to divide the computation into different sessions and process in parallel to improve computation efficiency (Healey et al. 1997; Hawick, Coddington, and James 2003; Zhao et al. 2016). The parallelization strategies are designed at two different levels. The first level is data division that divides geospatial data into parts to be loaded and processed in parallel. The data division strategy varies as data structure changes (Shook et al. 2016; Qin, Zhan, and Zhu 2014; Liu et al. 2014). The second is the division of computation which divides the computation process into parts to be carried out in parallel. The computation process is divided according to algorithm characteristics (Qin and Zhan 2012; Liu et al. 2016). The modelling environment also provides users with the access to high-performance computing resources like cluster computing and grid computing (Huang et al. 2011; Lecca et al. 2011; Hussain et al. 2013; Kim and Tsou 2013; Yang et al. 2011). For example, the Google Earth Engine (Padarian, Minasny, and McBratney 2015; Gorelick et al. 2017) uses cloud computing to allow users to conduct the geo-computation with high-performance computing resources over the internet, without the need to set up a local high-performance computing infrastructure.

The other aspect of modelling environment in the context of improving computation efficiency is to integrate heterogenous computing resources, even over the web. The modelling environment is not a constraint to one specific geo-computation platform as scripting and

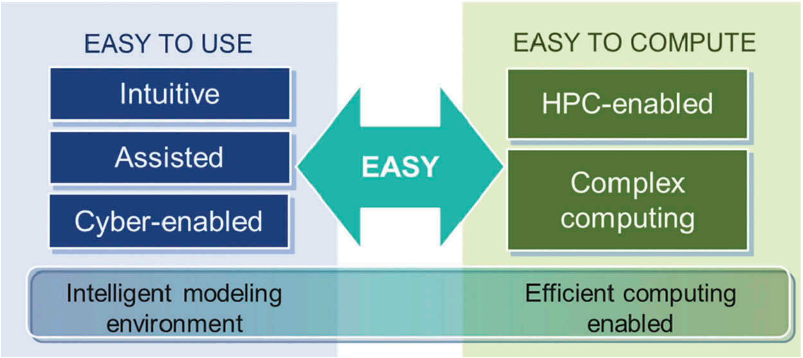

Figure 1. The idea of easy.

visualized modelling does. It integrates distributed geospatial services provided by various researchers and organizations. The modelling environments can integrate tools from existing platforms, as well as discover and utilize online services in the form of web services (Vaccari, Shvaiko, and Marchese 2009; Belete, Voinov, and Morales 2017; Chen et al. 2019; Evangelidis et al. 2014). More and more geospatial analysis tools are being published as web services these days (Zhao et al. 2012). To make this easier, the Esri and Microsoft work together to provide a cloud platform for users to manage their web services in the cloud (ESRI 2020b). Integration of these resources mediates the tools from different platforms to make them work under the same environment. For example, Belete, Voinov, and Morales (2017) developed a distributed model integration framework which integrates models and tools by searching the lexical database and semantics matching to connect these tools and models and their inputs/outputs, so as to construct a new workflow model. The integrated tools and models can be written in different languages, deployed in different hardware and software platforms, or made accessible online. Users can utilize the functionalities of various platforms under one gateway, which improves the efficiency by saving the time for software management, tools mediation and hardware configurations.

The capability of integrating various resources also makes sharing of geo-computing resources easier for developers in modelling environment. Nowadays, a developer can share the geospatial analysis methods and geospatial data with the form of standard web services, such as the Web Processing Service (WPS), the Web Map Service (WMS) and Web Feature Service (WFS) (Vretanos 2005; Schut 2007; Castronova, Goodall, and Elag 2013). Developers only need to provide the property and description of the web services. The modelling environments are designed with extensibility that easily allows the integration of these services (Liu, Padmanabhan, and Wang 2015; Jiang et al. 2016). One standard web service can be integrated into different modelling environments, which provide developers with an efficient way to share respective research innovations.

Modelling environment is a promising approach to ease the geo-computation process. However, current progresses are still fragmentary. For example, the algorithm or parameter selection methods are usually oriented for one specific model and cannot be applied to general situations. The parallelization is task-oriented and only limited geospatial tasks get accelerated. In addition, the description of expert knowledge on modelling in the form of semantic webs and the rule sets needs to be further enriched for automatically constructed workflow under different scenarios or different applications. Until now, no environment has been built to achieve all of the above features.

3. The basic idea of easy

We believe that in the process of overcoming the two digital divides in GIS 'easy' is the key. By 'easy' we mean: easy to use and easy to compute. Therefore, we argue that the next generation of GIS must be 'easy'.

To be easy to use, the realization of geographic analyses through GIS must be intuitive, assisted and cyber-enabled. 'Intuitive' means the process of geographic analysis (workflow) should be intuitive and easy to construct with little knowledge about specific techniques used. 'Intuitive' allows users to easily modify the settings (such as algorithms and their parameters) if they want and monitor the computation process. Users can easily become a part of the geo-computation through workflow construction, data provision and task execution if they so desire. Visual display of workflow in geometric shapes makes the geographic analysis more transparent, which will allow users to easily follow the construction of geographic analysis to ease the complication of geo-computing.

By 'assisted' it means that there is a need to develop a platform to help users with the execution of the geographic analysis, including workflow (analysis model) construction and data management. To help with workflow construction, the platform needs to contain modelling knowledge which will assist users in the construction of analysis model through suggesting or recommending to users the prospective modelling procedure, appropriate algorithms, and suitable parameters. To help with the data management, the platform should be able to bring the data into the same spatial/temporal scale and spatial/temporal domain, the same coordinate system, as well as at the same level of spatial granularity. In this way, users with insufficient geospatial analysis knowledge can also conduct geo-computation appropriately and easily.

'Cyber-enabled' means that the geo-computing process should be available over the internet and available over platforms that are able to utilize existing but heterogeneous computing resources across the cyberspace. By implementing the platform over the cyberspace, the computation performance does not rely on specific computing devices anymore. Users do not have to go through the cumbersome software installation process to participate in geo-computation. Instead, they can access the geo-computation online with any devices that are equipped with a browser and connected to the internet. By utilizing heterogeneous computing resources, the computation process can be distributed to different machines and platforms. In addition, cyber-enabled implementation of geo-computing would also make sharing of new methods easier because new methods can be wrapped into web services which can then be utilized through these cyber-enabled implementations. By designing the platform with these features, users do not have to be experts to conduct the geo-computation. The user-divide in existing GIS software will be much mitigated.

To be easy to compute, geo-computing has to be high-performance computing (HPC) enabled and complex computing capable. 'HPC-enabled' requires the platform to utilize various HPC resources such as cluster computing and grid computing. The utilization of HPC resources improves the computing efficiency in terms of hardware performance. 'Complex computing' is becoming an important feature of geo-computation that demands computing efficiency. Many of geo-computation tasks nowadays are usually complex due to the integrated nature of geographic analyses. Therefore, specialized parallel strategies need to be carefully designed to handle the acceleration of the complex computing. For example, in a fully sequentially dependent (spatially explicit) hydrological model, the overland flow routing and channel flow routing are performed sequentially from upstream to downstream simulation units. Apparently, there are dependent relationships among not only sub-basins but also basic simulation units within a basin. Therefore, they cannot be simply divided into separate units and simulated in parallel. Liu et al. (2016) designed a two-level layered approach to divide these sub-basins and simulation units into layers according to their dependent relationships, respectively. As we can see, strategies like this have to be developed to accommodate complex geographic computing.

In addition, coupled with 'cyber-enabled' implementation, 'easy to compute' is no longer limited to the computational resources provided in a uniform environment, and is made possible to utilize the heterogeneous computing resources available from different machines and platforms. The combination of the cyber resources together with HPC resources and specialized strategies for complex computing could create a new type of high-performance geo-computing platform. With this type of platforms, the computation divide in existing GIS software can be bridged or significantly mitigated.

With the aforementioned strategy, the goal of 'must be easy' for the next generation of GIS can be achieved through geo-computation platforms that are intelligent in modelling process, easy for users to participate, and efficient in computing (Figure 1).

4. Case studies

In this section, we use the easyGC, an implementation of easy geographic computing to illustrate the meaning of easy as discussed in Section 3. The system is available online through (http://www.easygeoc.net:8090/). The two cases were constructed under the easyGC to facilitate the computation in the domain of digital soil mapping and digital terrain analysis, respectively. The easyGC platform is an example of integrating some of the existing efforts into one GIS platform to make the process of geo-computing easier. The examples provided here are intended to elucidate the meaning of 'easy' from the perspective of GIS application users. The details on the development of the easyGC platform have been described elsewhere (Jiang et al. 2016; Qin et al. 2013) and beyond the scope of this paper. Clearly, research efforts are still needed to advance geo-computing platforms like easyGC.

4.1. CyberSoLIM

CyberSoLIM (Jiang et al. 2016) is a module of easyGC and was designed to simplify the computation in digital soil mapping (DSM). The general flow for conducting DSM in easyGC is shown in Figure 2 and can be categorized into six major steps. These steps were designed under the concept of 'easy' as discussed above and illustrated from the perspective of users when they need to conduct digital soil mapping.

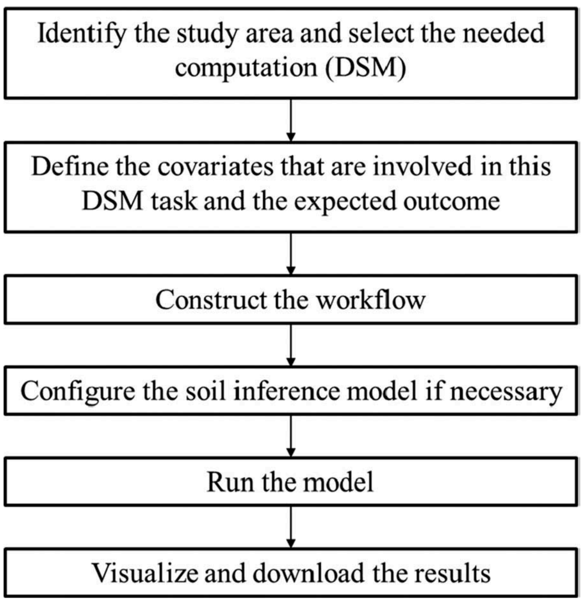

Figure 2. Flowchart of the process.

The first step in constructing digital soil mapping using the CyberSoLIM platform for a user is to specify the study area and the goal of the geo-computing (in this example it is DSM). Users can move to the area in the map view of the web page, then right-click and select 'Digital Soil Mapping' (Figure 3). A model building window with a soil inference method, the individual predictive soil mapping (iPSM, Zhu et al. 2015), will pop up.

Second, the user selects the environment covariates for this DSM task and define the expected mapping results. Click on the 'soil environment' to select the environmental variables to be used for soil mapping and these variables will be added as model inputs in the model building view. The output is designated to be soil property map and uncertainty map here (Figure 4).

Third, the user constructs the workflow. For the environment covariates that need to be derived from other data, the platform can automatically extend the workflow with services that generate these data. For example, when the user does not have slope data layer, he/she can right-click the 'slope' and choose 'automate', the 'slope service' and its input the data 'Filled DEM' (pit free DEM) will be automatically added into the soil mapping model. The 'Filled DEM' can be further automatedly retrieved by adding the 'Pit Remove Service' (Figure 5).

Fourth, the user configures the soil inference model if the user wants to adjust the parameters. The defaults are sufficient for most DSM applications. However, if the user wants, he/she can customize the parameters or choose a different method. Click on the 'individual

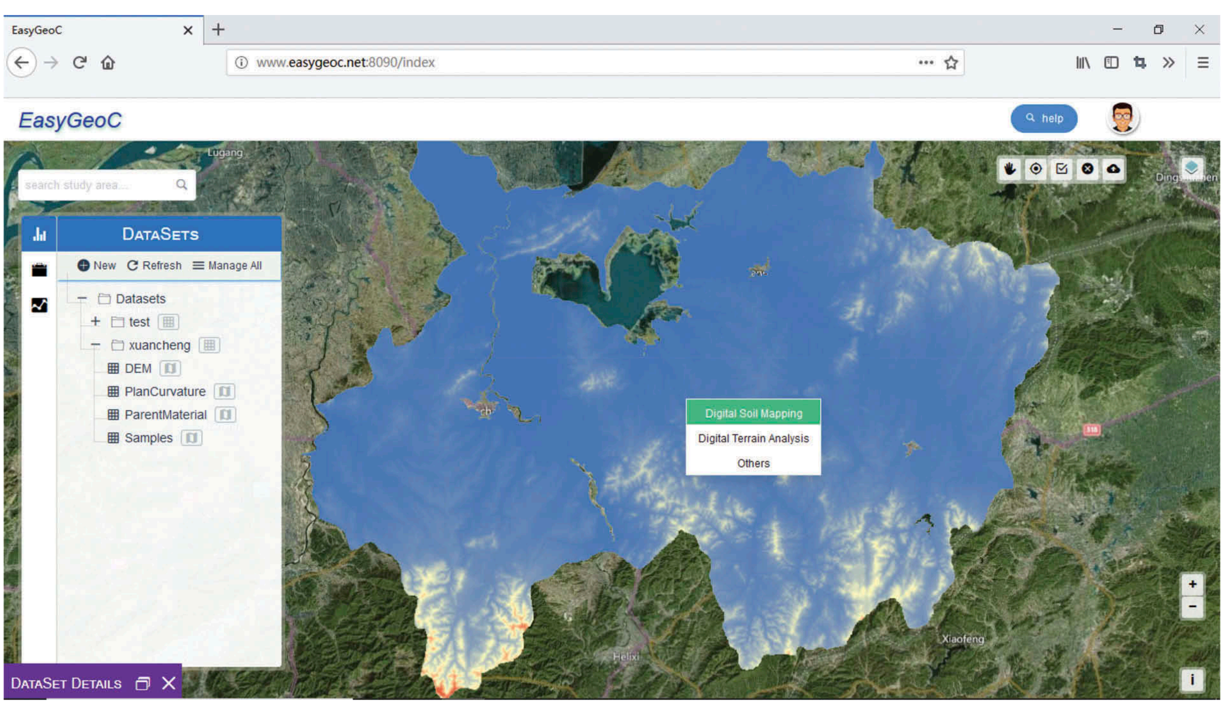

Figure 3. Identify the study area and select the needed computation (DSM).

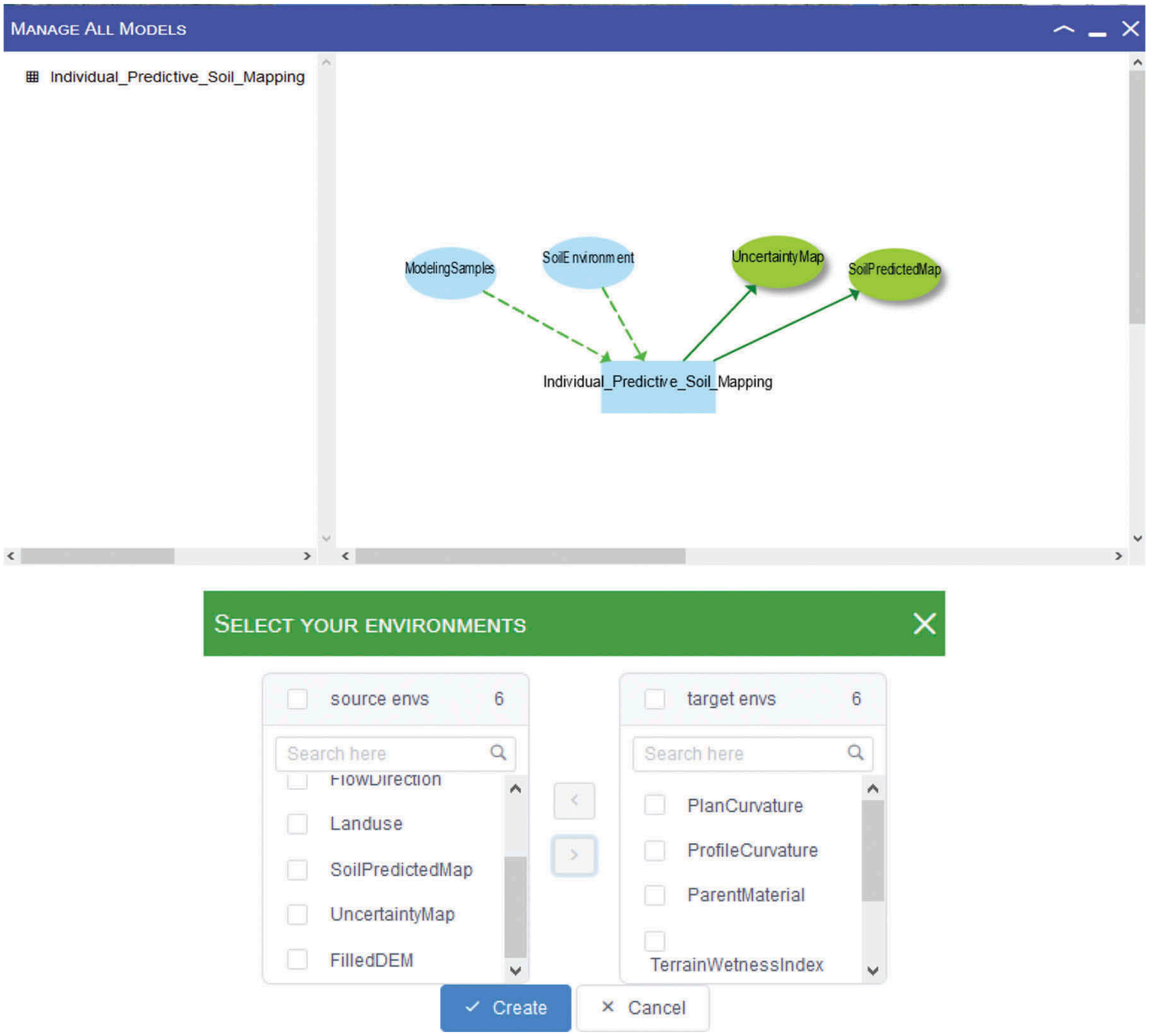

Figure 4. Select the involved environmental covariates and expected output.

Predictive Soil Mapping' to set the uncertainty threshold, category similarity integration method, and so on (Figure 6).

Fifth, the user runs the model. As soon as the user submits the configurations, the workflow and the model settings will be wrapped up as a workflow service and passed onto a high-performance computing service for execution. In this way, the user can really execute the workflow with one mouse click. During the workflow execution, the involved services can be provided by the platform, or heterogenous services over the cyber-space. The platform is designed with extensibility to allow the user integrates his/her own services. It enables the inclusion of new processing techniques as well as the utilization of high-performance computing resources. The involved services in this task are implemented as parallel computing running in a Linux cluster to accelerate the computation. Finally, the results are generated and stored on the platform. The user can view them on the web page or download them to his/her own computers (Figure 7).

The CyberSoLIM provides a DSM platform that is easy to use and easy to compute. On the 'easy to use', the modelling process is intuitive. During the workflow building, it uses graphics to visualize the soil inference and data preparation methods as well as the involved geospatial data. User can manage these workflow elements by clicking the graphics. The modelling process is also intelligently assisted (automated) with little inputs required from the user. User starts building the model by specifying his/her study area and goal. Then, he/she is assisted by the platform to expand the workflow to a complete task. The intelligence in workflow building is based on an ontology of GIS analysis, not based on a prescribed sequence of commands. Thus, this intelligence can be directly used in other application. The

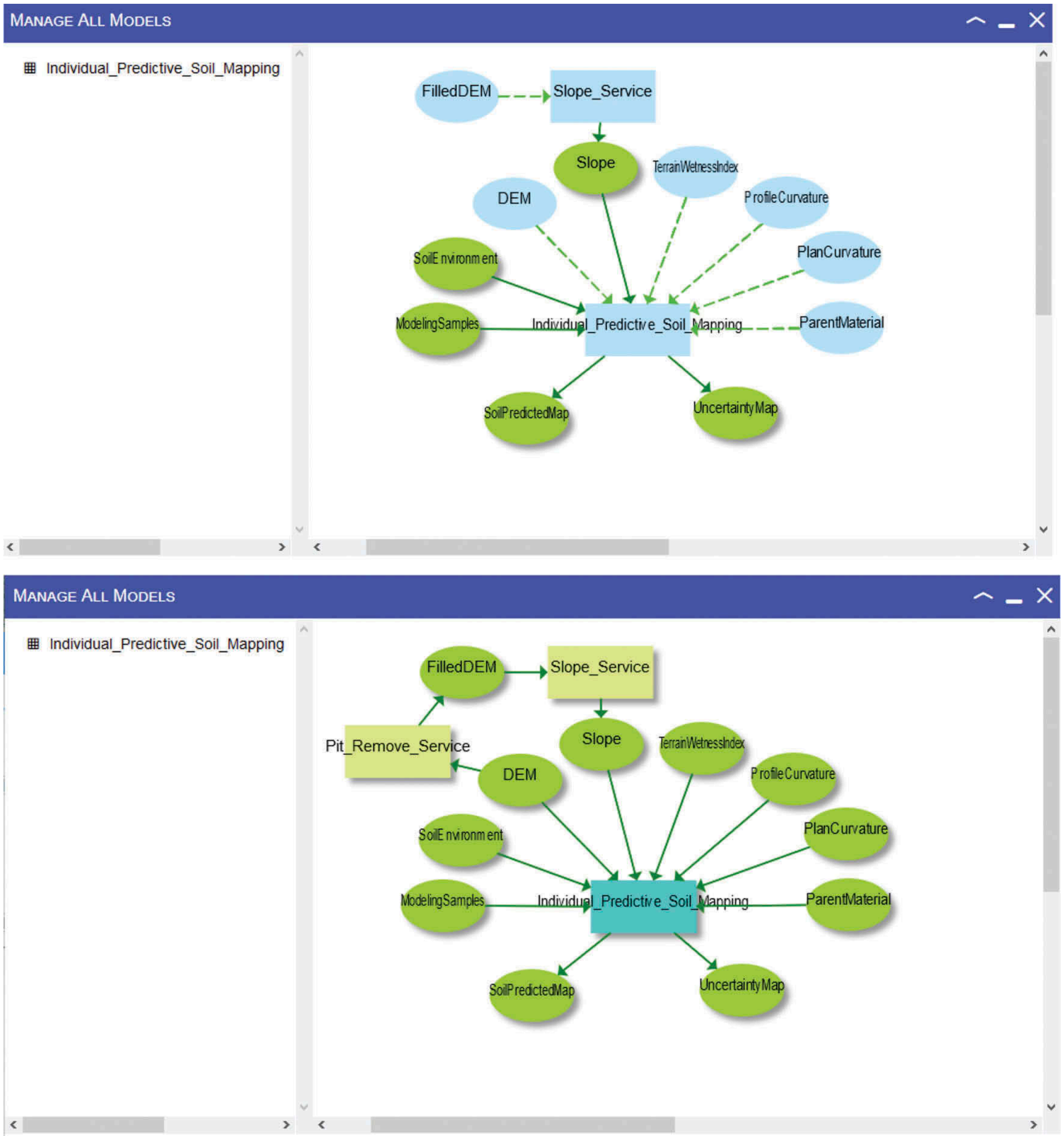

Figure 5. Construct the workflow.

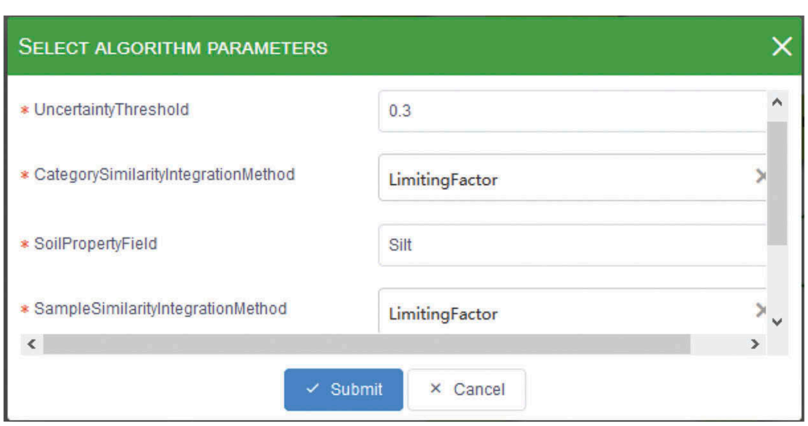

Figure 6. Configure the soil inference model.

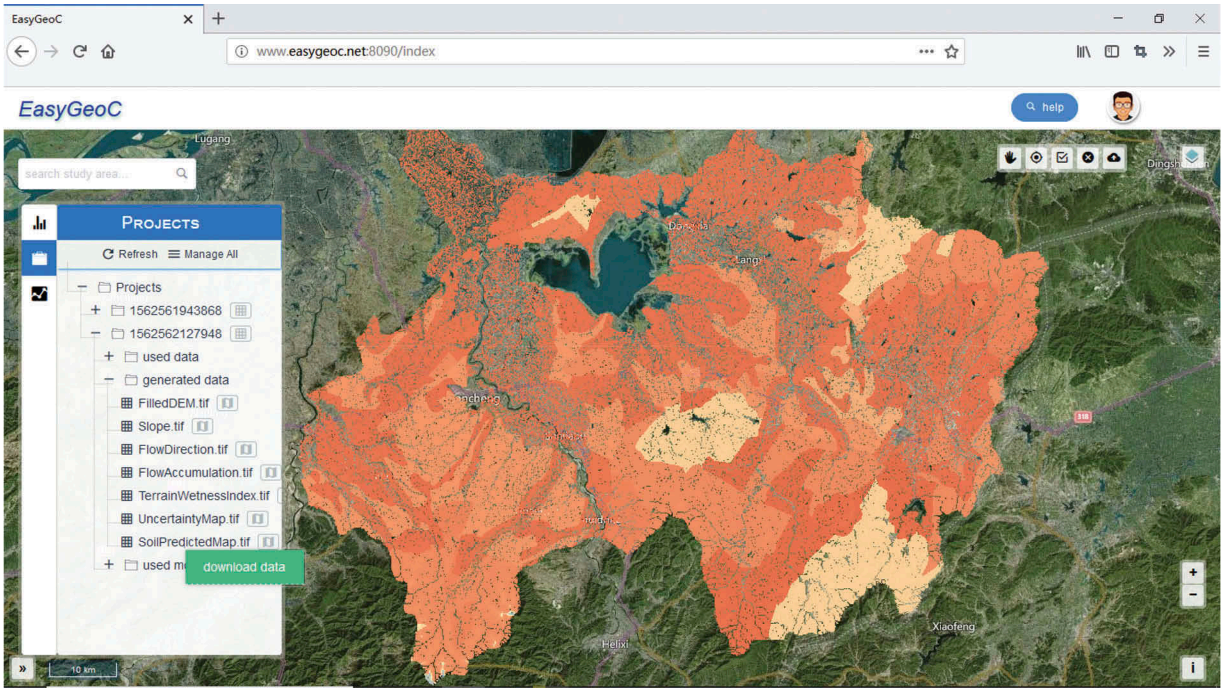

Figure 7. Visualize and download the results.

platform is cyber-enabled. It not only makes DSM available as a website but also provides a mean for users to share and use newly developed methods via web services. On the 'easy to compute', it is executed over Linux clusters and utilizes parallel computing to accelerate to a complex computing process. Thus, it is intelligent in modelling process and efficient in computing process.

4.2. *Application of easyGC for digital terrain analysis*

Digital terrain analysis (DTA) is another module of the easyGC (Qin et al. 2013). Here we use the computation of terrain wetness index (TWI) to illustrate the computation process. TWI is a common index which has been widely used and is computed through DTA. The workflow to compute TWI is shown in Figure 8. The computation process is rather complicated. In existing GIS software, the user needs to load the digital elevation model (DEM) data of the study area and use pit removing function to provide a pit free DEM (filled DEM). The algorithm for pit removing needs to be specified between incremental and decremental methods. Then, the filled DEM is used to compute specific catchment area (SCA) and slope gradient. To get the SCA, the user should first calculate the flow direction from the filled DEM, and then calculate the SCA from the flow direction. The user also needs to specify the algorithm for the flow direction between eight-direction (D8), Multiple Flow Direction (MFD), and D-Infinity (D-INF)

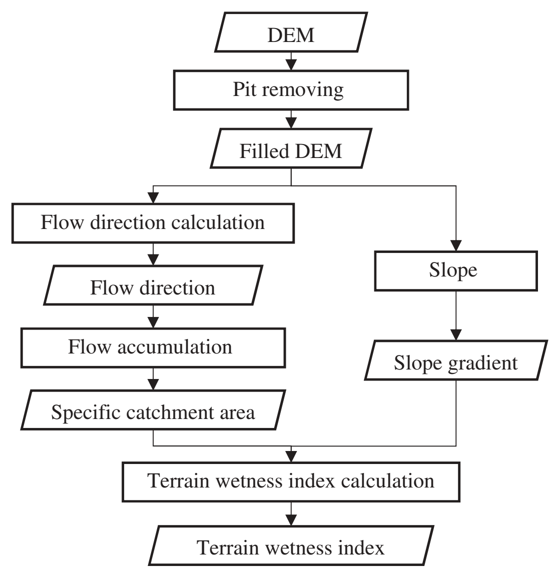

Figure 8. The workflow of calculating terrain wetness index.

methods. The slope gradient can be directly calculated from filled DEM using a slope calculation function with a second-order finite difference method or a third-order finite difference method. After that, the TWI can then be calculated from SCA and slope gradient. Usually, the computation process is quite complicated for non-expert users.

This complicated process can be simplified greatly in the easyGC platform. The users firstly specify the study area and expected results, which is TWI in this case.

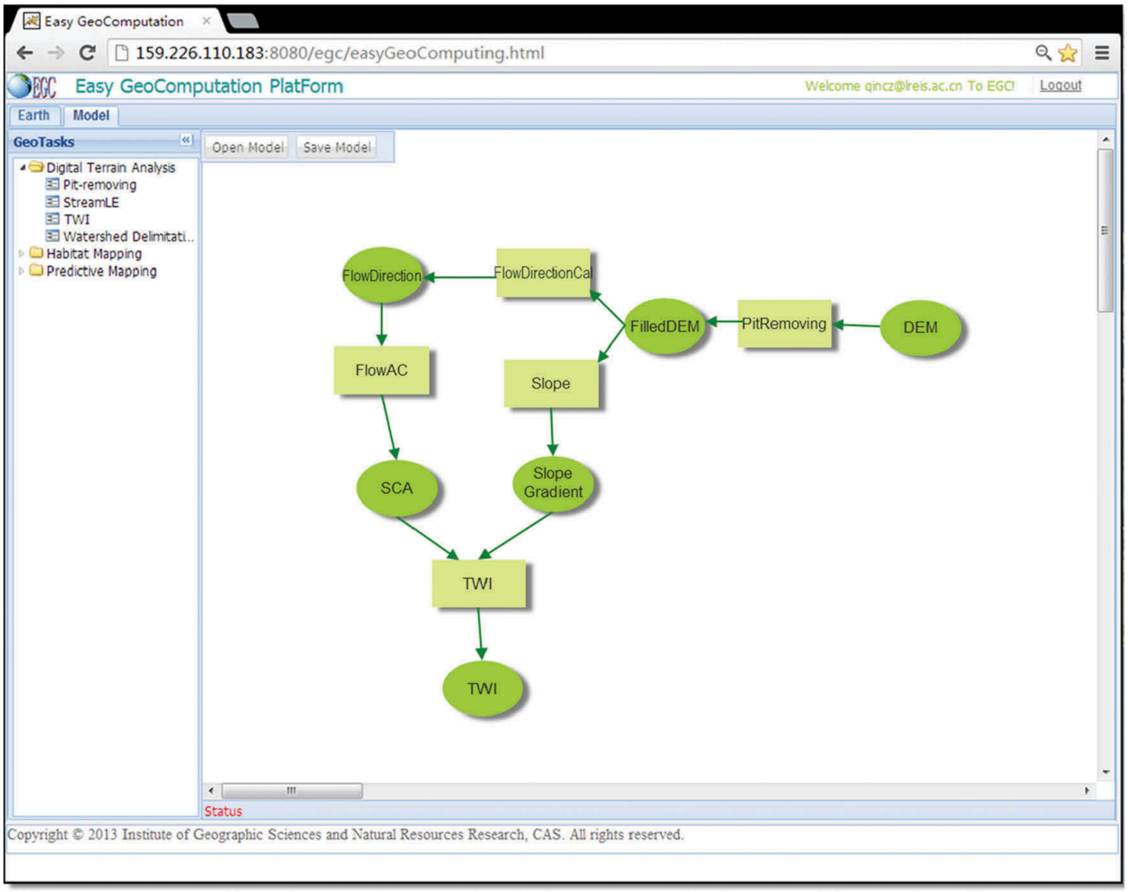

Figure 9. The modelling process to compute TWI in easyGC.

Then, the user can build the workflow in a visual modelling window with the help of inference engine based on the same intelligence as stated in the CyberSoLIM example. It can automate the construction of workflow by adding service that generates model input data. The platform is also able to choose the proper algorithm for a specific task. For example, when the algorithm for flow direction needs to be determined, the inference engine will choose the MFD algorithm when the flow direction is used for TWI calculation in a low-relief area at a finer scale. However, the D8 algorithm will be chosen if the flow direction will be used in watershed delineation task in a less spatially detailed scale. This process can be repeated until all input data can be found in the database or be calculated (Figure 9). After the model is built, the user can run the model with one-click. The computation is conducted in an HPC environment and many of the terrain analysis algorithms are implemented using parallel strategies to reduce the computing redundancy and improve the efficiency.

Just like what we have seen in CyberSoLIM, the DTA process, in this case, is both easy to use and easy to compute. To be easy to use, it uses graphical canvas to visualize the modelling process, allowing the process to be intuitive. It uses a knowledge base (general intelligence on the geographic analysis) to automate workflow expansion and algorithm selection process. It is also cyber-enabled by being accessible online and integrating other computing resources. To be easy to compute, it utilizes cluster computing to be HPC-enabled and is capable of complex computing with specialized parallel strategy. With this platform, users with little knowledge of digital terrain analysis can compute digital terrain attributes.

5. Lessons learned and future directions

5.1. Summaries on easyGC

The two cases shown above present how the easyGC platform makes the geo-computation for DSM and DTA easy. They demonstrate the possibility of resolving the user-divide and computation-divide in geographic analyses using GIS.

The 'easy' of geo-computation in esayGC is shown in both 'easy to use' and 'easy to compute'. In order to be easy to use, efforts for being 'intuitive', 'assisted' and 'cyber-enabled' have been made, respectively. First, the platform uses graphics to intuitively abstract the model elements, making the modelling process simple and easy. Users can easily understand what steps are used for getting which data. They can easily manage each step and each datum by clicking the graphics to alter the parameter or data sources if they wish. Second, it provides heuristics (intelligence) to automate the modelling process that is also open for users to modify the process with suggestions provided by the platform. Users can compute soil map or TWI without knowing what steps are involved in the model, how they are organized, or what algorithms should be chosen. Third, it is cyber-enabled so that it is accessible online which allows heterogeneous resources to be integrated into the form of web services. It frees users from the tedious and technical work of software management. It narrows the user-divide by allowing users with insufficient expert knowledge to participate in geo-computation.

In terms of 'easy to compute', the platform is HPC-enabled and capable of complex computing. It utilizes cluster computing resources to accelerate the computation. Together with the 'cyber-enabled' feature, users can access the functionalities of various platforms online and enjoy the high efficiency of HPC resources on their personal devices. The computation process gets accelerated without the constraints of the operating system. In addition, various parallel strategies are used to deal with the complex geo-computation. The parallel strategies divide computation into parallel parts to be processed simultaneously with multiple cores. It further improves the computing efficiency by making full use of HPC resources. Integration of heterogeneous resources together with specialized parallel strategies not only saves time for users but also allows for the efficient computation for complex geographic analysis, and thus address the issue of computation-divide in geographic analysis.

The easyGC well addresses the two digital divides faced in geographic analysis using existing GIS software. It achieved the goal of 'easy' in the domain of DSM and DTA by combining intelligent modelling environment and HPC-enabled computation.

In summary, ***to be easy*** future GIS software should have the following four characteristics:

1) ***Goal-driven***: the process-oriented workflow building process in existing GIS software needs to be changed to user goal-oriented. In existing software, users need to know every modelling detail before building the workflow. In easy geo-computing platform, users should be able to build the workflow only knowing what they want to achieve (the expected results). The platform should be able to help users build the path to get there. Users without detailed geographic analysis knowledge can also participate in geo-computing.

2) ***Intelligent***: the platform should have the knowledge to assist users during computation. There are three levels of intelligence. First, the easy geo-computing environment should have the knowledge of sequences of procedures. Second, it should possess the capability to choose the appropriate algorithms and parameters matching the given application context. Last but not least, it needs to be able to use new forms of data which may be spatially biased and unstructured in nature. The three levels of intelligence allow users, both experts and non-experts, to acquire expected results in a more simplified way.

3) ***Easily accessible***: easy geo-computation platform itself must be easily accessible. To be easily accessible, it is preferred to implement the computation environment over the cyberspace. Users can access to it anywhere by mobile terminals. In addition, the platform should integrate heterogeneous cyber resources to improve the computing efficiency.

4) ***Participatory***: the platform must allow users to contribute to data and algorithms easily. The low efficiency of sharing in existing GIS software must be changed. The platform must be designed with extensibility to allow for easy sharing. The platforms should have the ability to integrate the shared inventions in various forms. When sharing becomes easy, the platform becomes a community of sharing for new ideas or new methods on geo-computation.

Bear these four characteristics in mind, the future geo-computing platforms are supposed to be 'AAA', that is, ***Anybody*** can have access to geo-computing ***Anywhere*** and ***Anytime***. Users are able to conduct geo-computation without the constraints of expert knowledge, operating environment. In this way, the digital divides faced in geographic analysis using existing GIS software will be mitigated, even removed.

5.2. Future research directions

The easyGC provides an example of future geo-computation platform. The research for building easy geo-computation platforms is only at a starting stage. There are still many problems which need to be addressed and solved to further simplify the computation processes and extend the application of easy geo-computation to more domains and to more people.

In terms of goal-driven, the goals need to be expanded to include any geospatial problems of any forms, even speeches. The future geo-computing platforms should be able to translate various forms of goals into machine-processable geospatial problems.

To be more intelligent, it needs to be further improved from the following three aspects. The geospatial analysis task needs to be automatically chosen according to the geographic nature of the study area, the data, and the goals. The knowledge base for model settings, like the choice of the environmental covariate in CyberSoLIM, should be enriched to cover the settings for every involved model. The platform should be intelligent enough to find relevant data from various platforms in various forms and support the utilization of these heterogenous data to acquire more up-to-date and comprehensive information. The intelligence of the next generation of GIS should cover every aspect of the geo-computation process, from geospatial analysis task selection, to model setting, and to data provision and utilization.

When it comes to more easily accessible, the integration of heterogenous resources should be spontaneous. The platform should be able to spontaneously find and integrate the resources of data and algorithms over the cyberspace. The accessible resources should not be confined to those intentionally added by developers or users.

The next generation of GIS could also be more participatory by accepting a wider range of sharing. Web services are currently the most widely acceptable forms of sharing. Future GIS platforms should either support other forms of resources sharing or provide the functionality to automatically wrap other forms of sharing as web services. Aside from the sharing of data and algorithms, the content of sharing should also be extended to the knowledge base. The sharing should be more flexible to build a more active geo-computing community.

6. Conclusion

In existing GIS software, there are severe user divides between users of GIS and the target users of GIS software, and computation divide between computation need and the computation capability and efficiency provided by GIS software. There have been efforts in different stages to simplify the process of geo-computation, from scripting to visual modelling, and to modelling environment. Yet, building an easy geo-computing platform is the responsibility of the next generation of GIS so that these divides can be addressed by making geo-computation easy, including easy to use and easy to compute. The easyGC platform with cases in digital soil mapping and digital terrain analysis demonstrates the potential of being easy and the feasibility to fill the gaps. It narrows the user divide by building an intelligent modelling environment that is intuitive and automated. It narrows the computation divide by building an HPC-enabled computing environment that is cyber-enabled and complex computing capable. The easyGC presents an embryonic form of easy geo-computation for the next generation of GIS and proves the feasibility overcoming the two divides. The next generation of GIS should be built to be goal-driven, intelligent, easily accessible, and participatory. The ultimate objective for the next generation of GIS is to allow anyone, both experts and non-experts, to participate in geo-computation at anywhere and anytime.

Acknowledgements

This study was funded by the National Natural Science Foundation of China (41871300, 41431177), the National Basic Research Program of China (No. 2015CB954102), and the PAPD program of Jiangsu Higher Education Institutions. The support received by A-Xing Zhu through the Vilas Associate Award, the Hammel Faculty Fellow Award, the Manasse Chair Professorship from the University of Wisconsin-Madison are greatly appreciated. The support provided by China Scholarship Council (CSC) during a visit of Fang-he Zhao to the University of Wisconsin-Madison is acknowledged.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the National Natural Science Foundation of China [41871300, 41431177].

ORCID

A-Xing Zhu http://orcid.org/0000-0002-5725-0460
Fang-He Zhao http://orcid.org/0000-0001-8944-2427
Cheng-Zhi Qin http://orcid.org/0000-0002-5910-9807

References

Abdella, Y., and K. Alfredsen. 2010. "A GIS toolset for automated processing and analysis of radar precipitation data." *Computers & geosciences* 36 (4): 422–429. doi:10.1016/j.cageo.2009.08.008.

Alameh, N.. 2003. "Chaining geographic information web services." *IEEE internet Computing* 7 (5): 22–29. doi:10.1109/MIC.2003.1232514.

Allen, D. W. 2011. *Getting to know ArcGIS modelBuilder*. Redlands: Esri Press.

Belete, G. F., A. Voinov, and J. Morales. 2017. "Designing the Distributed Model Integration Framework – DMIF." *Environmental modelling and software* 94: 112–126. doi:10.1016/j.envsoft.2017.04.003.

Boroushaki, S., and J. Malczewski. 2008. "Implementing an extension of the analytical hierarchy process using ordered weighted averaging operators with fuzzy quantifiers in ArcGIS." *Computers & geosciences* 34 (4): 399–410. doi:10.1016/j.cageo.2007.04.003.

Brodaric, B. 2007. "Geo-Pragmatics for the Geospatial Semantic Web." *Transactions in GIS* 11 (3): 453–477. doi:10.1111/j.1467-9671.2007.01055.x.

Brown, J. L., J. R. Bennett, and C. M. French. 2017. "SDMtoolbox 2.0: the next generation Python-based GIS toolkit for landscape genetic, biogeographic and species distribution model analyses." *PeerJ* 5: e4095. doi:10.7717/peerj.4095.

Castronova, a. M., J. L. Goodall, and M. M. Elag. 2013. "Models as web services using the open geospatial consortium (OGC) web processing service (WPS) standard." *Environmental modelling and software* 41: 72–83. doi:10.1016/j.envsoft.2012.11.010.

Chau, K. W. 2007. "An ontology-based knowledge management system for flow and water quality modeling." *Advances in Engineering Software* 38 (3): 172–181. doi:10.1016/j.advengsoft.2006.07.003.

Chen, M., S. Yue, G. Lü, H. Lin, C. Yang, Y. Wen, R. Hou, D. Xiao, and H. Jiang. 2019. "Teamwork-oriented integrated modeling method for geo-problem solving." *Environmental modelling and software 119*: 111–123. doi:10.1016/j.envsoft.2019.05.015.

Clerici, A., S. Perego, C. Tellini, and P. Vescovi. 2006. "A GIS-based automated procedure for landslide susceptibility mapping by the Conditional Analysis method: the Baganza valley case study (Italian Northern Apennines)." *Environmental Geology* 50 (7): 941–961. doi:10.1007/s00254-006-0264-7.

Di, L. 2004. "Distributed geospatial information services-architectures, standards, and research issues." *The international archives of photogrammetry, remote sensing, and spatial information sciences,* vol. XXXV, Part 2. Commission II, Istanbul, Turkey, 7 (invited).

Di, L. 2005. "a Framework for Developing Web-Service-Based Intelligent Geospatial Knowledge Systems." *Geographic Information Sciences* 11 (1): 24–28.

Diallo, B. a. a., T. Badard, F. Hubert, and S. Daniel. 2018. "Towards Semantic Geo/BI: a Novel Approach for Semantically Enriching Geo/BI Data with OWL Ontological Layers (OOLAP and ODW) to Enable Semantic Exploration, Analysis and Discovery of Geospatial Business Intelligence Knowledge." *International Journal of Information Engineering and Electronic Business* 10 (6): 1. doi:10.5815/ijieeb.2018.06.01.

Dobesova, Z. 2011. "Visual programming language in geographic information systems." In *Proceedings of the 2nd international conference on Applied informatics and computing theory*, September, 276–280. World Scientific and Engineering Academy and Society (WSEAS). Stevens Point, WI.

Durante, K., and D. Hardy. 2015. "Discovery, management, and preservation of geospatial data using hydra." *Journal of map & geography libraries 11* (*2*): 123–154. doi:10.1080/15420353.2015.1041630.

Elwood, S., M. F. Goodchild, and D. Z. Sui. 2012. "Researching volunteered geographic information: Spatial data, geographic research, and new social practice." *Annals of the association of American geographers* 102 (3): 571–590. doi:10.1080/00045608.2011.595657.

ESRI. 1996. *Avenue — Customization and Application Development for ArcView*. Redlands, CA: Environmental Systems Research Institute Inc.

ESRI. 2019. "ArcGIS Desktop." Accessed 14 October. http://desktop.arcgis.com

ESRI. 2020a. "Location Analytics Software | ArcGIS Insight." Accessed 17 March 2020. https://www.esri.com/en-us/arcgis/products/arcgis-insights/overview?rmedium=esri_com_redirects01&rsource=/en-us/arcgis/products/insights-for-arcgis/overview

ESRI. 2020b. "Esri and Microsoft Cloud Deployment Options." Accessed 17 March. https://www.esri.com/en-us/about/esri-partner-network/our-partners/global-alliances/microsoft-esri/microsoft-cloud

Evangelidis, K., K. Ntouros, S. Makridis, and C. Papatheodorou. 2014. "Geospatial services in the Cloud." *Computers & geosciences 63*: 116–122. doi:10.1016/j.cageo.2013.10.007.

Gao, S., and M. F. Goodchild 2013. "Asking spatial questions to identify GIS functionality." In *2013 Fourth International Conference on Computing for Geospatial Research and Application*, July, 106–110. San Jose, CA: IEEE.

Gorelick, N., M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore. 2017. "Google Earth Engine: Planetary-scale geospatial analysis for everyone." *Remote sensing of environment 202*: 18–27. doi:10.1016/j.rse.2017.06.031.

Graser, A. 2013. *Learning QGIS 2.0*. Birmingham: Packt Publishing.

Griraa, J., Y. Bedard, S. Roche, and R. Devillersb. 2015. "Towards a collaborative knowledge discovery system for enriching semantic information about risks of geospatial data misuse." In Wenzhong, Shi, Bo Wu, and Alfred Stein (Eds.)., *Uncertainty Modelling and Quality Control for Spatial Data*, 233. Boca Raton, FL.

Gui, Z., C. Yang, J. Xia, K. Liu, C. Xu, J. Li, and P. Lostritto. 2013. "a performance, semantic and service quality-enhanced distributed search engine for improving geospatial resource discovery." *International Journal of Geographical Information Science* 27 (6): 1109–1132. doi:10.1080/13658816.2012.739692.

Hawick, K. A., P. D. Coddington, and H. A. James. 2003. "Distributed frameworks and parallel algorithms for processing large-scale geographic data." *Parallel computing* 29 (10): 1297–1333. doi:10.1016/j.parco.2003.04.001.

Healey, R., S. Dowers, B. Gittings, and M. J. Mineter, eds.. 1997. *Parallel processing algorithms for GIS*. Bristol: CRC Press.

Hofer, B., E. Papadakis, and S. Mäs. 2017. "Coupling Knowledge with GIS Operations: The Benefits of Extended Operation Descriptions." *ISPRS International Journal of Geo-Information* 6 (2): 40. doi:10.3390/ijgi6020040.

Hou, Z. W., C. Z. Qin, A. X. Zhu, P. Liang, Y. J. Wang, and Y. Q. Zhu. 2019. "From manual to intelligent: a review of input data preparation methods for geographic modeling." *ISPRS International Journal of Geo-Information* 8 (9): 376. doi:10.3390/ijgi8090376.

Huang, F., D. Liu, X. Tan, J. Wang, Y. Chen, and B. He. 2011. "Explorations of the Implementation of a Parallel IDW Interpolation Algorithm in a Linux Cluster-Based Parallel GIS." *Computers & geosciences 37* (*4*): 426–434. doi:10.1016/j.cageo.2010.05.024.

Hussain, H., S. U. R. Malik, a. Hameed, S. U. Khan, G. Bickler, N. Min-Allah, … J. Kolodziej. 2013. "a survey on resource allocation in high performance distributed computing systems." *Parallel computing 39* (*11*): 709–736. doi:10.1016/j.parco.2013.09.009.

Jiang, J., A. X. Zhu, C. Z. Qin, and J. Liu. 2019. "a knowledge-based method for the automatic determination of hydrological model structures." *Journal of Hydroinformatics 21* (6): 1163–1178. doi:10.2166/hydro.2019.029.

Jiang, J. C., A.-X. Zhu, C.-Z. Qin, T.-X. Zhu, J.-Z. Liu, F. Du, J. Liu, G.-M. Zhang, and Y.-M. An. 2016. "CyberSoLIM: a cyber platform for digital soil mapping." *Geoderma* 263: 234–243. doi:10.1016/j.geoderma.2015.04.018.

Kim, I. H., and M. H. Tsou. 2013. "Enabling Digital Earth simulation models using cloud computing or grid computing–two approaches supporting high-performance GIS simulation frameworks." *International Journal of Digital Earth 6* (*4*): 383–403. doi:10.1080/17538947.2013.783125.

Kuhn, W., and A. Ballatore. 2015. "Designing a language for spatial computing." In Bacao F., Santos M., Painho M. (eds.)., *AGILE 2015*, 309–326. Cham: Springer.

Lecca, G., M. Petitdidier, L. Hluchy, M. Ivanovic, N. Kussul, N. Ray, and V. Thieron. 2011. "Grid computing technology for hydrological applications." *Journal of Hydrology 403* (*1–2*): 186–199. doi:10.1016/j.jhydrol.2011.04.003.

Lemmens, R. 2006. "Semantic interoperability of distributed geo-services." *ITC*, September 1–2.

Liang, P., C.-Z. Qin, A.-X. Zhu, Z. Hou, N.-Q. Fan, and Y.-J. Wang. 2020. "a case-based method of selecting covariates for digital soil mapping." *Journal of Integrative Agriculture 19*: 2–11.

Liu, J., A. X. Zhu, Y. Liu, T. Zhu, and C. Z. Qin. 2014. "a layered approach to parallel computing for spatially distributed hydrological modeling." *Environmental modelling and software* 51: 221–227. doi:10.1016/j.envsoft.2013.10.005.

Liu, J., A.-X. Zhu, C.-Z. Qin, H. Wu, and J. Jiang. 2016. "a two-level parallelization method for distributed hydrological models." *Environmental modelling and software 80*: 175–184. doi:10.1016/j.envsoft.2016.02.032.

Liu, Y., A. Padmanabhan, and S. Wang. 2015. "CyberGIS Gateway for enabling data-rich geospatial research and education." *Concurrency and Computation: Practice and Experience 27* (*2*): 395–407. doi:10.1002/cpe.3256.

Lü, G. 2011. "Geographic analysis-oriented virtual geographic environment: framework, structure and functions." *Science China Earth Sciences* 54 (5): 733–743. doi:10.1007/s11430-011-4193-2.

Lutz, M., and D. Kolas. 2007. "Rule-Based Discovery in Spatial Data Infrastructure. Transactions in GIS, 11 (3),317-336.Lutz, M. (2007). Ontology-based descriptions for semantic discovery and composition of geoprocessing services." *Geoinformatica* 11 (1): 1–36. doi:10.1007/s10707-006-7635-9.

Magesh, N. S., N. Chandraseka, and S. Kaliraj. 2012. "a GIS based automated extraction tool for the analysis of basin morphometry." *Bonfring International Journal of Industrial Engineering and Management Science 2*: 32–35. Special Issue Special Issue on Geospatial Technology Development in Natural Resource and Disaster Management).

Maguire, D. J. 1991. "An overview and definition of GIS." In *Geographical Information Systems: Principles and Applications*, edited by D. J. Maguire, M. F. Goodchild, and D. W. Rhind, 9–20. Vol. 1. London: Longman Scientific and Technical.

McCarthy, J. D., and P. A. Graniero. 2006. "a GIS-based borehole data management and 3D visualization system." *Computers & geosciences* 32 (10): 1699–1708. doi:10.1016/j.cageo.2006.03.006.

Mericskay, B. 2018. "Automation of Workflows for the Installation of a Wind Farm." *QGIS and Applications in Territorial Planning 3*: 125–168.

Miller, S. N., D. J. Semmens, D. C. Goodrich, M. Hernandez, R. C. Miller, W. G. Kepner, and D. P. Guertin. 2007. "The automated geospatial watershed assessment tool." *Environmental modelling and software* 22 (3): 365–377. doi:10.1016/j.envsoft.2005.12.004.

Neitsch, S. L., J. G. Arnold, J. R. Kiniry, and J. R. Williams. 2011. *Soil and water assessment tool theoretical documentation version 2009*. Temple, TX: Texas Water Resources Institute.

Padarian, J., B. Minasny, and A. B. McBratney. 2015. "Using Google's cloud-based platform for digital soil mapping." *Computers & geosciences 83*: 80–88. doi:10.1016/j.cageo.2015.06.023.

QGIS Development Team. 2019. "QGIS Geographic Information System." Open Source Geospatial Foundation Project. http://qgis.osgeo.org

Qi, K., Z. Gui, Z. Li, W. Guo, H. Wu, and J. Gong. 2016. "An extension mechanism to verify, constrain and enhance geoprocessing workflows invocation." *Transactions in GIS* 20 (2): 240–258. doi:10.1111/tgis.12152.

Qin, C.-Z., J.-C. Jiang, L.-J. Zhan, Y.-J. Lu, and A.-X. Zhu. 2013. "A Browser/Server-based Prototype of Heuristic Modelling Environment for Digital Terrain Analysis." In *Geomorphometry'2013*, October 15–20. Nanjing: Nanjing Normal University.

Qin, C.-Z., X.-W. Wu, J.-C. Jiang, and A.-X. Zhu. 2016. "Case-based knowledge formalization and reasoning method for digital terrain analysis–application to extracting drainage networks." *Hydrology and Earth System Sciences* 20 (8): 3379–3392. doi:10.5194/hess-20-3379-2016.

Qin, C.-Z., and L. Zhan. 2012. "Parallelizing flow-accumulation calculations on graphics processing units—From iterative DEM preprocessing algorithm to recursive multiple-flow-direction algorithm." *Computers & geosciences* 43: 7–16. doi:10.1016/j.cageo.2012.02.022.

Qin, C.-Z., L. J. Zhan, and A.-X. Zhu. 2014. "How to apply the geospatial data abstraction library (GDAL) properly to parallel geospatial raster I/O?" *Transactions in GIS 18* (*6*): 950–957. doi:10.1111/tgis.12068.

Qiu, L., Z. Du, Q. Zhu, and Y. Fan. 2017. "An integrated flood management system based on linking environmental models and disaster-related data." *Environmental modelling and software* 91: 111–126. doi:10.1016/j.envsoft.2017.01.025.

Saquicela, V., L. M. Vilches-Blázquez, and O. Corcho. 2012. "Adding semantic annotations into (geospatial) restful services." *International Journal on Semantic Web and Information Systems (IJSWIS)* 8 (2): 51–71. doi:10.4018/jswis.2012040103.

Scheider, S., and A. Ballatore. 2018. "Semantic typing of linked geoprocessing workflows." *International Journal of Digital Earth* 11 (1): 113–138. doi:10.1080/17538947.2017.1305457.

Scheider, S., A. Ballatore, and R. Lemmens. 2019. "Finding and sharing GIS methods based on the questions they answer." *International journal of digital earth 12* (5): 594–613. doi:10.1080/17538947.2018.1470688.

Schut, P. 2007. *OpenGIS® Web Processing Service*. OGC 05-007r7Wayland, MA: Open Geospatial Consortium Technical Report.

Shook, E., M. E. Hodgson, S. Wang, B. Behzad, K. Soltani, A. Hiscox, and J. Ajayakumar. 2016. "Parallel cartographic modeling: a methodology for parallelizing spatial data processing." *International Journal of Geographical Information Science 30* (*12*): 2355–2376. doi:10.1080/13658816.2016.1172714.

Škerjanec, M., N. Atanasova, D. Čerepnalkoski, S. Džeroski, and B. Kompare. 2014. "Development of a knowledge library for automated watershed modeling." *Environmental modelling and software* 54: 60–72. doi:10.1016/j.envsoft.2013.12.017.

Sorokine, A. 2007. "Implementation of a parallel high-performance visualization technique in GRASS GIS." *Computers & geosciences* 33 (5): 685–695. doi:10.1016/j.cageo.2006.09.008.

SuperMap Software Co., Ltd. 2019. "SuperMap GIS for PC." Accessed 14 October. https://www.supermap.com/en-us/list/?7_1.html

Tarboton, D. G. 2005. *Terrain Analysis using Digital Elevation Models (TauDEM)*. Logan: Utah State University.

Thieler, E. R., E. A. Himmelstoss, J. L. Zichichi, and A. Ergul. 2009. "The Digital Shoreline Analysis System (DSAS) version 4.0-an ArcGIS extension for calculating shoreline change." *No. 2008-1278*. Reston, VA: US Geological Survey.

Vaccari, L., P. Shvaiko, and M. Marchese. 2009. "a geo-service semantic integration in spatial data infrastructures." *International Journal of Spatial Data Infrastructures Research 4*: 24–51.

Vahedi, B., W. Kuhn, and A. Ballatore. 2016. "Question-based spatial computing—a case study." In Sarjakoski T., Santos M., Sarjakoski L. (eds.)., *Geospatial Data in a Changing World*, 37–50. Cham: Springer.

Vretanos, P. 2005. "Web feature service implementation specification." Accessed on May 11, 2020. https://portal.ogc.org/files/?artifact_id=8339.

Wang, J., M. Chen, G. Lü, S. Yue, K. Chen, and Y. Wen. 2018. "a Study on Data Processing Services for the Operation of Geo-Analysis Models in the Open Web Environment." *Earth and Space Science* 5 (12): 844–862. doi:10.1029/2018EA000459.

Wang, S. 2010. "a cyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis." *Annals of the Association of American Geographers* 100 (3): 535–557. doi:10.1080/00045601003791243.

Wei, Y., S. K. Santhana-Vannan, and R. B. Cook 2009. "Discover, visualize, and deliver geospatial data through OGC standards-based WebGIS system." In *2009 17th International Conference on Geoinformatics*, August, 1–6. Fairfax, VA: IEEE.

Wiegand, N., and C. García. 2007. "a task-based ontology approach to automate geospatial data retrieval." *Transactions in GIS* 11 (3): 355–376. doi:10.1111/j.1467-9671.2007.01050.x.

Williams, W. A., M. E. Jensen, J. C. Winne, and R. L. Redmond. 2000. "An automated technique for delineating and characterizing valley-bottom settings." In Sandhu, S. S., Melzian, B. D., Long, E. R., Walton, B. T., & Whitford, W. G. (Eds.)., *Monitoring Ecological Condition in the Western United States*, 105–114. Dordrecht: Springer.

Wood, J. 2009. "Geomorphometry in landserf." *Developments in soil science* 33: 333–349.

Yang, C., M. F. Goodchild, Q. Huang, D. Nebert, R. Raskin, Y. Xu, M. Bambacus, and D. Fay. 2011. "Spatial cloud computing: How can the geospatial sciences use and help shape cloud computing?" *International Journal of Digital Earth 4* (*4*): 305–329. doi:10.1080/17538947.2011.587547.

Yin, Z., C. Zhang, D. W. Goldberg, and S. Prasad. 2019. "An NLP-based Question Answering Framework for Spatio-Temporal Analysis and Visualization." In *Proceedings of the 2019 2nd International Conference on Geoinformatics and Data Analysis*, March, 61–65, New York.

Yue, P., L. Di, W. Yang, G. Yu, and P. Zhao. 2007. "Semantics-based automatic composition of geospatial Web service chains." *Computers & geosciences* 33 (5): 649–665. doi:10.1016/j.cageo.2006.09.003.

Yue, S., M. Chen, C. Yang, C. Shen, B. Zhang, Y. Wen, and G. Lü. 2018. "a loosely integrated data configuration strategy for web-based participatory modeling." *GIScience & Remote Sensing* 56 (5): 670–698. doi:10.1080/15481603.2018.1549820.

Zhao, L., L. Chen, R. Ranjan, K. K. R. Choo, and J. He. 2016. "Geographical information system parallelization for spatial big data processing: a review." *Cluster computing 19* (*1*): 139–152. doi:10.1007/s10586-015-0512-2.

Zhao, P., T. Foerster, and P. Yue. 2012. "The geoprocessing web." *Computers & geosciences* 47: 3–12. doi:10.1016/j.cageo.2012.04.021.

Zhu, a.-X., J. Liu, F. Du, S.-J. Zhang, C.-Z. Qin, J. Burt, T. Behrens, and T. Scholten. 2015. "Predictive soil mapping with limited sample data." *European journal of soil science 66* (3): 535–547. doi:10.1111/ejss.12244.

Zhu, a.-X., C.-Z. Qin, P. Liang, and F. Du. 2018. "Digital Soil Mapping for Smart Agriculture: the SoLIM method and Software Platforms." *RUDN Journal of Agronomy and Animal Industries 13* (4): 317–335. doi:10.22363/2312-797X-2018-13-4-317-335.

Zhu, L. J., J. Liu, C. Z. Qin, and A. X. Zhu. 2019. "a modular and parallelized watershed modeling framework." *Environmental modelling and software* 122: 104526. doi:10.1016/j.envsoft.2019.104526.

Zhu, Y., and J. Yang. 2019. "Automatic data matching for geospatial models: a new paradigm for geospatial data and models sharing." *Annals of GIS* 25(4): 283-298.

Zhu, Y., A.-X. Zhu, M. Feng, J. Song, H. Zhao, J. Yang, Q. Zhang, K. Sun, J. Zhang, and J. Yao. 2017. "a similarity-based automatic data recommendation approach for geographic models." *International Journal of Geographical Information Science 31* (7): 1403–1424. doi:10.1080/13658816.2017.1300805.