International Journal of Applied Earth Observation and Geoinformation 124 (2023) 103520

Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

journal homepage: www.elsevier.com/locate/jag

Reproducing computational processes in service-based geo-simulation experiments

Zhiyi Zhu [a,b,c], Min Chen [a,b,c,*], Lingzhi Sun [d], Zhen Qian [a,b,c], Yuanqing He [a,b,c], Zaiyang Ma [a,b,c], Fengyuan Zhang [a,b,c], Yongning Wen [a,b,c], Songshan Yue [a,b,c], Guonian Lü [a,b,c]

[a] Key Laboratory of Virtual Geographic Environment (Ministry of Education of PRC), Nanjing Normal University, Nanjing, Jiangsu, China
[b] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing Normal University, Nanjing, Jiangsu, China
[c] State Key Laboratory Cultivation Base of Geographical Environment Evolution (Jiangsu Province), Nanjing Normal University, Nanjing, Jiangsu, China
[d] The Third Geoinformation Mapping Institute of Ministry of Natural Resource, Chengdu, Sichuan, China

ARTICLE INFO

Keywords:
Reproducibility
Geo-simulation experiments
Service-oriented strategy
Process document
Scientific workflow

ABSTRACT

Geo-simulation experiments (GSEs) are experiments allowing the simulation and exploration of Earth's surface (such as hydrological, geomorphological, atmospheric, biological, and social processes and their interactions) with the usage of geo-analysis models (hereafter called 'models'). Computational processes represent the steps in GSEs where researchers employ these models to analyze data by computer, encompassing a suite of actions carried out by researchers. These processes form the crux of GSEs, as GSEs are ultimately implemented through the execution of computational processes. Recent advancements in computer technology have facilitated sharing models online to promote resource accessibility and environmental dependency rebuilding, the lack of which are two fundamental barriers to reproduction. In particular, the trend of encapsulating models as web services online is gaining traction. While such service-oriented strategies aid in the reproduction of computational processes, they often ignore the association and interaction among researchers' actions regarding the usage of sequential resources (model-service resources and data resources); documenting these actions can help clarify the exact order and details of resource usage. Inspired by these strategies, this study explores the organization of computational processes, which can be extracted with a collection of action nodes and related logical links (node-link ensembles). The action nodes are the abstraction of the interactions between participant entities and resource elements (i.e., model-service resource elements and data resource elements), while logical links represent the logical relationships between action nodes. In addition, the representation of actions, the formation of documentation, and the reimplementation of documentation are interconnected stages in this approach. Specifically, the accurate representation of actions facilitates the correct performance of these actions; therefore, the operation of actions can be documented in a standard way, which is crucial for the successful reproduction of computational processes based on standardized documentation. A prototype system is designed to demonstrate the feasibility and practicality of the proposed approach. By employing this pragmatic approach, researchers can share their computational processes in a structured and open format, allowing peer scientists to re-execute operations with initial resources and reimplement the initial computational processes of GSEs via the open web.

1. Introduction

Reproducibility has received extensive attention across multiple scientific communities (Stodden et al., 2018; Tomasello and Call, 2011), as the reproducibility of studies embodies the reliability of their outcomes (Baker, 2016; Reinecke et al., 2022). As long as studies are reproducible, scientists in the present and future can stand on the shoulders of giants to evaluate and rely on previous research, allowing scientists to discover the truths of the world (Barba, 2018; Wilson et al., 2021).

Geography is a comprehensive discipline that investigates various aspects of the Earth's land surface, such as its structure, composition, processes, and history (Song, 2016). Geo-simulation experiment (GSE), functioning as a practical vehicle of geography, leverage numerous

* Corresponding author at: Key Laboratory of Virtual Geographic Environment (Ministry of Education of PR China), Nanjing Normal University, Nanjing 210023, China.

E-mail address: chenmin0902@163.com (M. Chen).

https://doi.org/10.1016/j.jag.2023.103520
Received 28 July 2023; Received in revised form 13 September 2023; Accepted 11 October 2023
Available online 19 October 2023
1569-8432/

resources, such as geo-analysis models (hereafter called 'models') and data, to produce outputs, typically numerical, that simulate the Earth and aid decision-making processes (Benenson and Torrens, 2004; Chen et al., 2021a; Chen et al., 2021b; Shashidharan, 2017). For example, one quintessential use of GSEs is the application of spatial analysis models to simulate urban layouts and environments, thereby yielding insights pertinent to policy-making (Qian et al., 2020; Zhang et al., 2021b; Zhu et al., 2023a). The reproducibility of GSEs not only ensures the reliability and trustworthiness of the findings but also forms a foundation for future research (Chen et al., 2023; Committee on Reproducibility and Replicability in Science et al., 2019; Kedron and Holler, 2022). This is mainly because reproducible GSEs allow peer researchers to build upon previous work with confidence, promoting the rectification of any previous errors and the potential revelation of new truths about the Earth (Barba, 2018; Goodchild et al., 2021).

Computational processes are the computational steps where researchers interact with different resources (i.e., models and data) in GSEs with computers (Little et al., 2010; Stodden et al., 2018; Waltemath et al., 2011). An example of such a computational process is the use of global climate models to predict future climatic conditions (Semenov and Stratonovitch, 2010). These processes tend to consist of a range of actions (such as data processing, model selection, parameter adjustment, and data validation), which are integral to the implementation of GSEs (Konkol et al., 2020; Ma et al., 2022). The reproducibility of computational processes refers to the ability to achieve consistent outputs with the same resources, which is the same as the definition of computational reproducibility of GSEs. The reproduction of computational processes lays the foundation for the reproduction of GSEs, given that GSEs are actualized through the computational processes inherent in different models (Chen et al., 2021b; Konkol et al., 2019; Stodden et al., 2018). The necessity for the reproducibility of GSEs has sparked efforts to promote the reproducibility of computational processes (Reinecke et al., 2022; Richardson et al., 2015).

The reproducibility of computational processes requires that the involved resources be fundamentally accessible and executable (Bolukbasi et al., 2013; Essawy et al., 2020; Gil et al., 2016). However, this requirement faces three typical obstacles. First, despite concerted efforts to share these resources, ensuring their accessibility remains a daunting task for many scientists (Choi et al., 2023; Reinecke et al., 2022). The difficulty often stems from issues such as proprietary rights, the management of large file sizes, and other assorted restrictions (Merz et al., 2020; Wilkinson et al., 2016; Wilson et al., 2021). Second, executing shared model resources poses another set of challenges due to the dependence on a specific software environment (such as an operating system, programming language version, or library package) (Davison, 2012). If the exact environment cannot be faithfully recreated, it can pose significant obstacles to the direct use of shared models and hence inhibit reproducibility (Knoth and Nüst, 2017; Nüst et al., 2020). Third, the configuration of resources plays a critical role in the accurate execution of models (Yue et al., 2019; Zhang et al., 2021b). If this is overlooked, models might fail to execute correctly due to data configuration errors within the model.

Recent advancements in computer technology, coupled with the push for increased openness and transparency in science, have spurred the development of service-oriented strategies (Lorscheid et al., 2012; Merz et al., 2020; Zhang et al., 2021a). These strategies, which encapsulate model resources in different forms (such as executable files and Python files and modules in software) as model-service resources online, promote the sharing, accessibility, and reuse of model resources (Yue et al., 2016; Zhang, 2019). These strategies lower the high technical barriers faced by researchers and other stakeholders, allowing them to concentrate on the transformation of dataflow through the models (He et al., 2023).

While service-oriented strategies offer promising solutions for reproducing computational processes through model-service resource sharing and reuse, they typically focus on the execution of individual model-service resources to generate outputs (Nüst and Pebesma, 2021; Wilson et al., 2021; Zaragozí et al., 2020). However, in reality, a typical GSE likely involves the use of multiple interlinked resources rather than reliance on a single model in isolation (Zhang et al., 2023). Current strategies ignore the associations and interactions among researchers' actions regarding the use of sequential resources (Chen et al., 2019; Ma et al., 2022; Zhang et al., 2021b). Ignoring the interconnected nature of resource utilization can lead to reproducibility failures.

In this study, an approach for reproducing computational processes by tracking researchers' actions (behavior of researchers when implementing different resources) online on the basis of the current service-oriented strategy is proposed. To better organize the computational processes, the actions in these processes are described as a collection of action nodes and corresponding logical links (node-link ensembles). An action node is the fundamental unit of a computational process and helps map the interactions between participant entities and resource elements (i.e., model-service resources and data resources), while logical links represent the logical relationships between action nodes. Based on this framework, the proposed approach encompasses three primary stages: the representation of actions with node-link ensembles, the formation of documentation with the operation of these actions, and the reimplementation of process documents based on a *vertex-edge* model. Notably, the accurate representation of actions promotes the correct execution of these operations, thereby facilitating the documentation of these operations; thus, this standardized documentation can be reproduced by visualization with a *vertex-edge* model. In this context, computational processes can be expressed by the specific dataflows, which can also be reproduced by peer scientists. Building upon this approach, a prototype system is designed to demonstrate the feasibility and practicality of the proposed approach, as illustrated by a case study. Through this approach, researchers can share their computational processes in an open and structured manner, while peer scientists are able to re-execute the initial resources and reimplement the initial computational processes of GSEs.

The remainder of this paper is organized as follows. Section 2 presents the fundamental foundation, organization, and framework of the proposed approach. Section 3 introduces details about the representation of actions, the formation of process documents, and the reproduction of process documents. The implementation of the proposed approach for a prototype system, along with one case study, is presented in Section 4. Finally, Section 5 provides the discussion and conclusion of the study.

2. Basic idea

This study mainly focuses on the track of researchers' actions by utilizing model-service resources in an ordered manner. Open Geographic Modeling and Simulation (OpenGMS, https://geomodeling.njnu.edu.cn/) is a well-known platform focusing on the sharing and reuse of models on the web (He et al., 2023). The OpenGMS service-oriented strategy is a typical method used to wrap models as components and then publish them as services on the open web (Chen et al., 2020). Inspired by this service-oriented strategy, the foundation of the proposed approach is the encapsulation of model resources. In addition, a method of organizing computational processes to show their internal content and structure is proposed, and a framework is built to facilitate the sharing and reproduction of computational processes on the open web.

2.1. Foundation of the proposed approach

The OpenGMS service-oriented strategy consists of the following four steps:

(1) First, users need to describe geo-analysis models via the Model Description Language (MDL). The MDL is a conceptual model that

abstracts and summarizes the execution process of a model as a set of *states* and the input or output data processing messages as *events* (Yue et al., 2016). The two main kinds of *events* correspond to the model's input and output information. Additionally, determining the correct parameters is crucial for successful operation; therefore, in this study, a parameter attribute is added to the *event* set to clearly present and use the model parameters.

(2) The second step is encapsulation. This step involves "wrapping" and "packaging" heterogeneous models into a standardized and structured format, which makes them easier to handle, share, and reuse.
(3) Third, in terms of deployment, the encapsulated models should be uploaded to the OpenGMS Wrapper System, a lightweight tool for model storage and online execution (Zhang, 2019). The OpenGMS Wrapper System is configured with the required runtime environment for the model, ensuring that models will be executed properly when requested.
(4) The final step is publishing. This makes the encapsulated models publicly available by exposing them to the open network environment, usually through a unique identifier. This accessibility allows other researchers and stakeholders to access and execute the models, fostering collaborative research and knowledge sharing (Zhang et al., 2021b).

By using the OpenGMS service-oriented strategy, models can be effectively wrapped into model-service resources that are easily accessible and executable by researchers and stakeholders, with corresponding outputs.

2.2. Organization of the proposed approach

The computational processes in this study are characterized as sequences of implementation actions that extensively utilize model-service resources and data resources, as shown in Fig. 1. These implementation actions can be viewed as a series of action nodes and their logical links (node-link ensembles).

The action node is the fundamental unit of the computational process and helps map the interactions between participant entities and resource elements, representing actions such as data preprocessing, model execution, parameter adjustment (Koo et al., 2020), and result analysis (Ma et al., 2021). Each action node wraps a distinct action or operation that is generally independent of others and can be implemented as a standalone task. In the context of the OpenGMS service-oriented strategy, these action nodes can correspond to the utilization of specific model-service resources.

A logical link (hereafter called a 'link') represents a logical relationship between action nodes, outlining the sequence or flow of actions in the computational processes. These signify the direction and order of the computational process, indicating which action node follows another. They can represent dependencies between action nodes, with one action node needing to be completed before another can begin, or they can indicate parallel action nodes for actions that can be carried out simultaneously. In some cases, logical links might also represent conditional relationships, where the outcome of one action node determines the next action taken.

This approach of mapping the computational process as a series of action nodes and their links provides a clear structured understanding of the computational processes. It can also improve transparency and facilitate the tracking of researchers' actions.

2.3. Framework of the proposed approach

Our study targets the reproduction of computational processes in GSEs by leveraging model-service resources and abstracting actions into node-link ensembles. To facilitate online sharing and the reproduction of actions, we propose a conceptual framework (shown in Fig. 2) consisting of three core components: the organization and presentation of actions with node-link ensembles; the formation of standardized documentation with the action operations; and the reimplementation of process documents based on a *vertex-edge* model.

(1) When trying to recognize implementation actions, it is crucial to accurately identify the action nodes and links involved in detail, which represent the basic content and structure of these actions. This study explores the related entities, elements, and interactions (in Section 3.1) to better understand and express these node-link ensembles. By appending the relevant details, researchers' behaviors can be abstracted and communicated in a clear and manageable way. This is an essential step for the subsequent stages of presenting and reproducing these actions.
(2) When applying certain resources, the process of formation and implementation of the organized actions should be documented and standardized. Through a visual or descriptive representation, the operation cycle of these actions can be recorded as a task-instance ensemble. To support the representation, the process document, an automated, standardized record of computational processes (Section 3.2), is designed.
(3) With dataflows, the process document should be expressed and visualized to be logically rational before reproduction by peer scientists. To facilitate this approach, a *vertex-edge* model is designed to facilitate documentation visualization and reimplementation (in Section 3.3). In the dataflow of this model, the relationships among different resources can be formulated and represented in a structured manner.

3. Detailed designs

3.1. Representation of actions with node-link ensembles

Actions in GSEs can be represented as node-link ensembles, where action nodes indicate a collection of entities, elements, and their interactions, and links refer to the various relationships among action

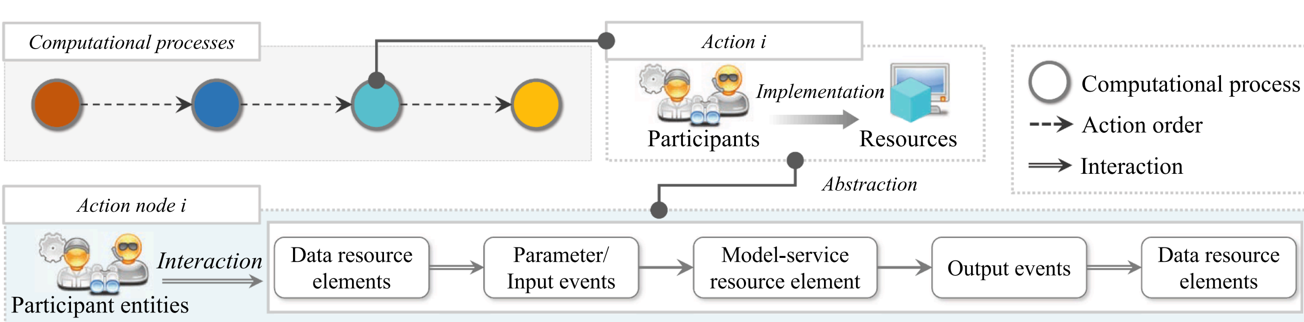

Fig. 1. An example of one action and its abstraction.

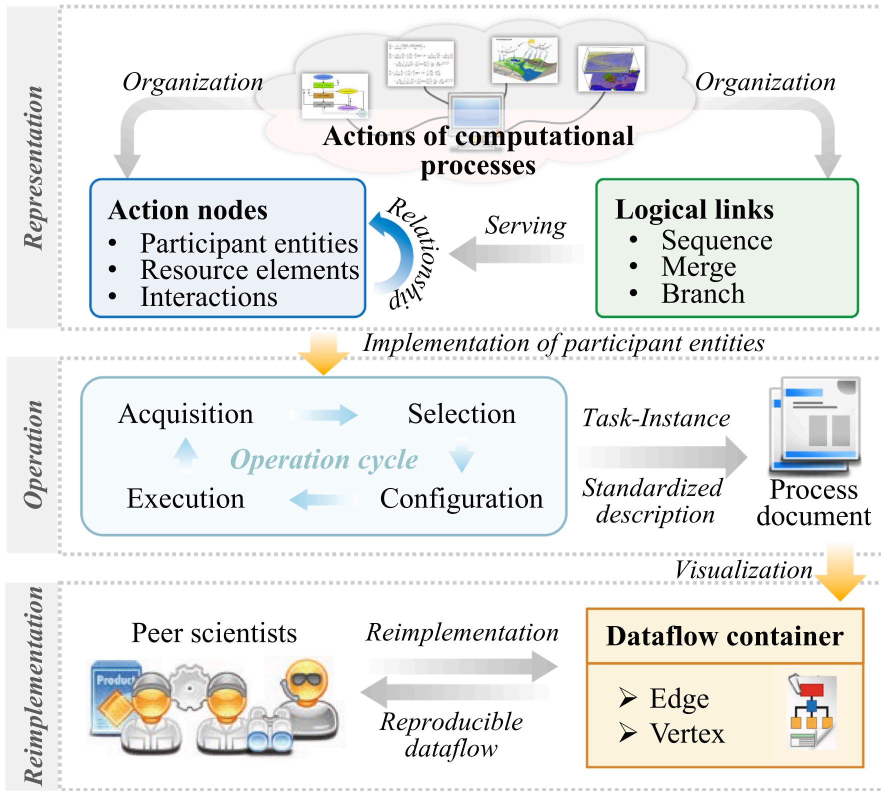

Fig. 2. Framework of the proposed approach.

nodes. Thus, in this section, the characteristics of action nodes and the features of links are introduced to clarify the detailed content and structure of node-link ensembles.

3.1.1. The characteristics of action nodes

Entities, elements, and their interactions are three components of action nodes. Specifically, entities represent participant entities (hereafter called participants) that are individuals, such as developers, stakeholders, decision-makers, or any other participants, involved in performing a computational process (Chirigati et al., 2013). Elements are model-service resource elements and data resource elements employed and generated in the experiment. Model-service resource elements (hereafter called model-service resources) are quantitative models or tools involving a set of equations or rules to simulate and analyze Earth surface processes, which are encapsulated as web services online based on the OpenGMS service-oriented strategy (Badham et al., 2019; Chen et al., 2021a; Zhang et al., 2021a). Data resource elements (hereafter called data resources) involve relevant data used and produced at each action node, including input data, parameters, intermediate outputs (temporary datasets generated during the process), and final outputs (Ma et al., 2022; Zhang et al., 2021a). In addition, there are some relationships between participant entities and resource elements at one action node: (1) a one-to-more relationship executed by one participant entity utilizing several resource elements (Fig. 3 (a)), and (2) the relationships between data resources and events related to one model-service resource (Fig. 3 (b)).

Interactions encompass the dynamic processes used to establish connections between the participant entities and resource elements, thereby facilitating the exchange of information or other relevant factors. These interactions can be categorized into two distinct types: (i) participant-resource interactions and (ii) resource-resource interactions. The detailed descriptions are as follows.

Participant-resource interactions. Such interactions include a set of operations by participants entailing the selection, configuration, execution, and acquisition of resources. (1) Selection is performed to choose the appropriate resources for analyses based on the given scientific questions and research objectives. (2) Configuration is the process of setting the initial parameters and data resources as inputs for the execution of model-service resources. This sets the foundation for discussions or explanations about how different models are configured or how various parameters and data inputs influence the outputs. (3) Execution is the behavior of participants invoking model-service

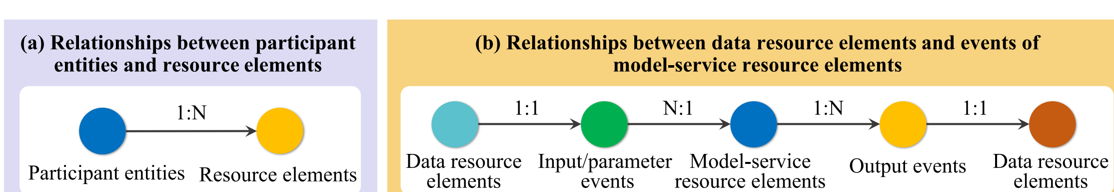

Fig. 3. Two types of dependencies through entities and elements. Fig. 3 (a) demonstrates that at each action node, a single participant entity (hereafter called a participant) may rely on or interact with multiple resource elements, highlighting the multifaceted relationship between the participant entity and resource elements. Fig. 3 (b) highlights that every model-service resource element (hereafter called model-service resource) is linked to multiple input/output/parameter events, and each of these events is specifically tied to a singular data resource element (hereafter called a data resource).

resources to perform calculations. Once the resources have been selected, participants need to run these model-service resources, applying them to the selected input data. (4) Acquisition involves acquiring the necessary output data resources of the model-service resources for the next actions, for final output analysis, or just accessing them on the local machine. Local access to resources allows participants to work offline, which can be beneficial in situations where an internet connection is unavailable or intermittent.

Resource-resource interactions. Such interactions refer to the interplay between data resources and model-service resources. There are two main types of this kind of interaction. First, existing data resources are prepared and filled with the appropriate information to ensure that input data resources are correctly positioned as input events to model resources. Second, output data resources generated from model-service resources need to be adapted and placed in the appropriate position in series of output events. A concise example of the multiscale geography weighted regression model provides views on how events in one model are related to resources (Fotheringham et al., 2017), as shown in Fig. 4.

3.1.2. The features of logical links

Links are used to organize the relationships between action nodes, tracing from the output resources of one previously executed action node to a subsequently executed action node. From the perspective of the participants, all the action nodes can be organized in a sequential order, reflecting the logical flow of their actions. That is, the action nodes are executed one by one in one direction. Sequence, merge, and branch are all unique types of links that represent different logical relationships between action nodes (Fig. 5). The details of these links are as follows:

Sequence. This type of link represents a linear, sequential order of execution actions, where one action node follows another. In a sequential relationship, the output data resources from one action node become the input for the next one, and the action nodes are executed one after the other. Sequences represent a straightforward progression from one action node to the next. They indicate a direct linear flow from one action to the subsequent action, illustrating the chronological order of tasks performed by participants.

Merge. This is a type of link for which multiple action nodes (two or more) converge into a single subsequent action node. This structure signifies that the outputs from several action nodes are combined and input into the next node. This process involves a specific type of sequence in which multiple parallel sequences are merged into one.

Branch. A branch dependency occurs when a single action node splits into multiple parallel action nodes (two or more). In this case, the output of one action node becomes the input for two or more other nodes, which are then executed concurrently. Similar to a merge, a branch is a special case of a sequence in which one action node leads to multiple subsequent nodes.

3.2. Formation of documentation for the operation of actions

When participants start to interact with several resource elements, an operation cycle in one action node can be described as shown in Fig. 6, beginning with (1) resource selection, (2) resource configuration, (3) resource execution, and (4) resource acquisition. To effectively map the operation cycle in action nodes, a conceptual model that maps these operations as transitioning from a task (pre-execution status) to an instance (post-execution status) is developed, as shown in equation (1).

$$\textit{Operation cycle} \sim < \textit{Task}, \ \textit{Instance} > \tag{1}$$

A task is a single unit that represents a model-service resource that needs to be simulated and generates corresponding outputs as well as the configuration of input events. A task is also in a waiting state before resources are executed. When participants exploit action nodes to represent the structure and components of actions, several tasks are formed, each corresponding to a specific action node.

A task corresponds to the status of a specific action node before its execution, and an instance corresponds to the status of an action node after the task has been executed. When participants start to execute the tasks, instances are formed, reflecting the actual implementation of the computational process. Instances encompass information about the inputs, outputs, and settings used during the execution of a task, providing valuable insights into the implementation action.

3.2.1. Establishment of a task and instance

Once resource elements are selected and implemented at an action node, a task is formed, which can also be regarded as the template of the chosen model-service resources and the configured data resources. The related resource elements and interactions involved are encapsulated as a basic unit used to carry out a specific task.

Given the potential for subsequent implementation and reimplementation, the method must include a systematic way to document and

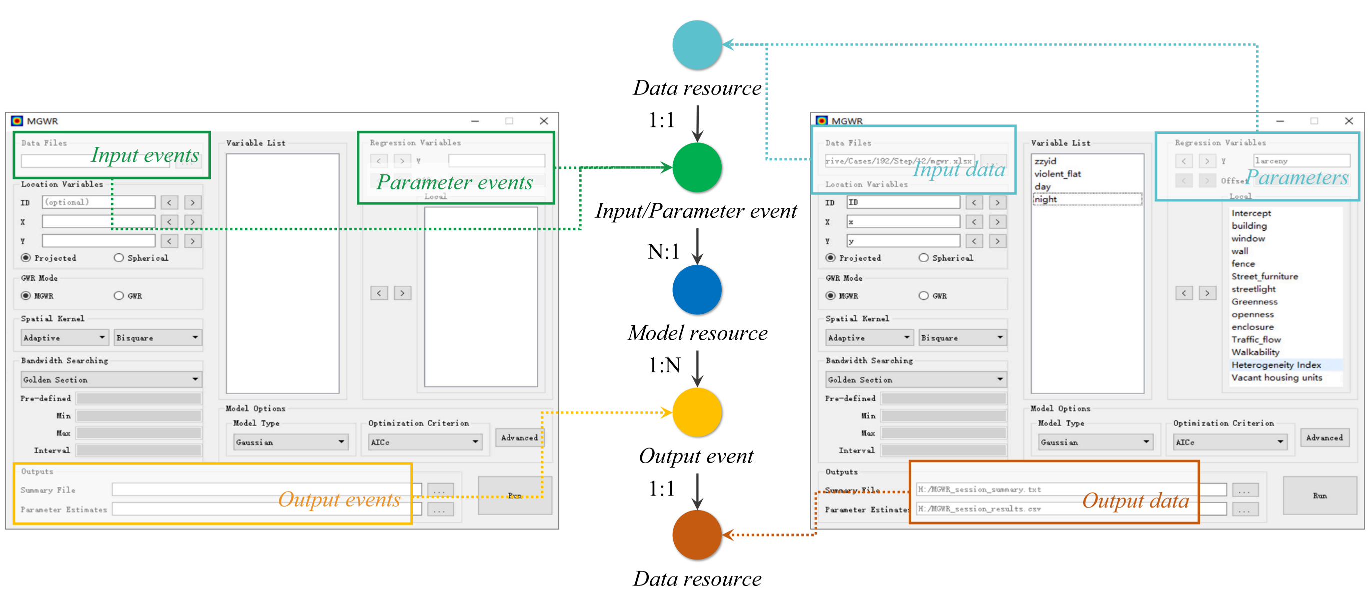

Fig. 4. Example of correspondence between events in one model and data resources. In this visualized model, several input events correspond to input data resources, parameter events correspond to the values of parameters, and two output events are related to the output data resources.

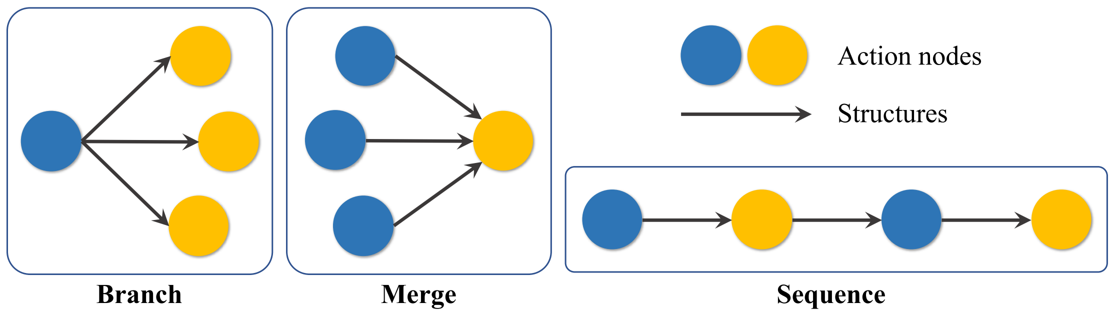

Fig. 5. Three types of link structures.

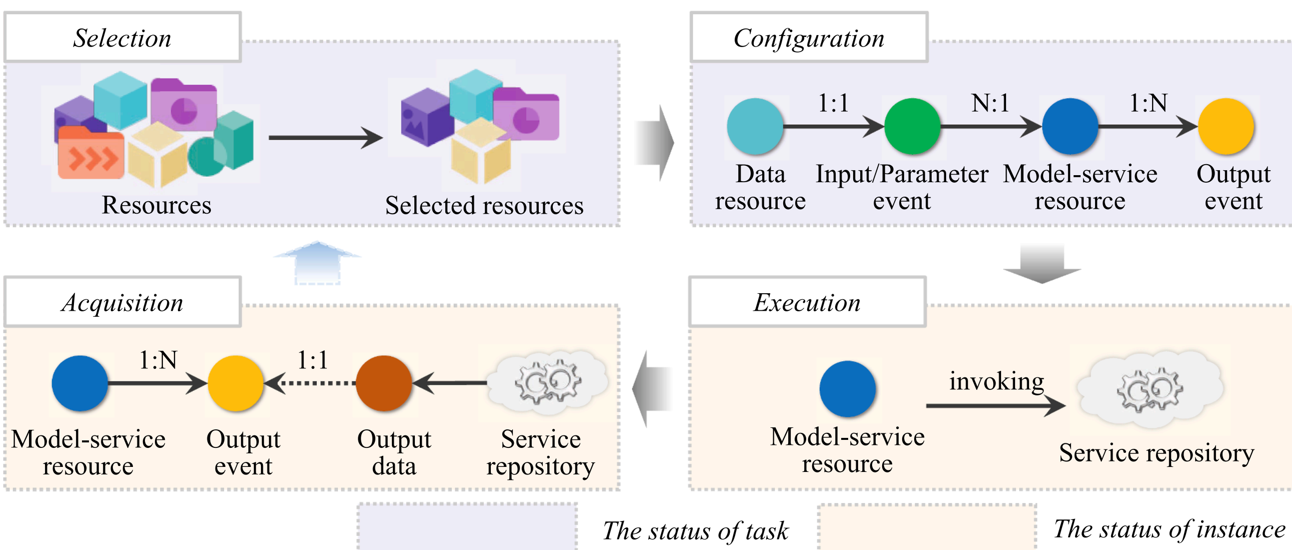

Fig. 6. An operation cycle in action nodes.

track these tasks. While metadata are the ideal choice for recording such information, it is still difficult for participants to record all the needed information. Thus, the recommendations provided for the GSEs in reference descriptions can be used to specify which information needs to be noted about tasks (Zhu et al., 2023b). The standardized description aims to capture critical details about the task, including the resource elements employed, parameters used, results generated, and any other important details about the execution of the task. This description consists of three primary parts (green parts in Fig. 7): general information, character information, and agent information, as illustrated in Fig. 7. Specifically, general information provides an overview of the resources used. Agent information identifies the individuals and agencies responsible for the resources, indicating their provenance (Chirigati et al., 2013; Fotheringham and Sachdeva, 2022). Character information comprises resource-specific details that vary depending on the type of resource.

An instance is formed after the start of the execution of tasks, which results in the generation of specific outputs. There are several steps required to achieve the desired outputs through configured resources between task requirements and the service repository, where model-service resources can be stored, managed, and executed. Fig. 8 illustrates the execution steps, which can be summarized as follows: 1) querying the model-service resources to achieve the model-service resource's behaviors; 2) querying the configured data resources and invoking the model-service resources; 3) computing the model entity and polling the output data; and 4) finalizing the calculation and obtaining the corresponding output data.

3.2.2. *Construction of process documents*

The process document is a standardized carrier used to record the operation cycle of actions, and it can be built automatically once instances are completed and bound to be reproduced by peer scientists. The process document captures the sequence of operations carried out during the experiment, from task formation to instance completion. This document is made available to both researchers and the platform to help them understand the implementation of action nodes.

Based on the referenced descriptions and the implementation of the operations, four main components can be seen in the top side of Fig. 9. (1) *General information.* General information refers to an overview of what an action node is and why it must be implemented, such as title, order, and type information (Gil et al., 2016). (2) *Agent information.* Agent information indicates the provenance of responsible participants (Chirigati et al., 2013). (3) *Implementation information.* Implementation information specifies task information and instance information about one action node. (4) *Dependency information.* Dependency information defines the logical structure of each action node.

To describe these attributes in a structured manner, the eXtensible Markup Language (XML) format is used for this document, as it is cross-platform and easy-to-exchange (Chen et al., 2019); this is shown in the bottom side of Fig. 9.

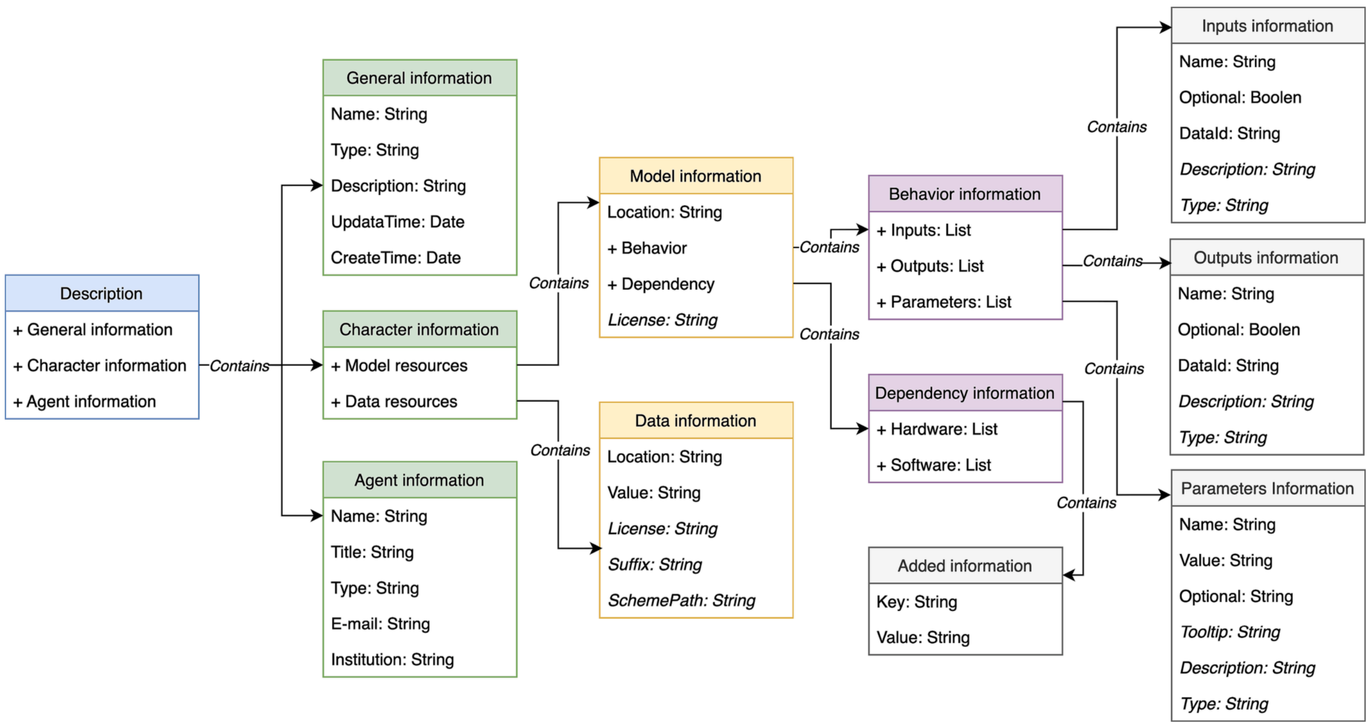

Fig. 7. Basic design of the standard description. In this design, different colors represent different hybrids. Italics indicate nonessential information.

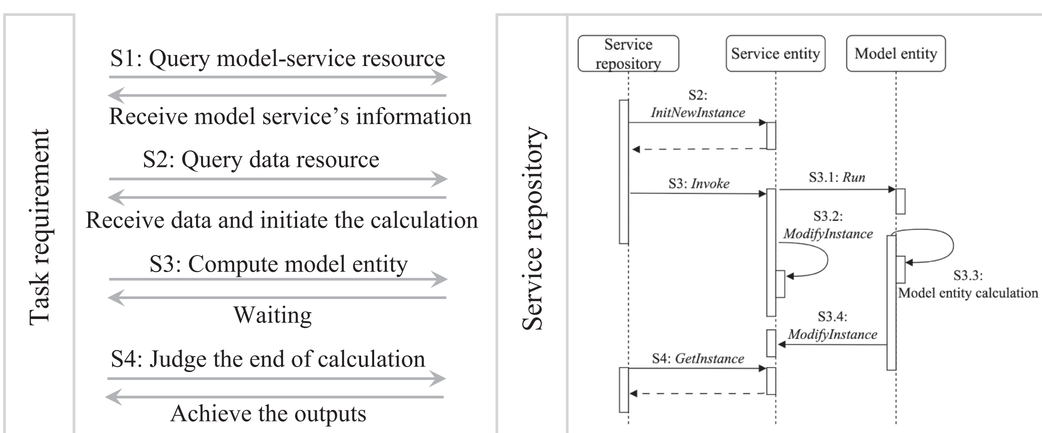

Fig. 8. Steps of executing a model-service resource based on requirements and the service repository.

3.3. Reimplementation of the process documents based on the vertex-edge model

3.3.1. Visualization of the dataflow

A scientific workflow is an effective means of representing different instances in an abstract graph (Beaulieu-Jones and Greene, 2017; Cerutti et al., 2021; Essawy et al., 2020). MxGraph has emerged as an effective tool through scientific workflows involving the creation of interactive graphs and charting applications that run on the web (https://jgraph.github.io/mxgraph/). It serves as the foundation for representing a multitude of complex structures, especially in applications in which there is a need to visualize connections, dependencies, and hierarchies. Inspired by MxGraph, a *vertex-edge* model is proposed in this context (as shown in equation (2)), focusing specifically on the visualization and reimplementation of computational processes based on the relevant process documents by creating a series of workflows. This model emphasizes the chaining of resources in a logical sequence, ensuring that the dataflow through computational processes remains consistent.

$$Process\ document \sim \{Vertex,\ Edge\} \tag{2}$$

The *vertex-edge* model includes five types of vertices: (i) input vertices, (ii) parameter vertices, (iii) intermediate output vertices, (iv) output vertices, and (v) model vertices. Specifically, the input vertices indicate the input event and the configured input data resources associated with an action node. The parameter vertices indicate the parameter event and its configuration. The intermediate output vertices indicate the output event used as the input event at another action node. The output vertices indicate the raw output events of a model. The model vertices indicate the model-service resources involved in an action node, such as the Soil & Water Assessment Tool (SWAT, https://swat.tamu.edu/), the Soil Water Atmosphere Plant (SWAP, https://swap.wur.nl/), and the unstructured grid Finite Volume Community Ocean Model (FVCOM, https://www.fvcom.org/). This *vertex-edge* model also includes edges to show the transformation relationships between vertices, with each edge arrow pointing in one direction.

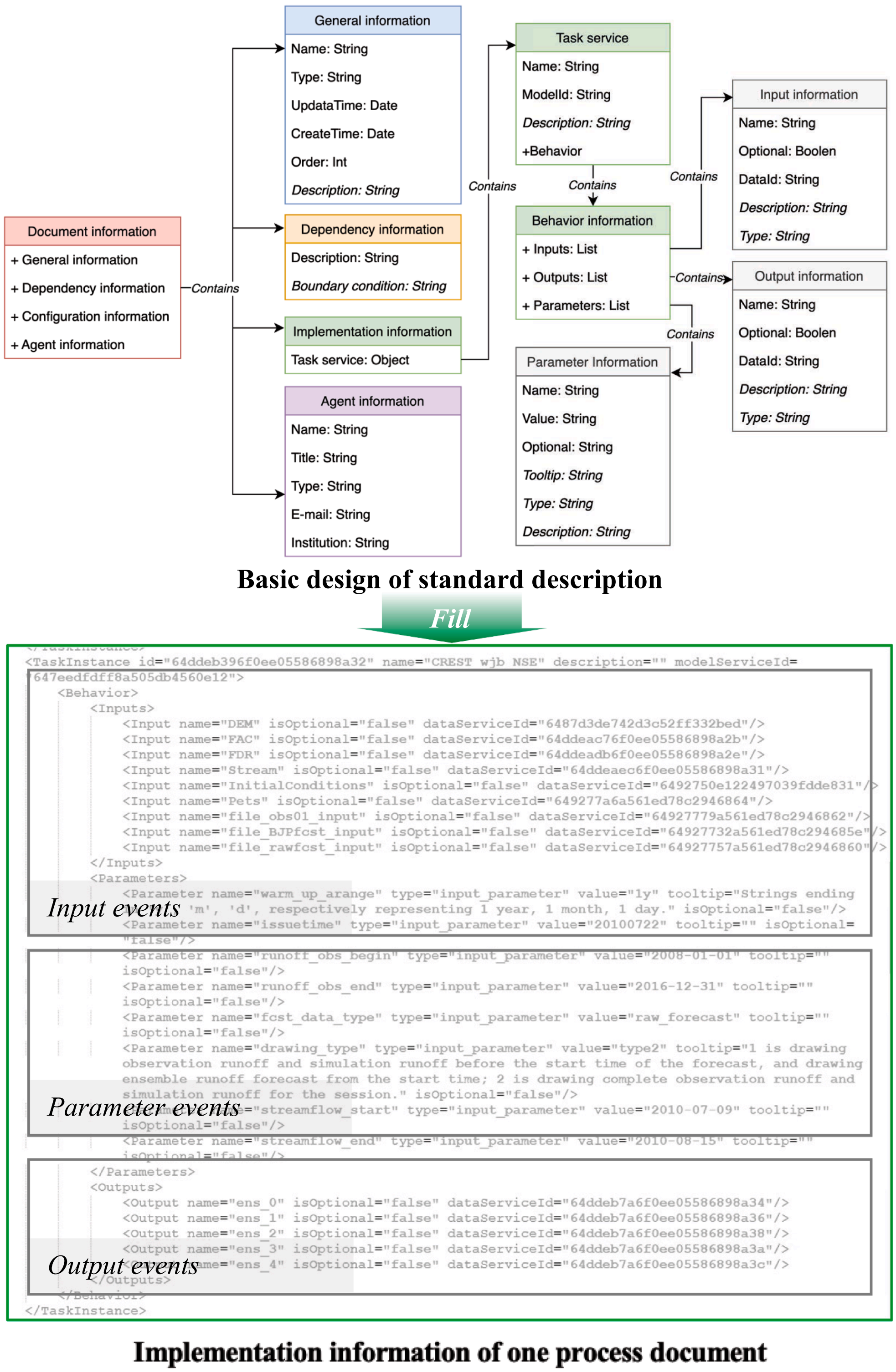

Fig. 9. Basic design and an example of documentation. In this design, different colors represent different hierarchies. Italics indicate nonessential information in the design.

Based on this model, a collection of instances can be mapped to a series of vertices with edges. In short, one instance contains one model vertex, several input vertices, parameter vertices, and output vertices with the initial data resources. Fig. 10 shows one example of a dataflow mapped based on instances. By combining this method with a scientific workflow based on the representation and organization of instances, each task in the original instance can be re-executed with initial data resources as re-executed new instances. Moreover, these regenerated outputs can be reused as inputs in other tasks.

3.3.2. Evaluation of reproducibility

An evaluation of the reproducibility of computational processes should be explicitly based on the evaluation of GSE reproducibility (Zhu et al., 2023b). In this study, the evaluation process involves determining

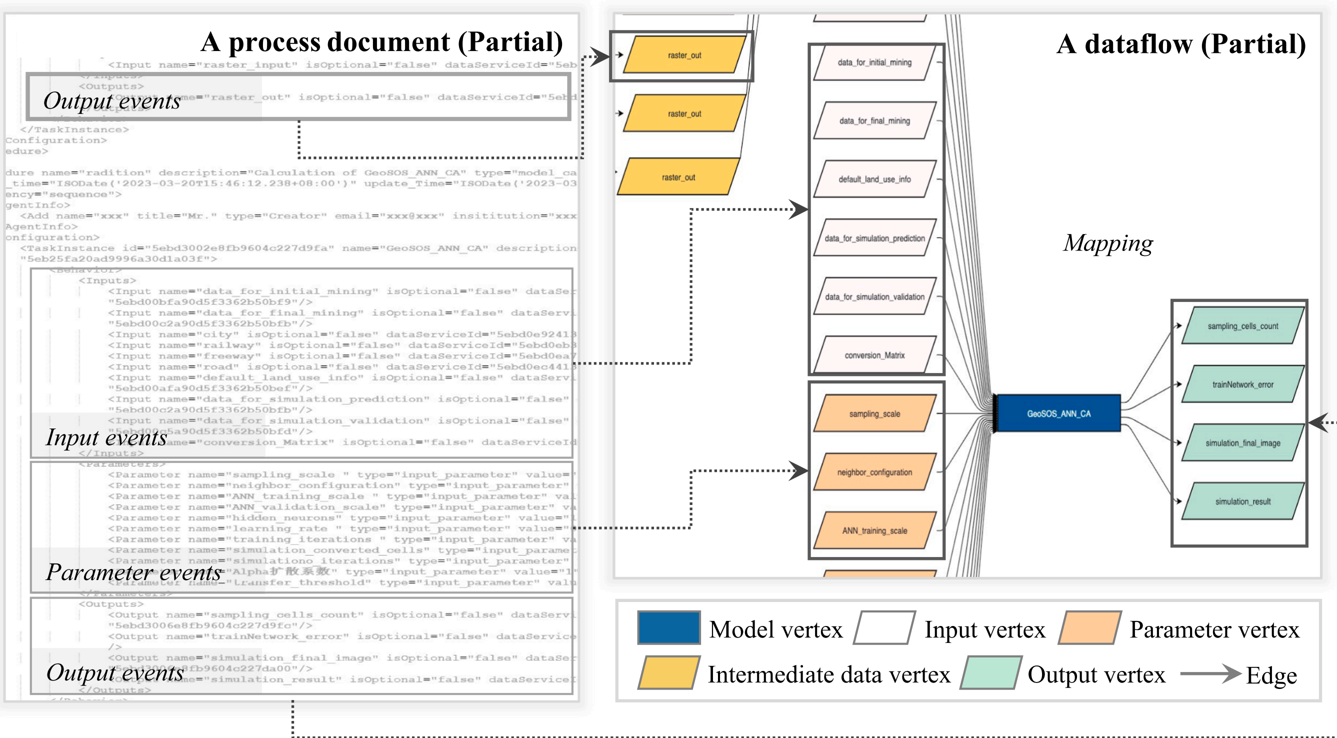

Fig. 10. Example of a partial mapping dataflow. The schematic showcases various blocks, each assigned a distinct color to represent different model, input, parameter and output vertices. The edges between each vertex depict the dependencies between them, and initial resource elements are encompassed by several vertices.

whether can be obtained for the same tasks, thereby forming corresponding instances across different participants. This study defines two levels of reproducibility in the context of the organization of computational processes. This classification approach helps participants understand the extent to which particular computational processes can be reproduced and provides a basis for evaluating and improving reproducibility in the field.

- *Reproducible computational processes*. This level of reproducibility implies that measured action nodes can be reimplemented, resulting in consistent outputs.
- *Irreproducible computational processes*. At this level, measured action nodes cannot be re-executed, or the consistent outputs cannot be achieved.

Statistical indicators can be used to quantify the magnitude of a difference to some extent (Open Science Collaboration, 2015). This study exploits the R-squared value (R^2), as well as the statistical indicators like the mean absolute error (MAE) and mean square error (MSE), to measure the consistency of outputs. MAE and MSE are used to evaluate the differences between the actual (initial output data) and predicted values (reproduced output data) (Chicco et al., 2021; Khajetoorians and Wiebe, 2014). R^2 is a statistical measure that can be used in regression analysis to assess the goodness of fit between initial output data and reproduced output data (Nakagawa et al., 2017). Thus, the MSE and MAE offer insight into the average magnitude of errors between predicted and observed values, and R^2 gives an indication of how closely the data fit the original regression line (Hodson, 2022).

4. Experiment

4.1. Design of the prototype system

In this study, a prototype system is established to validate the feasibility and effectiveness of the proposed approach in a practical way. The design of the prototype system is illustrated in Fig. 11, where the project center, user center, and resource center are the three main parts of the system.

The project center is designed to facilitate the implementation of computational processes in GSEs, enabling the creation of reproducible projects consisting of a single workspace, multiple tasks, and numerous instances. The workspace provides an environment for the formation of different tasks and the completion of instances. Upon creating a project, participants can utilize selected model-service resources within the workspace. When a model-service resource is chosen, a task is automatically formed. Execution of this task generates an instance, yielding outputs. To streamline the interactions among resource elements and their integration with other applications on the open web, the execution processes of task requirements are exposed as application programming interfaces (APIs), as illustrated in Table 1. After the execution of each task, when the related instances of each action node are bound, the process document can be formed, and computational processes can be visualized and reimplemented by peer scientists.

The resource center is responsible for organizing, collecting, and managing resource elements for computational processes in GSEs (Fig. 11). It comprises data and model-service resource repositories that store and organize these resources using unique identifiers. The resource center enables geoscientists to share and access existing data and model-service resources. Researchers can use all available resource elements in the project center to create a series of computational processes for solving geo-simulation problems. Peer scientists can also access and reproduce the initial model-service resources. The resource center provides a channel for both participants and peer scientists to upload, download, and invoke resource elements. The operations of containers include (a) resource upload, (b) resource download, (c) resource deletion and (d) resource execution.

The user center is a vital component in the prototype system and is responsible for managing all users involved through the construction

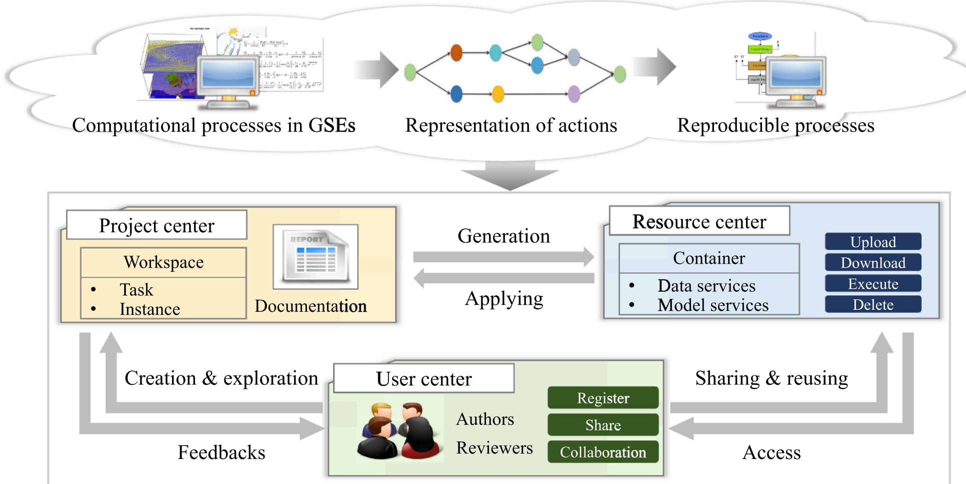

Fig. 11. Design of the prototype system.

Table 1
Interface of task requirements.

Router	Request	Operating instructions
/{type}/:id/getBehavior	GET	Obtaining the behavior of model-service resources in one task
/{type}/initNewInstance	POST	Initializing a new instance
/{type}/:instanceId/invoke	PUT	Invoking a task to form an instance
/{type}/:instanceId/modify	PUT	Modifying an instance
/{type}/:instanceId	GET	Querying an instance

and reproduction of computational processes. All the users are able to create and share their own projects in the project center and explore and reuse available resource elements in the resource center. The two primary user types in the system are participants and visitors. Participants are key stakeholders who contribute to various aspects of the project, such as developers, teammates, and decision-makers. Visitors, on the other hand, are researchers who visit the projects of others and can gain inspiration for their own research. Overall, the user center plays a critical role in facilitating collaboration, knowledge sharing, and reproducibility within the GSE computational process framework.

Within this system, researchers can share the reproducible computational processes of a GSE by constructing a reproducible project. In addition, peer scientists can visit and explore this project by reimplementing shared tasks and instances. All the codes of the prototype system are available in GitHub at the links below:

(1) https://github.com/orangemimi/reproducibility-opengms-front.
(2) https://github.com/orangemimi/reproducibility-opengms-back.

4.2. Case study

The Wangjiaba basin is in the upper part of the Huaihe River basin. This area is characterized by steep terrain, fast-flowing water, and concentrated precipitation during the rainy season, which makes flood resistance at the Wangjiaba gates very important. This case study is a hydrometeorological ensemble forecasting study of the Wangjiaba catchment supported by the proposed prototype system to simulate the probability distribution of runoff from 22 July to 2 August 2010. This reproducible experiment provides probability distributions for the occurrence of future flood events and helps decision-makers better understand the uncertainties and possible risks of flood events. Furthermore, it is possible to provide early awareness of flood events, which can help people prepare for and respond to these events more effectively.

As shown in Table 2, there are seven related models, including the fill sink, flow direction, flow accumulation, and coupled routing and excess storage (CREST) models. These models are represented in different formats, such as Python files, executable files, and software modules.

4.2.1. Action nodes for online sharing

First, these models were wrapped, deployed, and published online based on the OpenGMS service-oriented strategy. After encapsulation, the model resource packages were deployed and published in the

Table 2
Models involved in this case study.

Name	Description	Format
Fill sink	This model is used to fill the "sinks" or "pits" in a digital elevation model (DEM), as they might occur due to noise or errors in the data.	Module in ArcGIS Desktop
Flow direction	After the sinks are filled, this model can be used to create a map that represents the direction of flow from each cell in the DEM to its steepest downslope neighbor.	Module in ArcGIS Desktop
Flow accumulation	This model is used to accumulate the number of cells that flow into each downslope cell, which represents the contributing area for that cell.	Module in ArcGIS Desktop
River extraction	This is a raster calculator model used to extract river network data from the DEM, whose entity is the essentially formula.	Module in ArcGIS Desktop
TiffToAsc	This is a data processing model used to convert data from the Tagged Image File Format (TIFF) to the American Standard Code for Information Interchange (ASCII) format.	Python
AscToTiff	This is a data processing model used to convert data from ASCII format to TIFF format.	Python
Coupled routing and excess storage (CREST)	The CREST model is a hydrological model developed to simulate the spatiotemporal variations in water and energy fluxes and storages based on cell-to-cell simulation. It was developed by the University of Oklahoma in cooperation with NASA Goddard Space Flight Center scientists (https://servirglobal.net/Regions/E-S-Africa/Articles/Article/510/coupled-routing-and-excess-storage-crest-model).	Executable file

OpenGMS Wrapper System, which can be accesses in the resource center. Moreover, the datasets for the simulation contain valuable information regarding Wangjiaba, including elevation information in the digital elevation model (DEM) and initial conditions such as the initial soil moisture level and potential evapotranspiration (PET). These related datasets were also uploaded to the resource center in preparation for the implementation.

Second, to enable the construction of computational processes,

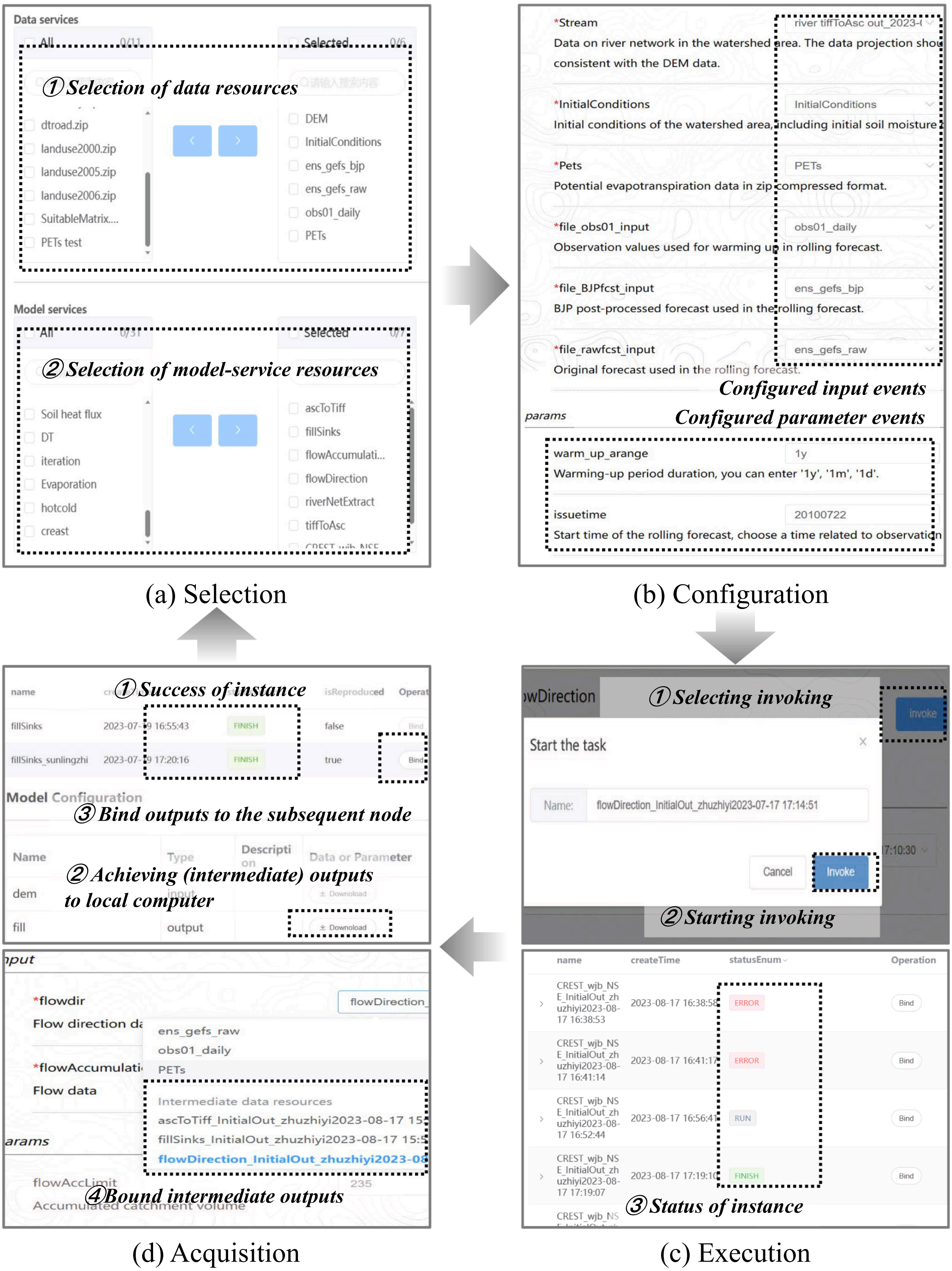

Fig. 12. The actions of participants.

researchers developed a project, dubbed "Hydrometeorological ensemble forecasting study of the Wangjiaba", within the prototype system. In the workspace of this project, the dataset and model-service resources from the resource center can be selected to execute the subsequent operations.

Third, the researchers started to perform actions on these resource elements in this workspace, as shown in Fig. 12. The researchers chose all the needed data resources and model-service resources first (Fig. 12 (a)). Then, the model-service resources, as well as the needed data resources, were selected to obtain the configuration later. In addition, the data resources were configured for the appropriate input or parameter events to form different tasks (Fig. 12 (b)). Then, the researchers began the execution of different tasks based on an invoking operation (Fig. 12 (c)). The execution of these model-service resources yielded numerous intermediate and final outputs. After the execution of the tasks, several instances were obtained with a "fail" or "success" status, and the results could be visualized by visitors (Fig. 12 (d)). Notably, all intermediate and final outputs generated from this project were also made available for reuse within the project. For instance, the researchers began by applying the *fill sink* model to the DEM to remove any depressions, as illustrated. The output data from this bound instance were then be passed to the next model, namely, the *flow direction* model.

Finally, after the selection and binding of instances, the prototype system automatically captured the researchers' actions presented in a process document. The entire process document could be shared with other researchers or stakeholders through the project center.

4.2.2. Tasks for peer scientists to reproduce online

When accessing the project, visitors can review the specific page of this case study (Fig. 13). The resource collection, service instances, and simulation dataflow are the three primary parts of content. The resource collection provides a comprehensive list of every resource element used within the dataflow. For the sake of reproducibility and further research, these data resources are made available for download. This allows visitors not only to reproduce the study with the same resources but also to reuse these resources for their own related projects. Service instances are instances selected and bound for display by participants. The simulation dataflow is a dataflow containing several instances that can also be re-executed by visitors. Peer scientists can not only review the initial outputs generated by the researchers but also re-execute these tasks to generate reproduced outputs.

To validate the authenticity and reproducibility of the case study, the original tasks were re-executed with the same data resources and model-service resources, and the reproduced outputs were generated (Fig. 14). Moreover, all the intermediate and final outputs were checked by calculating the MSE, MAE, and R^2 to evaluate the consistency between the reproduced and initial outputs. Based on MSE = 0 and MAE = 0, we found that every value in the reproduced outputs matched the corresponding value in the initial computational processes. An R^2 value of 1 confirmed this inference by indicating a perfect correlation. These results indicate a perfect equality between the initial and reproduced outputs. In summary, the reproduced outputs can be regarded as consistent with the initial outputs (Fig. 14).

5. Conclusion and outlook

In summary, this study introduces a strategic approach to enhance the reproducibility of computational processes in GSEs. Central to this strategy is the tracing of researchers' online actions, particularly concerning the utilization of model-service resources and the presentation of actions in the form of node-link ensembles. Consequently, this approach has three core components: the organization of actions with node-link ensembles, the formation of a process document containing the operation cycle, and the reimplementation of the process document based on a *vertex-edge* model.

By rendering the sequence of actions in a detailed and transparent manner, this strategy facilitates the reproduction of the computational processes underpinning the service-oriented strategy. It empowers researchers to share their computational processes effectively while simultaneously giving their peers the capability to re-execute initial resources and reimplement the initial computational processes of GSEs with enhanced assurance.

However, in light of the complexity and heterogeneity of implementing computational processes in various fields, future research on service-oriented GSEs is imperative. Specifically, attention should be given to the following factors:

(1) Developing tools aimed at enhancing the ease of use of the prototype system. While this study focuses on the successful implementation of resources, practical considerations such as construction costs, sharing security, and efficiency in handling massive datasets must also be considered. Thus, tools that simplify the encapsulation processes, enhance resource security, increase output visualization, and improve execution efficiency are needed to increase the usability and widespread adoption of this prototype system within the geoscience community.
(2) Designing tools aimed at facilitating other types of GSEs that may not be primarily service-based. While our primary focus is on service-based GSEs, the diversity of geo-analysis models must be taken into consideration. It is, therefore, imperative to broaden our toolset to accommodate GSEs that operate outside a service-oriented strategy. By developing and integrating tools that cater to nonservice-based GSEs, the inclusion, adaptation, and robustness of the proposed approach can be ensured.
(3) Exploring collaborative methods for constructing reproducible GSEs in an open web environment. Given that complex GSEs often involve multiple collaborating researchers, including developers, submitters, and decision-makers, author collaboration with distinct responsibilities must be involved (Chirigati et al., 2013). Furthermore, effective communication among participants and visitors should be encouraged, not only to provide feedback on outputs but also to provide real-time assistance to visitors experiencing difficulties throughout various computational processes.
(4) Conducting validation studies comparing the reproduced and initial outputs. Validation involves assessing the degree to which a measured object can be regarded as reproducible, given that threshold fluctuations are permitted when executing different model services. In complex geoscience, the definitions and evaluation criteria established may have broad and more complex implications.

CRediT authorship contribution statement

Zhiyi Zhu: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Min Chen:** Conceptualization, Methodology, Supervision, Writing – review & editing, Funding acquisition. **Lingzhi Sun:** Methodology. **Zhen Qian:** Methodology, Writing – review & editing. **Yuanqing He:** Methodology, Software, Writing – review & editing. **Zaiyang Ma:** Formal analysis, Writing – review & editing. **Fengyuan Zhang:** Formal analysis, Writing – review & editing. **Yongning Wen:** Methodology, Writing – review & editing. **Songshan Yue:** Formal analysis, Writing – review & editing. **Guonian Lü:** Formal analysis.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

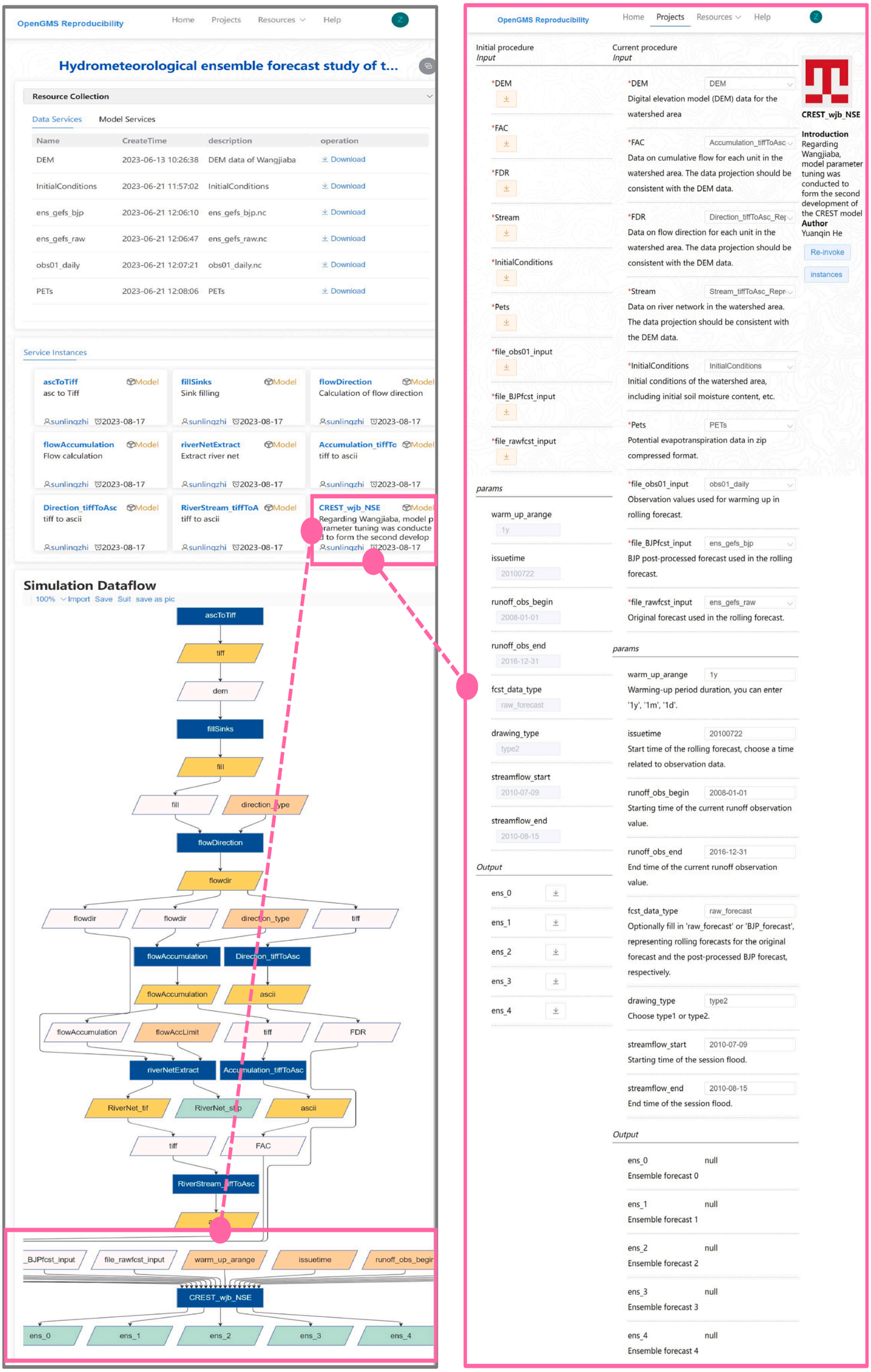

Fig. 13. Specific page of this case study.

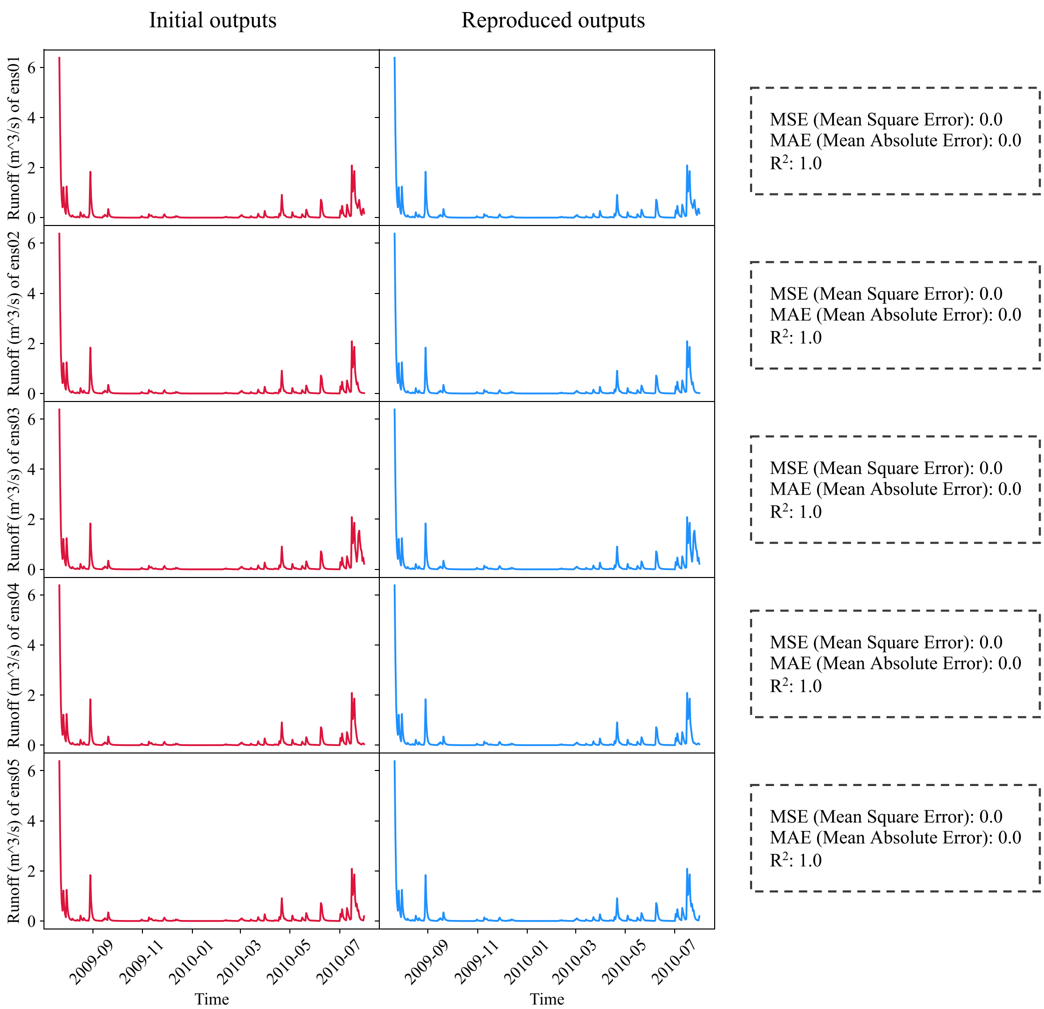

Fig. 14. Comparison between the initial and reproduced outputs.

Data availability

All the data resources that support the case study can be achieved with the link: https://doi.org/10.5281/zenodo.8190022. The codes are shared with links in Section 4.1.

Acknowledgements

We appreciate the detailed suggestions and comments from the editor and anonymous reviewers and express heartfelt thanks to the other members of the OpenGMS team. This work was supported by the National Natural Science Foundation of China [Grant No. 42071363].

Appendix A. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jag.2023.103520.

References

Badham, J., Elsawah, S., Guillaume, J.H.A., Hamilton, S.H., Hunt, R.J., Jakeman, A.J., Pierce, S.A., Snow, V.O., Babbar-Sebens, M., Fu, B., Gober, P., Hill, M.C., Iwanaga, T., Loucks, D.P., Merritt, W.S., Peckham, S.D., Richmond, A.K., Zare, F., Ames, D., Bammer, G., 2019. Effective modeling for integrated water resource management: a guide to contextual practices by phases and steps and future opportunities. Environ. Modell. Softw. 116, 40–56. https://doi.org/10.1016/j.envsoft.2019.02.013.

Baker, M., 2016. 1,500 scientists lift the lid on reproducibility. Nature 533, 452–454. https://doi.org/10.1038/533452a.

Barba, L.A., 2018. Terminologies for Reproducible Research. doi: 10.48550/arXiv.1802.03311.

Beaulieu-Jones, B.K., Greene, C.S., 2017. Reproducibility of computational workflows is automated using continuous analysis. Nat. Biotechnol. 35, 342–346. https://doi.org/10.1038/nbt.3780.

Benenson, I., Torrens, P., 2004. Geosimulation: Automata-based Modeling of Urban Phenomena. John Wiley & Sons.

Bolukbasi, B., Berente, N., Cutcher-Gershenfeld, J., Dechurch, L., Flint, C., Haberman, M., King, J.L., Knight, E., Lawrence, B., Masella, E., et al., 2013. Open data: crediting a culture of cooperation. Science 342, 1041–4042. https://doi.org/10.1126/science.342.6162.1041-b.

Cerutti, V., Bellman, C., Both, A., Duckham, M., Jenny, B., Lemmens, R.L.G., Ostermann, F.O., 2021. Improving the reproducibility of geospatial scientific workflows: the use of geosocial media in facilitating disaster response. J. Spat. Sci. 66, 383–400. https://doi.org/10.1080/14498596.2019.1654944.

Chen, Y., Lin, H., Xiao, L., Jing, Q., You, L., Ding, Y., Hu, M., Devlin, A.T., 2021b. Versioned geoscientific workflow for the collaborative geo-simulation of human-nature interactions – a case study of global change and human activities. Int. J. Digit. Earth 14, 510–539. https://doi.org/10.1080/17538947.2020.1849439.

Chen, M., Voinov, A., Ames, D.P., Kettner, A.J., Goodall, J.L., Jakeman, A.J., Barton, M. C., Harpham, Q., Cuddy, S.M., DeLuca, C., Yue, S., Wang, J., Zhang, F., Wen, Y., Lü, G., 2020. Position paper: Open web-distributed integrated geographic modelling and simulation to enable broader participation and applications. Earth-Sci. Rev. 207, 103223. https://doi.org/10.1016/j.earscirev.2020.103223.

Chen, M., Yue, S., Lü, G., Lin, H., Yang, C., Wen, Y., Hou, T., Xiao, D., Jiang, H., 2019. Teamwork-oriented integrated modeling method for geo-problem solving. Environ. Modell. Softw. 119, 111–123. https://doi.org/10.1016/j.envsoft.2019.05.015.

Chen, M., Lv, G., Zhou, C., Lin, H., Ma, Z., Yue, S., Wen, Y., Zhang, F., Wang, J., Zhu, Z., Xu, K., He, Y., 2021a. Geographic modeling and simulation systems for geographic research in the new era: Some thoughts on their development and construction. Sci. China Earth Sci. 64, 1207–1223. https://doi.org/10.1007/s11430-020-9759-0.

Chen, M., Qian, Z., Boers, N., Jakeman, A.J., Kettner, A.J., Brandt, M., Kwan, M.-P., Batty, M., Li, W., Zhu, R., Luo, W., Ames, D.P., Barton, C.M., Cuddy, S.M., Koirala, S., Zhang, F., Ratti, C., Liu, J., Zhong, T., Liu, J., Wen, Y., Yue, S., Zhu, Z., Zhang, Z., Sun, Z., Lin, J., Ma, Z., He, Y., Xu, K., Zhang, C., Lin, H., Lü, G., 2023. Iterative

integration of deep learning in hybrid Earth surface system modelling. Nat. Rev. Earth Environ. 4, 568–581. https://doi.org/10.1038/s43017-023-00452-7.
Chicco, D., Warrens, M.J., Jurman, G., 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput. Sci. 7, e623. https://doi.org/10.7717/peerj-cs.623.
Chirigati, F., Troyer, M., Shasha, D., Freire, J., 2013. A computational reproducibility benchmark. Bull. IEEE Comput. Soc. Tech. Committee Data Eng. 36, 54–59.
Choi, Y.-D., Roy, B., Nguyen, J., Ahmad, R., Maghami, I., Nassar, A., Li, Z., Castronova, A. M., Malik, T., Wang, S., Goodall, J.L., 2023. Comparing containerization-based approaches for reproducible computational modeling of environmental systems. Environ. Modell. Softw. 167, 105760 https://doi.org/10.1016/j.envsoft.2023.105760.
Committee on Reproducibility and Replicability in Science, Board on Behavioral, Cognitive, and Sensory Sciences, Committee on National Statistics, Division of Behavioral and Social Sciences and Education, Nuclear and Radiation Studies Board, Division on Earth and Life Studies, Board on Mathematical Sciences and Analytics, Committee on Applied and Theoretical Statistics, Division on Engineering and Physical Sciences, Board on Research Data and Information, Committee on Science, Engineering, Medicine, and Public Policy, Policy and Global Affairs, National Academies of Sciences, Engineering, and Medicine, 2019. Reproducibility and Replicability in Science. National Academies Press, Washington, D.C. doi: 10.17226/25303.
Davison, A., 2012. Automated capture of experiment context for easier reproducibility in computational research. Comput. Sci. Eng. 14, 48–56. https://doi.org/10.1109/MCSE.2012.41.
Essawy, B.T., Goodall, J.L., Voce, D., Morsy, M.M., Sadler, J.M., Choi, Y.D., Tarboton, D. G., Malik, T., 2020. A taxonomy for reproducible and replicable research in environmental modelling. Environ. Modell. Softw. 134, 104753 https://doi.org/10.1016/j.envsoft.2020.104753.
Fotheringham, A.S., Sachdeva, M., 2022. Modelling spatial processes in quantitative human geography. Ann. GIS 28, 5–14. https://doi.org/10.1080/19475683.2021.1903996.
Fotheringham, A.S., Yang, W., Kang, W., 2017. Multiscale geographically weighted regression (MGWR). Ann. Am. Assoc. Geogr. 107, 1247–1265. https://doi.org/10.1080/24694452.2017.1352480.
Gil, Y., David, C.H., Demir, I., Essawy, B.T., Fulweiler, R.W., Goodall, J.L., Karlstrom, L., Lee, H., Mills, H.J., Oh, J., Pierce, S.A., Pope, A., Tzeng, M.W., Villamizar, S.R., Yu, X., 2016. Toward the Geoscience Paper of the Future: Best practices for documenting and sharing research from data to software to provenance. Earth Space Sci. 3, 388–415. https://doi.org/10.1002/2015EA000136.
Goodchild, M.F., Fotheringham, A.S., Kedron, P., Li, W., 2021. Introduction: forum on reproducibility and replicability in geography. Ann. Am. Assoc. Geogr. 111, 1271–1274. https://doi.org/10.1080/24694452.2020.1806030.
He, Y., Chen, M., Wen, Y., Yue, S., Lü, G., 2023. A web-based strategy to reuse grids in geographic modeling. Int. J. Appl. Earth Obs. Geoinf. 116, 103170 https://doi.org/10.1016/j.jag.2022.103170.
Hodson, T.O., 2022. Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not. Geosci. Model Dev. 15, 5481–5487. https://doi.org/10.5194/gmd-15-5481-2022.
Kedron, P., Holler, J., 2022. Replication and the search for the laws in the geographic sciences. Ann. GIS 28, 45–56. https://doi.org/10.1080/19475683.2022.2027011.
Khajetoorians, A.A., Wiebe, J., 2014. Hitting the limit of magnetic anisotropy. Science 344, 976–977. https://doi.org/10.1126/science.1254402.
Knoth, C., Nüst, D., 2017. Reproducibility and practical adoption of GEOBIA with open-source software in Docker containers. Remote Sens. 9, 290 https://doi.org/10.3390/rs9030290.
Konkol, M., Kray, C., Pfeiffer, M., 2019. Computational reproducibility in geoscientific papers: Insights from a series of studies with geoscientists and a reproduction study. Int. J. Geogr. Inf. Sci. 33, 408–429. https://doi.org/10.1080/13658816.2018.1508687.
Konkol, M., Nüst, D., Goulier, L., 2020. Publishing computational research - a review of infrastructures for reproducible and transparent scholarly communication. Res. Integr. Peer Rev. 5, 10. https://doi.org/10.1186/s41073-020-00095-y.
Koo, H., Iwanaga, T., Croke, B.F.W., Jakeman, A.J., Yang, J., Wang, H.-H., Sun, X., Lü, G., Li, X., Yue, T., Yuan, W., Liu, X., Chen, M., 2020. Position paper: sensitivity analysis of spatially distributed environmental models- a pragmatic framework for the exploration of uncertainty sources. Environ. Modell. Softw. 134, 104857 https://doi.org/10.1016/j.envsoft.2020.104857.
Little, G., Chilton, L.B., Goldman, M., Miller, R.C., 2010. Exploring iterative and parallel human computation processes. In: Proceedings of the ACM SIGKDD Workshop on Human Computation, pp. 68–76. doi: 10.1145/1837885.1837907.
Lorscheid, I., Heine, B.-O., Meyer, M., 2012. Opening the 'black box' of simulations: increased transparency and effective communication through the systematic design of experiments. Comput. Math. Organ. Theory 18, 22–62. https://doi.org/10.1007/s10588-011-9097-3.
Ma, Z., Chen, M., Yue, S., Zhang, B., Zhu, Z., Wen, Y., Lü, G., Lu, M., 2021. Activity-based process construction for participatory geo-analysis. GISci. Remote Sens. 58, 180–198. https://doi.org/10.1080/15481603.2020.1868211.
Ma, Z., Chen, M., Zheng, Z., Yue, S., Zhu, Z., Zhang, B., Wang, J., Zhang, F., Wen, Y., Lü, G., 2022. Customizable process design for collaborative geographic analysis. GISci. Remote Sens. 59, 914–935. https://doi.org/10.1080/15481603.2022.2082751.
Merz, K.M., Amaro, R., Cournia, Z., Rarey, M., Soares, T., Tropsha, A., Wahab, H.A., Wang, R., 2020. Editorial: method and data sharing and reproducibility of scientific results. J. Chem. Inf. Model. 60, 5868–5869. https://doi.org/10.1021/acs.jcim.0c01389.
Nakagawa, S., Johnson, P.C., Schielzeth, H., 2017. The coefficient of determination R 2 and intra-class correlation coefficient from generalized linear mixed-effects models revisited and expanded. J. R. Soc. Interface 14, 20170213.
Nüst, D., Pebesma, E., 2021. Practical reproducibility in geography and geosciences. Ann. Am. Assoc. Geogr. 111, 1300–1310. https://doi.org/10.1080/24694452.2020.1806028.
Nüst, D., Sochat, V., Marwick, B., Eglen, S.J., Head, T., Hirst, T., Evans, B.D., 2020. Ten simple rules for writing Dockerfiles for reproducible data science. PLoS Comput. Biol. 16, e1008316 https://doi.org/10.1371/journal.pcbi.1008316.
Open Science Collaboration, 2015. Estimating the reproducibility of psychological science. Science 349, aac4716. doi: 10.1126/science.aac4716.
Qian, Z., Liu, X., Tao, F., Zhou, T., 2020. Identification of urban functional areas by coupling satellite images and taxi GPS Trajectories. remote Sens. 12, 2449 https://doi.org/10.3390/rs12152449.
Reinecke, R., Trautmann, T., Wagener, T., Schüler, K., 2022. The critical need to foster computational reproducibility. Environ. Res. Lett. 17, 041005 https://doi.org/10.1088/1748-9326/ac5cf8.
Richardson, D.B., Kwan, M.-P., Alter, G., McKendry, J.E., 2015. Replication of scientific research: addressing geoprivacy, confidentiality, and data sharing challenges in geospatial research. Ann. GIS 21, 101–110. https://doi.org/10.1080/19475683.2015.1027792.
Semenov, M.A., Stratonovitch, P., 2010. Use of multi-model ensembles from global climate models for assessment of climate change impacts. Clim. Res. 41, 1–14. https://doi.org/10.3354/cr00836.
Shashidharan, A., 2017. Computational steering for geosimulations. SIGSPATIAL Spec 8, 7–8. https://doi.org/10.1145/3100243.3100248.
Song, C., 2016. On paradigms of geographical research. Prog. Geogr. 35, 1–3. https://doi.org/10.18306/dlkxjz.2016.01.001.
Stodden, V., McNutt, M., Bailey, D.H., Deelman, E., Hanson, B., Heroux, M.A., Ioannidis, J.P.A., Taufer, M., 2018. Enhancing reproducibility for computational methods. Science 354, 1240–1241. https://doi.org/10.1126/science.aah6168.
Tomasello, M., Call, J., 2011. Reproducible research in computational science. Science 334, 1227–1228. https://doi.org/10.1126/science.1213443.
Waltemath, D., Adams, R., Beard, D.A., Bergmann, F.T., Bhalla, U.S., Britten, R., Chelliah, V., Cooling, M.T., Cooper, J., Crampin, E.J., Garny, A., Hoops, S., Hucka, M., Hunter, P., Klipp, E., Laibe, C., Miller, A.K., Moraru, I., Nickerson, D., Nielsen, P., Nikolski, M., Sahle, S., Sauro, H.M., Schmidt, H., Snoep, J.L., Tolle, D., Wolkenhauer, O., Le Novère, N., 2011. Minimum information about a simulation experiment (MIASE). PLoS Comput. Biol. 7, e1001122 https://doi.org/10.1371/journal.pcbi.1001122.
Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L.B., Bourne, P.E., Bouwman, J., Brookes, A.J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C.T., Finkers, R., Gonzalez-Beltran, A., Gray, A.J.G., Groth, P., Goble, C., Grethe, J.S., Heringa't, J., Hoen, P.A.C., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S.J., Martone, M.E., Mons, A., Packer, A.L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., Mons, B., 2016. The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data 3, 160018. https://doi.org/10.1038/sdata.2016.18.
Wilson, J.P., Butler, K., Gao, S., Hu, Y., Li, W., Wright, D.J., 2021. A five-star guide for achieving replicability and reproducibility when working with GIS software and algorithms. Ann. Am. Assoc. Geogr. 111, 1311–1317. https://doi.org/10.1080/24694452.2020.1806026.
Yue, S., Chen, M., Wen, Y., Lu, G., 2016. Service-oriented model-encapsulation strategy for sharing and integrating heterogeneous geo-analysis models in an open web environment. ISPRS-J. Photogramm. Remote Sens. 114, 258–273. https://doi.org/10.1016/j.isprsjprs.2015.11.002.
Yue, S., Chen, M., Yang, C., Shen, C., Zhang, B., Wen, Y., Lü, G., 2019. A loosely integrated data configuration strategy for web-based participatory modeling. GISci. Remote Sens. 56, 670–698. https://doi.org/10.1080/15481603.2018.1549820.
Zaragozí, B.M., Trilles, S., Navarro-Carrión, J.T., 2020. Leveraging container technologies in a GIScience project: a perspective from open reproducible research. ISPRS Int. J. Geo-Inf. 9, 138. https://doi.org/10.3390/ijgi9030138.
Zhang, F., 2019. Design and development of a service-oriented wrapper system for sharing and reusing distributed geoanalysis models on the web. Environ. Modell. Softw. 111, 498–509. https://doi.org/10.1016/j.envsoft.2018.11.002.
Zhang, F., Chen, M., Kettner, A.J., Ames, D.P., Harpham, Q., Yue, S., Wen, Y., Lü, G., 2021a. Interoperability engine design for model sharing and reuse among OpenMI, BMI and OpenGMS-IS model standards. Environ. Modell. Softw. 144, 105164 https://doi.org/10.1016/j.envsoft.2021.105164.
Zhang, F., Chen, M., Wang, M., Wang, Z., Zhang, S., Yue, S., Wen, Y., Lü, G., 2021b. A framework on task configuration and execution for distributed geographical simulation. Int. J. Digit. Earth 14, 1103–1125. https://doi.org/10.1080/17538947.2021.1949400.
Zhang, Z., Chen, M., Zhong, T., Zhu, R., Qian, Z., Zhang, F., Yang, Y., Zhang, K., Santi, P., Wang, K., Pu, Y., Tian, L., Lü, G., Yan, J., 2023b. Carbon mitigation potential afforded by rooftop photovoltaic in China. Nat. Commun. 14, 2347. https://doi.org/10.1038/s41467-023-38079-3.
Zhang, M., Yue, P., Hu, L., Wu, H., Zhang, F., 2023a. An interoperable and service-oriented approach for real-time environmental simulation by coupling OGC WPS and

SensorThings API. Environ. Modell. Softw. 165, 105722 https://doi.org/10.1016/j.envsoft.2023.105722.

Zhu, Z., Chen, M., Qian, Z., Li, H., Wu, K., Ma, Z., Wen, Y., Yue, S., Lü, G., 2023b. Documentation strategy for facilitating the reproducibility of geo-simulation experiments. Environ. Modell. Softw. 163, 105687 https://doi.org/10.1016/j.envsoft.2023.105687.

Zhu, R., Zhang, F., Yan, J., Ratti, C., Chen, M., 2023. A sustainable solar city: From utopia to reality facilitated by GIScience. TIG 100006. doi: 10.59717/j.xinn-geo.2023.100006.