Geoprocessing Appstore

Christin Henzen, Geoinformation Systems, TU Dresden, christin.henzen@tu-dresden.de
Johannes Brauner, Geoinformation Systems, TU Dresden, johannes.brauner@tu-dresden.de
Matthias Müller, Geoinformation Systems, TU Dresden, matthias_mueller@tu-dresden.de
Daniel Henzen, Geoinformation Systems, TU Dresden, daniel.henzen@tu-dresden.de
Lars Bernard, Geoinformation Systems, TU Dresden, lars.bernard@tu-dresden.de

Abstract

Spatial Data Infrastructures still focus geodata discovery and visualization. Web platforms for publishing, sharing and documenting geospatial algorithms are equally important, but do hardly exist. The Geoprocessing Appstore, a Web-platform respectively portal, provides a software solution for current algorithm discovery and access problems. It allows users to publish, search and download implementations of geospatial algorithms. Supplementary to the algorithm management and discovery components, the Geoprocessing Appstore comprises a knowledge platform to manage and cite semantic descriptions of geoprocessing functionality. Further, the Appstore's algorithm sandbox enables to test and execute published algorithms. The implementation of the Geoprocessing Appstore bases on various Open Source software frameworks and uses standard interfaces to ensure interoperability and integration into existing Spatial Data Infrastructures or software components.

Keywords: Discovery, Geoprocessing, Knowledge Platform, Spatial Data Infrastructure, Web Platform, Open Source Software

1 Introduction

Geodata discovery, visualization and download are still the primary focus of current Spatial Data Infrastructures (SDI). For sharing geodata, catalogue services with metadata for geodata and related Web Services, exist. Typically, these standardized catalogue interfaces (CSW) are used in geoportals, which provide user interfaces for metadata management and discovery as well as geodata visualization and download (publish, find, bind).

Equally important but often neglected are Web platforms or portals for publishing, sharing and documenting geospatial algorithms and simulation models [1]. Such platforms are expected to increase the transparency of scientific workflows and processing results but also facilitate reuse of existing algorithm implementations. Frequently used algorithms, such as spatio-temporal aggregation [2] or spatial data fusion, e.g. for linking crowdsourced and governmental data [3], should be published on the Web for further usage.

Web platforms and services, such as the workflow management system Kepler emerging in bioinformatics science [4] that support such tasks, hardly exist.

This paper proposes the Geoprocessing Appstore (GA), a community platform that acts as a Web based repository for geospatial algorithms and their implementations. Similar to catalogues for geodata, it provides management functions to publish, search and download implementations of geospatial algorithms. A supplementary sandbox allows users to test and execute the algorithms before downloading and embedding them in their own software environments. Furthermore the GA comprises a knowledge platform, which provides documentation and (semantic) metadata about geoprocessing functionality (geooperators). See Figure 2 for an overview of the GA components.

2 Geoprocessing Appstore concept

The GA is a Web-platform respectively a portal to share geospatial algorithms in a community, and provides a software solution for current algorithm discovery and access problems. Based on the core concept of the publish-find-bind paradigm for service-oriented architectures, it connects developers and users of geospatial software (Figure 1): an algorithm provider publishes algorithm source code including a description of the provided functionality with technical requirements to execute the source code. Eventually, users can discover algorithms according to their needs and finally bind, respectively apply them [5].

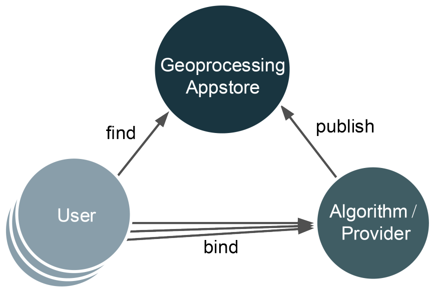


Figure 1: Publish-find-bind paradigm in the GA

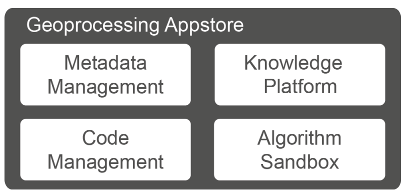


Figure 2: Components of the Geoprocessing Appstore

Beside this basic functionality required for the Publish-find-bind paradigm, the GA facilitates developers and users to share their experiences about algorithm usage, reproducibility of algorithms and software requirements. These experiences can be expressed by rating and commenting published algorithms or managing algorithm knowledge, e.g. geooperator metadata. In addition to these quality assurance aspects, the GA serves as testing environment respectively sandbox. Thereby, users can examine published algorithms and conduct dry runs before they are going to deploy them in their own infrastructures.

Furthermore, the GA provides several interfaces to 1) inform about new or updated algorithms, 2) to bind discovered algorithms in own infrastructures and 3) to reuse algorithms in own applications, which are described below. Own GA instances can also be set up in local environments.

Two concepts serve as fundamentals for the Geoprocessing Appstore: 1) Moving Code Packages as description format for geospatial algorithm code [6] and 2) geooperators as implementation independent description for geospatial algorithm semantics [7].

Moving Code is a concept to interchange reusable algorithms. A Moving Code Package is a software component that contains source code and description of a provided geospatial algorithm. The algorithm description is machine-readable and contains information about the provided functionality, software and hardware requirements in regard to platform, infrastructure, and exploitation rights (Figure 3). The description format uses the Web Processing Service (WPS) standard. Therefore, code packages can be enacted and executed via WPS [6]. In the GA, the Moving Code description format serves as metadata scheme for the geospatial algorithms. Moving Code Packages as self-describing software components are used to test or execute published algorithms on the Web via WPS.

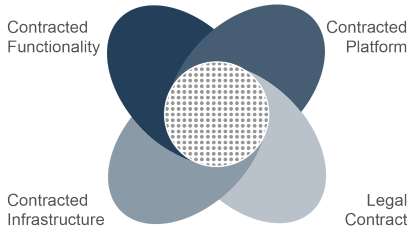


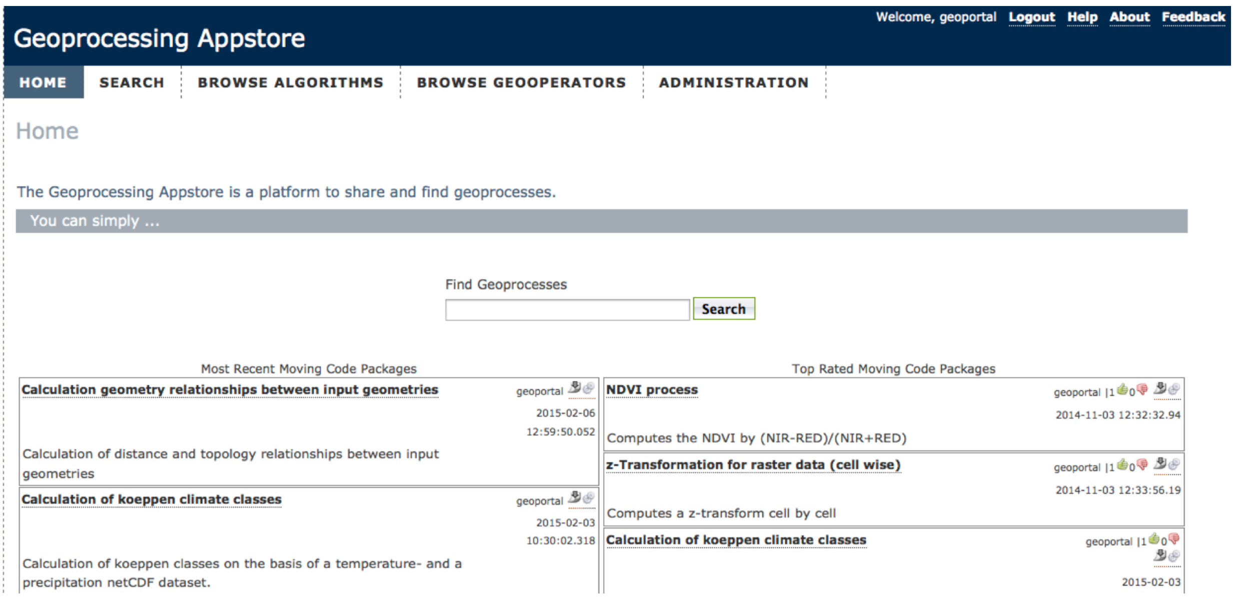
Figure 3: 4 Dimensions to describe an algorithm implementation [6]

Another core concept that drives the GA is represented by geooperators [7]. Geooperators provide building blocks for geospatial algorithms, which are semantically described. Beside semantic metadata, a geooperator description contains links to similar geooperators to enable an implementation independent creation of workflows and thus enhance interoperability for the developed algorithms. Furthermore, the geooperator descriptions provide links to further geooperators that are often used in the same context. In the GA, a geooperator browser is implemented as part of the knowledge platform, which provides several alternative perspectives on geooperators by a faceting browsing interface. The browser thus facilitates geooperator discovery and their integration into algorithms.

2.1 Management of algorithm descriptions and source code

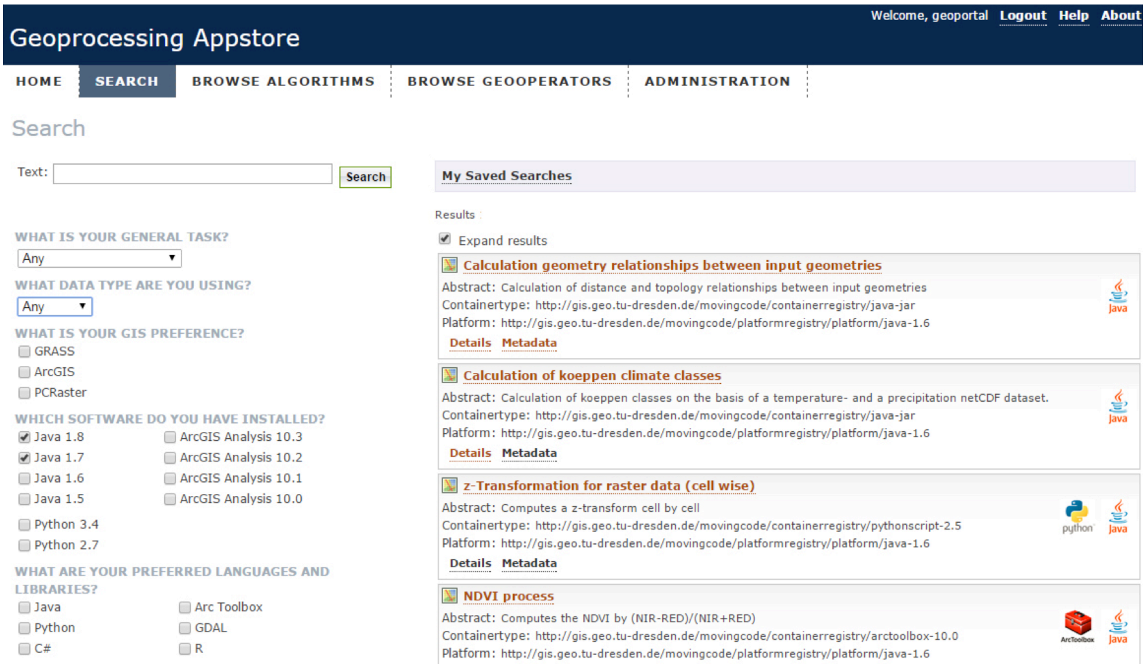
The management of algorithm descriptions and source code is one of the core features of the GA. The GA management section provides functionality for facilitating algorithm providers to publish, update or delete algorithm descriptions or source code. Publishing and updating algorithm descriptions including source code can be done as follows: 1) using the provided metadata editor to create the algorithm

Figure 4: Landing page of the Geoprocessing Appstore with recently published and top rated algorithms



description and upload the described source code (the guided

Figure 5: Advanced search in the Geoprocessing Appstore with text- and icon-based result list



approach based on pre-filled forms and including schema validation) or 2) uploading a moving code package – a well-defined file in zip-format that contains the algorithm description formalized as XML (Extensible Markup Language) file, and the described source code as zip file. Moreover, providers can tag algorithm descriptions with used geooperators, which facilitates users to compare algorithms on a conceptual level.

2.2 Discovery of algorithm descriptions

In contrast to the well-established geodata discovery, similar mechanisms are yet missing for geospatial algorithm discovery, especially in terms of standardized protocols and discovery modes. To address this issue, our GA provides several discovery modes for the published algorithms known from geodata discovery: 1) a simple keyword search, 2) an advanced search using multiple filters, 3) a browsing user interface and 4) an RSS (Rich Site Summary) feed. By providing established geodata discovery modes, users familiar with geoportals can easily find their way in the GA as well.

The GA landing page (Figure 4) offers the keyword search, and algorithm lists containing recently published and top rated algorithms. The advanced search combines keyword search with further search filters (Figure 5).

The browsing user interface (Figure 6) supports novice users to explore all available algorithm descriptions. Depending on individual requirements, users can browse by using different viewpoints.

In addition to the mentioned approaches to identify and discover algorithms via active searching, the GA provides an RSS feed. RSS feeds can be processed in various external RSS clients, e.g. Web-based applications, external browser-based newsreaders, or other subscriptions-enabled software components. Thereby, users can actively be informed about recently published or updated algorithms that match their individual configured filters in their RSS clients.

Figure 6: User interface to browse algorithm descriptions in the Geoprocessing Appstore

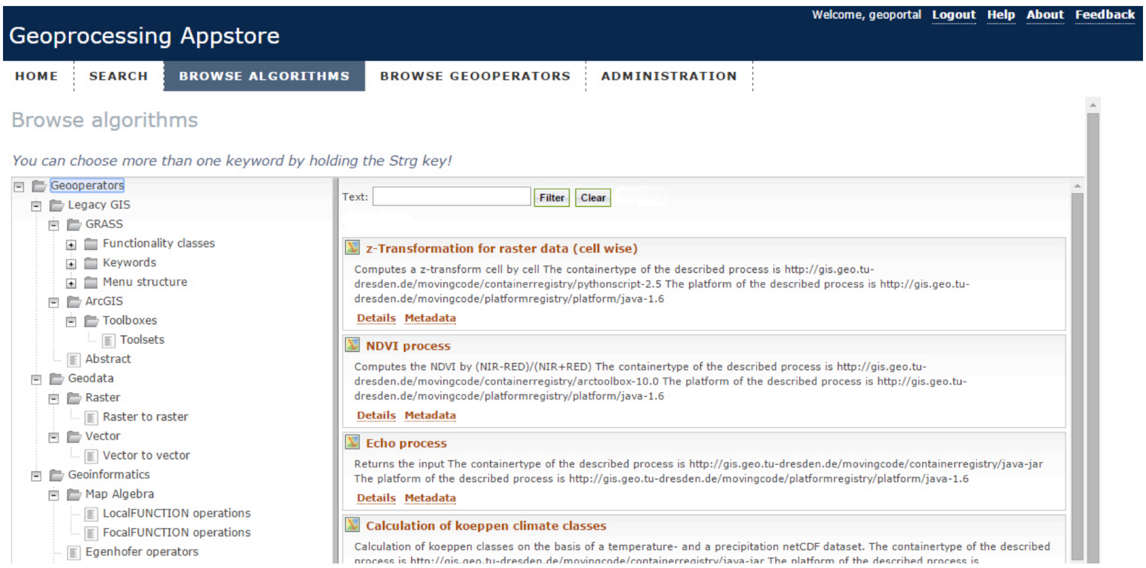
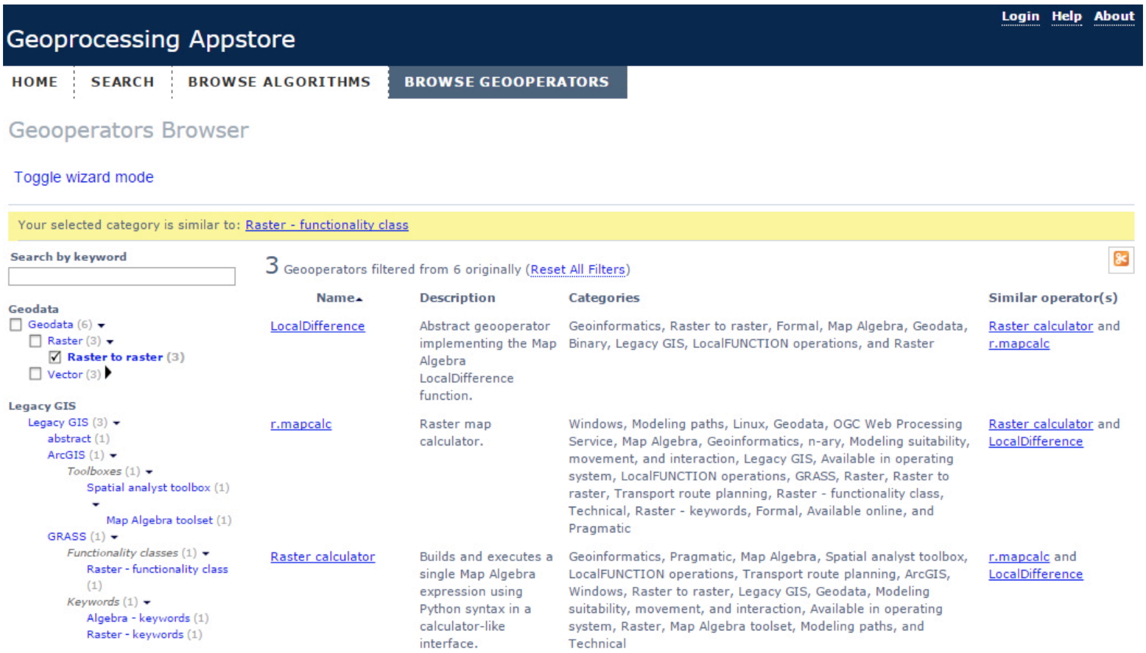


Figure 7: Geooperator Browser



2.3 Knowledge management and geooperator discovery

The GA is a community-based knowledge platform for geoprocessing in general and geospatial algorithms in particular. Community-driven information contained in the GA are 1) geooperators that semantically describe atomic or complex parts of geospatial algorithms, 2) container types that describe general runtime requirements (e.g. the algorithm language), and 3) platforms that describe implementation dependencies for runtime environment configurations (e.g. required geospatial libraries including their specific version). Container type and platform describe the required runtime environment and also further facilitate adaptability and enhancement for algorithms.

The Geooperator Browser (Figure 7), as core of the knowledge platform, acts as discovery tool and reference documentation for geooperators. For each geooperator, the browser provides a detailed (semantic) description, related categories, links to similar and otherwise related geooperators, and relevant scientific publications and documentations. To explore geooperators, users can simultaneously enter keywords, select hierarchical categories arranged by facets, and use the wizard mode for a guided discovery.

In the geooperator editor of the GA (Figure 8), users can describe and categorise geooperators and define relations between them, according to their individual expertise.

2.4 Geospatial algorithm sandbox

Beside discovering geospatial algorithms or downloading the related source code from the GA, a fundamental GA feature is to evaluate the algorithm's fitness for use by the algorithm's description, and to test the algorithm source code in a sandbox environment. In SDI, geoprocessing is realized with WPS. Typically, WPS are used to provide and execute

Figure 8: Geooperator management in the GA administration

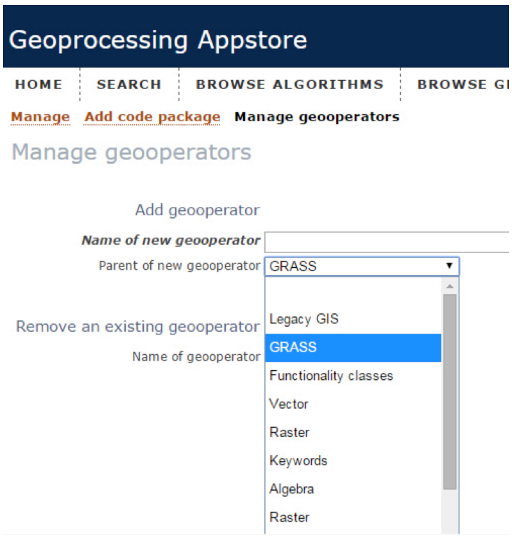


Figure 9: User interface of the algorithm sandbox as provided by the 52°North WPS-JS

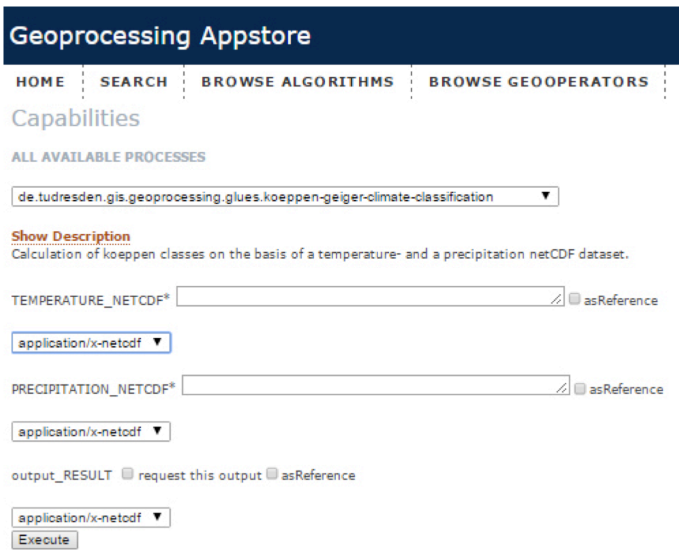
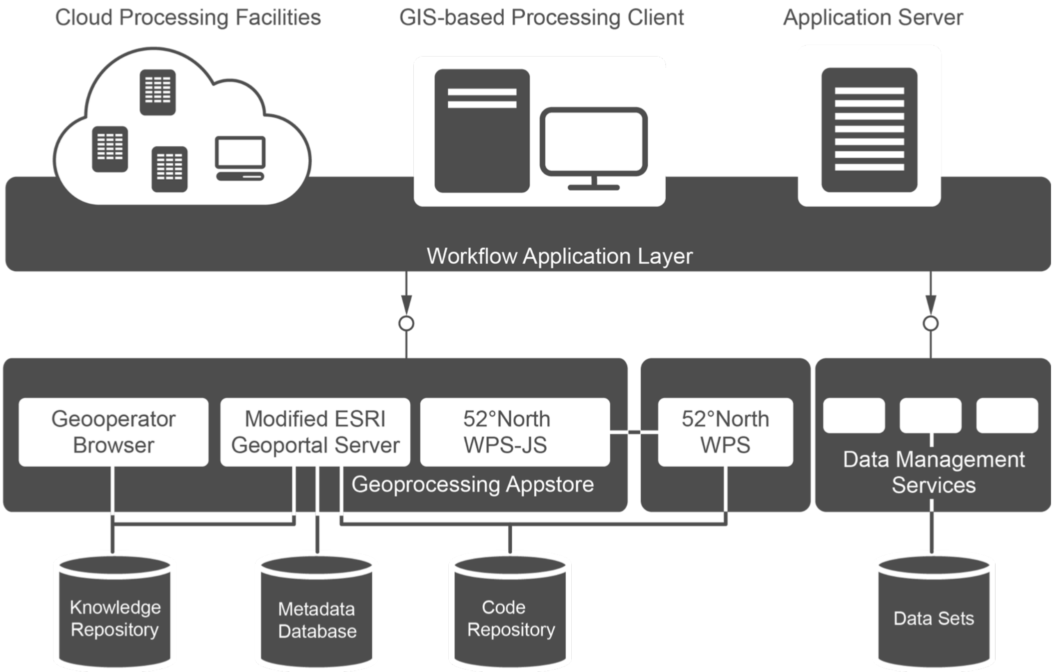


Figure 10: Components of the Geoprocessing Appstore in a distributed algorithm sharing environment



geospatial algorithms. In the GA, the WPS and the related WPS-JS standalone JavaScript WPS client provide the basis for a testing platform (sandbox) for supplied algorithms. To navigate to the respective algorithm sandbox section, the details page and the short description of an algorithm in the search result list provide corresponding links. Based on the algorithm metadata, the algorithm sandbox configuration page (Figure 9) automatically generates suitable user interface elements to configure all algorithm parameters individually. Due to the automatically generated user interface, non-technical users do not need to be aware of the WPS complexity, including manual configuration and request creation. After testing the algorithm source code and evaluating the given result, the user can subsequently decide whether to a) download the source code, b) bind the sandbox WPS in own infrastructures or c) refine and continue the previously performed algorithm search.

3 Geoprocessing Appstore implementation

As a Web platform with a community-oriented algorithm exchange approach, the GA offers common functionality for login, search, and a rating system. The implementation is based on Open Source frameworks, which provide the required functionality as extendable and modifiable source code. By submitting optimized and refactored source code back to the respective developer communities, the sustainability of the GA is further improved. Besides, the use of standard interfaces ensures interoperability and integration into existing SDI or software systems.

The GA uses the following products: the ESRI Geoportal Server for the management and discovery of algorithm descriptions, the 52°North WPS and WPS-JS for the algorithm execution functionality and the Geooperator Browser for geooperator discovery, semantic geooperator descriptions, and the visualization of geooperator relationships. The underlying algorithm descriptions are formalized in the Moving Code Schema [6] and the OGC WPS Standard. The semantic geooperator descriptions [7] and the knowledge section are based on the Geooperator Browser.

The ESRI Geoportal Server[1] project realizes typical geoportal functions (login, search, etc.) and different related interfaces and is implemented in Java, Java Server Pages (JSP), JavaScript and Hypertext Markup Language (HTML). The following features provided by the ESRI Geoportal Server are relevant for the GA implementation:

- Mechanisms to integrate new metadata schemes
- User interface and logic for metadata publication via editor or XML upload, and adaptability of the editor for a customized metadata schema
- User interface and logic for discovery and extensible search filters
- Various user interfaces to browse metadata
- Various REST (Representational State Transfer) interfaces including Atom feeds, HTML snippets, and JSON (JavaScript Object Notation)
- Mechanisms to rate metadata

The 52°North WPS[2] project enables Web-based deployment of geoprocessing functionality. It is based on WPS Profiles and implemented in Java, JSP and HTML. The 52°North WPS supports several established (geographical) computation backends, e.g. R, GRASS, ArcGIS and SEXTANTE, and thus enables algorithm providers to publish scripts in their preferred programming language and runtime environment.

[1] Project information and source code are available at: https://github.com/Esri/geoportal-server/

[2] Project information and source code are available at: https://github.com/52North/WPS

The related WPS-JS[3] project is a light-weight WPS Web client, which provides a user interface for WPS implementations. Both projects jointly provide the following functions relevant for the GA implementation:

- Pluggable framework for algorithms based on formal description files
- Wrapper for established functionality backends.
- Algorithm source code repository
- Parser and encoder for WPS requests
- User interfaces for WPS process descriptions
- User interface for WPS process execution and parameterization

The Geooperator Browser[4] is a Web client that visualizes geooperators and their semantic metadata. The light-weight JavaScript and HTML client facilitates the interactive discovery of geooperators. The following Geooperator Browser features are relevant for the GA implementation:

- Discovery of geooperators by underlying backend (e.g. GRASS, ArcGIS), by underlying data model (e.g. raster or vector), or by application context (e.g. for modelling hydrological simulations, etc.)
- Facilitating explorative search and search by keywords for geooperators and providing a wizard for guided discovery
- Showing geooperator relationships (e.g. similar geooperators for their interchangeability or otherwise related ones)

The GA[5] integrates the mentioned products and extends the provided application logic and user interface components for the management of the algorithm code packages, for a seamless algorithm discovery, for sandbox testing, as well as for geooperator knowledge management. Whilst Geooperator Browser and WPS-JS are completely integrated into the GA (Figure 10) the WPS is merely coupled.

Enabling the ESRI Geoportal Server to manage algorithm descriptions and source code is based on the implementation of the algorithm description scheme, and the creation of the code repository and upload mechanisms to publish and update the code. Since the ESRI Geoportal Server provides schema integration components, registering the Moving Code Package scheme as new algorithm description format is realized in several XML, XSD (XML Schema Definition) and XSL (Extensible Stylesheet Language) files for 1) schema definition and template, 2) validation information and 3) user interface definitions for details and edit views.

Integrating the code repository is more sophisticated, as there are two alternatives to publish algorithm source code and description (Section 2.1): 1) upload only the source code and add the description via the guided metadata editor, or 2) upload a preassembled Moving Code Package (zip file) containing algorithm description and the respective source code. To allow users to upload Moving Code Packages, a corresponding user interface is developed and the existing ESRI Geoportal Server upload mechanism is extended to enable zip file upload. Functionality for unpacking zip files and schema validation is realized. Further, code storage into the code repository along with linked algorithm description in the database is added.

For the discovery of published algorithms, search filters are implemented and search indices are defined. The implemented search filters are registered in the ESRI Geoportal Server and allow to filter results based on used geooperators, and required platform and container type. When a search is triggered, the Geoportal index is searched. Thus, the index is extended by the Moving Code Package metadata elements.

The knowledge repository managed by the GA administration view contains the structure and description of the geooperator, platform and container type metadata. Information stored in this database is used to generate the browse algorithm page and to generate GUI elements for the metadata editing menus. The latter usage is important for less error prone metadata as users select from predefined values.

To provide an algorithm sandbox, the GA is coupled with a 52°North WPS and the WPS-JS user interface. The GA links to the WPS-JS with the algorithm identifier and maps it to the WPS process identifier. WPS-JS is adapted to be invokable with this process identifier. Based on the algorithm description, WPS-JS generates the user interface to configure and execute the algorithm via WPS. The WPS comprises a configurable Moving Code Package drop-in folder, which is used to publish and execute algorithms on-demand. The GA code repository is configured to act as drop-in folder as well. Thereby, uploaded algorithms are also instantly referenced in the WPS.

The tight coupling between WPS and GA is advantageous as follows: 1) an additional file transfer to execute a published algorithm is not necessary and 2) published algorithms are immediately available and usable as WPS in external infrastructures. On the other hand, the WPS can handle only a limited number of simultaneously executed Moving Code Packages. To address such load issues, extended load tests and the integration of multiple parallel WPS instances are planned.

4 Summary and future work

The GA is a Web-based platform to publish, discover and use geospatial algorithms. To provide a valuable product to the geoprocessing community, the GA can be used in several ways: 1) as Web platform to manage algorithm descriptions, source code, geooperators and user feedback, 2) as interface to bind components in other applications e.g. by providing parameterized WPS or WPS-JS interfaces, and the RSS feed, 3) as separate instance for integration into external infrastructures and 4) as living project for active users implementing and providing the community with novel geoprocessing functionality. By basing the GA on the standardized WPS interface for geoprocessing, our approach

[3] Project information and source code are available at: https://github.com/52North/wps-js

[4] Project information and source code are available at: https://github.com/GeoinformationSystems/GeooperatorBrowser

[5] Project information and source code are available at: https://github.com/GeoinformationSystems/GeoprocessingAppstore

remains fully interoperable. Exclusively using Open Source software further adds to interoperability and transparency.

Currently our GA provides the core functionality to manage, search and test algorithms. To further improve the GA, several extensions represent future work: 1) simplify the algorithm description scheme and therefore the Web forms for algorithm publication, 2) extend the sandbox functionality, e.g. by a graphical model builder as provided by [8] or by providing respective test datasets, and 3) enhance the search logic and provide recommendations for similar algorithms.

On a conceptual level, the following ideas are attractive to be focused on: 1) crowdsourcing for algorithm and geooperator descriptions, 2) enhancing/linking algorithm descriptions with scientific publications, 3) integrating geodataset metadata to further facilitate the Web-based geodata and algorithm discovery and testing in the sandbox, and 4) development of an abstract geoalgebra that allows to define and describe the workflow of implemented algorithms in an abstract manner, thus ensuring i.a. a long term archiving of algorithms independent from their implementation.

Crowdsourcing as a community-oriented approach certainly helps to improve algorithm and geooperator descriptions. Crowdsourcing can be enabled by allowing users to tag existing algorithm and geooperator descriptions with further keywords and metadata. For instance, a published algorithm source code needs Python 2.7 to be executed, which is described in the algorithm description. A user successfully tests the source code with Python 3.3 and is able to update the tags correspondingly in the algorithm description. Furthermore, triggering the community to participate in the GA is crucial for its sustainability, timeliness and last but not least versatility in terms of offered algorithms.

5 Acknowlegdement

The authors thank Bernd Grafe and Jochen Lenz for their support in implementing Geoprocessing Appstore components.

This work is part of GLUES (Global Assessment of Land Use Dynamics, Greenhouse Gas Emissions and Ecosystem Services), a scientific coordination and synthesis project of the ‘Sustainable Land Management’ research programme. This research programme has been launched by FONA (Research for Sustainable Development) as a framework programme within the resources and sustainability field of action, and is funded by BMBF (German Federal Ministry of Education and Research). [Support code: 01LL0901A]

This research was partly funded by the Excellence Initiative by the German Federal and State Governments (Institutional Strategy, measure "support the best").

6 Literature

[1] Bernard, L.; Mäs, S.; Müller, M.; Henzen, C.; Brauner, J.: Scientific Geodata Infrastructures: Challenges, Approaches and Directions. In: International Journal of Digital Earth, 2013.

[2] Härtwig, Mario; Müller, Matthias; Bernard, Lars: A Generic Web Service for Ad-hoc Statistical Spatio-Temporal Aggregation. In: Transactions in GIS, 2014.

[3] Wiemann, Stefan; Bernard, Lars: Linking crowdsourced observations with INSPIRE. The 17th AGILE International Conference on Geographic Information Science, Castellón, 2014.

[4] Altintas, I.; Berkley, C.; Jaeger, E.; Jones, M.; Ludascher, B. and Mock, S.: Kepler: an extensible system for design and execution of scientific workflows. Scientific and Statistical Database Management, 2004, pp. 423-424.

[5] Kadner, D.; Müller, M.; Brauner, J.; Bernard, L.: Konzeption eines Marktplatzes für den Austausch von Geoprozessierungsimplementierungen. In: gis.SCIENCE, 25(3), 2012, pp. 118–124.

[6] Müller, M.; Bernard, L.; Kadner, D.: Moving code - Sharing geoprocessing logic on the Web. In: ISPRS Journal of Photogrammetry and Remote Sensing, 83, 2013. pp. 193–203.

[7] Brauner, J.:.Ad-hoc Geoprocessing in Spatial Data Infrastructures - Formalizations for Geooperators. In H. Pundt & L. Bernard (Eds.), Proceedings 1st AGILE PhD School - 13.-14.03.2012 Wernigerode/Harz (pp. 89–98). Aachen: Shaker Verlag.

[8] Bensmann, F.; Alcacer-Labrador, D.; Ziegenhagen, D.; Roosmann, R.: The RichWPS Environment for Orchestration. ISPRS Int. J. Geo-Inf. 2014, 3(4), 1334-1351; doi:10.3390/ijgi3041334.