Multidisciplinary Research on Geographical Information in Europe and Beyond
Proceedings of the AGILE'2012 International Conference on Geographic Information Science, Avignon, April, 24-27, 2012
ISBN: 978-90-816960-0-5
Editors: Jérôme Gensel, Didier Josselin and Danny Vandenbroucke

Moving Code – Sharing Geospatial Computation Logic on the Web

Matthias Müller
TU Dresden
Geoinformation Systems
01062 Dresden
matthias_mueller@tu-dresden.de

Daniel Kadner
TU Dresden
Geoinformation Systems
01062 Dresden
daniel.kadner@tu-dresden.de

Lars Bernard
TU Dresden
Geoinformation Systems
01062 Dresden
lars.bernard@tu-dresden.de

Abstract

Software reuse is a common paradigm for building software systems and has been proven to facilitate maintenance and increase productivity. Valuable algorithms are produced in many scientific projects, open source initiatives or commercial software manufacturers. Despite the success, reuse of computational logic in large distributed systems such as Spatial Data Infrastructures (SDI) remains an issue.

This paper presents an approach for describing, publishing and sharing geospatial computational logic on the Web as Moving-Code-Packages. These packages are service-oriented software components that contain algorithmic code and its description, i.e. the contracted functionality, platform and hardware plus basic information about the exploitation rights. We also provide a generic messaging mechanism to publish processing logic on the Web and to make users aware of new versions and updated algorithms. The suggested approach is expected to provide benefits in data-driven science, cloud computing environments and offers new possibilities to deal with data protection requirements.

Keywords: Geoprocessing, Moving Code, Service-orientation, Software reuse, Spatial Data Infrastructures

1 Introduction

Geoprocessing or geo-computation is the processing of implicitly or explicitly geo-referenced data and an important of geographic information systems (GIS). A basic set of geospatial operations for analysis and data curation tasks is part of any modern Desktop GIS. With the rise of Web-based Spatial Data Infrastructures (SDI) during the last decade there is also a growing need to also support geo-computation in a Web-based manner [3, 9]. However, the development of the tools and services to accomplish the latter has not kept pace with the growing amount of data sources. This is mainly due to two reasons: 1) A lack of well-established common understanding about the right functional granularity to support recurring geo-computation tasks and 2) a lack of technical mechanisms that support the exchange of processing functionality across different distributed computation platforms [5, 13]. This paper mainly deals with the second challenge but also promotes some ideas that may facilitate a solution for the first challenge.

We propose lightweight publication framework that contracts geoprocessing logic in four dimensions: A *functional description* provides a (1) a formalization of the input and output data with respect to structure and content and (2) a comprehensible description of the procedure that derives the outputs from the inputs. A *platform description* states the dependencies to other software that needs to be available for deploying and executing the logic. Such information allows choosing a proper software platform for deployment or probing the compatibility of existing configurations. Third, a *hardware description* states the hardware requirements to reliably execute that code. As a fourth dimension we recognize the *exploitation rights* associated with the computational logic.

The paper proceeds with a conceptual model for reusable geoprocessing logic based on the requirements of service-oriented software design and a multi-level description approach for implemented processing logic. This conception is then applied to a modern heterogeneous SDI and realized using open standards wherever possible.

2 A conceptual model for reusable geoprocessing logic

Processing logic in an SDI suits a variety of tasks which range from simple general tools for data massage to sophisticated simulation models. They may be used for change and anomaly detection, to conduct what-if analyses, data capturing and curation, general analysis or complex simulations as well as aggregation and transformation logic to convert and compare data that refer to different spatial, temporal and thematic granularities [3, 5, 6, 8]. Geoprocessing functionality serves a vast number of tasks in different disciplines each having their own concepts and processing tools. In this section we investigate the requirements for published processing functionality. These requirements comprise design guidelines as well as expressive description and metadata for interchangeable processing logic.

2.1 Service-oriented design guidelines

The paradigm of service-oriented computing provides a generic means to build and describe the functionality of software components in a user-centric way. A list of design principles geared towards the service orientation of software components been complied by [4]. These principles comprise guidelines for the development of interoperable process interfaces, coherent functional contract and implementation and an effortless service-oriented operation.

1) Functional abstraction and discoverability. An important principle when designing service-oriented software components is abstraction which demands a separation of the functional description, i.e. a “blueprint”, and the concrete implementation of the logic as a software component. The functional description is a formal definition of the provided logic, e.g. in a mathematical or procedural sense. Any implementation that is contracted to the formalized logic must provide a 1:1 match of the functional description. A separation of the functional description and the implementation has several advantages in a distributed system: 1) it enables self-describing software components, 2) it integrates nicely with catalogues that provide search interfaces for processing logic, and 3) allows to assess the correctness of implemented functionality [11]. The functional contract is to be formalized in compliance with existing metadata standards as far as possible to ensure readability by a large audience. To date there are hardly any useful and widely adopted metadata standards for the description of geoprocessing logic. Web service standards like WPS [10] or WSDL [12] are at least clear about input and output data types and provide metadata elements at the interface level that may be used to reference standardized or commonly agreed functionality.

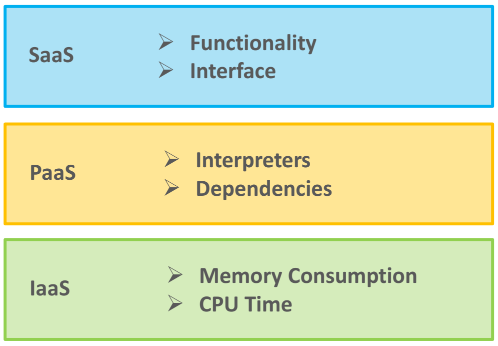
2) Service-oriented interfaces. A standardized service contract demands that the interface for a service-oriented software component is not to be derived from the implementation. This is often the case when scientific simulation models are published: Instead of retro-fitting the implemented logic to commonly used service interface specifications and data exchange formats users are often confronted with proprietary data formats or unusual interfaces. Uncommon interfaces and uncommon data formats are a major obstacle for the integration of valuable functionality into larger scale infrastructures. In contrast, the application of commonly used or even standardized data formats is a first step towards the creation of re-usable data processing tools. NetCDF (Network Common Data Form), GML (Geography Markup Language) and KML (formerly Keyhole Markup Language) are examples for such well-known, standardized data formats in the geospatial and environmental domain.

3) Service-oriented operation. A service-oriented view on processing logic includes the provision of a software component and the hardware resources required for execution. Typical scenarios are 1) the provision of processing logic as a Web service, 2) the deployment and execution of publicly available software components on a private system. In case the processing logic is provided as a Web service a mechanism for concurrent execution has to be provided that enables multiple parallel and independent ad-hoc executions as well as mechanisms for fault handling or even a failover infrastructure. In case of a mere deployment on a private target platform, the processing logic is provided in a self-descriptive, resource-oriented manner. A service-oriented operation of such a tool means the effortless parameterization and execution. In both cases the provided logic can be invoked into larger process chains to accomplish higher level computations. Possible applications are the execution of chained geoprocessing tools, simulation runs with a subsequent anomaly detection procedure or multiple parallel executions for sensitivity analyses in environmental modelling. In GIS it is common practice to execute functions in a chained manner.

2.2 Multi-level Description

With the advent of Web services in geoprocessing, research and development activities focused on service interfaces and discovery in the emerging distributed environment leaving the underlying implementation as black boxes. With the emerging cloud era it became clear that various aspects of service quality have to be treated more carefully to leverage software reuse [13]. In cloud computing, three different levels of service are commonly referred to: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS), each offering an independent layer to characterize the capabilities and quality of software components. Based on the functional abstraction that has been defined in the previous section, there may be multiple implementations, each running on a different platform (i.e. depend on different sets of libraries and interpreters) and requiring different hardware (Figure 1).

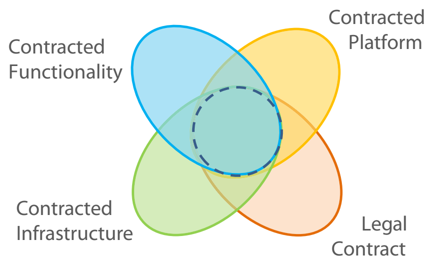
Figure 1: Contracted functionality, platform and infrastructure are independent aspects of implemented processing logic. They relate to different service layers that are usually distinguished in cloud environments.



2.3 Exploitation Rights

With the interchange of software that provides computational logic it becomes necessary to ship licenses or terms of use with such a software component. We shall refer to these as ‘exploitation rights’, which cover all legal aspects of the implemented logic from usage rights to liabilities. Producers can thus exclude certain use of their applications, e.g. to modify them, or demand them to share their derivate under the same conditions [1]. Users on the other hand are able to choose between different implementations and decide to use a product that satisfies their legal needs best. In combination with the multilevel description, the four-dimensional description model is obtained (Figure 2).

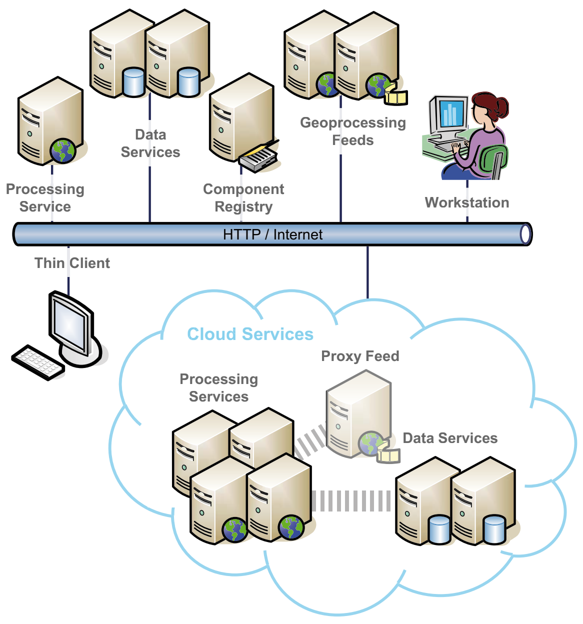
Figure 2: Conceptual description model for interchangeable processing logic. The four principal dimensions are considered independent and relate to different aspects associated with a reusable software component.



3 Operational Architecture

The proposed concept is envisaged to be operated in heterogeneous distributed systems. Figure 3 shows an architectural overview: Possible application contexts range from stand-alone applications in GIS workstations, via service-based applications where clients invoke remote services to perform GIS tasks, to cloud environments offering computational power and bandwidth to process large datasets. While the effective geoprocessing tasks are either conducted on servers or workstations, the required geoprocessing logic is provided by geoprocessing feeds that can be consumed like a typical newsfeed by humans or machines. The code is stored in Moving Code Packages that comply with the requirements from the previous section. All participants that actively perform geoprocessing can bind to one or more geoprocessing feeds and digest these Moving Code Packages.

Figure 3: Architectural overview.



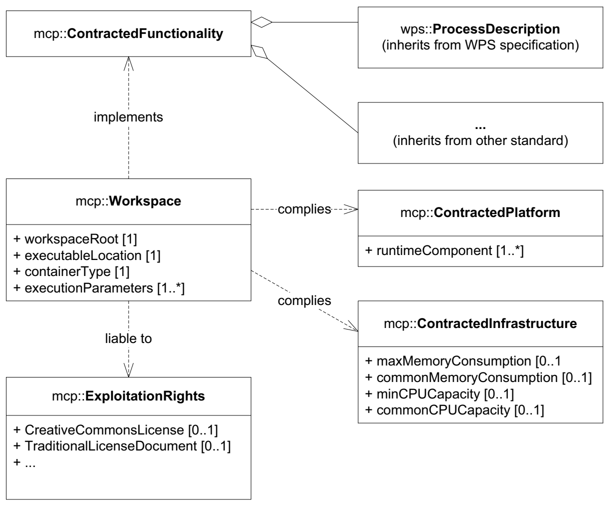
The presented approaches have been partly tested in a proof-of-concept implementation in the frame of the Open Geospatial Consortium's OWS-8 testbed and are further developed in the projects GLUES (Global Assessment of Land Use Dynamics, Greenhouse Gas Emissions and Ecosystem Services) and EO2HEAVEN (Earth Observation and Environmental modeling for the mitigation of Health risks).

3.1 Moving Code Packages

An initial approach to transport geoprocessing logic across system borders has been presented by Mueller et al. [9]. Their so-called Moving Code Packages are structured zip-files which contain an interoperable description of the processing logic and a compliant implementation. A mapping mechanism between the standardized functional description and a platform-specific interface is used to propagate data in and out. The application of standardized service interfaces satisfies the requirements 1) and 2) set out in section 2.1.

Figure 4 presents an enhanced Moving Code Package that enables the multi-level description and the statement of exploitation rights, which were proposed in section 2. The contracted platform describes high-level dependencies, such as 3rd party libraries and software packages that must be available at the target platform. The description of the contracted infrastructure allows testing the provided hardware resources to ensure a stable and possibly concurrent execution (requirement 3) in section 2.1).

Figure 4: A schematic view of an enhanced Moving Code Package (simplified)


mcp::ContractedFunctionality
wps::ProcessDescription (inherits from WPS specification)
... (inherits from other standard)
implements
mcp::Workspace
+ workspaceRoot [1]
+ executableLocation [1]
+ containerType [1]
+ executionParameters [1..*]
complies
mcp::ContractedPlatform
+ runtimeComponent [1..*]
complies
mcp::ContractedInfrastructure
+ maxMemoryConsumption [0..1
+ commonMemoryConsumption [0..1]
+ minCPUCapacity [0..1]
+ commonCPUCapacity [0..1]
liable to
mcp::ExploitationRights
+ CreativeCommonsLicense [0..1]
+ TraditionalLicenseDocument [0..1]
+ ...

The schema is designed for extensibility, allowing different functional representations of the provided logic. Currently we support the WPS standard as a basic means to communicate a declarative specification of the implemented logic but there are other standards like WSDL which is widely adopted outside the geo-community. Clearly also any of those extensions has to fulfill the requirements 1) and 2) from section 2.1. Functional descriptions may be provided in different formats and encodings. Different disciplines such as hydrology, environmental systems modeling, atmospheric or marine research may already have well established functional descriptions in place.

The contracted platform is formalized as a list of unique identifiers of software components that have to be installed at the target machine and thus form a runtime environment. Individual components are represented by unique identifiers which are further described at the component resolver shown in Figure 3. Components may also aggregate other components leading to more coarse grained runtime specifications. This facilitates the establishment of commonly available geoprocessing platforms and supplies PaaS providers with a registry of software environments that are commonly used for geoprocessing. It also provides a source of documentation for setting up a specific runtime environment required by a piece of processing logic.

The contracted infrastructure is described in terms of required CPU power and memory capacity as these factors are the primary determinants of a service-oriented operation in a target environment. When the deployment of a Moving Code Package is considered on the target platform, the execution engine may check for the availability of the required hardware resources and report warnings if it expects problems. Users may also decide to select resource-efficient implementations if multiple Moving Code Packages offer the same functionality.

An interoperable approach to the communication of exploitation rights is the Creative Commons license. This type of license is supported by the enhanced schema. However, the usual commercial license is not modularized or standardized, making it necessary to study legal texts. For such cases, a traditional license document may be referenced in the package description.

3.2 Geoprocessing Feeds

The Atom Syndication Format [7] is a lean extensible XML-based format for platform independent information exchange. It is applied to range of applications from simple exchange of text news, multimedia content or general data. News readers as well as other aggregator software may subscribe to Atom feeds and obtain new or updated content by a pull mechanism. The format's capabilities to exchange information in multiple representations for humans as well as machines and the provision of a simple versioning mechanism make it a suitable exchange vehicle for Moving Code Packages. Atom feeds delivering geoprocessing functionality are called geoprocessing feeds in this paper. The human-readable part of the feed's content allows users to study the provided functionality and pick those packages that suit their needs. The machine-readable part of the same feed can be invoked by processing services or GIS workstations (Figure 3). Additionally, after subscription these machines may occasionally synchronize with the geoprocessing feed to obtain updated logic.

For cloud environments geoprocessing feeds facilitate the distribution of logic across a network of processing nodes. The processing services in the cloud (Figure 3) provide well-known software platforms (listed in the component registry) and hardware. Their functionality is drawn from the original geoprocessing feeds outside the cloud. The proxy feed in the cloud aggregates all the functions that are to be distributed to the cloud's processing services. Additional to re-grouping or filtering the original feeds' contents, such a proxy feed may serve further purposes. Such an instance may offer a selection of processes that have passed additional security checks to ensure that the code is not harmful or that have been tested to deliver a certain Quality of Service (QoS) in the target environment. Hosting the Moving Code Packages in-house also increases resilience to external server outages.

4 Outlook

The outlined approach will help to better judge the fitness for purpose of geoprocessing logic. It is an important step towards the setup of a community repository of simulation and scientific computation logic and functionality as well as a powerful tool for a seamless and comprehensible documentation in geospatial data handling. Progressing in the design of well-defined descriptions of the geoprocessing logic is one of the future research challenges [2]. The four-dimensional contract for processing logic helps to share existing code with many users and removes obstacles that keep people from re-using each other's implementations. The recognition of legal issues provides a foundation to incorporate commercial participants into such a community.

The creation of a basis for such a community is our plan for the future. The community building process will be promoted based on a web platform approach like the Apple iTunes App Store or the Android Market. Providers of processing logic can publish their algorithms on such a marketplace, and all other users will be able to consume them. A key feature of this exchange platform is an integrated review process that allows all participants to comment and rate the published algorithms. Such a review process can be extended towards quality guarantees. The outlined concept already provides a lot of information that helps to create quality metrics for the functional description and the runtime behavior. An improved quality assessment that allows verifying these metrics also requires the integration of a testing environment. This requires test data to be shipped with the Moving Code Package. A virtual testing environment that is accessible though the community platform allows users to effortlessly run and test the algorithms. A new form of a trust layer needs to be implemented so that a newly submitted algorithm can be checked for containing viruses and/or other irregular malicious code.

A common description format and an exchange platform for computational logic can also facilitate the publication of scientific projects. Supplements of well-known journals already support the publication of related software. The proposed approach already contains the required information and maps easily to Software publishing styles of scientific journals.

Acknowledements

The research leading to these results has received funding from the German Federal Ministry of Education and Research under grant agreement n° 01LL0901C and the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 244100.

References

[1] Hal Abelson, Ben Adida, Mike Linksvayer and Nathan Yergler. ccREL: The Creative Commons Rights Expression Language. 2008.

[2] Johannes Brauner, Theodor Foerster, Bastian Schaeffer and Bastian Baranski. Towards a Research Agenda for Geoprocessing Services. In Proceedings of the 12th AGILE Conference, Hannover, 2009.

[3] Max Craglia, Michael Goodchild, Alessandro Annoni, Gilberto Camara, Michael Gould, Werner Kuhn, David Mark, Ian Masser, David Maguire, Steve Liang and Ed Parsons. Next-Generation Digital Earth: A position paper from the Vespucci Initiative for the Advancement of Geographic Information Science. International Journal of Spatial Data Infrastructures Research, 3, 2008.

[4] Thomas Erl. SOA Principles of Service Design: Prentice Hall International, 2007.

[5] Jim Gray. eScience: a transformed scientific method. In: Anthony J. G. Hey, Stewart Tansley and Kristin Michele Tolle, editors, The fourth paradigm: data-intensive scientific discovery, Microsoft Research, Redmond, WA, pages xvii-xxxi, 2009.

[6] Sören Haubrock, Falko Theisselmann, Henry Rotzoll and Doris Dransch. Web-based management of simulation models - concepts, technologies and the users' needs. In: R. S. Anderssen, R. D. Braddock and L. T. H. Newham, editors, Proceedings of the 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, pages 880-886, 2009.

[7] IETF. The Atom Syndication Format. 2005.

[8] ISO. Geographic information - Services. ISO 19119:2006, 2006.

[9] Matthias Müller, Lars Bernard and Johannes Brauner. Moving Code in Spatial Data Infrastructures - Web Service Based Deployment of Geoprocessing Algorithms. Transactions in GIS, 14(S1): 101-118, 2010.

[10] OGC. OpenGIS Web Processing Service, Version 1.0.0. OGC document 05-007r7, 2007.

[11] Stephen P. Prisley and Michael J. Mortimer. A synthesis of literature on evaluation of models for policy applications, with implications for forest carbon accounting. Forest Ecology and Management, 198(1-3): 89-103, 2004.

[12] W3C. Web Services Description Language (WSDL) Version 2.0. 2007.

[13] Hongji Yang and Xiaodong Liu. Software Reuse in the Emerging Cloud Computing Era: IGI Global, 2012.