

# KONZEPTION EINES MARKTPLATZES FÜR DEN AUSTAUSCH VON GEOPROZESSIERUNGSSIMPLEMENTIERUNGEN

Daniel Kadner, Matthias Müller, Johannes Brauner, Lars Bernard

**Zusammenfassung:** Die Wiederverwendung von Softwarekomponenten ist ein bekanntes Leitbild bei der Erstellung von effektiven und effizienten Anwendungen. Dieses Leitbild ist ebenso übertragbar auf die Erstellung von Geoprozessierungsalgorithmen, die in vielen wissenschaftlichen Projekten, Open-Source-Initiativen oder auch kommerziellen Produkten erzeugt und als ausführbarer Code implementiert werden. Auch hier gilt es mit möglichst geringem Zusatzaufwand den einfachen Austausch dieser Geoprozessierungslogik zwischen verschiedenen Instanzen zu ermöglichen. Dieser Artikel beschreibt ein Konzept, um Algorithmen über einen communitybasierten Marktplatz auszutauschen. Der vorgestellte Ansatz ermöglicht den Nutzern eines solchen Marktplatzes den einfachen Austausch von Algorithmen und Implementierungen und stellt Instrumente zur Qualitätssicherung bereit. Um das Angebot des Marktplatzes weiterzuentwickeln und kontinuierlich zu verbessern, werden Mechanismen für das Testen und Aktualisieren der angebotenen Algorithmen vorgeschlagen.

**Schlüsselwörter:** Moving Code, Geoprozessierung, Marktplatz, Community-Portal, GDI

## DESIGNING A MARKET PLACE FOR THE EXCHANGE OF GEOPROCESSING IMPLEMENTATIONS

**Abstract:** Software reuse is a well-known paradigm for building effective and efficient software systems. This paradigm is transferable for the creation of software algorithms in a geospatial context, as well. These algorithms and the respective executable implementations are produced in many scientific projects, open source initiatives or commercial products. A lightweight approach for exchanging these components between different instances is desirable. This paper presents a concept for exchanging algorithms on a community marketplace. The approach enables users to interchange their algorithms and perform basic quality management. To evolve the marketplace's offerings we also propose mechanisms for testing and updating implemented algorithms.

**Keywords:** Moving code, geoprocessing, market place, community portal, SDI

### Autoren

Dipl.-Medieninf. Daniel Kadner  
 Dipl.-Geogr. Matthias Müller  
 Dipl.-Geoinf. Johannes Brauner  
 Prof. Dr. Lars Bernard  
 Technische Universität Dresden  
 Professur für Geoinformationssysteme  
 Helmholtzstraße 10  
 01069 Dresden

E: [daniel.kadner@tu-dresden.de](mailto:daniel.kadner@tu-dresden.de)  
[matthias\\_mueller@tu-dresden.de](mailto:matthias_mueller@tu-dresden.de)  
[johannes.brauner@tu-dresden.de](mailto:johannes.brauner@tu-dresden.de)  
[lars.bernard@tu-dresden.de](mailto:lars.bernard@tu-dresden.de)

## 1 GEOPROZESSIERUNG

Viele der bereits heute vorhandenen Daten haben einen expliziten oder impliziten Raum- und Zeitbezug: Ortsbezogene Statistiken, Schadstoffmessungen, Abschätzungen zur Lärmbelastung, Flusspegel oder Verkehrszählungen. Solche unter anderem frei zugänglichen Geodaten (Open Geodata) haben ein enormes Mehrwertpotenzial für Bürger, Unternehmen und Entscheidungsträger sowie die Gesellschaft insgesamt. In vielen Fällen muss dieser Mehrwert, also die konkret benötigte Geoinformation, durch geeignete Datenverarbeitungsalgorithmen erst noch generiert werden. „Geoprocessing“ ist ein Sammelbegriff für eine Reihe von Verfahren beginnend bei der einfachen Datentransformation über Funktionen zur Datenverdichtung und Datenveredlung bis hin zu komplexen Analysefunktionen.

Diese Funktionen zur Geodatenverarbeitung sind meist Bestandteil heute eingesetzter Desktop-GIS-Anwendungen. Während es zwischen den unterschiedlichen Systemen eine Schnittmenge hinsichtlich des verfügbaren Funktionsumfangs gibt, bestehen zumeist Unterschiede im fachlichen Fokus (Hydrologie, Raumplanung, etc.) sowie der Art der Implementierungen (Rechengeschwindigkeit, Parametrisierbarkeit etc.).

Funktionalitäten zur Geoprozessierung werden auch als Internetdienste in Geodateninfrastrukturen (GDI) zur Weiterverarbeitung der angebotenen Geodaten zur Verfügung gestellt. Sind diese Dienste räumlich oder organisatorisch voneinander getrennt, so spricht man von einer verteilten Geoprozessierung (Brauner 2008, Friis-Christensen et al. 2007, Stollberg et al. 2007). Zusammen mit den bereits etablierten Internetdiensten zur Bereitstellung von Geodaten (Datendienste) haben diese verteilten Geoprozessierungsdienste (Prozessierungsdienste) einen hohen potenziellen Mehrwert, da sie GIS-Funktionen mit anderen, eventuell sogar fachfremden Systemen integrierbar machen und Ad-hoc-Analysen auf verteilten Datenbeständen ermöglichen. Dennoch stellen heutige Geodateninfrastrukturen kaum Prozessierungsdienste öffentlich bereit. Gründe sind beispielsweise:

1) In einer GDI können Datendienste und Prozessierungsdienste von unterschiedlichen Institutionen bereitgestellt werden.

Geodaten müssen dann erst zu den verarbeitenden Institutionen transportiert werden (Friis-Christensen et al. 2007). In heutigen GDI-Implementierungen findet sich überwiegend dieser stark datengetriebene Ansatz, meist in Kombination mit einer synchronen Datenkommunikation. Für datenintensive Anwendungen entstehen so zum Teil sehr hohe Netzlasten und Wartezeiten, die eine operationelle Anwendung verhindern können und sehr anfällig gegen Netzausfälle sind.

- 2) Die Verarbeitung von Geodaten ist meist rechenintensiv, belegt also entsprechend Hardwareressourcen und kostet Energie. Das erfordert beispielsweise die Einführung und Durchsetzung von Abrechnungsmechanismen (pay per use) oder eine rigorose Begrenzung von Rechenkapazitäten. Beides verringert die Attraktivität und Einsetzbarkeit von Prozessierungsdiensten.
- 3) Prozessierungsdienste sind durch ihren mehrschichtigen Aufbau nur aufwendig anzupassen und zu warten. Neben einer eventuell vorhandenen und durch den Entwickler des Diensts vorgegebenen, minimalen Möglichkeit zur Parameterübergabe an den Dienst, ist es aus der Sicht des Anwenders daher meist nicht möglich, diesen Dienst an die eigenen Bedürfnisse und eventuellen Gegebenheiten anzupassen.
- 4) Sicherheitsrelevante Daten, welche die Persönlichkeitsrechte berühren oder im Sinne der öffentlichen Sicherheit vertraulich sind, dürfen nur innerhalb einer bestimmten Institution transportiert und verwendet werden. Eine Übertragung zu einem räumlich entfernten Ort ist entweder nicht erlaubt oder nur unter Einhaltung hoher Sicherheitsstandards möglich.
- 5) Standardisierungsbestrebungen für Prozessierungsdienste (z. B. durch das OGC – Open Geospatial Consortium) sind noch nicht vollständig abgeschlossen; beispielsweise in Bezug auf Genauigkeit und Vollständigkeit der Prozessbeschreibung (u. a. formalisierte Semantik).
- 6) Im Gegensatz zu den beispielsweise durch die INSPIRE-Direktive (EC 2007) verpflichtend bereitgestellten Daten- und Visualisierungsdiensten besteht eine derartige Verpflichtung für

Prozessierungsdienste zurzeit nicht. Darüber hinaus ist die Nachfrage in der Praxis noch gering. In den aktuellen Entwicklungen zu Durchführungsbestimmungen für INSPIRE Spatial Data Services werden allerdings auch Prozessierungsdienste diskutiert, sodass hier möglicherweise bei gleichzeitig ausgereifteren Standards künftig eine verstärkte Nachfrage zu erwarten ist.

Ein Mechanismus zum einfacheren, systemübergreifenden Austausch von Geoprozessierungsalgorithmen würde zur Lösung der meisten genannten Probleme beitragen: Die Algorithmen können innerhalb der eigenen Infrastruktur veröffentlicht und auf eigener Hardware betrieben werden. Die Notwendigkeit des Datentransports zu entfernten Rechnerknoten entfällt, sicherheitsrelevante Daten können weiterhin intern prozessiert werden. Rechenintensive Prozesse können auf eigener oder gemieteter Infrastruktur laufen.

Bisher mangelt es an technischen Standards zum reibungslosen Austausch von Funktionalitäten zur Datenprozessierung in einer verteilten Umgebung (Hey et al. 2009, Müller et al. 2010, Yang et al. 2012). Gängige praktische Konzepte für Marktplätze verlangen bereits vorab eine Kopplung an bestimmte Plattformen (z. B. Apple AppStore für iOS oder Google Play für Android; myExperiment.org oder crowdlabs.org für scientific workflows), Produkte (CRAN für die Statistiksoftware R), Programmiersprachen oder domänenspezifische Sprachen (DSL – Domain-specific languages). Das Auffinden identischer Funktionen über Plattform- und Sprachgrenzen hinweg ist damit nicht vorgesehen.

Um die Idee eines freien und plattformunabhängigen Marktplatzes für den Austausch von Geoprozessierungsalgorithmen zu beschreiben, wird in diesem Beitrag auf die bereits im Vorfeld entwickelte Moving-Code-Idee (Müller et al. 2012, 2010) zurückgegriffen. Statt feingranular Abhängigkeiten zwischen verschiedenen Softwarebibliotheken abzubilden, ermöglicht das Konzept die plattformunabhängige Beschreibung von plattform-spezifischen Implementierungen. Damit versetzt es Nutzer in die Lage, eine für ihre Plattform passende Implementierung eines Algorithmus zu finden und dabei die funktionale Äquivalenz mit alternativen Implementierungen sicherzustellen.

## 2 GEOPROZESSIERUNGS-ALGORITHMEN AUSTAUSCHEN

Die Möglichkeit, bestehende Geoprozessierungsfunktionen in eigenen Umgebungen zu nutzen, setzt in der Regel das Vorhandensein einer bestimmten Geoprozessierungsplattform und eine entsprechenden Laufzeitumgebung voraus. Das Moving-Code-Konzept schafft eine Möglichkeit, diese konkrete Verknüpfung zu lösen und die Algorithmen unabhängig von deren Umgebung zu betrachten und austauschbar zu machen (Müller et al. 2012, 2010). Wird im Folgendem von Algorithmen gesprochen, so ist die Rede von *Implementierungen* der dargestellten Algorithmen. Das Prinzip des „Moving Code“ kapselt diese realisierten Algorithmen in Form von Paketen, die für verschiedenen Plattformen verfügbar sind, aber plattformunabhängig aufgefunden und ausgetauscht werden können.

Für die Beschreibung eines Algorithmus werden vier generelle Aspekte unterschieden: Funktionalität, Laufzeitumgebung (Plattform), benötigte Hardware und Nutzungsrechte. Eine Verschmelzung dieser Aspekte stellt die Beschreibung eines Moving-Code-Packages dar (siehe Abbildung 1).

Zur Beschreibung der *Funktionalität* müssen notwendige Eingabe- und Ausgabeparameter beschrieben werden, die zur Ausführung des Algorithmus benötigt werden. Als Ausgangspunkt für eine solche Schnittstellenbeschreibung wird die Web Processing Service (WPS) Spezifikation und hier das Schema der Prozessbeschreibung genutzt (Schut 2007).

Eine *Plattformbeschreibung* gibt Auskunft zu den softwaretechnischen Abhängigkeiten der Implementierung: Algorithmen werden in der Regel nicht von Grund auf neu programmiert, sondern nutzen bestehende Bibliotheken und Programmierschnittstellen für Datenimporte und Aufrufe grundlegender Informationen. Die Plattformbeschreibung kapselt Informationen zu benötigten Softwarekomponenten und Laufzeitumgebungen, die zur Ausführung des Algorithmus auf dem Zielsystem vorhanden sein müssen. Unter Einsatz eines Identifikatorsystems für derzeit gängige Softwareplattformen werden diese Beschreibungen kodiert. Im derzeitigen Stadium der hier vorgestellten Entwicklung kann u. a. eine Anbindung an die Spatial Analyst Tools von ArcGIS oder eine GDAL/Python-Umgebung erfolgen.

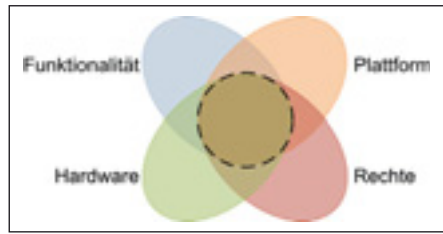


Abbildung 1: Kernaspekte des Moving-Code-Konzepts

Die *Hardwarebeschreibung* macht eine Vorabbeurteilung zur Lauffähigkeit des Algorithmus in einer bestimmten Infrastruktur möglich. Speicherverbrauch und Prozessorleistung sind hier die wesentlichen Merkmale. Binäre Implementierungen können zudem an bestimmte Architekturen (32 oder 64 bit) gekoppelt sein. Für rechenintensive, parallelisierte Algorithmen ist die Zahl der

Spezifikation des Open Cloud Computing Interface Standard (OCCI). Für die Rechtebeschreibung (*exploitation rights*) können neben eigenen Lizenzmodellen auch bereits existierende benutzt werden (z. B. Creative Commons). Für die Plattformbeschreibung (*contracted platform*) gestaltet sich die Formalisierung schwierig, da es hier an Standards fehlt. Aus diesem Grund wurden eigene Spezifikationen für Pakete und Paketbeschreibungen erstellt. Mittels dieser definierten XML-Schemata können insgesamt valide XML-Dokumente kreiert werden, welche die Implementierung des Algorithmus genau beschreiben und einen Austausch der Pakete zwischen verschiedenen Instanzen durchführbar macht. Diese selbstbeschreibenden Pakete werden als Moving-Code-Package bezeichnet.

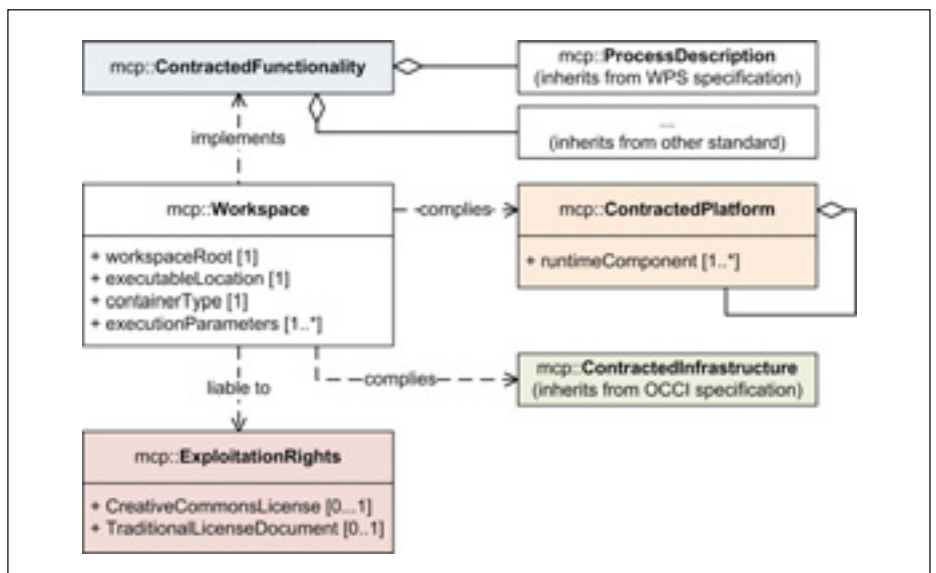


Abbildung 2: Schematische Beschreibung eines Moving-Code-Packages

Kerne interessant. Damit erfolgt die Hardwarespezifikation eng am OCCI-Standard für Rechnerknoten in Cloud-Umgebungen.

Lizenzaspekte werden von der Rechtebeschreibung abgedeckt. Hier kann spezifiziert werden, unter welchen Nutzungsbedingungen der Algorithmus veröffentlicht wird und welche Rechte bei der Anwendung durch Dritte bestehen.

Diese vier Bestandteile zur Beschreibung eines Algorithmus werden wie in Abbildung 2 zusammengefasst. Die Beschreibung der Funktionalität (*contracted functionality*) basiert auf der Spezifikation des aktuellen WPS (1.0.0) und die der Hardware (*contracted infrastructure*) auf der

Da nicht der Code direkt eingebettet wird, sondern *Workspaces* „verpackt“ werden, können in den Algorithmen Datenquellen eingebettet sein. Das ist insbesondere dann sinnvoll, wenn das Verfahren auf unveränderliche Daten (z. B. ein Höhenmodell oder ein Flussnetz) zugreift.

## 3 EIN GEOPROZESSIERUNGS-MARKTPLATZ

Der hier vorgestellte Marktplatz ist eine webbasierte Plattform beziehungsweise ein Portal zum Austausch von Geoprozessierungsalgorithmen in einer größeren Community. Grundkonzept ist das Publish-Find-Bind-Prinzip aus den serviceorien-

tierten Architekturen (siehe Abbildung 3): Nutzer von Algorithmen können danach verschiedene Anbieter und deren angebotene Algorithmen für die gesuchte Funktionalität finden.

Der Anbieter publiziert oder registriert einen neuen Algorithmus auf dem Geoprozessierungsmarktplatz. Ein Nutzer kann auf dem Markt nach einem auf seine Problemstellung passenden Algorithmus suchen. Bei erfolgreicher Suche entscheidet er sich für die Nutzung eines Algorithmus' aus der Ergebnismenge („Binden“).

Die verschiedenen Akteure und Bestandteile im Kontext eines Marktplatzes sind im Wesentlichen der Markt, die Entwickler der Algorithmen, die Anwender, Prozessierungsdienste (siehe Abbildung 4) und – im erweiterten Kontext – Datenserver.

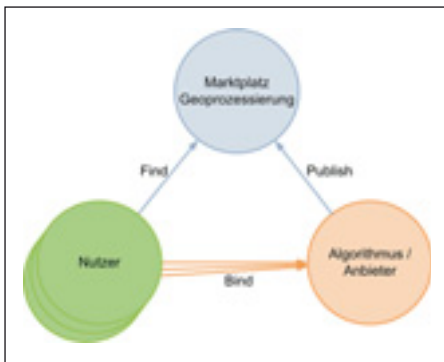


Abbildung 3: Publish-Find-Bind-Prinzip Marktplatz Geoprozessierung

Die zentrale Komponente ist der Markt selbst. Dieser stellt im Sinne einer Registry die Funktionen zum Verwalten der Algorithmen zur Verfügung. Das Ziel ist es dabei, einen Bestand an Algorithmen zur Geodatenverarbeitung aufzubauen. Zum Austausch von Algorithmen bzw. den zugehörigen Geodaten werden die im vorigen Absatz beschriebenen Moving-Code-Packages genutzt. Mithilfe der Beschreibung zu den benötigten Laufzeitumgebungen wird es dem Nutzer ermöglicht, zwischen beispielsweise ressourcenintensiven und ressourcenschonenden Algorithmen zu unterscheiden.

Ein reiner Markt dient allein dem Austausch von Algorithmen zwischen Anbietern und Nutzern. Ein erweiterter Markt (im Sinne eines Community-Portals) bietet Entwicklern und Nutzern auch die Möglichkeit, sich über Anforderungen, Er-

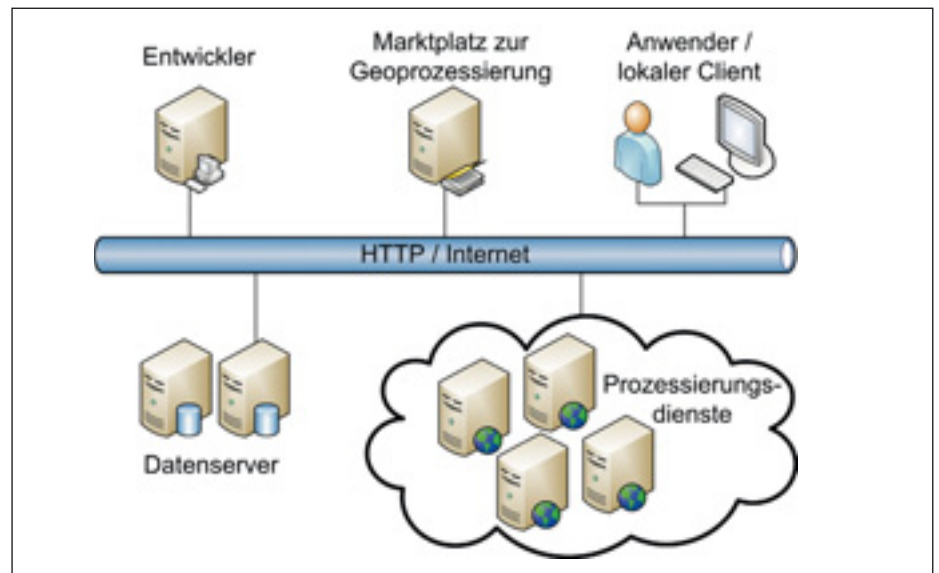


Abbildung 4: Akteure und Bestandteile eines Marktplatzes

fahrungen und Ideen auszutauschen und dadurch neue Produkte zu generieren oder bestehende Produkte zu verbessern. Im Folgenden werden die entsprechenden Erweiterungen beschrieben.

### 3.1 QUALITÄTSSICHERUNG

Die Verfügbarkeit qualitativ hochwertiger Algorithmen entscheidet über Erfolg und Misserfolg eines Marktplatzes. Ohne eine ausreichende Sicherstellung der Qualität der eingebrachten Algorithmen ist das Modell eines Community-basierten Ansatzes nicht attraktiv. Erst ein Mindestmaß an Qualität führt zur Akzeptanz der Anwendung bei den Nutzern (Davis 1993, Ahn et al. 2007). Die grundlegendste Art der Qualitätssicherung beinhaltet die automatische Überprüfung der eingereichten Algo-

rithmen auf Schadsoftware. Erst bei bestandener Prüfung kommt eine Publikation oder Weiterverbreitung überhaupt erst infrage. Eine nächste Stufe in der Qualitätssicherung ist eine Begutachtung der Algorithmen durch gewählte Experten. Wichtige Prüfkriterien sind Vollständigkeit und Aussagekraft der Beschreibung (Was berechnet/analysiert der Algorithmus? Wofür kann er verwendet werden?), die Korrektheit der zugesicherten Laufzeitumgebung sowie der stabile Betrieb für die angegebenen Inputdaten. Gerade in der Aufbauphase für den anvisierten Produktiveinsatz des Marktplatzes ist diese Maßnahme als sinnvoll zu betrachten. Bei steigender Anzahl verfügbarer Algorithmen oder häufigen Verbesserungen skaliert dieses Verfahren jedoch recht schlecht: Die manuelle Überprüfung



Abbildung 5: Beispiele für Review-Systeme (Quellen: a) amazon.de, b) holidaycheck.de)

erfordert sowohl Zeit als auch Fachwissen. Eine hohe Veröffentlichungsquote verlangt damit auch viele Experten.

Besser skalieren Community-basierte Review-Systeme, wie sie bereits im Umfeld von vielen Webanwendungen bekannt sind (siehe Abbildung 5). Communities bestehen aus einer Vielzahl von Individuen welche entweder Inhalte beisteuern oder diese aktiv nutzen. Beide Gruppen können Qualitätsbewertungen aus ihrer jeweiligen Sicht abgeben. Obwohl sie nicht die Kohärenz eines Expertengremiums in ihren Bewertungen erreichen, funktionieren sie in der Praxis dennoch recht gut.

Abbildung 5a) zeigt das wohl bekannteste Beispiel des Amazon-Onlineshops. Hier kann ein Nutzer eine Bewertung für einen Artikel abgeben sowie diesen ebenfalls kommentieren. Allerdings lässt sich das hier angewandte aggregierte Bewertungsschema schlecht interpretieren: Was bedeuten fünf Sterne? Daher wird eine detailliertere Bewertungsschema favorisiert (siehe Abbildung 5b). Denkbare Kriterien sind beispielsweise Rechenperformanz des Algorithmus, effizienter Ressourcenverbrauch oder dass die Entwicklung für eine vom Rezensenten bevorzugte Softwareplattform erfolgt. Hier gilt es in künftigen Arbeiten geeignete Rubriken zu definieren.

**3.2 TESTEN DER ALGORITHMEN**

Neben der Qualitätssicherung durch Automatismen, Experten sowie die Community ist die Möglichkeit, angebotene Algorithmen auszuprobieren, ein weiterer wichtiger Bestandteil des erweiterten Marktplatzes. Eine durch den Marktplatz bereitgestellte Testumgebung (Sandbox) erlaubt die manuelle Begutachtung der angebotenen Algorithmen für kleinere, exemplarische Anwendungsfälle. Die Sandbox ermöglicht

den Anwendern einen angebotenen Algorithmus zu testen, ohne diesen gleich in die eigene Infrastruktur transferieren zu müssen. Dafür können sowohl vorbereitete Testdatensätze als auch vom Nutzer definierte externe Datenquellen eingebunden werden. Hier sind Entwickler zu motivieren, in die Moving-Code-Packages auch geeignete Testdatensätze bzw. Verweise auf entsprechende Datendienste zu integrieren. Anwender, die eigene Daten als Grundlage für Tests nutzen, sollten diese Testdaten auch anderen Anwendern sowie den Entwicklern zur Verfügung stellen können. Das Vorhandensein offenen Quellcodes bietet erfahrenen Benutzern die Möglichkeit, die angebotenen Algorithmen im Detail zu analysieren. Die Anwendung eines in der Softwareentwicklung üblichen Fehlerprotokollierungssystem (bug tracking) bietet zusätzlich die Möglichkeit, die bei der Begutachtung und Ausführung entdeckten Fehler und weitere Verbesserungsvorschläge zu verwalten und an die Entwickler zu kommunizieren. Neben der reinen Fehlerbehebung kann ein solches System auch zur allgemeinen Qualitätssteigerung für Algorithmen auf dem Marktplatz eingesetzt werden.

**3.3 BENACHRICHTIGUNGEN**

Mechanismen zur Benachrichtigung haben auf dem Marktplatz vier Funktionen:

1. sie informieren über neu erschienene Algorithmen;
  2. sie informieren über aktualisierte Algorithmen;
  3. sie informieren über die Stabilität der Algorithmen;
  4. sie schlagen relevante Algorithmen vor.
- Die Punkte 1. bis 3. beziehen sich auf den mehrphasigen Lebenszyklus eines Moving-Code-Packages (Abbildung 6). Sobald

ein Paket einen stabilen Zustand erreicht hat, kann es von anderen Anwendern genutzt werden. Durch den Entwickler nachfolgend vorgenommene Veränderungen am Algorithmus (etwa Fehlerbeseitigung, Performanzverbesserung, Ressourcenschonung) führen zu einer Aktualisierung des Moving-Code-Packages. Werden Pakete gar nicht angenommen oder nicht weiter gepflegt, können sie ggf. vom Marktplatz entfernt werden.

Ein Benachrichtigungssystem muss den Nutzer eines Algorithmus über Aktualisierungen informieren. Das Benachrichtigungssystem des hier konzipierten Marktplatzes besteht aus zwei Bestandteilen. Eine einfach realisierbare Form der Benachrichtigung findet durch einen Newsletter statt. Hierbei werden alle innerhalb eines zeitlich definierten Rahmens veröffentlichten oder aktualisierten Anwendungen aufgeführt und per E-Mail an die Nutzer gesendet. Der Nutzer hat dabei die Möglichkeit, diesen Newsletter zu individualisieren, das heißt nur für ihn interessante Thematiken aufführen zu lassen, oder auch komplett abzubestellen. Neben dieser Option kann der Nutzer ebenso Nachrichten in Form von Atom-Feeds abonnieren. Auch diese können individualisiert und an die Anforderungen des Nutzers angepasst werden. Ein Atom-Feed benachrichtigt den Anwender unmittelbar bei der Anpassung des Algorithmus durch den Entwickler über diese Veränderung und stellt somit eine Verbesserung des Newsletters dar, der nur in periodischen Abständen vom System erzeugt wird. Zusätzlich beinhalten die Atom-Feeds maschinenlesbare Beschreibungen der Moving-Code-Packages. Damit können physische Prozessierungsinstanzen wie interne Dienste oder lokale Softwareinstallationen über einen Pull-Mechanismus automatisch aktualisiert werden (Müller et al. 2012). Die Atom-Feeds sind damit insbesondere zur Verteilung von Algorithmen auf dezentrale Prozessierungsknoten geeignet.

Ein letzter Bestandteil des Benachrichtigungssystems sind Empfehlungssysteme (Resnick et al. 1997, Schafer et al. 1999, Adomavicius et al. 2011, McSherry 2005). Ein sehr bekanntes Beispiel für ein solches System ist ebenfalls bei Amazon zu finden. Es werden auf Basis der Auswertung der Benutzermodelle von anderen Nutzern Empfehlungen für einen bestimmten Nutzer ausgesprochen („Kunden, die

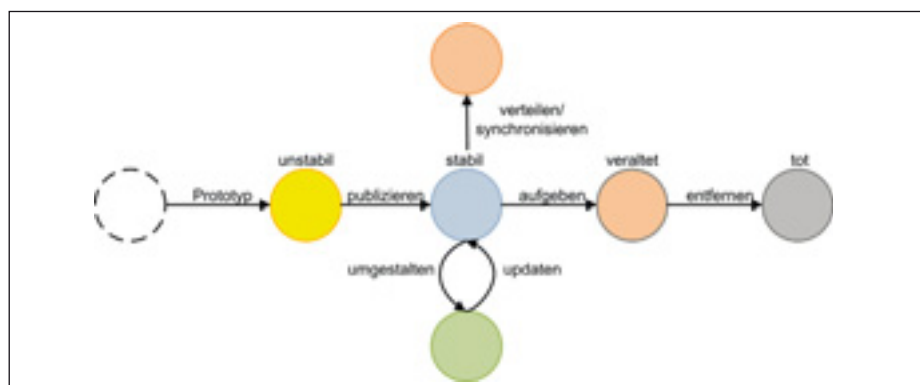


Abbildung 6: Lebenszyklus eines Moving-Code-Packages

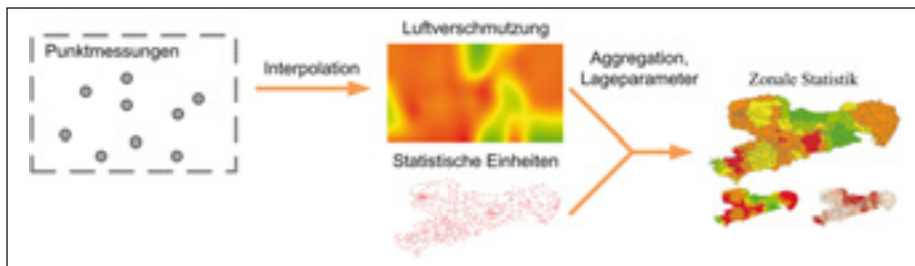


Abbildung 7: Anwendungsbeispiel Umweltstatistik in EO2HEAVEN

diesen Artikel gekauft haben, kauften auch:"). Neben diesem auf den Nutzerprofilen basierenden Filter der Vorschläge ist auch ein inhaltsbasiertes Vorgehen denkbar. Dabei werden die bereitgestellten Algorithmen auf mögliche Ähnlichkeiten und Verknüpfungen hin untersucht und bewertet. Beide Ansätze erlauben es, den Nutzer auf Grundlage seiner aktuellen Auswahl an Algorithmen auf weitere für ihn möglicherweise relevante Algorithmen hinzuweisen (Sarwar et al. 2001).

Als Distributionswege sind sowohl Newsletter als auch Atom-Feeds geeignet. Darüber hinaus kann das Empfehlungssystem als alternatives Discovery-Instrument auf der Website des Marktplatzes eingesetzt werden.

#### 4 PROTOTYPISCHE UMSETZUNG UND PRAKTISCHE ANWENDUNG

Im Rahmen des Projekts EO2HEAVEN (<http://www.eo2heaven.org>) wurden Teile der hier vorgestellten Konzepte prototypisch realisiert und Anwendungsbeispiele entwickelt. EO2HEAVEN zielt auf die Untersuchung des Zusammenhangs zwischen beobachtbaren Umweltparametern und möglichen Effekten für die menschliche Gesundheit und hier insbesondere auf die Entwicklung von geeigneten Informationssystemen für diese Analysen. Beispielhaft werden dazu Szenarien für Sachsen und Durban (Südafrika) zur Untersuchung von Luftqualität und Atemwegs- bzw. Herz-Kreislauferkrankungen entwickelt.

Um mögliche Effekte der Luftverschmutzung, u. a. durch Verkehr, Industrie und Haushalte zu untersuchen, werden entsprechende Schadstoffausbreitungskarten benötigt. Durch entsprechende Interpolationsverfahren werden die Schadstoffausbreitungskarten auf Grundlage öffentlich verfügbarer Umweltmessdaten, ergänzender Fernerkundungsdaten und weiterer Geodaten erzeugt (Wiemann et al.

2012). Diese Schadstoffausbreitungskarten werden genutzt, um Risikokarten für eventuell betroffene Bevölkerungsgruppen wie beispielsweise Asthmatiker zu erstellen. Weiterhin werden die interpolierten Schadstoffverteilungen mit vorhandenen Gesundheitsdaten über das Auftreten von Atemwegserkrankungen abgeglichen und auf eventuell existierende Korrelationen untersucht. Die Gesundheitsdaten werden dabei von den Krankenkassen in einer anonymisierten Form für eine statistische Einheit

tion, kann innerhalb des Marktplatzes für Geoprozessierungsalgorithmen nach entsprechenden Algorithmen gesucht werden. Mittels eines im Projekt lokal aufgesetzten Web Processing Service, der in diesem Szenario als ein Client für den Marktplatz fungiert, können die gewünschten Algorithmen über entsprechende Atom-Feeds abonniert und für die Benutzung heruntergeladen werden. Als lokale WPS-Instanz wurde die WPS Referenzimplementierung von 52°North genutzt. Die zur Konfiguration des WPS von 52°North bereitgestellte Weboberfläche wurde dahingehend erweitert, dass eine einfachere Integration der Moving-Code-Packages möglich ist. Abbildung 8 zeigt einen Ausschnitt dieser Anpassungen, in welchem bereits ein hier „Remote Repository“ genanntes Atom-Feed hinzugefügt wurde.

Die WPS-Erweiterung benötigt zur Ausführung die Definition eines entfernten Aufbewahrungsorts für einen Geoproces-

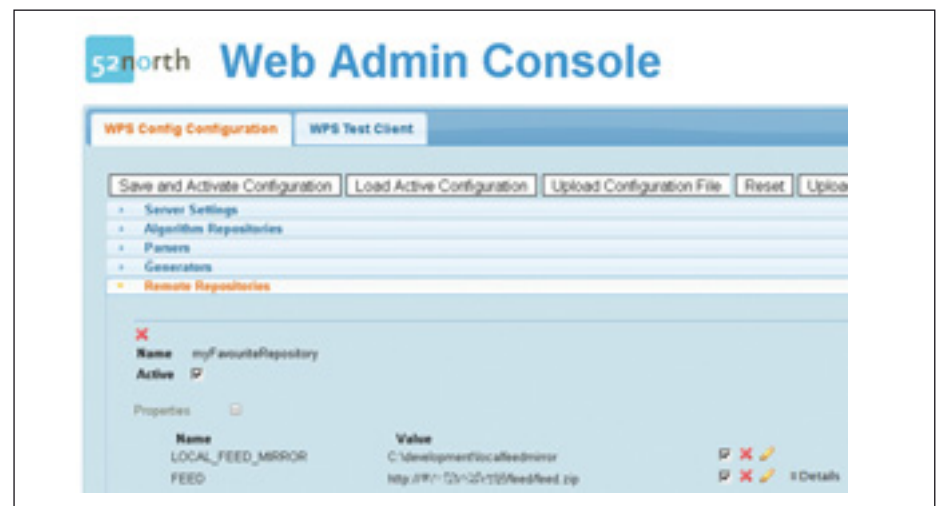


Abbildung 8: Webbasierte Konfigurationskonsole des 52°North WPS

bereitgestellt. Je nach Grad der Anonymisierung sind die Daten auf Landkreis- oder Gemeindeebene aggregiert. Die ermittelten Schadstoffausbreitungen müssen daher ebenfalls für diese statistischen Einheiten aggregiert werden. In weiteren Schritten werden aus dieser zonalen Statistik und weiteren Gesundheitsdaten dann Risikokarten für potenziell betroffene Bevölkerungsgruppen erzeugt. Abbildung 7 zeigt den Ablauf dieser Geoprozessierungsschritte im Überblick.

Für die darin benötigten Verfahren, wie der Interpolation und der räumlichen Aggrega-

tion-Feed. Hier reicht die Angabe des entsprechenden Netzwerkpfads und der Algorithmus wird automatisch in ein lokal angegebenes Verzeichnis kopiert. Auf dieses Verzeichnis besitzt die lokale WPS-Instanz Zugriffsrechte und kann dadurch den darin enthaltenen Algorithmus als eigenen Dienst anbieten. Die manuelle Eingabe der Netzwerkadresse wird innerhalb der nächsten Entwicklungsiteration durch ein halbautomatisches Verfahren ergänzt, sodass es auch möglich wird, einen spezifischen WPS mit neuer Prozessierungsfunktionalität direkt aus dem Marktplatz heraus anzureichern.

5 FAZIT UND AUSBLICK

Die hier präsentierten Konzepte zeigen die grundlegenden Aspekte eines Marktplatzes für Algorithmen zur Geoprozessierung. Auf Basis der Moving-Code-Idee erlaubt ein solcher Marktplatz anders als eine heutige GDI die Algorithmen zu den Geodaten zu bringen, und so speziell bei den meist datenintensiven Geonanwendungen die Prozessierungslogik auszutauschen, ohne große Datenmengen über das Netz transportieren zu müssen.

Erste prototypische Umsetzungen belegen die Machbarkeit, aber die Arbeiten stehen noch am Anfang und es bleiben zahlreiche Arbeitsgebiete für die weitere Forschung und Entwicklung; beispielhaft genannt seien:

- ▶ Ansätze für einen standardisierten Austausch von algorithmischer Funktionali-

tät, etwa durch eine verbesserte Formalisierung für Geooperatoren (Semantik).

- ▶ Geeignete Beschreibungs- und Suchstrategien für die Geoprozessierungsalgorithmen.
- ▶ Exaktere und standardisierte Beschreibung der benötigten Laufzeitressourcen.
- ▶ Verbesserte Nutzbarkeit in Cloud-Infrastrukturen, als Bereitstellung entsprechender Infrastruktur (Infrastructure as a Service – IaaS) oder ausführender Plattformen (Platform as a Service – PaaS).
- ▶ Intelligente Kombination mit Systemen zur Offline-Nutzung.
- ▶ Automatische Auslieferung aktualisierter Algorithmen an die Abonnenten.

Die Summe dieser Arbeiten dient der Entwicklung geeigneter Strategien, um für die in Zukunft immer umfangreicher zu

erwartenden Open-Geodata-Angebote rasch effektive Geoprozessierungswerkzeuge bereitstellen zu können. Durch die Möglichkeit, Algorithmen zu den Daten transportieren zu können, wird die Analyse von sensiblen Daten (z. B. von Gesundheitsdaten) innerhalb von Sicherheitsbereichen als ein weiterer Einsatzbereich erschlossen. Dies erlaubt eine effiziente Analyse und damit erst die Generierung der eigentlichen Mehrwerte aus diesen Angeboten.

*Ein Teil der hier beschriebenen Forschungsarbeit wurde durch das 7. EU-Forschungsrahmenprogramm (FP7/2007-2013) gefördert (grant agreement no. 244100).*

Literatur

Adomavicius, G.; Tuzhilin, A. (2011): Recommender Systems Handbook. Edited by Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P. B. Springer US, Boston, MA.

Ahn, T.; Ryu, S.; Han, I. (2007): The impact of Web quality and playfulness on user acceptance of online retailing. In: Information & Management, 44, S. 263-275.

Brauner, J (2008): Anbindung von GIS-Funktionalitäten an eine Geodateninfrastruktur über eine Web Processing Service Schnittstelle. In: GIS – Zeitschrift für Geoinformatik, 3/2008, S. 18-25.

Craglia, M.; Goodchild, M. F.; Annoni, A.; Camara, G.; Gould, M.; Kuhn, W.; Mark, D.; Masser, I.; Maguire, D. (2008): Next-Generation Digital Earth: A position paper from the Vespucci Initiative for the Advancement of Geographic Information Science. In: International Journal of Spatial Data Infrastructures Research, 3/2008, S. 146-167.

Davis, F. (1993): User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. In: International Journal of Man/Machine Studies, 38, S. 475-487.

EC (2007): Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for

Spatial Information in the European Community (INSPIRE). <http://eur-lex.europa.eu/JOHtml.do?uri=OJ:L:2007:108:SOM:EN:HTML>.

Friis-Christensen, A., Ostländer, N., Lutz, M., Bernard, L. (2007): Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions. In: Transactions in GIS, 11 (6), S. 799-818.

Hey, T.; Tansley, S.; Tolle, K. (2009): The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, Redmond, WA.

McSherry, D. (2005): Explanation in Recommender Systems. In: Artificial Intelligence Review, 24, S. 179-197.

Müller, M.; Bernard, L.; Brauner, J. (2010): Moving Code in Spatial Data Infrastructures – Web Service Based Deployment of Geoprocessing Algorithms. In: Transactions in GIS, 14, S. 101-118.

Müller, M.; Kadner, D.; Bernard, L. (2012): Moving Code – Sharing Geospatial Computation Logic on the Web. In: Proceedings of the 15th AGILE International Conference on Geographic Information Science, Avignon, 2012, S. 1-6.

Resnick, P.; Varian, H. R. (1997): Recommender systems. In: Communications of the ACM, 40, S. 56-58.

Sarwar, B.; Karypis, G.; Konstan, J. (2001): Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, S. 285-295.

Schafer, J.; Konstan, J. (1999): Recommender systems in e-commerce. In: Proceedings of the 1st ACM conference, S. 158-166.

Schut, P. (2007): Open Geospatial Consortium Inc. OpenGIS ® Web Processing Service. [http://portal.opengeospatial.org/files/?artifact\\_id=24151](http://portal.opengeospatial.org/files/?artifact_id=24151).

Stollberg, B.; Lutz, M.; Ostländer, N.; Bernard, L. (2007): Geoprozessierung in Geodateninfrastrukturen – Aufgaben für die nächste Generation. In: GIS – Zeitschrift für Geoinformatik, 4/2007, S. 24-29.

Wiemann, S.; Richter, S.; Karrasch, P.; Brauner, J.; Pech, K.; Bernard, L. (2012): Classification-driven air pollution mapping as for environment and health analysis. In: Proceedings of the 6th International Congress on Environmental Modelling and Software, Leipzig, 2012.

Yang, H.; Liu, X. (2012): Software Reuse in the Emerging Cloud Computing Era. Idea Group, U.S.