
Studienprojekt: Entwurf und Implementierung eines Workflows zur Untersuchung von Hochwasserereignissen

Eduard Gette (610533)
Student

Dr.-Ing. Daniel Eggert
Betreuer

Prof. Dr. Doris Dransch
Betreuerin

Kontext des Projektes

Dieses Studienprojekt findet im Rahmen des „Digital Earth“ - Projektes der acht Helmholtz-Zentren des Forschungsbereichs Erde und Umwelt statt.

Erklärtes Ziel des „Digital Earth“ - Projektes ist es die Entwicklung eines holistischen Erdsystemverständnisses durch Integration von Daten und Wissen unterschiedlicher Disziplinen der Erdwissenschaften voranzutreiben.

Zur Erreichung dieses Ziels werden unter Einsatz aktueller Konzepte und Methoden aus der Informatik und der Datenwissenschaft, Software-Komponenten zur Unterstützung integrativer, datengetriebener Wissenschaften entwickelt. Die Schwerpunkte liegen dabei auf „Smart Monitoring“ und „Integrative Data Exploration“. [3]

Im Fokus dieses Studienprojektes steht die Entwicklung eines Software-Moduls für den „Digital Earth Flood Event Explorer“, einer Anwendung, welche die Untersuchung der Entstehung, Entwicklung und Auswirkungen von Flutereignissen ermöglichen soll.[2]

Anforderungen

An das zu erstellende Modul wurden zu Beginn des Projektes die folgenden Anforderungen festgelegt:

1. Der Quellcode des Moduls soll in der Programmiersprache „Python“ verfasst sein.
2. Die Routine des Moduls soll vom Nutzer über einen Knopf im Benutzer-Interface gestartet werden können.
3. Die Routine soll die folgenden Teilschritte umfassen. Alle Arbeitsschritte sind wenn möglich im Arbeitsspeicher ohne Zwischenspeicherung auf dem Harddrive zu erledigen.
 - (a) Erstelle eine Liste aller verfügbaren Flutereignisse auf der Webseite des Copernicus Emergency Management Service (Rapid Mapping) [1]. Die verfügbaren Flutereignisse sollen gefiltert werden können. Die Filter werden in der Konfiguration des Moduls festgelegt.
 - (b) Erfasse für jedes Flutereignis alle auf der Website des Flutereignisses verfügbaren Daten, insbesondere Meta-Daten und Download-Links zu verfügbaren Produkt-Downloads.
 - (c) Aktualisiere/ Erstelle alle Datenspeicher auf der GeoServer-Instanz gemäß den erfassten Flutereignissen. Die GeoServer-Instanz wird in der Konfiguration des Moduls festgelegt. Die für die Erstellung/ Aktualisierung notwendigen Dateien werden aus den entsprechenden Produkt-Downloads herausgefiltert und alle nicht benötigten Dateien werden verworfen. Bei den gewünschten Dateien handelt es sich

um Shapefiles, welche Umrisspolygone für alle überfluteten Bereiche enthalten.

- (d) Für jeden aktualisierten/ neuen Datenspeicher sollen die Ortsdaten mit Hilfe eines Digitalen Geländemodells (DGM) um Höhendaten ergänzt werden. Anschließend soll für jeden so ergänzten Datensatz ein neuer Datenspeicher erstellt werden. Sprünge in den Höhendaten sollen mit Hilfe eines Kernels geglättet werden können. Die Punktdichte der Umrisspolygone soll mindestens der Auflösung des DGM entsprechen. Der Kernel wird in der Modul-Konfiguration festgelegt.

4. Der Benutzer soll im Benutzer-Interface über den Fortschrittsverlauf der Routine informiert werden.

Implementierung

In diesem Abschnitt werden die wichtigsten Details zur Implementierung, insbesondere wichtige Klassen und die Architektur, erläutert. Anschließend werden eventuelle Abweichungen bezüglich der spezifizierten Anforderungen dargestellt.

Architektur

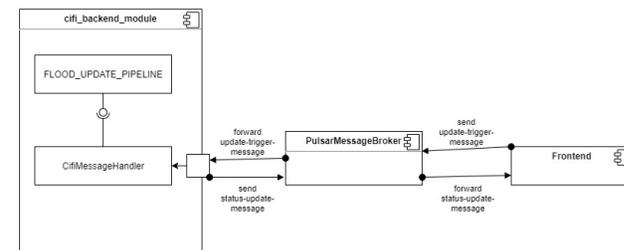


Abbildung 1: Schematischer Überblick über das neue Modul und seine Anbindung an das Frontend

In [Abbildung 1](#) ist zu erkennen, dass das neue Modul sich einfach durch Registrierung bei der aktuellen Instanz des Pulsar Message Brokers zu dem bestehenden System hinzufügen lässt. Die Routine wird hierbei durch eine Service Anfrage aus dem Frontend ausgelöst. Die Weitergabe der Fortschrittsberichte erfolgt in analoger Art und Weise durch die Übergabe eines ProgressReport - Objektes an die Instanz der FLOOD_UPDATE_PIPELINE.

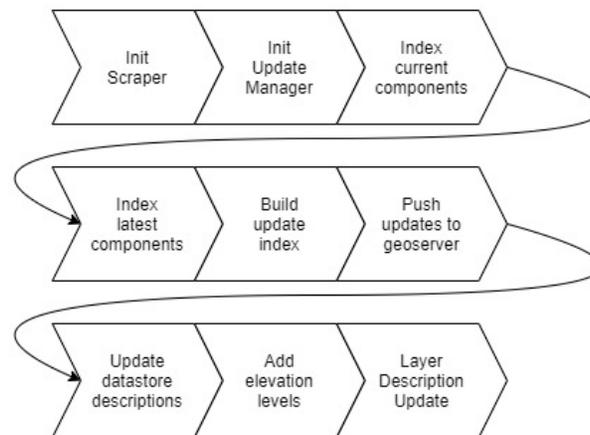


Abbildung 2: Schematischer Überblick über alle Stufen der FLOOD_UPDATE_PIPELINE

[Abbildung 2](#) zeigt, dass bei der Implementierung der gewünschten Routine eine einfache lineare Pipeline-Struktur verwendet wurde, bei der jede Stufe eine klar definierte Teilaufgabe erledigt und anschließend ihre Ergebnisse an die nachgelagerte Stufe weiterreicht. Im Detail besteht die aktuelle Pipeline aus den folgenden acht Stufen:

1. Init Scraper: Prüft die Verbindung zu der EMS Webseite von Copernicus und fragt eine Liste der die jeweiligen Filter verfügbaren Flutereignisse ab
2. Init Update Manager: Initialisiert das UpdateManager-Objekt und prüft die Verbindung zum Geoserver
3. Index current components: Fragt alle in dem jeweiligen Arbeitsbereich auf dem Geoserver verfügbaren Datenspeicher an und baut mit Hilfe der in der Beschreibung gegebenen Meta-Informationen einen Index über alle vorhandenen Flutereignisse auf
4. Index latest components: Erstellt einen Index über alle verfügbaren Flutereignisse durch Abfrage aller verfügbaren Informationen für alle Einträge in der in (1) erstellten Liste
5. Build update index: Erstellt einen Index der zu aktualisierenden Datenspeicher durch Abgleich der Listen aus (3) und (4)
6. Push updates to geoserver: Lädt alle relevanten Produkte herunter, lädt gewünschte Dateien im Arbeitsbereich des GeoServers hoch und erstellt ggf. neue Datenspeicher
7. Update datastore descriptions: Hinterlegt die relevanten Meta-Informationen in den Beschreibungen der aktualisierten/ neuen Datenspeicher
8. Add elevation levels: Reichert die Umrisspolygone der Flutereignisse wie in Abschnitt (Anforderungen 3.d) erläutert um Höheninformationen an
9. Layer Description Update: Kopiert die Meta-Daten in den Beschreibungen der aktualisierten

Datenspeicher in die Beschreibungen der zugehörigen Layer, sodass diese vom Frontend abgerufen werden können

Wichtige Klassen

1. Stage: Abstrahiert eine Stufe in der Pipeline. Erhält mindestens einen Namen und eine Funktion, welche die Funktionalität der Stage darstellt als Argumente. Durch Übergabe einer Vorverarbeitungsfunktion können die Outputs der Vorgängerstufe beliebig angepasst werden. Die Übergabe einer Evaluationsfunktion erlaubt die Bewertung ob die Stage erfolgreich ausgeführt werden konnte.
2. LinearPipeline: Erlaubt die Aneinanderreihung mehrerer Stages und deren sequentielle Ausführung. Sollte eine der Stages nicht erfolgreich ausgeführt werden, so bricht die Pipeline ab und ihr Output entspricht dem Output der zuletzt erfolgreich ausgeführten Stage.
3. CopernicusScraper: Erlaubt erleichterten Zugriff auf alle relevanten Ressourcen auf der Copernicus EMS Webseite, z.B. die Anfrage von Produkt-Downloads oder verfügbarer (Flut-)Ereignisse
4. FloodUpdateManager: Regelt alle für die Aktualisierung notwendigen Indexierungsvorgänge und prüft die Eignung von Produkt-Downloads als Updates
5. GeoserverWeb(Coverage/Feature)Service: Erleichtern die Abfrage von WCS/ WFS des GeoServers durch Automatisierung der Konfiguration, z.B. Ermittlung der Reihenfolge der Achsen des Referenzsystems, und weitere Funktionalitäten

Bekannte Abweichungen von der Spezifikation

1. In der „Add elevation levels“ Stufe ist es momentan notwendig ein temporäres Verzeichnis anzulegen, in welches die für die Erstellung der aktualisierten Shapefile notwendigen Dateien zwischengespeichert werden, bevor sie auf dem GeoServer hochgeladen werden. Dieses temporäre Verzeichnis wird direkt nach Ende des Prozesses vollständig gelöscht.

Konfiguration

Zur Steuerung der Funktionsweise der FLOOD_UPDATE_PIPELINE stehen folgende Parameter zur Verfügung, welche im CifiMessageHandler im Attribut `_cifi_pipeline_config` als Key-Value Paare gesetzt werden können:

1. `base_url`: [str] URL der Startseite des Copernicus EMS
2. `activations_list_url`: [str] URL unter welcher die Liste der verfügbaren Aktivierungen von Ereignissen zu finden ist
3. `activation_filters`: [dict]
 - (a) `affected_countries`: [list[str]] Default: [] Liste von Länderabkürzungen. Bestimmt von welchen Ländern Flutereignisse in die Ereignisliste aufgenommen werden sollen. Wenn dieser Filter nicht gesetzt wird, werden Ergebnisse für alle verfügbaren Länder ausgegeben
 - (b) `event_time_utc_start / event_time_utc_end`: [str] Default: ". Format „JJJJ-MM-TT“. Filter zur Beschränkung des Zeitraums für welchen Ergebnisse ausgegeben werden sollen.

Weglassen des Startwertes führt zur Ausgabe aller Ergebnisse ab dem ersten Eintrag bis zum Endwert, Weglassen des Endwertes führt zur Auflistung aller Ergebnisse ab dem Startwert bis zum aktuellen Tag + 1

- (c) field_event_type_tid: [int] Default-ID: 17 (Flutereignisse). Filter zur Bestimmung welche Art von Ereignissen in der Ergebnisliste ausgegeben werden sollen. Mögliche Aktivierungstypen mit zugehöriger ID:
- i. 'Drought': 617,
 - ii. 'Epidemic': 618,
 - iii. 'Extreme temperature': 616,
 - iv. 'Humanitarian': 520,
 - v. 'Infestation': 619,
 - vi. 'Mass movement': 580,
 - vii. 'Storm': 597,
 - viii. 'Transportation accident': 620,
 - ix. 'Volcanic activity': 615,
 - x. 'Wildfire': 535,
 - xi. 'Forest fire, wild fire': 16,
 - xii. 'Flood': 17,
 - xiii. 'Windstorm': 18,
 - xiv. 'Earthquake': 19,
 - xv. 'Industrial accident': 20,
 - xvi. 'Other': 21
- (d) activation_status: [str] Default: 'ALL'. Mögliche Werte: 'ALL', '0', '1' ('0' steht für Open, '1' für Closed) Filter bestimmt welchen Aktivierungs-Status Ergebnisse haben dürfen.
- (e) title: [str] Default: ". Engt die Suche je nach in title_op spezifizierter Operation ein. Nur

Aktivierungen mit einem Titel für welchen die Operation den Wert 'WAHR' annimmt werden ausgegeben.

- (f) title_op: [str] Default: 'contains'. Für weitere mögliche Werte siehe Copernicus EMS Webseite.
- (g) geoserver_url: [str] URL der Geoserver-Instanz
- (h) username: [str] Benutzer-Name zum Login auf Geoserver-Instanz
- (i) password: [str] Passwort zum Login auf Geoserver-Instanz
- (j) workspace: [str] Ziel-Arbeitsbereich auf Geoserver-Instanz
- (k) digital_elevation_model: [str] vollständiger Titel des Digitalen Geländemodell-Layers auf der Geoserver-Instanz, welches für die Anreicherung mit Höhendaten verwendet werden soll. Bsp: 'digitalearth:de_dem25m'
- (l) enforce_xy_order: [bool] Vom Geoserver durch WFS abgerufene Daten sind konform mit dem EPSG-Standards und achten somit deren Achsen-Reihenfolge. Einige verwendete Bibliotheken gehen jedoch von einer festen X-Y (bzw. Longitude-Latitude) Reihenfolge aus. Dieser Parameter erzwingt X-Y-Achsenreihenfolge wenn True.
- (m) min_recency_in_days: [int] Kann an Stelle der Filter in (b) verwendet werden, um nur Ergebnisse der letzten n Tage anzuzeigen. VORSICHT: die Filter in (b) haben Vorrang!

Screenshots

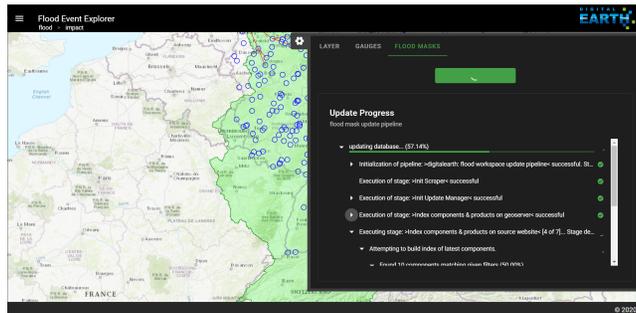


Abbildung 3: Einbettung des Update-Knopfes in den Flood Event Explorer und Beispiel Fortschritts-Bericht

Literatur

- [1] Copernicus Emergency Management Service.
<https://emergency.copernicus.eu/index.html>.
- [2] Digital Earth: Flood Showcase.
<https://www.digitalearth-hgf.de/en/pages/workflow-show-case-flood>.
- [3] DigitalEarth at a Glance. <https://www.digitalearth-hgf.de/en/pages/project>.