Hierarchical process profiles for interoperable geoprocessing functions

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

Abstract

Process profiles for geoprocessing services have the potential to serve as a starting point for the definition of interoperable geoprocessing functions. Potential applications are process cataloguing, retrieval and comparison. Although process profiles are an integral part of the Web Processing Service (WPS) specification, they have found little uptake in spatial data infrastructures.

This paper presents a novel framework for the definition of process profiles. It is shown that there are different notions of a process profile which can be distinguished and related in a hierarchical structure. The upper levels in this hierarchy are used to specify the functionality of the process. Concrete interfaces, as proposed by the WPS specification, are considered at lower levels. A hierarchical profiling approach shows how concrete implementations can be derived from abstract functional definitions. Additionally, the hierarchical levels allow simple comparison tasks between different processes.

Based on that framework, previous works on process profiles are analysed and valued, highlighting differences and complementary concepts. Subsequently, a gap analysis proposes some modifications to current standards that are required for wide-range implementation. Furthermore, thoughts on possible applications of process profiles including search, retrieval and data provenance modelling are presented.

Keywords: Geoprocessing, WPS, Process profile, Functionality, Spatial Data Infrastructures

1 Introduction

Geoprocessing means processing of spatial or geo-referenced data and any geoinformation system (GIS) provides geoprocessing functions. Reuse and comparability of existing functions to assemble larger workflows as well as the detection of proper analysis functions is a persisting topic in GIS science and application. In service-oriented geoprocessing, process profiles, as suggested by the Web Processing Service (WPS) specification, are generally useful to standardise, reference and catalogue common processing functions. They can be simply used for searching particular processes or to compare two different processes in order to assess whether they provide the same functionality. However, previous work on this topic has shown major shortcomings in the original definition of a process profile and several authors have identified additional requirements to improve applicability and usefulness of the concept [8, 12, 18].

This paper presents a framework for contract-based process profiles and suggests a hierarchical approach to process definitions with different levels of abstraction. In parallel, the proposed concepts separate concerns related to the general functional definition of a process (i.e. what the process actually does and how it works) and interoperability arrangements for implementers of processing services and clients (i.e. agreements on technical formats for data exchange). A general framework is developed that specifies processing functions at four different levels of abstraction and specifies the associated inheritance mechanisms.

The paper is organised as follows: The next section lays out process profiles, highlighting differences, advantages and complementary solutions. By taking the current set of standards a gap analysis is conducted to identify existing implementation issues for an effective use of process profiles.

2 Functional contracts and process descriptions: From abstract definitions to concrete implementations

Four levels of abstraction for geoprocessing functions are defined in this section: Concepts, abstract profiles, concrete profiles and process implementations. These hierarchical levels and their properties are illustrated with an exemplary specification of a process that calculates the Normalised Differenced Vegetation Index (NDVI). This process is slightly more complex than a basic geo-operator (i.e. like those defined in Tomlin’s map algebra [17]) and helps highlight critical design decision in the specification process for process profiles. The NDVI is a commonly used indicator that can be used to assess the presence of live green vegetation in remote sensing images [16]. It is defined as follows:

\[
NDVI(\text{NIR}, \text{RED}) := \begin{cases} 
0, & \text{NIR} + \text{RED} = 0 \\
\frac{\text{NIR} - \text{RED}}{\text{NIR} + \text{RED}}, & \text{NIR} + \text{RED} \neq 0
\end{cases}
\]

with NIR and RED being defined as reflectance values in the near infrared and red spectrums.

Such a mathematical definition or otherwise formal or definite description specifies the generic concept of some process on a very abstract level. Nothing is yet said about the structure of the input and output data which makes the concept suitable for domain-agnostic specifications of
functionality. The concept must be placed in a persistent, traceable location so that derived profiles can make reliable references. In practice, a concept should not just contain mathematics but also provide further explanatory material to illustrate applications and contexts for use.

At the next level the abstract profile of a process is specified by its generic interface or structure, i.e. the specification of input and output elements. Depending on the domain-specific implementation of a certain function there may be different conceptual data models that directly affect the number and the type of the input data (Figure 1).

For our NDVI process there are two (and possibly much more) abstract profiles. Obviously the process can be described as having two inputs (near infrared and red measurements) and one output (the calculated NDVI value). A different interface specification may assume a commonly used data product (i.e. from a remote sensing data provider) as an input. An NDVI process specified for Landsat imagery takes a multi-spectral Landsat TM scene as an input. This product specific implementation performs additional pre-processing to acquire “apparent” reflectance values from the regular product (Landsat multispectral imagery is composed of 2D arrays of digital numbers ranging from 0 to 255 which need to be converted to reflectance values prior to any further computation). In this case the process takes a single input containing the required reflectance values but also provides a similar output as in the former case. In the given example the generic and product-specific profiles share the same output type definition.

The strong interconnection between the number and type of process arguments requires that abstract profiles provide abstract data type definitions for these arguments. For the generic NDVI profile, red and near-infrared input could be treated as 2-dimensional coverages. Both coverages may have an arbitrary domain and co-domain as long as they are identical for both inputs. The LandsatTM5 profile depends on a more complex multispectral data model imposed by the provider of Landsat imagery. It contains not just data for the two desired bands but also contains data for additional spectral bands. For both profiles the output can be specified as a single-valued (decimal), 2-dimensional coverage with a value range of [-1, 1].

With abstract data types and domain-specific consideration in place it is possible to specify the process even further by adding a so-called contract, which has been proven useful in object-oriented and service-oriented software component development ([2, 10]). In brief, a contract is a binding agreement between a component provider and consumer that explicitly lists obligations in terms of pre- and postconditions for a component invocation and may specify additional information, e.g. the provided precision of the calculation. For abstract profiles, the description of the concrete calculation procedure including pre- and postconditions for input and output data are stored in the contract document. Similarly to a concept, it must also be stored in a persistent, traceable location.

Concrete profiles directly extend generic interfaces. They specify concrete data types and formats for data encoding and transfer and represent interoperability arrangements that enforce the utilisation of particular, usually standardised, data
formats for process inputs and outputs (Figure 2). A concrete profile is a binding agreement between service providers and service consumers. For providers, it is an immediate template and interface definition for a process implementation. In return, clients can use a concrete profile to prepare processing requests to an arbitrary WPS implementing that profile.

A process implementation realises a concrete profile. It must implement the specified interface and also comply with the contract document and abstract data types defined by the abstract profile (Figure 3). However, an implementation may extend the interface in ways that do not interfere with the original contract or interface syntax. One such case could be the addition of further optional parameters, which trigger an alternative behaviour of the process. But the most frequent case for this extension mechanism is probably the support for additional input and output data formats which are not covered by the profile (Figure 4, NDVI Process A). In a very similar way, process implementations may realise two or more concrete profiles. In this case these concrete profiles must stem from the same abstract profile (Figure 4, NDVI Process B). The complete hierarchical inheritance mechanism for process profile is given in the Annex.

3 Comparison with related work

Some early ideas on process profiles have been presented by Nash [12]. In his work he discusses the structuring of URN namespaces for process profiles with the examples of basic GIS functions and (possibly complex) domain-specific specialised functions. These process profiles are immediately declared at a concrete level and are specific about the required technical data formats. Nash also discusses inheritance among profiles but in a different way: For process profiles that share many common elements and only diverge in a few, he suggests some sort of “truncated profile” that contains only the shared parameters. Since this approach focusses on syntax rather than behaviour, such generalisation could easily yield degraded process definitions which have apparently no functional purpose. (For the above NDVI profiles in Figure 2 such a trunk profile would have no inputs at all and only contain the shared output specification.) In contrast, this paper suggests starting from a functional definition, the concept, and then progressing towards abstract and concrete profiles. This

Figure 3: Relationship between concrete profile and process implementation.

Figure 4: Extension mechanism for process implementations. Process implementations may implement more than one concrete profile or also extend the original interface in non-harmful ways.
chain of steps will always yield purposeful process implementations. In his paper, Nash also suggests an inheritance mechanism that allows extending a profile by additional supported data formats. This requirement and has also been realised in the presented concept for concrete profiles and process implementations.

Taking the example of 3D terrain data processing and analysis, Lanig & Zipf [8] suggest an ordering scheme for geoprocessing functions. Similarly to Nash they provide some URN templates to structure and identify process profiles, but do not specify inheritance mechanisms. A profile is defined in a sense that resembles to what this paper defines as a concrete profile, containing ready-to-use blueprints for process interfaces. In a similar way Walenciak & Zipf [18] introduce the idea of grouped profiles that collect individual process profiles in domain-specific process “compilations”. This would make a WPS instance to a single point of access for problem-specific geoprocessing functions. A WPS would then indicate at the service level (i.e. in the service capabilities) that it features a particular set of well-known domain-specific functions. Clearly this idea does not interfere with the approach taken in this paper but rather discusses an orthogonal concept. In their conclusion the authors point out the inability to relate a mathematical model or formula to a particular, data-dependent spatial analysis algorithm. – An issue that has been dealt with in this paper.

An engineering report form the Open Geospatial Consortium (OGC) [14] and Foerster & Schäffer [4] highlight the importance of a comprehensible textual description in a process profile for inspecting the semantics of a particular process. They suggest the viewpoints from the Open Distributed Processing Reference Model (RM-ODP, [7]) to specify additional XML elements for in a process description. Since RM-ODP was rather designed for developing distributed systems and infrastructures but not individual services this approach is arguabe. None of these documents provides an example so it remains unclear what problem is actually solved. Although RM-ODP does mention service contracts and behaviour, it is not specific how it should be specified. However, the initial point of having additional human-readable process documentation is still valid and was incorporated into the proposed concept on hierarchical process profiles.

In the semantic web services domain, several authors have highlighted the need for specifying the behaviour of processing services to enable qualified searches for particular functions (e.g. [3, 9]). The definitions of functions, properties and relationships as well as associated data types are declared in domain ontologies. This resembles the idea of abstract or conceptual data types for abstract profiles. Theoretically, ontologies can be used to characterise a profile. Moreover, ontological approaches may become useful to relate process profiles considering various aspects of them and thus extend the rather static ordering approaches suggested by [8, 12]. The downside is that pure ontology-based behavioural specification of processes will not be an option for process description and retrieval in the near future. One issue is the lack of completeness in the required domain and data type ontologies. Probhibitively time-consuming reasoning in the discovery process for suitable processing services is another. Therefore Anselin [1] suggests putting more effort in research on structured metadata describing processing functions. Nevertheless, the previous works highlight the need for specifying the behaviour of processing services to enable qualified searches for particular functions and to provide a reference for checking the correctness of service implementations. This idea is endorsed in the proposed framework.

4 Gap analysis: Standardisation requirements for using process profiles

Concepts and abstract profiles contract the functionality or behaviour of a process and are a central piece to create trust in process implementations. They may thus foster reuse of processing services by pointing to a commonly agreed functional contract. Since a uniform standardised way of writing down a functional contract is currently not in sight, these contracts have to be captured in human-readable documents which have to be carefully written with precision and comprehensibility in mind. Such a written contract document would be an integral part of a process profile and can be unambiguously referenced. Multiple process implementations, that intent to support this contract, may use this reference and can thus be considered equal at the given level of abstraction.

Similarly to a registry on process profiles, a registry on abstract data types is required that serves as a stable reference for abstract data type formalisations and definitions. Existing approaches on databases and registries for coordinate reference system (CRS) definitions use URNs, URLs or well-known text (WKT, [15]) representations for this purpose. Progressing on a standardised approach to abstract data type definitions is crucial to achieve machine-readable abstract profiles. Considering the INSPIRE data specification efforts, domain-specific process profiles can already reference the developed data specifications [5]. A repository of well-known analysis processes for INSPIRE data would make a perfect INSPIRE geoprocessing use case. For more generic specifications the data models provided in the ISO 19100 series would make a good starting point for further formalisation.

Since the discussion about process profiles has a strong link to WPS, the proposed concept should have a technical implementation within this standard. Realising multiple concrete profiles in one process implementation is generally foreseen by the current WPS specification. The next step would be the support for profiles that reside on different hierarchical levels. This could be achieved by either augmenting the current profile element with additional attributes or by shifting the declaration of profiles to the process metadata section, which supports role-specific URL references. Here, the role attribute could be used to identify the hierarchical level of a referenced profile. To realise an abstract profile specification for WPS processes, an abstract process description structure must be implemented in the WPS standard. In contrast to a regular process description, this structure must tolerate missing details about input and output data. In addition, it is necessary to specify an inheritance mechanism in future WPS specifications to ensure navigability between the different levels of abstraction in a hierarchy of process profiles.
Process profiles can also be used in metadata records to document the lineage of processing results. Expressing the result of a processing workflow as a chain of process profiles (and their original data inputs) is a realisation of a tree-like provenance structure as stipulated by ISO 19115-2 [6] or Open Provenance Model (OPM, [11]). Currently, for geodata, much of the detailed information on processing steps is buried inside cumbersome metadata elements. The ability to reference process profiles can be used to produce clearer, leaner, and less ambiguous metadata statements by keeping detailed information about the applied algorithms in the profile description and focussing on functional contracts in the metadata record. With hierarchical profiles, inspectors of provenance information can decide whether they wish to inspect the detailed functionality of an abstract profile (i.e. for data verification) or simply choose to look at the concept to get a broad sense of how the data was computed. Both OPM and the current ISO 19115-2 metadata standard would support references to process profiles through unique identifiers, i.e. through URLs. Ideally, these references should point to the abstract profile which provides sufficient information about the applied processing algorithms.

5 Summary and outlook

This paper has presented a novel approach for profiling geoprocessing services. Instead of focussing on one particular interpretation of a profile, a hierarchical framework was developed that defines the functionality of a process at different levels of abstraction. The declaration of a detailed contract in a process profile is more than an implementation guideline for process providers and consumers: In representing a reliable agreement on the provided functionality, process profiles with contracts have the chance to increase trust in using third party implementations and thus spur the use of processing services. Besides, process profiles offer interesting opportunities for the following applications:

- Process search and retrieval of processing services by the inspection of abstract profiles and concepts as well as a perspective towards catalogues for processing functionality spatial data infrastructures
- The definition of reliable interoperability arrangements between service providers and consumers through concrete profiles
- Automatic generation metadata records for processing workflows; effective and efficient representation of data provenance
- Validation and sanity checks of processing workflows by inspection of concepts and abstract profiles
- Definition of standardised “added value” processing functions for standardised data products as e.g. defined by INSPIRE
- Support for conceptual modelling in processing workflows through abstract profiles

Since the creation and maintenance of process profiles incorporate both service providers and consumers some governance mechanism has to be established. Desired and provided functionality may evolve or be eventually replaced; possibly occurring ill-defined contracts will have to be retreated and substituted by corrected versions. Suitable governance mechanisms will have to be established to meet these requirements. Depending on the size and the composition of the intended user group standardisation bodies or semi-formal communities could be governing institutions.

Last not least, hierarchical process profiles help to address critique often made of the WPS specification which is its syntactic generality paired with a lack of semantic interoperability. Process profiles that contract the behaviour of a process rather than just defining the syntactic interface provide these additional semantics and may thus help to overcome the interoperability issue.

Acknowledgements

The research leading to these results has received funding from the German Federal Ministry of Education and Research under grant agreement n° 01LL0901C.

References


## Annex

Descriptive elements declared at each hierarchical level. An *X* marks the initial definition and use of an element, *restrict* and *extend* indicate alterations of elements at inferior levels.

<table>
<thead>
<tr>
<th>Element</th>
<th>Hierarchical Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Concept</td>
</tr>
<tr>
<td>Concept definition</td>
<td>X</td>
</tr>
<tr>
<td>document</td>
<td></td>
</tr>
<tr>
<td>Functional contract</td>
<td>X</td>
</tr>
<tr>
<td>documentation</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>X</td>
</tr>
<tr>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>Abstract</td>
<td></td>
</tr>
<tr>
<td>Metadata</td>
<td>X</td>
</tr>
<tr>
<td>Input</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>X</td>
</tr>
<tr>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>Abstract</td>
<td></td>
</tr>
<tr>
<td>Metadata *</td>
<td>X</td>
</tr>
<tr>
<td>Multiplicity</td>
<td>X</td>
</tr>
<tr>
<td>Data Format</td>
<td></td>
</tr>
<tr>
<td>Maximum Size</td>
<td>X</td>
</tr>
<tr>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>X</td>
</tr>
<tr>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>Abstract</td>
<td></td>
</tr>
<tr>
<td>Metadata *</td>
<td>X</td>
</tr>
<tr>
<td>Multiplicity **</td>
<td>X</td>
</tr>
<tr>
<td>Data Format</td>
<td></td>
</tr>
<tr>
<td>Maximum Size</td>
<td>X</td>
</tr>
</tbody>
</table>

* also references abstract data type  
** output multiplicity is not available in WPS 1.0 but planned for WPS 2.0  
*** additional optional input (output)