

# Automatic workflow composition in the geospatial domain: an application on sea-level rise impacts analysis

Samih Al-Areqi  
Institute of Computer Science  
University of Potsdam  
Potsdam, Germany  
alareqi@uni-potsdam.de

Anna-Lena Lamprecht, Tiziana Margaria  
Lero – The Irish Software Research Centre  
University of Limerick  
Limerick, Ireland  
anna-lena.lamprecht/tiziana.margaria@lero.ie

## Abstract

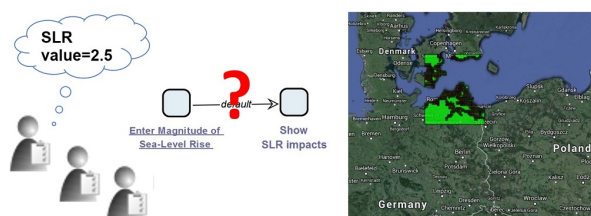
The geospatial domain has recently seen a trend towards migrating data analysis software processes from predefined static systems to purpose-specific compositions of existing services, often in the form of workflows. However, the technicalities involved in a correct and adequate workflow design and service composition frequently impose great challenges on researchers working in the geospatial application domain, especially if they are not IT experts or trained programmers. The PROPHETS plugin of the jABC workflow modeling framework facilitates synthesis-based (semi-) automatic composition of services into workflows and can thus help to reduce the amount of technical knowledge that is required for this task. In this paper, we show how it can be applied for the design of workflows for analyzing the impacts of sea-level rise. We also address the preceding domain modeling, which comprises the design of adequate services and the provisioning of semantic meta-information about the services in terms of service interface descriptions, as well as ontology models for the classification of the types and services in the domain.

*Keywords:* scientific workflows, automatic workflow composition, domain modeling, geospatial services, climate impact analysis

## 1 Introduction

With plenty of general-purpose geospatial services (such as data access services, portrayal services, and data manipulation services) and specific-purpose services designed to address particular geospatial applications available, it can be very hard and time-consuming for users to identify adequate (combinations of) services manually. As an illustrative example, consider the scenario depicted in Figure 1, where an arbitrary user wants to analyze the impacts of a sea-level rise of 2.5 m for a particular region. He knows what the initial data in this situation is (magnitude of sea-level rise, and the region in question) and what he wants to see in the end (for instance a map showing the flooded areas), but he does not know which computational steps are needed to carry out the analysis that would yield this result. A possible approach to overcome this situation is the use of semantics-based, (semi-) automatic workflow design techniques, which require the available services to be annotated with machine-readable metadata, and are then able to automatically derive workflows for a given specification.

Figure 1: Example scenario.



Most workflow management systems used in the geospatial domain, such as GeoJModelBuilder [17] and

Kepler [5], only simplify the (manual) workflow composition process syntactically. Learning how to apply these technologies to build a system based on services remains complex for application experts, in particular with the interoperability challenges of geospatial data. Recently, different attempts were undertaken towards semantic simplification through automatic geospatial service composition using AI planning and synthesis techniques [3, 15], and several works used OWL and OWL-S techniques to describe the functional capabilities of geospatial services [4, 16]. The successful application of all these techniques for (semi-) automatic workflow composition depends on an adequate domain modeling, that is, the provisioning of appropriate meta-information about the involved technical entities (services, data types) of the target domain. Ontologies or taxonomies are frequently used structures to represent and organize this information [2, 8].

In our work, we combine an intuitive graphical formalism for the manual composition of services into workflows with additional functionality that allows for making use of a synthesis algorithm to combine services automatically according to an abstract specification, again embedded in a very intuitively usable plugin. In this paper we focus on the application of this framework on an example that deals with the (semi-) automatic composition of workflows for the analysis of the impacts of sea-level rise. Since no application-specific ontological models about the services and types in this domain were available, we designed the required taxonomies ourselves. In contrast to many related approaches, which either support OGC or not, they comprise OGC-compliant as well as non-OGC terms. Thus, the domain model enables the user to consider a greater range of user objectives, perspectives and input/output preferences during the synthesis process.

The paper is structured as follows: Section 2 introduces the jABC workflow modeling framework including the PROPHETS plugin. Section 3 demonstrates how PROPHETS can be applied on the example of workflows for the analysis of sea-level rise impacts, that is, how the domain model is designed, and how workflows can be specified and synthesized. Finally, Section 4 discusses conclusions and plans for future work.

## 2 The jABC Framework and PROPHETS

The multi-purpose process modeling and execution framework jABC [14] is the current reference implementation of the eXtreme Model-Driven Design (XMDD) paradigm [11]. The service concept of jABC is very close to an intuitive understanding of service that requires to be ubiquitously accessible (location-agnostic) and mechanically configurable [6]. The term “service” is also used within jABC to refer to its functional building blocks (the so-called „SIBs”), which are viewed as independent from their location, the program entity, and hardware-platform that provides them. The jABC provides a comprehensive and intuitive graphical user interface in which users easily develop workflow applications by composing SIBs into hierarchical (flow-) graph structures (called Service Logic Graphs, or SLGs), which are executable models of the application. The workflow development process is furthermore supported by a set of plugins providing additional functionalities.

One of the plugins providing additional functionality to the jABC is PROPHETS (Process Realization and Optimization Platform using Human-readable Expression of Temporal-Logic Synthesis) [12], which follows the loose programming paradigm [7] to facilitate semantics-based

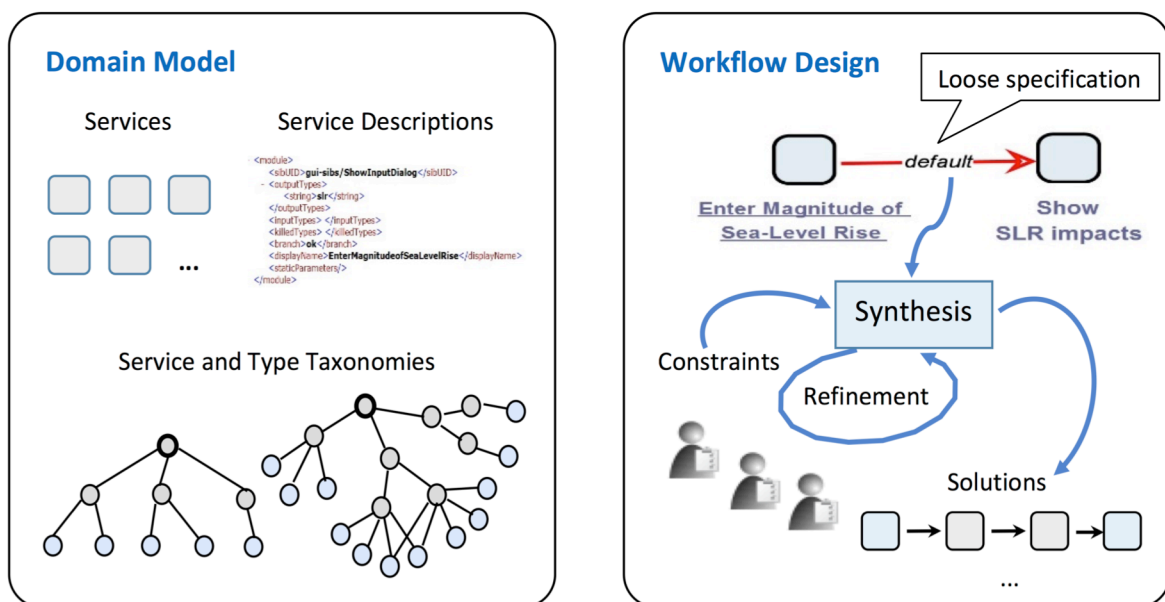
semi-automatic workflow design in addition to manual workflow construction.

As shown in Figure 2, working with PROPHETS consists of basically two phases: *domain modeling* and *workflow design*. In the domain modeling phase, domain experts provide resources (services, data) and the corresponding metadata such as service descriptions and taxonomies. In the workflow design phase, the workflow designer can then mark one or more branches between SIBs as loosely specified and apply the synthesis framework provided by PROPHETS to replace them by appropriate concrete service sequences. He can also define additional constraints to be taken into account by the synthesis. For this purpose, PROPHETS provides a constraint editor with natural-language constraint templates that users can apply without having knowledge of the underlying logic. PROPHETS automatically transforms the specification into a Semantic Linear Time Logic (SLTL) [13] formula that is the input of the synthesis algorithm.

## 3 Example: Synthesis of Workflows for Assessing Impacts of Sea-Level Rise

Analyzing and assessing potential impacts of climate change are critical and challenging tasks that require the processing of large and heterogeneous datasets. These analyses are particularly demanding because of the multi-scale and multi-objective nature of environmental modeling for climate change impact assessment [9]. For the example of sea-level rise (SLR) that we focus on in this paper, climate change is assessed with respect to the potential loss of agricultural production, calories available and effect for food security, but also with respect to properties of rural and urban damage functions. In this section, we discuss how we

Figure 2: Automatic workflow design with PROPHETS.



used PROPHETS to apply techniques for semi-automatic workflow design to sea-level rise impact analysis. First, Section 3.1 describes the design of a domain model for the SLR impact applications, before Section 3.2 focuses on performing the actual SLR workflow design.

### 3.1 Domain Modeling

The domain modeling comprises the provisioning of services and their descriptions in terms of inputs and outputs, the design of taxonomies for classifying the types and services in the domain, and the formulation of constraints.

**Services:** The first step in setting up the domain model for the SLR impacts analysis workflows was to turn the SLR tools used in the ci:grasp climate information platform (<http://www.cigrasp.org>) into services adequate for (re-)use as SIBs in the jABC workflow framework. We used the jETI (Java Execution Tool Integration) platform [10] for the service integration, which was a straightforward process once the desired functionalities had been identified.

The semantic meta-information about the services is stored in a separate file, `modules.xml`, where each module corresponds to one service entity as seen from the perspective of the synthesis algorithm. Each module is linked to a concrete SIB that provides the implementation, but the modules' names and the names used to describe input and output data types for the synthesis algorithm are symbolic. In addition to the modules we have defined based on the standard SIB libraries of jABC (such as the data input modules `Define-area-coordinates` and `Enter-magnitude-of-sea-level-rise`), we defined 82 different modules based on 22 SLR services (SIBs). Table 1 shows the names, descriptions, and input/output information for 12 of the modules.

**Taxonomies:** Service and type taxonomies are used to provide abstract classifications for the terms used in the module descriptions, which are in particular useful for the formulation of constraints about groups of data types or

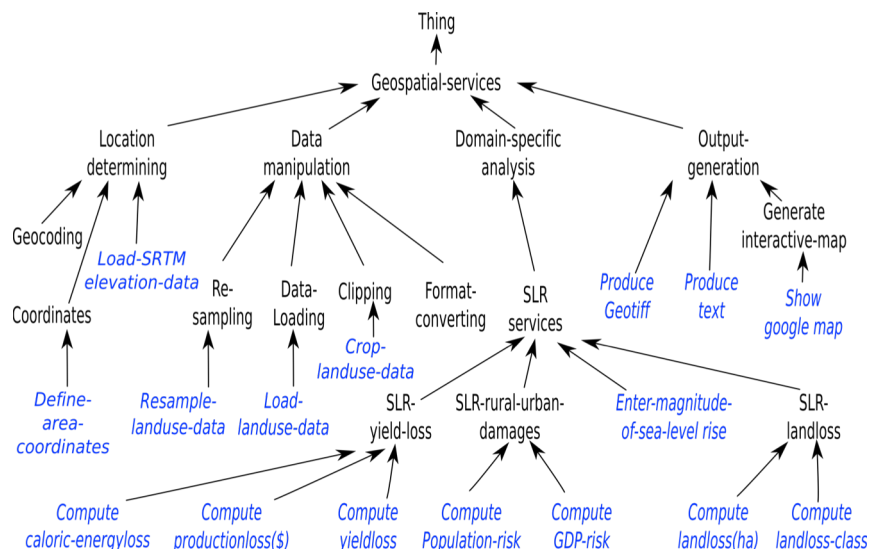
Table 1: Selected modules of the SLR application example.

Name	Inputs	Outputs
Load SRTM elevation data.	Coordinates	SRTM-data
Compute flooded areas	Elevation data	slrlandloss
Load-landuse data	Landuse-data.tiff	landuse-data
Crop-landuse data	landuse-data, slrlandloss	slrlanduse-data
Resample-landuse data	slrlandloss, slrlanduse-data	slrlanduse-sample
Compute landloss-class	slrLandloss, slrlanduse-sample	slrlandclassloss
Load population data	Population-data.tiff	Population-data
Resample population data	slrLandloss, Population-data	slrPopulation-sample
Compute potential landloss (ha)	slrLandloss	slrlandlossha
Compute population at risk of migration	Population-data, slrPopulation-sample	slrPopulationrisk
Produce GeoTIFF output	outputdata	Geo-referenced-file
Show google map	outputdata	google-map
...	...	...

services. In addition to particular features of the concrete applications, it is often also useful to incorporate knowledge from the application domain in general, in this case the geospatial application domain.

Figure 3 illustrates the service taxonomy that we defined for the SLR application example. Due to the limited space in the paper, we only depict the excerpts of the taxonomy that is relevant to the examples discussed in the next section.

Figure 3: SLR service taxonomy



Under the general *geospatial services* class, it defines four subclasses: *domain-specific analysis services*, *data manipulation services*, *location determining services* and *output generation services*.

While *data creation*, *location determining* and *output generation* services could be reused in the whole domain of geospatial applications, the *SLR services* class, which is a subclass of *domain-specific analysis services*, comprises services specifically for SLR impacts analysis. Thus, it contains SLR-related specific application services such as *SLR-landloss*, *SLR-urban-rural-damages* and *SLR-yieldloss*. The leaves of the taxonomy tree correspond to concrete modules of the domain model, as described above.

We have also defined a type taxonomy for the SLR application example that addresses major characteristics of geospatial data in order to handle users’ preferences and constraints regarding input/output data such as, formats, resolution, georeferencing systems and domain specific data. However, due to the limited space of the paper, the type taxonomy is not presented.

**Constraints:** During the domain modeling, general constraints, which apply to the whole application domain (in this case SLR impacts analysis), are defined. For instance, we defined constraints to avoid redundancy of common services (such as “Do not use module *Enter magnitude of sea-level rise* more than once.”), and we also defined a constraint to include an output generation service as last service in the workflow (“Use module *output-generation* as last module in the solution.”).

### 3.2 Workflow Synthesis

For demonstrating workflow synthesis with PROPHETS, we use the example from Figure 1 again. In fact, the process sketch shown in the figure can be interpreted as loose specification by the synthesis plugin. According to what is shown in Figure 2, the synthesis process with PROPHETS consists of the following steps: (1) interpreting the branch between *enter magnitude of sea-level rise* SIB at the beginning and the *show SLR impacts* SIB at the end as loose specification, (2) adding constraints to refine the solutions, (3) generating the possible solutions, and (4) selecting and inserting one solution.

Typically there are many different possibilities for workflows implementing the specification. When we start the synthesis for the example with the default constraints defined in the domain model only, we obtain a total of 850 possible implementations of the specification when exploring solutions up to a length of 10. Through several refinement steps, users can incrementally apply additional constraints to finally reach adequate solutions that match their particular objectives, perspectives and preferences.

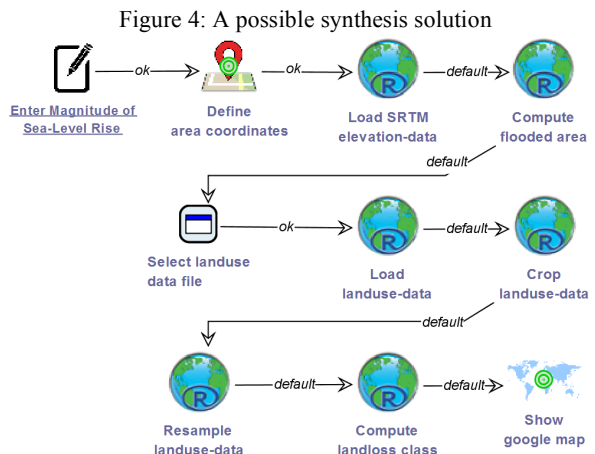
As shown in Table 2, with the constraints of the first refinement example, significantly less, namely 455 solutions are returned. This refinement specifies the intended goal of the SLR analysis by pointing the synthesis algorithm to one of the main classes of SLR analysis services, such as land loss, by adding the constraint that enforces the use of at least one module from the *SLR-landloss* class from the service taxonomy in the solution.

Table 2: Impact of constraints on synthesis solutions.

Refinement	Constraints	Solutions
1	Enforce the use of module <i>SLR-landloss</i>	455
2	Enforce the use of module <i>compute-landloss-class</i> . Enforce the existence of type <i>map</i> . Do not use the module <i>SLR-rural-urban-damages</i> . Do not use the module <i>SLR-yield-loss</i> .	24
3	Enforce the existence of type <i>coordinates</i> . Enforce the use of module <i>generate-interactive-map</i> . Enforce the existence of type <i>SRTM data</i> . Enforce the existence of type <i>landuse-data</i> .	1

In the second refinement step, we use constraints to explicitly include and exclude particular (groups of) modules from the solutions. This reduces the number of obtained solutions further, and an already manageable set of 24 possible implementations is obtained. Finally, we add constraints expressing preferences of data input and output. With those constraints only one solution, which adequately matches the user’s objectives and preferences, is returned.

The final solution, composing the required services to show the impact of sea-level rise over a specific class of land use, is depicted in Figure 4. These services have been composed based on their interface descriptions (see Table 1) and taking into account the constraints entered during the workflow design phase. More information about these services can be found in [1].



## 4 Conclusion

In this paper presented a brief demonstration of how the PROPHETS synthesis plugin of the jABC workflow modeling framework can facilitate the (semi-) automatic composition of workflows for analyzing the impacts of sea-level rise. In this setup, workflow designers are not required any more to implement the entire workflow manually. Instead, they can just provide a sketch of the intended workflow, together with a set of constraints that further specify the analysis objectives. PROPHETS then applies a synthesis algorithm to this abstract specification, and returns a set of possible implementations to the user.

To evaluate the effectiveness of the proposed ontology further, we have also performed various experiments with SLR applications other than land loss. In particular, yield loss and calories loss analyses are more data-intensive, and constraints can have a greater impact for them. The experiments also showed that it would also be beneficial for users to be able to define their preferences regarding data scales (spatial resolutions) and georeferencing systems, and then to automatically include resampling and transformation services in the designed workflow. In the scope of future work, we are therefore going to improve our current domain model by considering different resolutions of geospatial data and georeferencing systems in the scope of future work. Generally, a major part of our future work is going to focus on the further evaluation of existing and on the creation of new domain-specific ontologies in collaboration with experts from the application domain.

## Acknowledgments

The authors would like to thank Steffen Kriewald, Dominik Reusser, and Markus Wrobel from the climate change and development group at the Potsdam Institute for Climate Impact Research (PIK), who provided us with the tools and data sets required for the scenario. Thanks also go to Hartmut Asche from the department of Geography at University of Potsdam for his valuable feedback.

This work was supported, in part, by Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Irish Software Research Centre ([www.lero.ie](http://www.lero.ie)).

## References

- [1] Al-areqi, S., Kriewald, S., Lamprecht, A., et al.: Agile Workflows for Climate Impact Risk Assessment based on the ci:grasp Platform and the jABC Modeling Framework. In: *7th Int. Congress on Environmental Modelling and Software* (2014).
- [2] Bai, Y., Di, L., Wei, Y.: A taxonomy of geospatial services for global service discovery and interoperability. *Computers & Geosciences* 35(4), 783–790 (2009).
- [3] Farnaghi, M., Mansourian, A.: Automatic composition of WSMO based geospatial semantic web services using artificial intelligence planning. *Journal of Spatial Science* 58(2), 235–250 (2013).
- [4] Granell, C., Lemmens, R., Gould, M., et al.: Integrating semantic and syntactic descriptions to chain geographic services. *IEEE Internet Computing*, 10(5), 42–52 (2006).
- [5] Jaeger, E., Altintas, I., Zhang, J., et al.: A Scientific Workflow Approach to Distributed Geospatial Data Processing using Web Services. In: *Int. Conference on Scientific and Statistical Database Management..* 87–90. (2005).
- [6] Jung, G., Margaria, T., Nagel, R., et al.: SCA and jABC: Bringing a service-oriented paradigm to web-service construction. *Leveraging Applications of Formal Methods, Verification and Validation*, Springer CCIS 17, 139-154 (2009).
- [7] Lamprecht, A.L., Naujokat, S., Margaria, T., Steffen, B.: Synthesis-based loose programming. *7th Int. Conference on the Quality of Information and Communications Technology*, IEEE Conference Publications (2010).
- [8] Li, H., Wang, Y., Cheng, P.: Semantic description for the taxonomy of the geospatial services. *Boletim de Ciências Geodésicas* 21(3) (2015).
- [9] Lissner, T.K., Reusser, D.E., Schewe, J., Lakes, T., Kropp, J.P.: Climate impacts on human livelihoods: where uncertainty matters in projection of water availability. *Earth Syst. Dynam.*, 5, 355-373 (2014).
- [10] Margaria, T., Nagel, R., Steffen, B.: jETI: A Tool for Remote Tool Integration. *Tools and Algorithms for the Construction and Analysis of Systems*, Springer LNCS 3440, 557-562. (2005).
- [11] Margaria, T., Steffen, B.: Service-Oriented: Conquering Complexity with XMDD. In: *Conquering Complexity*, 217–236, Springer London (2012).
- [12] Naujokat, S., Lamprecht, A.L., Steffen, B.: Loose Programming with PROPHETS. *Fundamental Approaches to Software Engineering*. Springer LNCS 7212, 94–98 (2012).
- [13] Steffen, B., Margaria, T., Freitag, B.: Module Configuration by Minimal Model Construction. *Tech. rep. University Passau* (1993).
- [14] Steffen, B., Margaria, T., Nagel, R., Jörges, S., Kubczak, C.: Model-Driven Development with the jABC. *Hardware and Software, Verification and Testing*, Springer LNCS 4383, 92–108 (2007).
- [15] Yue, P., Baumann, P., Bugbee, K., Jiang, L.: Towards intelligent giservices. *Earth Science Informatics* 8(3), 463–481 (2015).
- [16] Yue, P., Di, L., Yang, W., Yu, G., Zhao, P.: Semantics-based automatic composition of geospatial web service chains. *Computers & Geosciences* 33(5), 649–665 (2007).
- [17] Zhang, M., Yue, P.: GeoJModelBuilder: A java implementation of model-driven approach for geoprocessing workflows. *2<sup>nd</sup> IEEE Int. Conference on Agro-Geoinformatics*, 393–397 (2013).