# Ontology patterns for function modeling with GFO

Patryk Burek[1,*], Frank Loebe[2,*] and Heinrich Herre[3]

[1]*Polish-Japanese Academy of Information Technology, Faculty of Computer Science, ul. Koszykowa 86; 02-008 Warsaw, Poland*

[2]*Leipzig University, ScaDS.AI Dresden/Leipzig, Humboldtstrasse 25, 04105 Leipzig, Germany*

[3]*Leipzig University, Institute of Medical Informatics, Statistics and Epidemiology (IMISE), Haertelstrasse 16-18, 04107 Leipzig, Germany*

### Abstract

Function modeling is a relevant aspect of many branches of conceptual modeling as well as in ontology engineering. In the current paper we briefly outline the results provided in the framework of the General Formal Ontology to support the modeling of functions. We report on work in progress on the development of OWL ontologies serving as ontology patterns for modeling functions.

### Keywords

function modeling, functional decomposition, ontology patterns, Web Ontology Language (OWL)

## 1. Introduction

Functions are first-class citizens in many scientific and engineering disciplines, as it is common to describe the elements of a domain in terms of the functions that they exhibit, e.g., organizations by business functions, drugs by therapeutic functions, organs by biological functions and technical artifacts by the functions that they have been designed for. Similarly, many conceptual modeling areas including systems, business-, software- and ontology engineering involve representing functions, in a variety of ways.

Besides its importance and wide applicability, function modeling is no easy enterprise and poses numerous challenges for a modeler. It is a first challenge to determine how to represent various aspects of functions, such as their teleological character and involved participants, in a coherent framework that avoids an implementation/realization bias. Secondly, function models are often complex, composed of hundreds or even thousands of elements. As an example consider the newest release of the Gene Ontology (GO) [1], which currently organizes 11228 functions into a hierarchical model. Handling models of such size requires adequate function decomposition techniques.

The objective of the current paper is to outline the results of the ontological analysis and the development of methods and tools devoted to function modeling in the ecosystem of the

*ER2023: Companion Proceedings of the 42nd International Conference on Conceptual Modeling: ER Forum, 7th SCME, Project Exhibitions, Posters and Demos, and Doctoral Consortium, November 06-09, 2023, Lisbon, Portugal*

*Corresponding author.

pburek@pjwstk.edu.pl (P. Burek); frank.loebe@informatik.uni-leipzig.de (F. Loebe); heinrich.herre@uni-leipzig.de (H. Herre)

0000-0001-8165-0272 (P. Burek); 0000-0003-4537-5212 (F. Loebe); 0000-0001-5343-9218 (H. Herre)

General Formal Ontology (GFO). The next section covers results obtained in the framework of GFO's first release, whereas in section 3 our work in progress on OWL ontologies of functions is discussed, which will be incorporated into the current GFO 2.0 release. The application of these ontologies is briefly illustrated with examples from the Gene Ontology.

## 2. Ontology of functions in GFO

The General Formal Ontology (GFO) is a formal top-level ontology (TLO), most recently exemplified in [2]. A function ontology is an outstanding aspect of GFO since its first major version. Except for GFO and the TLO YAMATO [3], even where other TLOs include a notion of function, they do not provide full-fledged support for function modeling. Subsequently, we summarize major milestones of developing the ontology of functions within GFO.

Motivated by the aim to provide greater accuracy and expressiveness to biomedical ontologies regarding the representation of functions, the authors in [4] provide means to capture existing functional knowledge in a more formal manner. There functions (and related notions) are introduced as an additional layer of concepts that can be added to the existing biomedical ontologies without requiring substantial changes to those domain ontologies.

Like in other areas, cf. e.g. [5], and not only since [4] for GFO, functions are intimately tied to the notion of goal, insofar that function modeling aims at representing the entities of the modeled domain in terms of goals that those entities achieve or are intended to achieve.

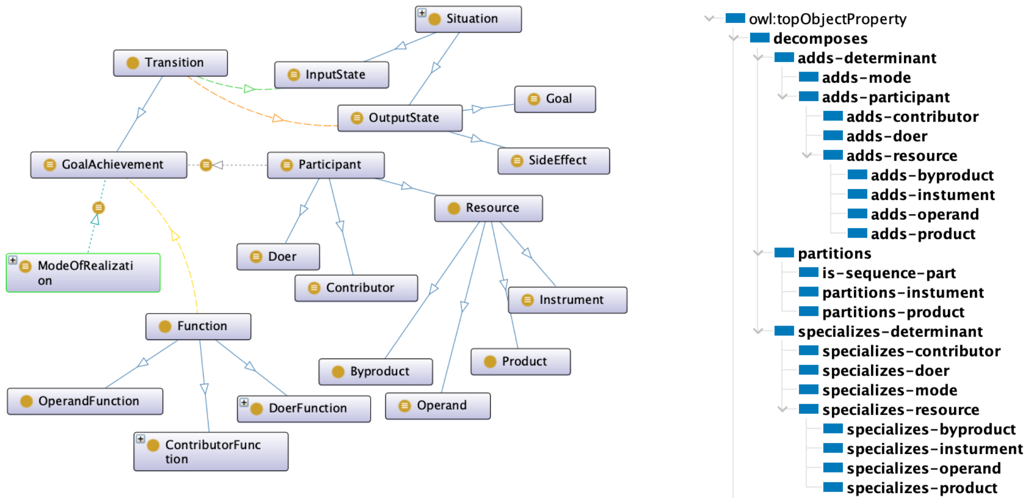
Besides functions as such, the second pillar of function modeling is functional decomposition, which is a popular notion, but unfortunately it is not clear-cut in the literature and thus can create ambiguities in models using it. In modeling systems, functional decomposition often refers to construing sub functions in a top-down manner, e.g. analyzing the overall function of the system (a general teleological component) by determining functional child components that contribute to achieving the goal of the parent function. In [6] we laid a foundation for the ontological analysis of function decomposition, understanding it as a family of relations rather than a single, uniform relation. In accordance with this view, function decomposition yields different kinds of subsumption links in taxonomic hierarchies of functions (as extensively provided in GO, for example).

In [7] the authors merged and extended the previous results and introduced a consistent framework of function modeling covering both function structure and function decomposition. The same work provides a partial formalization in FOL and introduces a UML profile, called Functional Modeling Language (FueL), which has been applied for the refactoring of GO.

## 3. GFO 2.0 function modeling patterns

The results referred to in section 2 have been adapted and integrated into the GFO 2.0 release following modularization principles [8]. Three sub ontologies constitute the functions module in GFO 2.0, each of which is dedicated to a specific task and comes with its own OWL serialization.

The modular approach fosters the building of functional models and ontologies without the need of importing complex and big ontologies. Instead, it permits a lightweight approach – sub modules are developed as ontology patterns dedicated to specific tasks.




(a) `gfo-functions-core` (b) `gfo-functions-decomposition`

**Figure 1:** Main elements of two out of the three OWL sub modules of module `gfo-functions`.

The first sub module, `gfo-functions-core`, provides a blueprint for specifying functions themselves. That includes the template for specifying the basic units of functional description, called Goal Achievements (GAs). The second module, `gfo-functions-decomposition`, supports decomposition models through a family of function decomposition relations making explicit the tacit rationale behind functional decomposition links. The third sub module, `gfo-functions-assignment`, comes with basic relations for specifying entities of the domain in terms of their functions and malfunctions. That last module falls victim to space limitations, whereas OWL ontologies of the first two are the subject of the current section.

## 3.1. Function and goal achievement

Figure 1a depicts the ontology of the `gfo-functions-core` sub module, in which Function is defined as a role of an entity in the context of some Goal Achievement (GA). The concept of goal achievement is defined as a purposeful transition – a transition to a situation that is (considered to be) a goal, whereas a Goal is an intended output of this transition, distinguished from an unintended or accidental output which is represented by the notion of SideEffect. Another common view on functions is focused on the participants involved in their realization. Besides the Doer role, which is responsible for conducting the goal achievement, `gfo-functions-core` introduces several other types of participant roles, both active (e.g. Contributor) as well as passive ones (e.g. Instrument, Product and Byproduct).

Finally, functions are often specified not only in terms of what is to be achieved or who is involved in the realization, but also in terms of how the goal is to be achieved, by which means of realization. In GFO the concept of ModeOfRealization specifies the GA component that indicates a particular means of realization.

## 3.2. Patterns for functional decomposition

The ontological analysis of functional decomposition conducted for GFO 1.0 led to distinguish-ing several decomposition relations, the use of which reveals tacit assumptions in function decomposition models. Figure 1b depicts the hierarchy of these relations as integrated into the `gfo-functions-decomposition` sub module of GFO 2.0.

These relations are based on several patterns occurring in function decomposition models across domains and industries. One common pattern is the decomposition of a function by adding a function determinant, i.e., by adding a mode of realization or a participant. For instance, in GO the function `transmembrane transporter activity` is decomposed/specialized into `active transmembrane activity` by adding the mode of realization, namely `active transport`. Note further that the addition of a mode frequently comes with the specification of a particular contributor or instrument.

Another pattern of function decomposition is not concerned with adding a determinant, but instead results from specializing it. For instance, `carbohydrate transmembrane transporter activity` specializes the operand `solute` of its parent function `transmembrane transporter activity` to `carbohydrate`. A third group of patterns is based on a partition of either the flow of realization or of certain determinants, e.g. of products manufactured or of instruments used.

The application of these patterns makes the subsumption links between functions explicit, which is a crucial enhancement especially for complex models. For illustration, listing 1 employs `gfo-functions-decomposition` on a GO snippet (represented in the RDF syntax Turtle).

**Listing 1:** Examples of function decomposition statements applied to Gene Ontology.

```
@prefix obo: <http://purl.obolibrary.org/obo/> .
@prefix gfofd: <http://onto-med.de/gfo/2.0/functions/gfo-functions-
decomposition.owl#> . obo:GO_0005402
  rdfs:label "carbohydrate:monoatomic cation symporter activity" ;
  rdfs:subClassOf obo:GO_0015144 ,
                  obo:GO_0015294 ;
  gfofd:adds-mode obo:GO_0015144 ;
  gfofd:adds-operand obo:GO_0015144 ;
  gfofd:specializes-operand obo:GO_0015294 .
obo:GO_0015144
  rdfs:label "carbohydrate transmembrane transporter activity" ;
  rdfs:subClassOf obo:GO_0022857 ;
  gfofd:specializes-operand obo:GO_0022857 .
obo:GO_0022857
  rdfs:label "transmembrane transporter activity" .
```

The implicit assumptions of subsumption links between biological functions are made explicit by the annotation with relations of `gfo-functions-decomposition`. In the example, `GO_0005402` has not only two parent concepts `GO_0015144` and `GO_0015294`, but the former subsumption link involves two differences: the addition of the operand `monoatomic cation` and the addition of the mode of realization `symporter`. Making such assumptions explicit not only exposes the rationale behind complex functional models, but it further enables manual or even automatic model annotation and validation.

As an example, consider the (semi-SWRL) rule in listing 2. It grasps the intuition that for a function F1 and its parent function F2, if the operand of F1 is a subclass of that of F2, then F1 subsumes F2 by operand specialization. An example of that case is the above relation between

GO_0005402 and its parent function GO_0015294, where the operand of the former (carbohydrate) is a subclass of the operand of the latter (solute).

**Listing 2:** Rule example reflecting properties of function decomposition relations.

```
@prefix gfofc: <http://onto-med.de/gfo/2.0/functions/gfo-functions-core.owl#> .
@prefix gfofd: <http://onto-med.de/gfo/2.0/functions/gfo-functions-
decomposition.owl#> . rdfs:subClassOf(F1, F2) and rdfs:subClass(F2, gfofc:Function)
and
gfofd:has-operand(F1, O1) and gfofd:has-operand(F2, O2) and rdfs:subClass(O1, O2)
-> gfofd:specializes-operand(F1, F2)
```

## 4. Conclusions

The paper presents results obtained within the framework of the General Formal Ontology (GFO) aimed at supporting function modeling and the representation of functional knowledge. Two of three OWL modules, which are currently under development, have been indicated. The first module provides means for the representation of teleological components, by specifying a blueprint of structured specifications. The second module contains a family of relations tailored for the construction of functional decomposition models that make tacit assumptions underlying decomposition links explicit. These modules can be used as OWL patterns for building OWL ontologies, as well as they yield opportunities for the validation and refactoring of function ontologies.


## Acknowledgements

Frank Loebe acknowledges the financial support by the Federal Ministry of Education and Research of Germany and by the Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the program Center of Excellence for AI research “Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig”, project id: ScaDS.AI


## References


[1] The Gene Ontology Consortium, The Gene Ontology knowledgebase in 2023, Genetics 224 (2023). doi:10.1093/genetics/iyad031.

[2] F. Loebe, P. Burek, H. Herre, GFO: The General Formal Ontology, Applied Ontology 17 (2022) 71–106. doi:10.3233/AO-220264.

[3] R. Mizoguchi, S. Borgo, YAMATO: yet-another more advanced top-level ontology, Applied Ontology 17 (2022) 211–232. doi:10.3233/AO-210257.

[4] P. Burek, R. Hoehndorf, F. Loebe, J. Visagie, H. Herre, J. Kelso, A top-level ontology of functions and its application in the Open Biomedical Ontologies, Bioinformatics 22 (2006) e66–e73. doi:10.1093/bioinformatics/btl266.

[5] H.-E. Eriksson, M. Penker, Business Modeling with UML – Business Patterns at Work, Wiley, 2000.

[6] P. Burek, H. Herre, F. Loebe, Ontological analysis of functional decomposition, in: H. Fujita, V. Marík (Eds.), New Trends in Software Methodologies, Tools and Techniques, volume 199 of *FAIA*, IOS Press, 2009, pp. 428–439. doi:10.3233/978-1-60750-049-0-428.

[7] P. Burek, F. Loebe, H. Herre, FueL: Representing function structure and function dependencies with a UML profile for function modeling, Applied Ontology 11 (2016) 155–203. doi:10.3233/AO-160167.

[8] P. Burek, F. Loebe, H. Herre, Towards GFO 2.0: Architecture, modules and applications, in: B. Brodaric, F. Neuhaus (Eds.), Proceedings of the 11th International Conference, FOIS 2020, Cancelled / Bozen-Bolzano, Italy, September 14-17, 2020, volume 330 of *FAIA*, IOS Press, 2020, pp. 32–45. doi:10.3233/FAIA200658.