A FAIR Model Catalog for Ontology-Driven Conceptual Modeling Research

Pedro Paulo F. Barcelos[1(✉)], Tiago Prince Sales[1], Mattia Fumagalli[1], Claudenir M. Fonseca[1], Isadora Valle Sousa[1], Elena Romanenko[1], Joshua Kritz[2], and Giancarlo Guizzardi[1,2]

[1] Conceptual and Cognitive Modeling Research Group (CORE), Free University of Bozen -Bolzano, Bolzano, Italy
{pfavatobarcelos,tprincesales,mfumagalli,cmoraisfonseca,ivallesousa, eromanenko}@unibz.it
[2] University of Twente, Enschede, The Netherlands
{j.kritz,g.guizzardi}@utwente.nl

Abstract. Conceptual models are artifacts representing conceptualizations of particular domains. Hence, multi-domain model catalogs serve as empirical sources of knowledge and insights about specific domains, about the use of a modeling language's constructs, as well as about the patterns and anti-patterns recurrent in the models of that language cross-cutting different domains. However, to support domain and language learning, model reuse, knowledge discovery for humans, and reliable automated processing and analysis by machines, these catalogs must be built following generally accepted quality requirements for scientific data management. Especially, all scientific (meta)data—including models—should be created using the FAIR principles (Findability, Accessibility, Interoperability, and Reusability). In this paper, we report on the construction of a FAIR model catalog for Ontology-Driven Conceptual Modeling research, a trending paradigm lying at the intersection of conceptual modeling and ontology engineering in which the Unified Foundational Ontology (UFO) and OntoUML emerged among the most adopted technologies. In this initial release, the catalog includes over a hundred models, developed in a variety of contexts and domains. The paper also discusses the research implications for (ontology-driven) conceptual modeling of such a resource.

Keywords: Ontology · Ontouml · Data catalog · Fair · Linked data

1 Introduction

Conceptual models are concrete artifacts representing conceptualizations of particular domains. Ontology-Driven Conceptual Modeling (ODCM) is a trending paradigm that lies at the intersection of conceptual modeling and ontology

J. Ralyté et al. (Eds.): ER 2022, LNCS 13607, pp. 3–17, 2022.
https://doi.org/10.1007/978-3-031-17995-2_1

engineering. ODCM is frequently about the use of foundational ontologies, i.e., axiomatic ontological theories (in the philosophical sense) to improve conceptual models, modeling languages, and tools [26].

In this context, the Unified Foundational Ontology (UFO) and the UFO-based conceptual modeling language OntoUML [5,8,12,13] have emerged among the most used approaches in the field [26]. Over the years, UFO and OntoUML have been adopted by research, industrial, and governmental institutions worldwide to create ODCM models in different domains [13,26]. In this context, conceptual models are created either by directly extending UFO's categories (e.g., having the type *Agent* specializing the UFO type *Object*, or having the type *Action* specializing the UFO type *Event*) or, more frequently, by using OntoUML stereotypes for classes and relations—which also reflect UFO's ontological distinctions (e.g., decorating the type *Action* with the ≪ event ≫ stereotype).

Multi-domain model catalogs serve as sources of empirical knowledge and insights about: (i) how specific domains are modeled (ii) the use of a modeling language's constructs, and (iii) domain-independent patterns that emerge from the use of a language. However, to support domain and language learning, model reuse, knowledge discovery for humans, and reliable automated processing and analysis by machines, these repositories must be built following generally accepted quality requirements for scientific data management. In particular, all scientific (meta)data—including models—must be created using the FAIR principles, which are: **F**indability, **A**ccessibility, **I**nteroperability, and **R**eusability [27].

In this paper, we report on the construction of an ODCM catalog, henceforth termed the **OntoUML/UFO Catalog**. This is, to the best of our knowledge, the first FAIR catalog of ontology-driven conceptual models. It is a structured, collaborative, and open-source catalog that contains UFO-grounded models—the vast majority of which are represented in OntoUML [5,8,12].

The OntoUML/UFO Catalog has two goals. First, we want to provide curated structured data to support empirical research in OntoUML/UFO, specifically, and on conceptual modeling in general. For example, this can provide high-quality data on *why*, *where*, and *how* these approaches are used, which can enable researchers to understand the evolution of the language and its foundations. It can also serve as a repository for patterns and anti-patterns detection [20], as well as a benchmark against which, e.g., language transformation models [2] and complexity management techniques [11,19] can be assessed. Additionally, it can support novice modelers who want to learn ODCM in OntoUML/UFO, as well as advanced users who want to reuse existing models as *seed models* [3].

The first catalog release offers a diverse collection of 127 models obtained from academic and industrial sources, created by modelers with varying modeling skills, for a range of domains, and for different purposes. These models are available in the JavaScript Object Notation (JSON) and Turtle machine-readable formats, and are accessible via permanent Uniform Resource Identifiers (URIs).

The rest of this paper is organized as follows. Section 2 discusses the process we followed and the tools we have used to create the catalog. Section 3

presents the catalog's structure and introduces the vocabularies we used to build it. Section 4 briefly discusses some statistics on the current release of the catalog. Section 5 evaluates the catalog with relation to the FAIR principles. Section 6 elaborates on the importance of the catalog for the community and on the different research endeavors it facilitates or enables. Section 7 positions our work with relation to other catalogs and datasets available to the modeling community. Finally, Sect. 8 makes some final considerations and discusses future works.

2 Methods and Materials

The OntoUML/UFO Catalog was conceived to be open and easily accessible to all members of the modeling community, to allow collaborative work, and to be easily maintainable. These are important requirements in this context, since we envision a continuous growth of the catalog in years to come. To reach these goals, we created a GitHub repository for hosting the catalog and associated it with a permanent URL[1].

The first activity for the catalog creation was the definition of a set of governance rules, an activity done by a team of OntoUML and Linked Data specialists. To be accepted into the catalog, a dataset submission had to comply with these rules. For example, a basic rule regarded what exactly constituted a submission: a *dataset submission* should include three files: the UFO-based/OntoUML model itself, a file with the model's metadata information, and the model's associated bibliography (when available). The catalog rules are formalized and made available for contributors on its GitHub's wiki page.

Once these rules were established, we could then populate the catalog. We encouraged the participation of the conceptual modeling and ontology engineering communities through public invitations for collaboration. Researchers were asked to submit their models to the catalog. Intending to familiarize them with the catalog structure and content, we also requested their cooperation for the migration of existing data to the catalog. Seventeen experienced modelers contributed to this activity.

To cover as many models as possible and to reduce the chances of receiving duplicates, we elaborated a list of all OntoUML/UFO-based models that we could find in a broad non-systematic literature search and in personal databases (e.g., OntoUML/UFO-based models developed by students during academic courses)—the final list contained more than 300 models.

After collecting these models, our collaborators began the data migration phase, which comprised adapting the original models to the catalog standards. The collaborators obtained images of OntoUML/UFO-based models from papers and technical reports and manually rebuilt them on the latest version (v16.3) of the Visual Paradigm (VP) modeling editor[2] using the ontouml-vp-plugin[3]. This

[1] https://purl.org/ontouml-models.

[2] Despite being a commercial tool, VP (https://www.visual-paradigm.com/) has a free community version that could be used by our contributors.

[3] Downloadable from https://purl.org/ontouml-vp.

plugin is an extension to VP that offers several important modeling services [6]. For example, it allows the use of OntoUML constructs (stereotypes for classes, associations, and attributes) when building UML class diagrams, and supports syntax verification, model serialization in JSON and Web Ontology Language (OWL), as well as model modularization and abstraction. Models already available in editable format were imported into VP.

We provided instructions to modelers to harmonize systematically their design decisions, including a Frequent Asked Questions (FAQ) page in the GitHub catalog's wiki for specifying topics that could lead to inconsistencies. As we want the migrated models to be as truthful as possible to the original ones, collaborators were advised not to reinterpret the model to be submitted and, e.g., they were asked to: preserve the original OntoUML stereotypes used, keep syntactical errors, maintain the original diagram layout as much as possible, and preserve the original terminology used in the model.

Files containing BibTeX references and, especially, rich metadata (see discussion in Sect. 3) for the submitted datasets were produced. As these files are fundamental parts of a dataset submission, the rules on how they should be produced were also detailed in the catalog's wiki.

To ensure the catalog's consistency, every new submission was subject to peer evaluation by the catalog curators (a team of OntoUML/UFO experts). The evaluation included a manual analysis of the files composing the dataset, checking for errors, and verifying their compliance with the defined catalog's governance rules. Once approved, the dataset was included in the catalog, where complementary files associated with it were derived. By the end of this process, the submitted dataset was then included in the catalog.

3 Catalog Structure

Following the recommendations for implementing the FAIR principles [15], the OntoUML/UFO *catalog schema* (see Fig. 1) reuses classes and properties from the following vocabularies:

- Data Catalog Vocabulary (DCAT)[4]
- Dublin Core Terms (DCT)[5]
- Simple Knowledge Organization System (SKOS)[6]
- Metadata for Ontology Description and Publication (MOD)[7]
- Friend of a Friend (FOAF)[8]

The catalog (`dcat:Catalog`) is maintained by a community of users (`foaf:Agent`), composed of a set of models (instances of `mod:SemanticArtefact`), which are described by the following metadata (asterisks show mandatory items):

[4] https://www.w3.org/TR/vocab-dcat-2/.
[5] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/.
[6] https://www.w3.org/TR/skos-reference/.
[7] https://w3id.org/mod/2.0.
[8] http://xmlns.com/foaf/0.1.

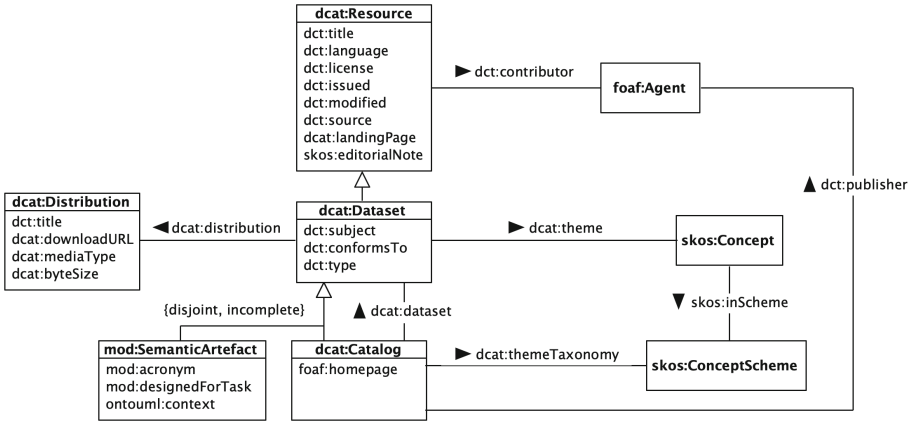

Fig. 1. The OntoUML/UFO Catalog schema in a UML class diagram representation.

- `dct:title`*: the name of the model. E.g., `"Common Ontology of Value and Risk"`, `"Reference Ontology of Trust"`.
- `mod:acronym`: the acronym one can use to refer to the model. E.g., `"RDBS-O"`, `"COVER"`, `"ROT"`.
- `dct:issued`: the year when the model was first published. E.g., 2022.
- `dct:modified`: the year of the most recent publication of the model (in a scientific publication, a technical report, a website, etc.). E.g., 2018.
- `dct:contributor`: a list of URIs of people who contributed to the development of the model. If possible, we recommend using a contributor's persistent URI from DBLP (e.g., https://dblp.org/pid/96/8280) or ORCID (e.g., https://orcid.org/0000-0003-2736-7817).
- `dct:subject`*: a list of strings that identify the domains covered by the model. E.g., `"robotic"`, `"technology"`, `"services"`, `"risk"`.
- `dcat:theme`*: the central theme of the model according to the Library of Congress Classification (LCC) system.[9] E.g., `"Class S - Agriculture"`, `"Class T - Technology"`. LCC is available as a `skos:ConceptScheme` and each of its classes as instances of `skos:Concept`.
- `skos:editorialNote`: general notes on the model documentation process. E.g., `"The ontology was originally designed in Portuguese"`.
- `dct:type`*: the list of types in which the model can have categories. Since OntoUML and UFO are frequently used for building `core` and `domain` ontologies, these are among the allowed values for this field (the other allowed value is `application`).
- `dct:language`*: the language in which the lexical labels of the model are written (using the IANA Language Sub Tag Registry[10]). E.g., `"en"`, `"pt"`.
- `mod:designedForTask`*: the list of goals that motivated the development of the model. The allowed values are `"conceptual clarification"`, `"data`

[9] https://www.loc.gov/catdir/cpso/lcco/.
[10] https://www.iana.org/assignments/language-subtag-registry/.

`publication`, `"decision support system"`, `"example"`, `"information retrieval"`, `"interoperability"`, `"language engineering"`, `"learning"`, `"ontological analysis"`, and `"software engineering"`.
- `ontouml:context`*: the list of contexts in which the model was developed. The allowed values for this field are: `"research"`, `"industry"`, `"classroom"`.
- `dct:source`: a list of URIs of the resources that contain, present, or significantly influenced the model. We recommend the use of persistent URIs, such as the Digital Object Identifier (DOI) or DBLP's URI, whenever possible. E.g., https://doi.org/10.3233/AO-150150, https://dblp.org/rec/journals/ao/Morales-Ramirez15.
- `dct:conformsTo`*: the list of representation styles adopted in the model. The allowed values for this field are: `"ontouml"`, for models that use OntoUML's stereotypes; and `"ufo"`, for pure UML models that specialize UFO's types and relations.
- `dcat:landingPage`: a URL of a web page to gain access to the ontology, its distributions and/or additional information. E.g., https://www.model-a-platform.com.
- `dct:license`: a URI of the model's license. E.g., https://creativecommons.org/licenses/by/4.0/.

Each model, after being added to the catalog, is available via three distributions (`dcat:Distribution`), namely a JSON distribution, a Turtle distribution, and a distribution in the format of the modeling tool used to represent the model. Each distribution is described by a `dcat:title` and a `dcat:downloadURL`.

A GitHub repository hosts the whole catalog. Its root directory has: (a) a `catalog.ttl` file[11] that is the file encoding the catalog itself—i.e., the aggregated data of all datasets that are part of the catalog; (b) a `metadata.ttl` file, which provides (in a triple-based format) all the catalog's metadata listed above, and which aggregates all metadata from its composing datasets; and (c) a list of folders—the datasets, each one including all the information related to an OntoUML/UFO-based model. We structured the dataset folders as from Fig. 2, namely:

- `ontology.vpp`: the Visual Paradigm project of the model;
- `ontology.json`: contains the JSON serialization of the model exported via the ontouml-vp-plugin;
- `ontology.ttl`: uses the OntoUML Metamodel in OWL[12] to map the model's data. This is a vocabulary designed to support the serialization and exchange of OntoUML models in compliance with the ontouml-schema[13], which is a specification of how to serialize OntoUML models as JSON objects [6]. This file provides a specific URI for all data from the model[14], and its publication allows anyone to access and manipulate all the model's instances;

[11] https://purl.org/ontouml-models/catalog.
[12] https://purl.org/ontouml-models/vocabulary.
[13] https://purl.org/ontouml-schema.
[14] These URIs are generated according to the following template: `https://purl.org/ontouml-models/<folder name>`.

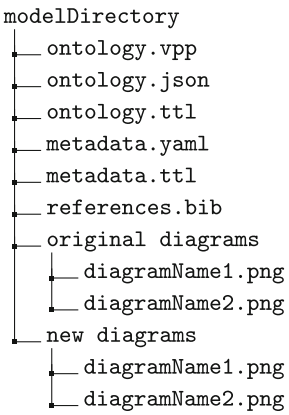

Fig. 2. Folder structure for each model in the catalog.

- `metadata.yaml`: contains the model's metadata;
- `metadata.ttl`: is an Resource Description Framework (RDF)-based version in Turtle syntax of `metadata.yaml`.
- `references.bib`: contains the BibTeX citation data for each publication about the model (this file is not required for unpublished models).
- `original diagrams/`: is a folder containing images in PNG format of the diagrams created by the authors of the model.
- `new diagrams/`: is a folder containing images in PNG format of all diagrams rebuilt on Visual Paradigm (keeping the names matching the original ones).

Note that the `vpp`, `yaml`, and `bib` files are the ones provided by the collaborators. In contrast, the OntoUML plugin for Visual Paradigm automatically generates the `json` file for each model. The `png` files hosted in the `new diagrams` folder are automatically generated; the same occurs for all the `ttl` files.

4 Catalog Statistics

4.1 Statistics on the Models

Table 1 presents some basic statistics in the current state of the catalog, considering the number of diagrams, classes, domain associations, and generalizations relations. These statistics give us an idea of the dimension of the models there included. In its current version, the catalog has 127 models, which have 656 diagrams, 7223 classes, 5392 associations, and 5474 generalizations. The size of the represented models varies, ranging from simple models with only 7 classes or models with no domain association (e.g., models that are mere taxonomies) to large and complex models with more than a thousand classes. While mean and median values show us the medium size of the models in the catalog, the mode value indicates that most of these models are small. The standard variation,

Table 1. Statistics on concepts from the catalog ontologies

	Diagrams	Classes	Associations	Generalizations
Sum	656	7223	5392	5474
Minimum	1	7	0	0
Maximum	138	1222	655	1119
Median	2	32	26	19
Mean	5,17	56.87	42.46	43.1
Mode	1	18	15	2
Standard Deviation	15.52	118.44	71.34	115.47
Sample Variance	240.84	14028.79	5089.44	13332.43
Standard Error	1.38	10.51	6.33	10.25

sample variance, and standard error values indicate that the catalog comprises models of varied sizes. This is a positive feature, demonstrating that the catalog collects a heterogeneous model sample, which can be useful for supporting different empirical analyzes.

4.2 Statistics on the Metadata

The 127 models included in the catalog have been created between 2005 to 2022 (coded in the `issued` metadata field). The models represent 161 different domains (`subject` metadata), being the most frequent ones *software engineering*, with 6 occurrences, followed by *finance* and *safety* with 5 occurrences each, and *value*, *economics*, and *education* with 4 occurrences each. Regarding the metadata field `conformsTo`, 115 (90,6%) of these models are represented using OntoUML stereotypes, and 18 (14,2%) of them directly extending UFO. Note that the sum of these values is over 127 because this field can assume multiple values—the same happens to the metadata fields `language`, `type`, `designedForTask`, and `context`. The metadata `theme` (which codes a library classification) can assume only one value, and therefore, the sum of its occurrences must be equal to the number of datasets that are in the catalog.

Considering the field `language`, the catalog has 119 (93,7%) models that use lexical terms in English (`en`), followed by Brazilian Portuguese (`pt-br`) with 10 (7,9%) occurrences, and Dutch (`nl`) with a single item (0,8%). Considering the `type` of these models, 112 (88,2%) of them are classified as domain ontologies, 12 (9,4%) as core ontologies, and 6 (4,7%) as application ontologies. Regarding their `context`, 94 (74,0%) models were created within a *research* environment, 28 (22,0%) within a *classroom* environment, and 7 (5,5%) within an *industry* setting. In terms of their purposes(`designedForTask` property), we have representatives in all the ten available classifications categories, distributed as follows: 85 (66,9%) *conceptual clarification*, 23 (18,1%) *learning*, 20 (15,7%) *interoperability*, 13 (10,2%) *software engineering*, 9 (7,1%) *ontological analysis*, 3 (2,4%)

decision support system, 3 (2,4%) *example*, 2 (1,6%) *information retrieval*, 2 (1,6%) *data publication*, and 2 (1,6%) *language engineering*.

Finally, regarding the library classification of the domains represented in these models (`theme`), of the 21 possible LCC classes, we have exemplars of 13 of them, distributed this way: 44 (34,6%) of *Social Sciences (Class H)*; 36 (28,3%) of *Technology (Class T)*; 8 (6,3%) of *Science (Class Q)*; 7 (5,5%) of *Medicine (Class R)*; 6 (4,7%) of *Geography, Anthropology, and Recreation (Class G)*; 6 (4,7%) of *Education (Class L)*; 5 (3,9%) of *Philosophy, Psychology, Religion (Class B)*; 5 (3,9%) of *Political Science (Class J)*; 4 (3,1%) of *Law (Class K)*; 3 (2,4%) of *Agriculture (Class S)*; 1 (0,8%) of *Music (Class M)*; 1 (0,8%) of *Military Science (Class U)*; and 1 (0,8%) of *Bibliography, Library Science, and General Information Resources (Class Z)*.

5 FAIRness Evaluation

In this section, we discuss how our catalog complies with the FAIR principles for scientific data management put forth by Jacobsen *et al.* [15].

Findable. The first FAIR principle refers to the importance of making (meta)data easily findable to both humans and computers. To accomplish this, the following more specific requirements are laid out [15]: (F1) the (meta)data must have "a globally unique and persistent identifier" (F2) the data must be "described with rich metadata" (F3) the metadata "clearly and explicitly include the identifier of the data they describe", and (F4) (meta)data must be registered/indexed in a searchable resource. The catalog uses persistent identifiers for all its resources (F1). Our data is described with rich metadata accessible to users (F2) and correctly referencing identifiers (F3). Finally, all (meta)data is hosted on a public GitHub repository, guaranteeing that they are indexed and findable by web search tools (F4).

Accessible. The second principle, accessibility, regards authentication and authorization. It requires that (A1) the (meta)data must be retrievable by their identifier using a standardized communications protocol that is open, free, universally implementable, and that allows authentication and authorization procedures, where necessary. It also requires that the (A2) metadata must be "accessible, even when the data are no longer available". We hosted the catalog in a public GitHub repository. Thus, all its resources are accessible to anyone with a browser and an internet connection (A1). We store data and metadata about each model in different files in our catalog (e.g., the `ontology.ttl` and `metadata.ttl` files in a model directory). Thus, even if an author removes their ontology from our catalog, its metadata will remain there (A2).

Interoperable. An important principle, in which ontology-driven conceptual models play an essential role [9], is interoperability. This principle states that the data should be able to integrate with other data, applications, or workflows. To achieve this goal (I1) the (meta)data must "use a formal, accessible, shared, and broadly applicable language for knowledge representation" (I2) the (meta)data

must "use vocabularies that follow FAIR principles", and (I3) also must "include qualified references to other (meta)data".

The datasets in the catalog are available using open, free, and standardized semantic web and syntax definition languages, such as JSON and RDF-based languages (I1). The catalog's metadata is described using FAIR vocabularies, such as DCAT, DCT, SKOS, and MOD. Our custom vocabulary, built to describe the models in our catalog, is also FAIR compliant, being accessible via a permanent URI, specified in RDF/OWL, hosted on its own GitHub repository, and with a clear license for reuse. Our metadata reuses identifiers from other data providers, such as the DBLP's author identifier, DOI, and LCC, thus paving the way for the integration with additional datasets (I3).

Finally, the interoperability of the models comprising the catalog is facilitated by having each of these models grounded on a foundational ontology (UFO), i.e., by having the domain concepts and relations in these models explicit connected to UFO's basic ontological categories.

Reusable. The last principle addresses data reuse, requiring that (meta)data should be structured and well-described, enabling them to be replicated or combined. More specifically (R1) the (meta)data must be "richly described with a plurality of accurate and relevant attributes" (R1.1) it must be "released with a clear and accessible data usage license" (R1.2) the (meta)data must be "associated with detailed provenance", and (R1.3) must "meet domain-relevant community standards". Our catalog metadata is extensive in the description of each piece of metadata, containing all relevant attributes for understanding each model, like name, source, modeler, and domain (R1). Moreover, the catalog registers the usage license for the included models (R1.1), each model in the catalog has its original source presented (publications, files, and diagrams) (R1.2), and for the whole catalog we used well-established vocabularies when defining the metadata standards (R1.3).

6 Relevance for Research

This catalog paves the way for research in the areas of ontology engineering, software design, and conceptual modeling, but also in the areas of machine learning, and, more precisely, relational learning, where the focus is to address prediction or information induction tasks by reusing knowledge encoded in a graph-structured format. We grouped some main usage examples as follows.

Algorithm Evaluation. An obstacle often found by OntoUML developers is how to evaluate their algorithms' effectiveness and performance. By lacking a reliable dataset for testing, authors most of the time rely on toy examples use-cases, or on unrealistic scoped domains. Most of the time, when testing on already published models, modelers must manually rebuild these models from their image files to produce a machine-readable version of the model. Using this catalog, interested users can find a significant amount of reliable data (since all input on the catalog is peer-reviewed) already in the desired format, thus creating a beneficial scenario for testing algorithms. To cite just two examples, algorithms that

would directly benefit from this catalog include those for automated clustering [11] and abstraction of conceptual models [19].

Language Evolution. For over fifteen years, the OntoUML research community has contributed to the development of the modeling language and to the evolution of its foundational ontology, UFO. By observing how OntoUML has been used over the years, by several groups, and in a variety of domains, one can derive fruitful empirical observations about the language. Previous works have already observed several diverse ways in which people systematically bent the syntax of the language, triggering its designers to evolve the language [12]. These *systematic subversions* refer to recurrent model fragments appearing in models produced by different users that albeit grammatically incorrect signal (to a language's creation) a design limitation of the language. The observation and analysis of these subversions in OntoUML have already been used as input to evolve UFO's theory of relations [5] and its theory of types [12]. Being a structured source of models' data in a machine-readable format, the OntoUML/UFO Catalog can be queried, and its data can be used for the identification of these subversions for further analysis.

Language Design and Evaluation. "How much language is enough?". With this question in mind, Muehlen and Recker analyzed Business Process Modeling Notation (BPMN) models using mathematical and statistical techniques and, among other findings, discovered that less than 20% of BPMN's vocabulary is regularly used—a piece of information that has implications for the entire language ecosystem and community [16]. Researchers can perform an equal analysis of OntoUML using our catalog. Such an analysis could benefit the OntoUML community by helping teachers and students to create improved pedagogical strategies. The results of such analyzes could also drive future researchers' efforts, allowing them to focus on the most used language concepts. Examples of these include [22], which proposes ontology-based rules for designing the concrete syntax of visual modeling languages, and [10], which proposed a canvas for ontology modeling. Both approaches aimed to address subsets of OntoUML/UFO categories. With this catalog, such design choices can be evidence-based.

Empirical Discovery of Modeling (Anti-)Patterns. A straightforward way to exploit the catalog is to use its information to understand how people use OntoUML in practice. This involves the empirical discovery of patterns or antipatterns [20], as good or bad modeling practices, which can be used to evolve OntoUML or by modelers that need to create new models. Note that, since OntoUML is a profile for UML class diagrams, the catalog offers an opportunity for researchers interested in the discovery patterns in that language as well.

Application Development. The availability of OntoUML models encoded in a uniform and processable format supports the development of new model engineering techniques and the improvement of existing ones. For instance, through an analysis of how people create diagrams (i.e., what are the cognitive steps in model construction), new editing and automatic layout services could be devised. Many datasets (or parts of them) may be used directly to design database

schemes for working applications of related domains, i.e., high-quality models there included can serve as *seed models* [3] for future developments.

New Source of Information for Machine Learning Set-Ups. Recently, machine learning approaches have taken advantage of graph-structured data to address specific tasks [17]. In this context, the catalog offers ODCM data that can be easily exploited and assessed in different scenarios and domains. The combination of domain-level and top-level knowledge, which is structural in OntoUML, represents an added information that is crucial sometimes. For instance, exploiting background knowledge with top-level information is recognized to be of remarkable significance for cross-domain transfer learning tasks [1,7]. Similarly, having a large catalog with domain data related to top-level data may significantly improve tasks in which schemes must be matched according to a reference standard [23]. It can also provide training data for ontology matching tasks leveraging on data annotated with categories coming from foundational ontologies—a still unexplored approach to ontology matching [24]. Finally, our catalog can be exploited as a training set for machine learning prediction tasks, where the goal is to predict the correct foundational category of a given class, thus providing automated support to build new models and define their scope.

7 Related Work

Our contribution here builds primarily on the large amount of work in recent years on the generation of repositories for maintaining and reusing knowledge resources, such as ontologies, conceptual models, and vocabularies. Based on an analysis of the collected data and their organization, we identified some initiatives that are close to ours, which we discuss in the sequel. Considering the different scope, we excluded from this section domain-specific catalogs and catalogs for artifacts different from models or ontologies (e.g., design patterns catalogs).

In the past, some of us have made a first attempt to gather and organize OntoUML models [20]. This effort gathered 54 models, most of which were (or will eventually be) included in the catalog presented here. This repository of models—which is no longer available or maintained—was created with the specific goal of supporting the empirical discovery of ontological anti-patterns [20]. Differently from the catalog described in this paper, that repository was not built in compliance with the principles represented in FAIR, or with the goal of fostering open and collaborative community participation.

The *Linked Open Vocabularies (LOV)* [25] is a platform that provides access to a catalog of OWL vocabularies. Starting in 2011, LOV is now hosted by the *Open Knowledge Foundation*, and it currently offers almost eight hundred vocabularies. LOV is based on some quality requirements, including URI stability and availability on the web. It relies on standard formats and publication best practices, quality metadata, and documentation. As a distinctive feature, LOV shows indicators that are not provided by other catalogs, such as the interconnections between vocabularies, the versioning history along with past and current editors (individual or organization). LOV is a catalog of vocabularies and/or lightweight

ontologies, i.e., semantic web models focused on web-based information sharing and computability issues. Ours, in contrast, focuses on ontology-driven conceptual models, i.e., models focused on expressivity and domain appropriateness, and capturing the result of ontological analyzes [8].

An example of a repository of ontologies as logical specifications is OntoHub [4], which collects over 20.000 specifications organized in almost 150 repositories. Most of the ontologies there are also lightweight models. We can say the same for the LOV-inspired Linked Open Vocabularies for the Internet of Things (LOV4IoT) [14], a domain-specific repository. In this sense, it is similar to the BioPortal [21], which includes almost 1.000 in the life sciences. In contrast to these other approaches, the OWL models in BioPortal usually are based on the Basic Formal Ontology (BFO) foundational ontology and, in principle, have a similar focus (e.g., with relation to domain appropriateness) to the ones in our catalog. However, despite their firm grounding, these models are subject to OWL's expressivity limitations and, hence, leave out unrepresented many important ontological nuances (e.g., related to modality, multi-level structures). Additionally, since these ontologies are rendered as textual (sentential) logical specifications, they do not provide data for supporting the study of diagrammatic/visual aspects of domain representations (e.g., model layout, visual patterns and anti-patterns).

Last, another initiative comparable to ours is the one by G. Robles *et al.* [18], who built an extensive catalog of UML models. Their approach was to gather UML models automatically from sparse GitHub projects and put them into a reference hub. The output is a catalog with over 93.000 UML diagrams from over 24.000 projects. Their catalog is clearly much larger than ours, but our goals are also different. First, in scope, they include any UML models, while we focused on ODCM models. Second, their emphasis is on quantity, while striving for a minimal quality threshold for the models and for the homogeneity of the data. All our models are available in the same formats and are described with rich and linked metadata, making our catalog much easier to reuse and analyze.

8 Final Considerations

In this paper, we presented the first FAIR Model Catalog for Ontology-Driven Conceptual Modeling Research. We provide a structured, collaborative, and open-source catalog of ODCM models designed with the OntoUML language (or by extending the UFO ontology). This resource shall support the conceptual modeling and ontology engineering communities with many important empirical tasks. These include language design, understanding, and evolution; machine-learning research over model data; testing model manipulation (e.g., code generation, mining, modularization, abstraction); and model reuse.

The catalog currently contains 127 models, but we expect it to grow, especially considering the UFO and OntoUML relevance to the ODCM field and considering that the catalog is open to receive contributions from the community. Instructions on how to collaborate are available on the repository's GitHub

page. We have identified many models that still have to be rebuilt in the Visual Paradigm to be included in the catalog. Given that this is a laborious task, we intend to investigate ways in which this process can be at least partially automated (e.g., by automating the normalization of data, partially generating models from figures). Additionally, we intend to create a service in the OntoUML plugin for modelers to submit their models directly to the catalog.

Inspired by the LOV initiative (which claims to be a high-quality catalog of reusable vocabularies), we envisage the creation of a Linked Open OntoUML Models (LOOM). Differently from LOV, LOOM would organize the space of ontologically well-founded domain models, i.e., a space of conceptual models grounded on a foundational ontology and, thus, having deeper ontological semantics by design.

Even though our catalog is restricted to UFO/OntoUML conceptual models, its metadata schema could be easily adapted to accommodate models built following other foundational ontologies. Nonetheless, to the best of our knowledge, UFO is the only mainstream foundational ontology [26] that has an ODCM language that is explicitly associated to it, i.e., in a technical sense: (i) having the modeling primitives of the language directly reflecting the distinctions of the ontology; (ii) having the grammatical constraints of the language explicit representing the axiomatization of the ontology.

Finally, as previously discussed, the models are included in this catalog in their original form, i.e., preserving the original modeling choices made by their creators. This is important to study how the language is actually used in practice, what are the most common modeling errors and anti-patterns, how different users subvert the grammatical rules of the language signaling possible evolution trends, etc. However, as a direct consequence, the catalog shall contain models that are of a variety of quality levels, including models bearing syntactic, semantic, and pragmatic problems. This hinders the potential (re)use of these models (e.g., as Seed Models or reusable modeling components). As future work, we intend to address this issue by investigating methodological and computational mechanisms for assessing some quality aspects of these models (e.g., with relation to syntactical correctness, presence of anti-patterns, visual pragmatics, among others).

Acknowledgements. We would like to thank Accenture Israel Cybersecurity Labs for supporting this work, as well as Ítalo Oliveira, Thomas Derave, Tim van Ee, Cristiano Silva, and Lucas Maddalena for their contributions to the catalog. We especially thank András Komáromi for his contribution and inspiring passion for modeling in OntoUML.

References

1. Al-Halah, Z., Stiefelhagen, R.: How to transfer? Zero-shot object recognition via hierarchical transfer of semantic attributes. In: WACV, pp. 837–843 (2015)
2. Barcelos, P.P.F., et al.: An automated transformation from OntoUML to OWL and SWRL. In: Seminar on Ontology Research in Brazil, vol. 1041 (2013)
3. Blaha, M.: Patterns of Data Modeling, vol. 1. CRC Press (2010)

4. Codescu, M., et al.: Ontohub: a semantic repository engine for heterogeneous ontologies. Appl. Ontol. **12**(3–4), 275–298 (2017)
5. Fonseca, C.M., et al.: Relations in ontology-driven conceptual modeling. In: Conceptual Modeling. ER 2019, vol. 11788, pp. 28–42 (2019)
6. Fonseca, C.M., et al.: Ontology-driven conceptual modeling as a service. In: JOWO 2021. The Joint Ontology Workshops, vol. 2969 (2021)
7. Fumagalli, M., et al.: Ontology-driven cross-domain transfer learning. In: Formal Ontology in Information Systems, vol. 330, pp. 249–263 (2020)
8. Guizzardi, G.: Ontological foundations for structural conceptual models (2005)
9. Guizzardi, G.: Ontology, ontologies and the "I" of FAIR. Data Intell. 2(1–2) (2020)
10. Guizzardi, G., Sales, T.P.: "As simple as possible but not simpler": towards an ontology model canvas. In: Proceedings of the Joint Ontology Workshops (2017)
11. Guizzardi, G., et al.: Automated conceptual model clustering: a relator-centric approach. Softw. Syst. Model (2021)
12. Guizzardi, G., et al.: Types and taxonomic structures in conceptual modeling: a novel ontological theory and engineering support. Data Knowl. Eng. **134**, 101891 (2021)
13. Guizzardi, G., et al.: UFO: unified foundational ontology. Appl. Ontology **17**(1), 167–210 (2022)
14. Gyrard, A., et al.: LOV4IoT: a second life for ontology-based domain knowledge to build semantic web of things applications. In: IEEE FiCloud (2016)
15. Jacobsen, A., et al.: FAIR principles: interpretations and implementation considerations. Data Intell. **2**(1–2), 10–29 (2020)
16. Muehlen, M., Recker, J.: How much language is enough? Theoretical and practical use of the business process modeling notation. In: CAiSE 2008, pp. 465–479 (2008)
17. Nickel, M., et al.: A review of relational machine learning for knowledge graphs. Proc. IEEE **104**(1), 11–33 (2016)
18. Robles, G., et al.: An extensive dataset of UML models in GitHub. In: 14th International Conference on Mining Software Repositories, pp. 519–522 (2017)
19. Romanenko, E., et al.: Abstracting ontology-driven conceptual models: objects, aspects, events, and their parts. In: RCIS (2022)
20. Sales, T.P., Guizzardi, G.: Ontological anti-patterns: empirically uncovered error-prone structures in ontology-driven conceptual models. DKE **99**, 72–104 (2015)
21. Salvadores, M., et al.: Bioportal as a dataset of linked biomedical ontologies and terminologies in RDF. Semantic Web **4**, 277–284 (2013)
22. da Silva Teixeira, M.D.G.: An ontology-based process for domain-specific visual language design. Ph.D. thesis, Ghent University (2017)
23. Sleeman, J., Finin, T., Joshi, A.: Entity type recognition for heterogeneous semantic graphs. AI Mag. **36**(1), 75–86 (2015)
24. Trojan, C., et al.: Foundational ontologies meet ontology matching: a survey. Semant. Web (2021)
25. Vandenbussche, P., et al.: Linked open vocabularies (LOV): a gateway to reusable semantic vocabularies on the web. Semant. Web **8**(3), 437–452 (2017)
26. Verdonck, M., Gailly, F.: Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling. In: ER (2016)
27. Wilkinson, M.D., et al.: The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data **3**(1) (2016)