Assembly Line for conceptual models

Karel Klima [a], Petr Kremen [b], Martin Ledvinka [b], Michal Med [b], Alice Binder [b], Miroslav Blasko [b] and Martin Necasky [a]

[a] *Department of Software Engineering, Charles University in Prague, Czechia*
E-mails: karel.klima@matfyz.cuni.cz, martin.necasky@matfyz.cuni.cz
[b] *Department of Computer Science, Czech Technical University in Prague, Czechia*
E-mails: kremep1@fel.cvut.cz, ledvima1@fel.cvut.cz, medmicha@fel.cvut.cz, binder.alice@fel.cvut.cz, blaskmir@fel.cvut.cz

Abstract. Assembly Line is an open-source web platform for collaborative development of ontologies. The platform is generic and domain independent by design, and we demonstrate its capabilities within the domain of Czech public administration – in this case Assembly Line facilitates building a common Czech eGovernment ontology. In our previous work we introduced the semantic government vocabulary (SGoV), an ontologically-founded layered vocabulary reflecting the diversity of public administration domains and legal concepts important in these domains. We presented SGoV as a rich conceptual model to describe the meaning of open government data published by different public authorities using shared vocabularies. Assembly Line is a natural continuation of that work. It is a platform designed to create, manage, and publish SGoV. The platform includes tools that enable various offices within public administration to manage their own vocabularies, while ensuring semantic integrity of the SGoV as a whole. We demonstrate the impact of the Assembly line on two data publication scenarios in the Czech public administration domain, the registry of rights and obligations and the electronic code of laws.

Keywords: ontology engineering, conceptual modeling, thesaurus management

1. Introduction

Today, many organizations have their data spread across many data sources. The data sources are heterogeneous in their syntax, structure (schemas), and semantics [1]. Heterogeneity leads to a wide variety of data, and it is one of the main challenges for organizations today [2, 3]. Heterogeneous data sources are usually called *data silos* and their integration poses a significant challenge to the organization [4]. A possible approach is to employ an organization's ontology as a semantic layer on top of the data silos. This approach is commonly called *Ontology-Based Data Access (OBDA)* [5]. Many practical applications of OBDA have recently appeared [6]. Their goal is to allow users to access data in silos in a uniform manner.

In [6], several successful applications of OBDA in the public administration sector are described. A public administration comprises various public authorities that manage data in their own data silos. Therefore, it could also benefit from OBDA. According to the original OBDA paper [5], the main reason to apply OBDA is to provide high-level services on top of the data silos. The most important service is query answering according to [5] and more recent publications [3, 6, 7]. This could be too ambitious for public administration. However, other services could also be useful. The semantic layer can serve as a data map used for navigation in data silos managed by different public authorities. It can help identify and locate silos where a certain semantic concept is represented or find out how two silos are semantically related to each other.

The semantic layer comprising ontological concepts is crucial for providing these services. In our work, we focus on enabling the public administration to build such a layer. Developing an ontology on the national government scale has certain specifics. First, the semantic layer should include important concepts from many different public administration domains ranging from citizen evidence through the vehicle registry to the hop farm registry. Second, the core source of these concepts is legislation, which also defines public authorities responsible for particular domains and their competences in these domains.

Example 1 (The model of car life-cycle in Czechia): The life-cycle of a car from the public administration's point of view starts when it is produced and registered. Then it is sold to the first owner and registered under their name. The owner first needs to buy mandatory insurance and then uses the car for some time. Once every few years the car has to undergo a regular mandatory technical inspection. During its operation, the car may be a subject of other inspections and checks. For example, when a police officer stops the driver for speeding, they may also need to check the technical state of the car. When driving on a highway, the car is checked for the paid toll. Similarly, in a city center which is a low emission zone, the car may be checked for its technical parameters recorded during its last mandatory technical inspection. Finally, when the car is too old or damaged in an accident, it is decommissioned and becomes a wreck that must be disposed of ecologically. From the legislative point of view, the car life-cycle is a collection of separate administrative actions, each possibly governed by a different legal act and respective public administration body. The actions and respective acts are shown in Table 1.

The example shows how the definition of a single concept can be separated by different legal acts. Each act then defines the competences of different public authorities. This needs to be reflected when building an ontology for the public administration. We cannot count on an upper authority to take responsibility for all the other authorities to semantically describe their data. We need to deal with the independence of the individual authorities and enable them to build the semantic layer in mutual cooperation without influencing each other beyond the law. The semantic layer must consist of semantic models governed independently by respective authorities.

In our previous work we introduced the semantic government vocabulary (SGoV) [8], an ontologically founded layered vocabulary reflecting the diversity of public administration domains and legal concepts important in these domains. We presented SGoV as a rich conceptual model to describe the meaning of open government data published by different public authorities using shared vocabularies.

Since its publication, we presented our approach on several occasions to representatives of Czech public authorities including system architects responsible for the core architecture of the Czech eGovernment. They were interested in the proposal and requested a modeling environment that would respect the specifics of public administration.

Motivated by their needs we proceeded to design and develop such environment which we call the Assembly Line (AL). As the contribution of this paper, we introduce the AL, its features, architecture and impact. The paper

Table 1
Overview of legal acts related to a car life-cycle in Czechia

Actions	Legal Act
Car production, registration and regular technical inspections	Act on Conditions of Operation of Vehicles on Roads (56/2001)
Driving a car and car inspections on the road	Road Traffic Act (361/2000)
Decommissioning a car and its ecological disposal	End of Life Products Act (542/2020)
Highway tolls and registering car for the paid tolls	Decree on the use of toll roads (470/2012)
Low emission zones and permissions to enter them with a car	No act has been issued yet

is structured as follows. In Section 2 we introduce necessary preliminaries. In Section 3 we present functional and qualitative requirements on AL. In Section 4 we describe the process of collaborative vocabulary design in AL. Section 5 provides an overview of the architecture of AL. In Section 6, we evaluate AL from two key perspectives – by providing a comprehensive user testing and demonstrating impact of AL by an outline of real-world scenarios in the governmental domain. Section 7 describes other state-of-the-art tools and compares them with AL. We conclude in Section 8.

2. Preliminaries

The Assembly Line (AL) serves to design the Semantic Government Vocabulary (SGoV) [8], a set of interlinked semantic vocabularies describing open data sets originating in Czech legislation. The SGoV semantic vocabularies consist of SKOS-based [9] thesauri and OWL-based [10] ontologies representing OntoUML [11] conceptual models. Since more detailed examples can be found in [8], we will only present these technologies informally, on a running example – the reader might want to refer to section 6 for a more detailed explanation of the context of the running example. Last, we show how the architecture of the semantic vocabularies look like.

2.1. SKOS

Simple Knowledge Organization System (SKOS) is a well-known standard for representing thesauri on the semantic web. The main value of thesauri lies in their simplicity and understandability by domain experts who are able to define new concepts, find their normative references and organize them in hierarchies. An example of a SKOS concept in Turtle syntax[12] is shown below[1]:

```
l-sgov-222-2016:legal-act a skos:Concept ;
  skos:prefLabel "Legal Act"@en ;
  skos:altLabel "Act"@en ;
  skos:inScheme l-sgov-222-2016:thesaurus ;
  skos:definition
    "A law, a ministerial decree,
     a constitutional law,
     a statutory measure of the
     Senate, a government
     regulation, or an
     implementing legislation"@en ;
  skos:broader eli:LegalResource .
```

It says that Act 222/2016 Coll. introduces a concept with an IRI of `l-sgov-222-2016:legal-act` and a preferred English label of *Legal Act*. The concept has an alternative label (a synonym) of *Act* and belongs to the thesaurus of Act 222/2016 Coll. Also, the concept has a narrower meaning than the concept `eli:LegalResource`.

[1]The following prefixes are used – `eli:` denotes `http://data.europa.eu/eli/ontology#`, `skos:` denotes `http://www.w3.org/2004/02/skos/core#` and `l-sgov-222-2016:` denotes `https://slovník.gov.cz/datový/sbírka/pojem/`

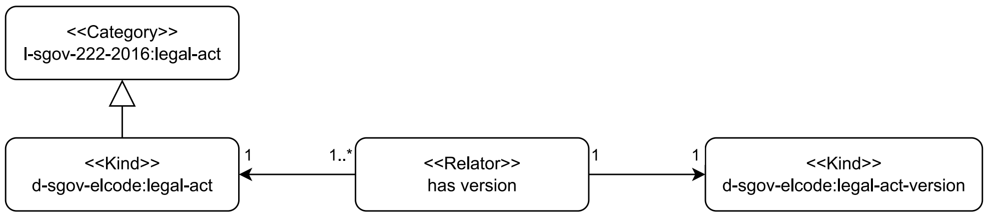

Fig. 1. Simple conceptual OntoUML model with stereotypes. *Category* (a rigid mixin) says the class can have instances with different identification schemes, while *Kind* (a rigid sortal) says the class defines a unique identification scheme for its instances. In either case the class is rigid (not contingent) – an instance belongs to a rigid class all the time, e.g. a legal act 222/2016 cannot cease to be a legal act. *Relator* represents a reified relationship, a dependent entity which exists as long as its participants exist.

2.2. Unified Foundational Ontology and OntoUML

Unified Foundational Ontology (UFO)[11], is a top-level ontology designed for conceptual modeling. OntoUML, an extension to UML, has been designed to validate conceptual models using UFO constraints (e.g., an antirigid type, like Approver, cannot be specialized by a rigid type, like Physical Person).

Consider the concept introduced in Section 2.1 together with three other concepts (which might have a similar SKOS representation).[2]:

- `d-sgov-elcode:legal-act`
- `d-sgov-elcode:has-version`
- `d-sgov-elcode:legal-act-version`

Based on these concepts, a simple conceptual model can be created. E.g., one might want to express that a legal act has a version and each such version is actually an instance of the class `d-sgov-elcode:legal-act-version`, as shown in Figure 1.

2.3. OWL

OWL is a W3C standard for representing web ontologies. Although it is suitable for automated reasoning and compliant with semantic web standards, it is a low-level language, not easily understandable even by many knowledge modelers. Yet, it is powerful enough to represent expressive conceptual models. Several partial translations from OntoUML models to OWL have been defined in [13]. Taking the example in Figure 1, one possible transformation to an OWL ontology can look like the listing in Figure 2.

2.4. Architecture

SGoV vocabularies [8] are defined in layers and can inherit from each other. Namely,

Basic Vocabulary defines top-level domain-independent common-sense terms, introduced by UFO, e.g. `Object` or `Relator`.

Public Sector Vocabulary describes shared concepts across public sector. Examples of concepts involve e.g. `LegalSubject` or `PublicAuthority`.

Legal Vocabularies describe terms of a single legal act. Example concept of the Act No 222/2016 is the concept `Legal Act`.

Data Vocabularies describe concepts used in data schema of a particular dataset (e.g. column names of a CSV table).

[2]The prefix `d-sgov-elcode:` denotes the namespace `https://slovník.gov.cz/datový/sbírka/`

```
d-sgov-elcode:has-version
  a owl:ObjectProperty, z-sgov:Aspect .
d-sgov-elcode:legal-act-version
  a owl:Class, z-sgov:Phase .
l-sgov-222-2016:legal-act
  a owl:Class, z-sgov:Kind ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty
      d-sgov-elcode:has-version ;
    owl:someValuesForm
      d-sgov-elcode:legal-act-version
  ] ;
  rdfs:subClassOf [
    a owl:Restriction ;
    owl:onProperty
      d-sgov-elcode:has-version ;
    owl:allValuesForm
      d-sgov-elcode:legal-act-version
  ] .
```

Fig. 2. Example of OWL representation of a OntoUML model.

3. Requirements

As discussed in Section 1, public administration in the Czech Republic comprise independent authorities, each responsible for some governmental domain. Yet, data produced by these agendas are inherently semantically connected. This distributed setup poses requirements that are formulated next in this section.

3.1. Functional requirements

Based on the use cases we have identified the following requirements:

1. **Vocabulary modeling tools**. The Assembly Line should let users create vocabularies in a user-friendly way by providing a set of modeling tools that would guide the user through the process of creating a vocabulary, ensuring its consistency and linking it to other vocabularies. Comprehensive knowledge of underlying ontology-related technologies should not be required to successfully create or modify a vocabulary.
2. **Collaborative workflow support**. Creating a vocabulary consists of three major stages. First, a thesaurus of terms is compiled by domain experts, followed by building a conceptual model (connecting the terms) on top of the glossary by data modeling experts. Last, it is necessary to peer-review the vocabulary and confirm that it is compatible with common SGoV principles and other vocabularies. The Assembly Line should accommodate all of these roles by providing specialized set of tools tailored to particular use cases, while taking the user knowledge and experience level into account.
3. **Publishing of vocabularies**. After a new vocabulary is created (or an existing one updated) and reviewed, it needs to be published properly in order to be reused by other vocabularies as a dependency or data sets as data schema. The vocabulary must be published under a permanent URL to provide long-term accessibility.
4. **Automated syntax and constraints checking**. The Assembly Line should provide means of assisting users with automated evaluation of produced vocabularies. Whereas it is difficult to automatically verify factual accuracy of vocabulary glossary and model (as domain knowledge is needed), the tools for creating and reviewing vocabularies need to be able to verify formal qualities of a vocabulary, that is, check that the vocabulary syntax and schema are valid and that connections to other vocabularies or other resources are valid as well.
5. **Versioning**. Majority of particular vocabularies within the SGoV we proposed are based on existing legal acts. When a law changes, the corresponding vocabulary may need to be updated as well, which may lead to

conflicts. Therefore the Assembly Line needs to define and implement a versioning support so that the SGoV can be safely extended and updated, while maintaining backwards compatibility.

6. **Decentralized vocabulary governance**. The SGoV is designed to be a singular coherent mass of interlinked vocabularies, but in order to enable its adoption across various data publishers, the governance of sections of SGoV needs to be decentralized. First and foremost, primary users of SGoV will be state offices and bureaus with separate competencies and agendas defined by law; if such actors publish agenda-specific data using vocabularies from SGoV as data schema, it is highly efficient for them to maintain the vocabularies themselves, as they usually have all the domain knowledge already; in addition they do not need to get blocked by an independent review and publishing endeavour, should the governance be centralized. Last but not least, such actors are usually reluctant to give up any more of their independence than they have to, so we can safely conclude that having a centralized governance system would at the very best slow the adoption of SGoV, or likely make it impossible altogether. It should therefore be possible for a data publisher to host and govern its own set of vocabularies and run its own Assembly Line. With that said, coherence of vocabularies within SGoV is of paramount importance and needs to be maintained rigorously. To accomplish that, a central authority to govern over core vocabularies needs to be established.

3.2. *Qualitative requirements*

1. **Modularity, composability and extensibility**. The architecture of the Assembly Line should be modular and its components loosely coupled so that all of its parts can be swapped if needed. For example it should be possible to replace an authentication module or a graph database. It is therefore desirable that the interface of components follows common existing standards and best practices whenever possible. The feature set of the Assembly Line should also be extensible so that users can create and set up custom applications (e.g. editors or visualization tools) for the vocabulary data. In order to support extensibility the architecture should provide a common interface to the core services and data operations.
2. **Scalability and distributability**. Our ambition is to have the Assembly Line solution distributed across all the relevant authorities. We need to be able to deploy multiple interdependent instances. Each instance would host its own set of vocabularies, but users may choose to select dependencies to vocabularies hosted in other instances. For example, one authority may host an important vocabulary that other vocabulary producers may reuse.

4. Collaborative workflow design

This section describes the process of updating the SGoV, either by updating existing vocabularies or creating new ones.

As described in [8], a vocabulary consists of a thesaurus that collects terms and a conceptual model that describes links between terms. Both the thesaurus and the model usually refer to terms from other vocabularies as well, creating one coherent graph. An update of SGoV therefore consists of modification of a thesaurus or a model of one or more vocabularies. The high-level process of updating a vocabulary is illustrated in Figure 3.

4.1. *Designing a vocabulary*

Users of the Assembly Line perform changes to vocabularies through *workspaces*. A workspace is an isolated sandbox-like data space that hosts writable draft versions of vocabularies. In order to create a new vocabulary, a user creates a workspace first and then creates a new vocabulary within; similarly in order to edit an existing vocabulary that is already a part of SGoV, it is possible to import said vocabulary as a writable copy to the workspace. Number of vocabularies within a workspace is not limited – when designing a vocabulary it is often necessary to make some adjustments to related vocabularies as well, and users can do just that by having multiple vocabularies in a single workspace. The data structure of workspaces is further described in Section 5.2.

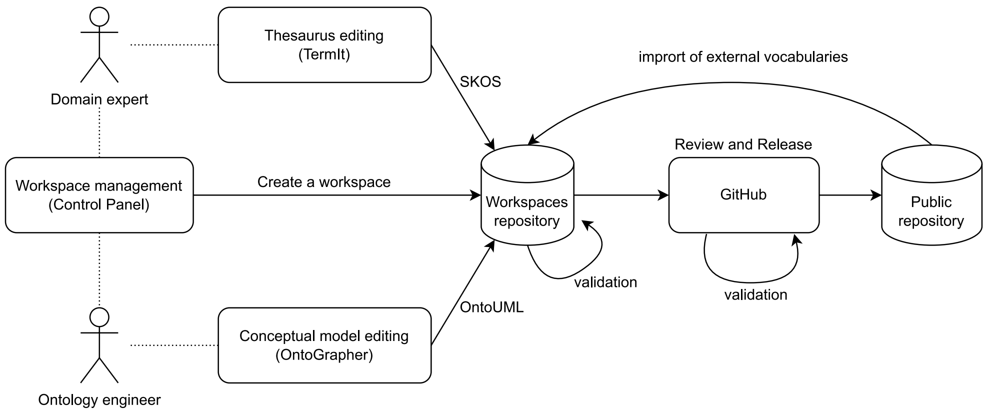

Fig. 3. Collaborative workflow process. In order to commence vocabulary modeling, a user must create a workspace, which is an isolated environment that hosts user changes. After that the user may either import an existing vocabulary to the said workspace, or create a new one. After the user is done with editing, the changes are exported to GitHub for review. After the review passes, the changes are published and incorporated into the original vocabulary. After that the workspace may be removed.

The Assembly Line provides users (both domain experts and ontology engineers) with several modeling tools for vocabulary editing and all of the tools operate on whole workspaces, which means that the users are able to modify multiple vocabularies at once. In addition, users may share workspaces they create with other users in order to collaborate on vocabularies.

4.2. Review of proposed changes

After a user is done with creating or editing vocabularies within a workspace, they can initiate a publishing process, comprising manual and automated checks. Domain experts should review updates to vocabulary thesauri, whereas ontology engineers should verify changes in model. Such manual review should be complemented by automated verification where possible, according to the requirements outlined in Section 3. Any issues identified within the review process must be addressed by the contributors to the vocabularies, and after that the review process must be restarted.

4.3. Publishing of modifications

If the review passes, then the vocabularies residing in the workspace are incorporated into SGoV – existing original vocabularies are replaced with the modified versions and new vocabularies are created. After that, the workspace may be safely deleted; the new or modified vocabularies may be further edited using another workspace when needed.

To ensure efficient collaboration, workspaces should be short-lived and the changes in vocabularies should not diverge much from the original SGoV content, most importantly when editing an existing vocabulary. First, it may be difficult to peer-review the vocabularies and ensure compatibility with SGoV if there is too much new content, especially if the reviewed vocabularies are interlinked with multiple other vocabularies from SGoV. Secondly, there is a risk of non-trivial merge conflicts since one vocabulary may be edited in several workspaces in parallel – e.g. users may create duplicate terms or link to terms that may have been modified or removed.

5. Architecture

We describe the architecture of AL from two different perspectives - first, an introduction to the component structure of the framework is given. Next, we describe the information architecture of the solution.

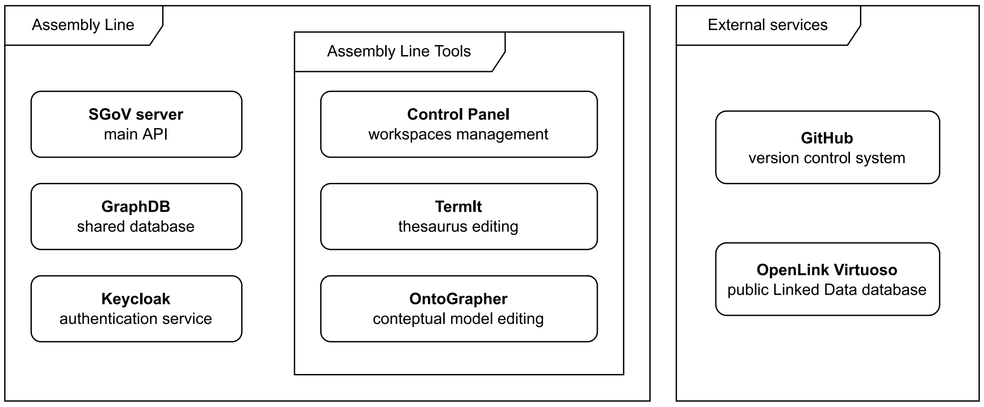

Fig. 4. Architecture of the Assembly Line. AL consists of three shared core services and a set of tools that interface with the core services. In addition, there are two external dependencies - a version control system and a public database for publishing vocabularies.

5.1. Components

The Assembly Line comprises of three kinds of components: Core components, Assembly Line Tools, and External services. Overview of the components is given in Figure 4.

5.1.1. Core components

Core components facilitate the fundamentals of the Assembly Line operation and together comprise the main infrastructure and interface that can be used by other components.

1. **SGoV server**: provides common API for validation of workspaces and vocabularies and for publishing workspaces as SGoV increments; facilitates mirroring of data between GitHub and the graph database.
2. **Graph database**: storage for SGoV vocabularies and workspaces.
3. **Authentication service**: OIDC-based authentication service.

5.1.2. Assembly Line Tools

AL Tools are user-facing components that facilitate management and editing of vocabularies. The tools are loosely coupled and use the shared data-centric interface provided by the core components. The platform is designed to accommodate as many tools as needed, as long as they adhere to the interface.

The three most important tools that have been developed so far are:

1. **Control Panel**: provides front-end interface to create and edit user workspaces, as well as create or add existing vocabularies to a workspace and publish them.
2. **TermIt**: lets domain experts view and edit thesauri of vocabularies within a context of a workspace.
3. **OntoGrapher**: lets ontology engineers view and edit conceptual models of vocabularies within a context of a workspace.

Table 2

Important GitHub repositories of Assembly Line components

Component	Repository URL
SGoV vocabularies	https://github.com/opendata-mvcr/ssp
OntoGrapher	https://github.com/opendata-mvcr/ontoGrapher
TermIt	https://github.com/opendata-mvcr/termit
SGoV server	https://github.com/opendata-mvcr/sgov
Control Panel	https://github.com/opendata-mvcr/mission-control
SGoV deployment	https://github.com/opendata-mvcr/sgov-assembly-line

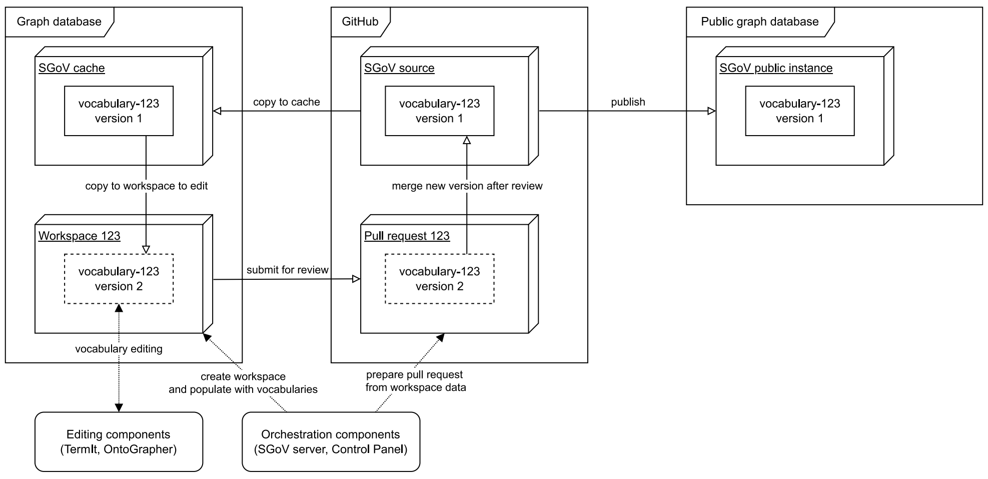

Fig. 5. Information architecture including data flow. Data are stored in graph databases and a GitHub repository. Editable versions of vocabularies are stored within workspaces, which are accessible by Assembly Line editing tools. The updates that are made to vocabularies are validated by SGoV server and then the whole workspace is prepared for review in form of a GitHub pull request.

5.1.3. External services

The Assembly Line depends on two external services that are incorporated into the vocabulary editing and publishing process, but are otherwise standalone.

1. **GitHub**: provides basic collaborative tools to review and merge vocabulary increments into the vocabulary system.
2. **Public graph database**: official public repository of SGoV vocabularies where vocabulary modifications made within AL are published.

As per the requirements, all the components mentioned above are loosely coupled and whenever possible use industry standard interfaces so that it is possible to swap particular components if needed, for example replace the authentication service with a different kind.

5.2. Information architecture

To accommodate the fundamentals of the collaborative workflow requirements, we decided to design the collaboration process around GitHub. GitHub provides a solid collaboration framework and infrastructure out of the box, and its recommended workflow of feature branches and pull requests corresponds to the SGoV collaborative workflow design outlined in the previous chapter.

GitHub is best for collaborating on simple text file based data, which is convenient, as the SGoV can be serialized to the Turtle format and stored as such. Although in theory the whole SGoV could be stored only in a GitHub repository and the updates realized through pull requests, when it comes to editing or reviewing vocabularies, it is not user-friendly nor convenient to work directly with text files. Instead, a use of a graph database is preferred to leverage powerful data querying using industry standard languages (SPARQL), to be used either directly or indirectly via Assembly Line Tools. As a result, we designed a cache layer consisting of a graph database that contains a copy of SGoV. We treat the GitHub repository hosting SGoV as the source of truth, mirroring the data to the graph database cache automatically whenever the SGoV repository changes. Thanks to this dual structure, we can get the best of both worlds when it comes to graph data structures – we get the version history, diff and review tools provided by GitHub, and at the same time by having a graph database we can use existing tools to easily access the data.

The overview of data architecture and interaction between AL components is illustrated in Figure 5.

In addition to the convenience copy of SGoV, the graph database serves as a repository of *workspaces* – structured graphs that contain editable copies of existing vocabularies or new vocabulary candidates. Workspaces are managed by AL orchestration tools (SGoV server, Control Panel). On top of creating, editing and validating workspaces, these tools also facilitate the publishing process. When a workspace is ready to be published (a user is finished with editing of its content), all the vocabularies in the workspace are serialized to text and committed in the form of a feature branch and a corresponding pull request to GitHub repository for review. After the review the new changes may be incorporated to SGoV by merging the pull request, resulting in old versions of vocabularies being overwritten by new versions contained in the pull request.

Finally, we maintain an official SGoV public repository, which is a separate graph database that gets automatically updated upon changes in SGoV.

6. Evaluation and Impact

We present two different aspects of evaluation of Assembly Line:

1. **User testing**: to verify usability and user satisfaction of Assembly Line.
2. **Impact**: to present modeling scenarios where AL was successfully applied in the real world.

6.1. User testing

In order to validate our assumptions regarding usability of Assembly Line, we have conducted a user testing comprising of various modeling scenarios. The scenarios were designed to cover all important features of AL needed to successfully create interconnected vocabularies based on real world artifacts.

The user testing covered following areas:

1. Creating a workspace
2. Creating a vocabulary
3. Adding terms to a glossary of a vocabulary
4. Adding relations between terms (a model) of a vocabulary
5. Connecting multiple vocabularies together
6. Inferring relations and properties of terms from vocabulary diagrams

Users of Assembly Line may vary significantly in terms of ontology modeling skills, so we decided not to measure quality of the outcome of the testing scenarios (the ontologies that the test users created), but instead focus on usability and user satisfaction. Therefore we have opted for System Usability Scale (SUS) [14] as an industry standard tool to measure software usability, comprising of:

1. **effectiveness**: can users successfully achieve their objectives.
2. **efficiency**: how much effort and resource is expended in achieving those objectives.
3. **satisfaction**: was the experience satisfactory.

The test user group consisted of *sixteen* participants that were asked to complete five test scenarios and fill in the SUS questionnaire. Most of the respondents were real future users of AL – clerks from various sectors of public administration that had undergone one day modeling training prior to the user study and were given basic introduction to AL. The rest of the respondents were ontology engineers with limited or no previous experience with Assembly Line.

The resulting SUS score for Assembly Line is **68,3**[3], which can be characterized as **good** [15] and acceptable for general use.

On top of the standard SUS questionnaire, we also asked the test participants to provide details of what they like or not regarding the Assembly Line. Several of the respondents appreciated the fact that AL connects multiple distinct tools in a productive way. Some respondents with prior ontology modeling experience claimed that AL may be superior to other modeling solutions that they had used in the past. A few of the responses included actionable constructive feedback regarding user experience and application design and how to improve it.

[3]Full responses to the SUS questionnaire are available at https://github.com/opendata-mvcr/sgov-assembly-line/blob/1cdec56d5322c0ad170ce9f3ef8b2e54f029e6fb/docs/AssemblyLine_SUS_questionnaire_results.csv

6.2. *Impact*

We present two different modeling scenarios where AL was successfully applied to demonstrate the impact of AL to the real world. We also show how AL facilitated the integration of both cases.

6.2.1. *Registry of rights and obligations*

The Czech eGovernment is built around *base registries* which are authoritative sources of truth about certain kinds of entities. Any governmental information system is required to have its data synchronized with data in the base registries. There are 4 base registries:

– Registry of persons
– Registry of residents
– Registry of territorial identification, addresses and real estate
– Registry of rights and obligations

The Ministry of Interior, which is responsible for the registry of rights and obligations, decided to publish the content of the registry as open data. They needed to publish the content in a machine readable format split into several data sets due to the size of the registry. It was necessary to document for each data set which part of the registry is available in that data set. The idea was to provide logical data schemas of these data sets and their documentation which would explain the semantics of the data schemas and also semantics of the relationships between the data sets.

The ministry identified main types of entities and decided that there will be a separate data set for each entity type. This led to a decision to publish the content of the registry in 33 data sets. This required to create 33 data schemas, their documentation and also documentation of relationships between the data sets. The content of the registry is defined by the legal act on base registries (Base Registries Act, no. 111/2009) and it was important for the ministry to document also the relationships between the data sets and respective sections of the act. Moreover, the act is the subject of amendments so it is necessary to reflect the changes in the published content, schemas, documentation and relationships. To illustrate this, there were 7 amendments of the act in the last 2 years and 4 of them had impact to the semantics and structure of the content of the registry.

We can identify identify three stakeholders in this scenario:

– registry administrator - the company delivering and administering the information system of the registry for the ministry, responsible also for exporting the content of the registry to a representation which will be published as open data
– open data publisher - the ministry responsible for the content of the registry, designing the structure of published datasets, their logical schemas and documentation
– open data consumer - a third party which downloads the data sets published as open data and creates own services using the data

We were asked by the ministry to design the data sets, their logical schemas and documentation. The requirement was to enable domain experts from the ministry to discuss with us about the designed content of the data sets even without their technical knowledge of database and web technologies, JSON, etc. Therefore, instead of direct design of JSON schemas and writing the documentation, we used AL to design the semantic vocabulary of respective parts of the Base Registries Act. It has shown that some parts of the registry are more detailed than specified by law so we also created a complementary vocabulary for these parts. The resulting vocabulary has 508 concepts. We then defined for each data set the corresponding part of the semantic vocabulary and used it as the semantic model of the data set. The advantage for the ministry as the open data publisher was that using the semantic vocabulary their non-technical people were able to discuss with us about the designed data sets, relationships between them and relationships to the Base Registries Act.

Based on this semantic model, we defined a hierarchical JSON structure in which the data set would be published. We expressed the structure as a JSON schema for each data set. We then mapped each its structural elements to the vocabulary using a JSON-LD context. Using the vocabulary and JSON-LD context we generated the documentation of each data set with a script. The documentation presents the concepts relevant for a data set in a human readable

HTML form. Moreover, the script also generates examples of queries on the published data sets as a part of the documentation. They are SPARQL queries on top of the RDF representation which can be easily obtained by open data consumers by applying the JSON-LD contexts. The process reduces unnecessary human work and ensures that all technical artefacts for each data set are consistent with each other and with the actual legislation. Therefore, it reduces the effort for the open data publisher. It is also useful for the registry administrator who uses the artifacts for defining a correct exporting mechanism and also for a open data consumer who receives not only documentation of data structures with some human readable explanations but exact semantic specification linked with the relevant legislation. This increases confidence about correct data processing on both sides.

Regarding the legislation changes, whenever an amendment to the act is published we update the semantic vocabulary manually. This allows us to to easily analyze the impact of the changes to the structure of the data sets and identify which parts of the JSON schemas and JSON-LD contexts need to be updated. Their necessary updates need to be done manually but the documentation is updated automatically.

The created semantic vocabulary is valuable for the ministry on its own. For example, it can be viewed and browsed in the Assembly line by employees of the ministry which need to know the semantics of the content of the registry of rights and obligations. Before the existence of the vocabulary, there was only the technical documentation of the internal relational database structure which was incomprehensible to non-tech savvy people. They can even use the model for other purposes, e.g. to define necessary non-public data outputs from the registry for the internal purposes, analyze the impact of prepared amendments of the Base Registries Act, etc.

6.2.2. Electronic Code of Law

For over a decade, Czech legislation has been issued in the form of PDF documents containing the wordings of approved bills (of legal acts and other governmental legal documents). These PDF documents are published online and can be retrieved based on their identifier, or keywords in their textual annotations. The actual content of the bills is not searchable. Also, any analysis of the actual structure of the legislation is limited, as no machine readable representation of the Czech legislation is available. These problems have been tackled by a tender (worth a few dozen million euros), aiming at delivering a web portal, APIs as well as Open Data of the Electronic Code of Law (ECL), called e-Sbírka in Czech.

Open data shall be delivered to the general public in two forms – as a set of JSON documents, and as a SPARQL service endpoint populated by JSON-LD versions of these documents. Furthermore, the data should be compliant with the European Legislation Identifier (ELI) standard [16]. Thus we can identify three stakeholders participating in this scenario:

- open data designer – responsible for the design and actual delivery of the services of the ECL, including the design of the open data sets.
- open data publisher – responsible for publishing the open data, making them understandable, explorable and understandable. Open data publisher uses the Assembly line to design the conceptual model of the ECL and publish the open data in compliance with the resulting OWL version of the model.
- open data consumer – general public, data journalists who want to analyse Czech legislation.

In this scenario, the assembly line has been used to design the ontological model of the open data. Key parts of the ontological model are sketched in Figure 6. It shows, for example, that legal act versions do not have a direct counterpart in the relevant Czech legislation, while they do in ELI. Modeling the semantic mismatches between the data models and the conceptual models of the legislation are one of the benefits of the presented approach.

The ontological model has been designed from two directions:

bottom-up based on cca 30 JSON documents (draft versions of the open data sets) describing e.g. legal acts, their versions, validity, their decomposition to chapters, paragraphs and articles, we designed their ontological model, relating its entities to existing semantic vocabularies (e.g. the public sector vocabulary) as well as new vocabularies reflecting the existing legislation (e.g. the Act No. 222/2016 Coll. on Electronic Code of Law) .

top-down based on the ELI ontology, we identified the key parts needed to describe the open data of ECL, this involved mainly the legal acts, their parts and versions, leaving out some other parts which are not easily mapable to ELI (like attachments, or links to external classifications).

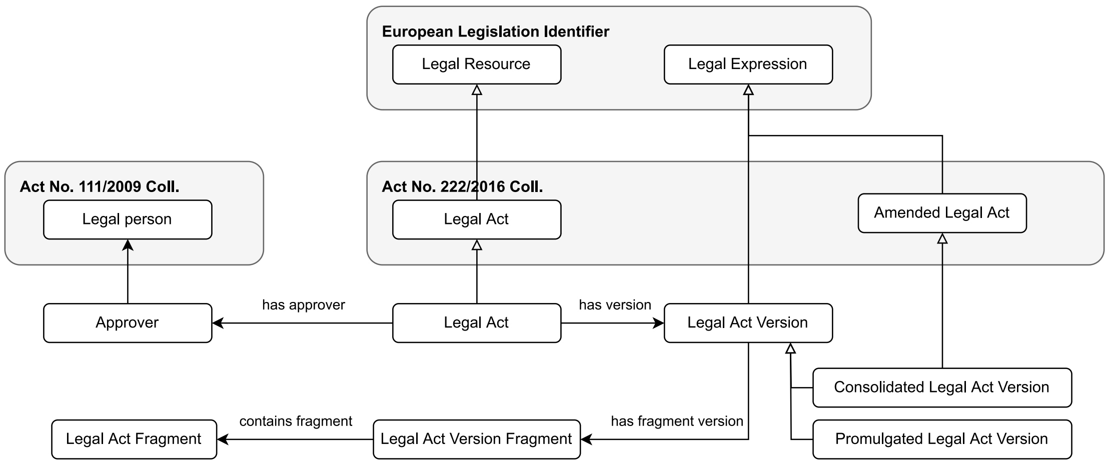

Fig. 6. Key concepts from the model of the Electronic Code of Law (out of boxes) and their mapping to the Czech legislation and ELI.

The Assembly line has been used to design the respective vocabularies and interlinking them. The resulting model has cca 200 concepts (incl. OWL classes, properties and individuals) and interlinks to three more vocabularies.

6.2.3. *Interlinking registry of rights and obligations with electronic code of law*

Individual rights and obligations registered in the registry are linked to the respective legislation, i.e to respective legal act fragments. The electronic code of law is in its final stage of development and should be available at the beginning of 2022. Therefore, the links from the registry to the legislation are expressed as proprietary textual references and need to be changed to proper links to the electronic code of law.

The semantic vocabulary of the registry of rights and obligations covers also these proprietary references. As it was created before the model for the electronic code of law it contains its own simple vocabulary of legal acts and their fragments. As a part of the preparations to start the electronic code of law at the beginning of 2022, it is necessary to replace proprietary references in the registry of rights and obligations to fragments of legal acts with proper links to the electronic code of law. Also the model of the registry and technical artefacts described in Section 6.2.1 need to be updated appropriately.

AL makes the changes at the semantic layer easy. To change the model of the registry it is possible to reuse the model of legal acts and their fragments from the model of the electronic code of law. This is an easy replacement operation which can be performed without any technical knowledge in AL. We can then automatically propagate this change to the documentation and JSON schemas of the data sets published from the registry.

7. Related work

Assembly Line is both a web-based platform facilitating collaboration process and a tool for modeling domain-specific ontologies and vocabularies. As a consequence it can be compared to quite large number of tools, depending on a particular use case. Instead of providing a comprehensive survey of all the tools, we decided to focus on those most impactful and recognized systems produced within academic and scientific community in recent years.

To provide a relevant comparison with similar projects, we employed the following selection criteria:

– The project must be presented in form of a paper describing requirements and architecture.
– The paper must be recognized within academic community by being cited in related papers.
– The project must not be deprecated.
– The project must be either open-source or must provide a runnable demo-version so that it can be evaluated.

One of the most well-known vocabulary editors is WebProtégé [17], an open-source Web application for editing OWL ontologies. The project started as a light-weight web-based alternative to Protégé desktop editor, but has grown into a complex standalone product over the years. The application consists of a default simple editing interface,

Table 3
Overview of features of related tools

Feature	Assembly Line	WebProtégé	VocBench	VoCol	OnToology
Vocabulary modeling tools	✔	✻	✻		
Collaborative workflow support	✔	✔	✔	✻	✻
Publishing of vocabularies	✔		✻	✔	✔
Automated syntax and constraints checking	✔		✔	✻	✔
Versioning of vocabularies	✻	✻	✻	✻	✻
Decentralized vocabulary governance	✻				
✔ supported					
✻ partially supported					

which provides access to commonly used OWL constructs. It is accompanied by several features to aid collaboration, including support for threaded discussions of issues and full change tracking and revision history. WebProtégé does not include features facilitating publication of created ontologies.

VocBench [18] is a web-based, multilingual, collaborative development platform for managing OWL ontologies, SKOS thesauri, Ontolex-lemon lexicons and generic RDF datasets. It includes features for publishing and validating vocabularies. VocBench does not provide advanced visual modeling tools and although its ultimate goal is to provide a general modeling solution, it focuses mostly on SKOS thesauri editing.

Similar to Assembly Line, VoCol [19] and OnToology [20] platforms base their collaboration workflow on top of a Git repository. Both of these platforms integrate various tools to facilitate particular features like syntax validation, visualization or documentation generation. However, neither of the two provide user-friendly editing capabilities, so a deep ontology and Linked Data knowledge is required to use the two respective tools.

Table 3 provides a comparison of the aforementioned tools within the context of Assembly Line requirements. Features that are fully supported within the respective tools are marked with " ✔ " symbol, and those that are partially supported are marked with " ✻ ".

The table shows that none of the existing tools satisfy all the requirements outlined in this paper.

Vocabulary modeling tools is fully supported only by the Assembly Line. Both WebProtégé and VocBench offer user-friendly form-based UI for managing ontologies, which is useful for simple thesauri, but not convenient for complex models. The tools provide components for ontology visualization, but in both cases it is read only and cannot be configured. VoCol and OnToology do not provide any modeling tools at all, as users are required to design the models in external editors.

Collaborative workflow support is included in the Assembly Line, WebProtégé and VocBench tools. All of these platforms provide user-friendly and role-based collaboration utilities, allowing for some form of concurrent editing of vocabularies. VoCol and OnToology support this feature through integration with a git or GitHub repository respectively. Collaboration in VoCol is realized by pushing commits to a main branch, whereas OnToology employs pull request based approach specific to GitHub platform.

Publishing of vocabularies is fully supported within the Assembly Line, VoCol and OnToology. All of these platforms employ a continuous integration and continuous deployment ontology development, which means that with every vocabulary increment the respective vocabulary is tested and published within the designed pipeline. Besides publishing vocabularies to a triple store or as a downloadable file containing RDF serialization of the vocabularies, the platforms are also able to generate additional artifacts like graph visualization or documentation. VocBench supports on-demand export to file or publishing to a configurable triple store; in addition it supports programming a custom deployer. WebProtégé only allows users for saving an ontology as a file.

Automated syntax and constraints checking is provided by the Assembly Line, OnToology and VocBench. Besides basic syntax checking of created vocabularies, these tools allows users to define advanced ontology constraints using SHACL shape validation. According to the documentation, VoCol supports syntax validation and some form of constraints checking as well using a custom component. WebProtégé does not include an explicit ontology validation beyond standard form error handling within its UI interface.

Neither of the researched tools fully support stable *versioning of vocabularies*. VoCol and OnToology realize the versioning using the git-based approach, while VocBench and WebProtégé let users create a project snapshot, that however cannot be easily published as a separate version that would eventually exist alongside other versions. To create a truly versioned ontology, users need to assign version and publish the artifacts manually. The Assembly Line currently combines the approaches of the other tools, but the requirement as outlined in this paper is not completely satisfied yet.

Decentralized vocabulary governance is not supported by any of the aforementioned platforms. This requirement is partially satisfied by the Assembly Line by having multiple instances set up, having all of these instances based on a common set of fundamental vocabularies. The platform is still missing a feature to publish vocabularies from all the instances to a common data set catalog.

8. Conclusion and Future Work

Assembly Line is a platform that provides means to create, manage and publish SGoV, which should eventually serve as a semantic backbone of the Czech eGovernment. While still in the proof-of-concept phase, the tool is being gradually adopted by various Czech public offices that need common semantic framework for data publishing and sharing. The user base is growing steadily, and so does the actual content of SGoV. As indicated, Assembly Line provides core feature for vocabularies management and publishing, but it does not yet satisfy all the requirements outlined in previous sections. The following aspects of Assembly Line will be improved further in the next months.

8.1. Decentralized data governance

In this paper we presented the Assembly Line platform, or rather a single instance of it. The outlined solution is limiting because it assumes that the whole SGoV resides in a single GitHub repository, which will likely be governed by single agent, which may be a bottleneck. To satisfy the requirement of decentralized data governance, the platform will need to be transformed to a multi-instance architecture of self-sufficient, but inter-connected instances of the Assembly Line. Our vision is that each owner, office or institution will run its own instance of the Assembly Line, governing a subset of SGoV vocabularies tied to the owner's agenda.

The key requirement is to ensure the data integrity of SGoV, to facilitate that there will need to be one authority instance that will contain core vocabularies which are the most common dependencies shared by all vocabularies within SGoV. But apart from that all the instances of Assembly Line should be as independent as possible.

We considered developing a multi-tenant single instance platform instead, but that architecture does not guarantee sufficient independence of governance for our use case.

8.2. Versioning

One of the most difficult problems we will need to solve is versioning of SGoV and its sub-vocabularies. As mentioned in the requirements section, some vocabularies are based on legal acts, and when the law changes, the related vocabularies will likely need to change as well. We cannot just change the vocabulary without impacting the already published data, since the vocabularies will be used to annotate open data though. Therefore we will need to introduce some form of versioning strategy.

The method that seems promising is the versionless approach, recently pioneered by GraphQL. GraphQL takes a firm stance on avoiding versioning by providing the tools for the continuous evolution of a GraphQL schema. In the context of SGoV, it means that we will need to make sure not to introduce breaking changes when updating a vocabulary, which can be fortunately achieved quite easily. As all the terms and links within vocabularies are identified by IRI, we just need to make sure not to delete any IRIs, but only add new information to vocabularies. However, this does not prevent changes like adjusting labels of resources or marking some resources as deprecated.

Nevertheless, we will need to do further research in this area to decide what is the best strategy.

8.3. Universal Assembly Line for various domains

The SGoV is based on formal languages and core ontologies based on legal theory and real legal acts. However, those building blocks can be adjusted and the whole Assembly Line generalized by introducing different sets of core ontologies, while maintaining the same principles of governing vocabularies. As a result, the Assembly Line may be reused in a distinct domain, fulfilling the role of universal tool for creating and maintaining sets of related conceptual models.

References

[1] A.P. Sheth, *Changing Focus on Interoperability in Information Systems:From System, Syntax, Structure to Semantics*, in: *Interoperating Geographic Information Systems*, M. Goodchild, M. Egenhofer, R. Fegeas and C. Kottman, eds, Springer US, Boston, MA, 1999, pp. 5–29. ISBN 978-1-4615-5189-8. doi:10.1007/978-1-4615-5189-8$_2$.

[2] M.N. Mami, D. Graux, S. Scerri, H. Jabeen, S. Auer and J. Lehmann, Squerall: Virtual Ontology-Based Access to Heterogeneous and Large Data Sources, in: *The Semantic Web – ISWC 2019*, C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois and F. Gandon, eds, Springer International Publishing, Cham, 2019, pp. 229–245. ISBN 978-3-030-30796-7.

[3] S. Schmid, C. Henson and T. Tran, Using Knowledge Graphs to Search an Enterprise Data Lake, in: *The Semantic Web: ESWC 2019 Satellite Events*, P. Hitzler, S. Kirrane, O. Hartig, V. de Boer, M.-E. Vidal, M. Maleshkova, S. Schlobach, K. Hammar, N. Lasierra, S. Stadtmüller, K. Hose and R. Verborgh, eds, Springer International Publishing, Cham, 2019, pp. 262–266. ISBN 978-3-030-32327-1.

[4] M. Stonebraker and I.F. Ilyas, Data Integration: The Current Status and the Way Forward., *IEEE Data Eng. Bull.* **41**(2) (2018), 3–9.

[5] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini and R. Rosati, Linking Data to Ontologies, in: *Journal on Data Semantics X*, S. Spaccapietra, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 133–173. ISBN 978-3-540-77688-8.

[6] G. Xiao, L. Ding, B. Cogrel and D. Calvanese, Virtual Knowledge Graphs: An Overview of Systems and Use Cases, *Data Intell.* **1**(3) (2019), 201–223. doi:10.1162/dint_a_00011.

[7] O. Corcho, F. Priyatna and D. Chaves-Fraga, Towards a new generation of ontology based data access, *Semantic Web* **11**(1) (2020), 153–160.

[8] P. Křemen and M. Nečaský, Improving discoverability of open government data with rich metadata descriptions using semantic government vocabulary, *J. Web Semant.* **55** (2019), 1–20. doi:10.1016/j.websem.2018.12.009.

[9] A. Miles and S. Bechhofer, SKOS Simple Knowledge Organization System Reference, W3C Recommendation, W3C, 2009. https://www.w3.org/TR/2009/REC-skos-reference-20090818/.

[10] B. Motik, B. Parsia and P.F. Patel-Schneider, OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax, W3C Recommendation, W3C, 2009. http://www.w3.org/TR/2009/REC-owl2-syntax-20091027.

[11] G. Guizzardi, C.M. Fonseca, A.B. Benevides, J.P.A. Almeida, D. Porello and T.P. Sales, Endurant Types in Ontology-Driven Conceptual Modeling: Towards OntoUML 2.0, in: *Conceptual Modeling - 37th International Conference, ER 2018, Xi'an, China, October 22-25, 2018, Proceedings*, J. Trujillo, K.C. Davis, X. Du, Z. Li, T.W. Ling, G. Li and M. Lee, eds, Lecture Notes in Computer Science, Vol. 11157, Springer, 2018, pp. 136–150. doi:10.1007/978-3-030-00847-5_12.

[12] G. Carothers and E. Prud'hommeaux, RDF 1.1 Turtle, W3C Recommendation, W3C, 2014, https://www.w3.org/TR/2014/REC-turtle-20140225/.

[13] J. Bourguet, G. Guizzardi, A.B. Benevides and V. Zamborlini, Empirically Evaluating Three Proposals for Representing Changes in OWL2, in: *Proceedings of the Joint Ontology Workshops 2017 Episode 3: The Tyrolean Autumn of Ontology, Bozen-Bolzano, Italy, September 21-23, 2017*, S. Borgo, O. Kutz, F. Loebe, F. Neuhaus, K. Adrian, M. Antovic, V. Basile, M. Boeker, D. Calvanese, T. Caselli, G. Colombo, R. Confalonieri, L. Daniele, J. Euzenat, A. Galton, D. Gromann, M.M. Hedblom, H. Herre, I. Hinterwaldner, A. Janes, L. Jansen, K. Krois, A. Lieto, C. Masolo, R. Peñaloza, D. Porello, D.P. Radicioni, E.M. Sanfilippo, D. Schober, R. Stufano and A. Vizedom, eds, CEUR Workshop Proceedings, Vol. 2050, CEUR-WS.org, 2017. http://ceur-ws.org/Vol-2050/DEW_paper_4.pdf.

[14] J.R. Lewis, The System Usability Scale: Past, Present, and Future, *International Journal of Human–Computer Interaction* **34**(7) (2018), 577–590. doi:10.1080/10447318.2018.1455307.

[15] A. Bangor, P. Kortum and J. Miller, Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale, *J. Usability Studies* **4**(3) (2009), 114–123–.

[16] T. Francart, J. Dann, R. Pappalardo, C. Malagon and M. Pellegrino, The European Legislation Identifier, in: *Knowledge of the Law in the Big Data Age, Conference 'Law via the Internet 2018', Florence, Italy, 11-12 October 2018*, G. Peruginelli and S. Faro, eds, Frontiers in Artificial Intelligence and Applications, Vol. 317, IOS Press, 2018, pp. 137–148. doi:10.3233/FAIA190016.

[17] M. Horridge, R.S. Gonçalves, C.I. Nyulas, T. Tudorache and M.A. Musen, WebProtégé: A Cloud-Based Ontology Editor, in: *Companion of The 2019 World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, S. Amer-Yahia, M. Mahdian, A. Goel, G. Houben, K. Lerman, J.J. McAuley, R. Baeza-Yates and L. Zia, eds, ACM, 2019, pp. 686–689. doi:10.1145/3308560.3317707.

[18] A. Stellato, M. Fiorelli, A. Turbati, T. Lorenzetti, W.V. Gemert, D. Dechandon, C. Laaboudi-Spoiden, A. Gerencsér, A. Waniart, E. Costetchi and J. Keizer, VocBench 3: A collaborative Semantic Web editor for ontologies, thesauri and lexicons, *Semantic Web* **11**(5) (2020), 855–881. doi:10.3233/SW-200370.

[19] L. Halilaj, N. Petersen, I. Grangel-González, C. Lange, S. Auer, G. Coskun and S. Lohmann, VoCol: An Integrated Environment to Support Version-Controlled Vocabulary Development, in: *Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings*, E. Blomqvist, P. Ciancarini, F. Poggi and F. Vitali, eds, Lecture Notes in Computer Science, Vol. 10024, 2016, pp. 303–319. doi:10.1007/978-3-319-49004-5_20.

[20] A. Alobaid, D. Garijo, M. Poveda-Villalón, I. Santana-Pérez and Ó. Corcho, OnToology, a tool for collaborative development of ontologies, in: *Proceedings of the International Conference on Biomedical Ontology, ICBO 2015, Lisbon, Portugal, July 27-30, 2015*, F.M. Couto and J. Hastings, eds, CEUR Workshop Proceedings, Vol. 1515, CEUR-WS.org, 2015. http://ceur-ws.org/Vol-1515/demo3.pdf.

[21] C. Ghidini, M. Rospocher and L. Serafini, MoKi: a Wiki-based Conceptual Modeling Tool, in: *Proceedings of the EKAW2010 Poster and Demo Track, Lisbon, Portugal, October 11 - 15, 2010*, J. Völker and Ó. Corcho, eds, CEUR Workshop Proceedings, Vol. 674, CEUR-WS.org, 2010. http://ceur-ws.org/Vol-674/Paper56.pdf.

[22] S. Arts, gist upper level ontology, 2021. https://www.semanticarts.com/gist/.

[23] P.-Y. Vandenbussche, G.A. Atemezing, M. Poveda-Villalón and B. Vatant, Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web, *Semantic Web* **8**(3) (2017), 437–452.