



# ISWC 2019 RESOURCES TRACK: quality attribute checklists for specific resource types

*Adapted by Maria Maleshkova and Vojtech Svatek from the 2018 version of Resources Track instructions made by Valentina Presutti and Mari Carmen Suárez de Figueroa, which had been, in turn, adapted from the 2017 version by Freddy Lecue and Valentina Tamma*



## Focus of the Resources Track

Papers describing **resources**: high-quality information artifacts that are reusable in novel contexts. They include, but are not limited to:

- Datasets
- Ontologies and vocabularies
- Ontology design patterns
- Workflows
- Evaluation benchmarks or methods
- Services and APIs
- Software frameworks
- Crowdsourcing task designs
- Methodologies
- Protocols and metrics



## Quality criteria of resources

When considering the general quality criteria valid for all kinds of resources (impact, reusability, technical quality and availability), both **authors** and **reviewers** may benefit from **checklists** of typical quality attributes of the different types of resources, by the experience from the previous editions of the Resources Track. The following slides present such **fine-tuned checklists** for one resource type at a time (only for such resources there has been sufficient experience with; *methodologies, workflows, protocols* and *metrics* are not covered)



# Ontologies and vocabularies

- Methodological soundness
- Clarity of the domain and requirements being addressed by the ontology or the vocabulary
- Clarity of modelling problems encountered
- Soundness of modelling choices and motivations including validation of SPARQL queries over possible evaluation scenarios
- High quality design: e.g. no hacks and workarounds, no redundancy
- Logical correctness: e.g. logical consistency, correct use of the modelling language primitives
- Meaningful and motivated reuse of other resources
- Reuse of ontology design patterns
- Validation in a real use case
- Quality of the resource documentation: rich annotations accompany and are included in the resource e.g. competency questions, rdfs:comment, reports, guidelines.



# Ontology Design Patterns (ODP)

- Methodological soundness
- Clarity of the requirements being addressed by the ODP
- The ODP is sufficiently general to be interesting for reuse (better: there is evidence of reuse in a number of independent ontologies)
- Soundness of modelling choices and motivations, including: validation of SPARQL queries over possible evaluation scenarios and axioms for supporting interoperability
- Limits and advantages of the ODP are clearly explained
- High quality design: e.g. no hacks and workarounds, no redundancy
- Logical correctness: e.g. logical consistency, correct use of the modelling language primitives
- Reuse of other ontology design patterns, if applicable (e.g. specialisation)
- Alignment to existing, relevantly related and widely used ontologies, if applicable. Or sound and convincing comparison with them
- Quality of the ODP documentation: rich annotations accompany and are included in the ODP e.g. specific ODP annotations, examples of reuse, competency questions.



# Synthetic Datasets

- The dataset is easy to access and query
- The model used to represent the data is clear
- The methodology to produce the data is sound
- The data generator is scalable
- The data capture important characteristics of the equivalent real-world data



# Annotated Datasets

- The dataset is easy to access and query
- The model used to represent the data is clear
- The assumptions behind the annotations are sufficiently described
- The methodology used for producing the annotations is sound



## Other Datasets

- The dataset is easy to access and query
- The ontology/vocabulary used to represent the data is clear
- The dataset provide a significant coverage of the domain it targets and it can be meaningfully used for real world applications and/or for supporting scientific experiments
- The methodology to produce the data is sound



# Software Frameworks

- Complexity of the implemented functionalities: the framework allows others to save significant coding effort
- The chosen abstractions are useful and likely to generalize to other problems
- The framework differs from existing ones that cover similar requirements and the difference addresses relevant requirements
- Quality and performance of the tool/system. Papers should include a clear evaluation of the performance of the tool/system according to relevant measures such as speed, usability, efficiency, etc.
- Community: e.g. active mailing list, issue trackers, can be (or better is being) used by others



# Services and APIs

- The functionality of the service is clear and important features of the service are published
- The service/API differs from existing ones that cover similar requirements and the difference addresses relevant requirements
- Relevant metrics about the service are provided, e.g. uptime of the service, service levels
- The service is well documented to enable use, e.g. availability of tutorials, code snippets.
- The API is documented in a machine processable way



# Benchmarks

- The benchmark measures something significant, it is relevant and sufficiently general
- The proposed performance metrics are sufficiently broad and relevant
- The tasks are well motivated in terms of testing the system or mimicking real-world scenarios
- The scale of the dataset is appropriate and it be scaled on appropriate metrics
- It differs significantly from existing benchmarks developed for similar purposes and the difference addresses a relevant shortcoming
- Others can use the data and software of the benchmark
- The benchmark has been run on at least three different systems (not variants of the same system)
- The coverage of systems is reasonable and a suitable baseline has been provided
- Sufficient experimental details are provided to enable interpretation of the results and replication of the experiments (e.g. software version numbers, hardware details)
- Good experimental protocols have been followed (e.g. warm-up periods, multiple runs, standard errors reported)
- The results are discussed and explained sufficiently



# Crowdsourced tasks and designs

- The crowdsourced task is clearly described and sound
- If the task is composed of more than one task, the workflow (sequence of tasks involved) is described and related designs and code are provided
- The task(s) template(s) design is clearly explained and the code available for reuse
- The setting for the crowdsource platform is provided: number of workers, restrictions, etc.
- Generality of the template to be applied to different data
- The template is easy to adapt to other platforms or data formats
- Sample of input data and result data is provided or alternatively, if data are not shareable, comprehensive examples and explanations
- Limits or potential weak points are pointed out