Resource Inventory Model

Daniel Schopper

# Resource Inventory Model

## Version 0.2

|  |  |
| --- | --- |
| Author(s): | Daniel Schopper |
| Version: | 0.2 |
| Last Changed: | 2022-04-24 |
| Documentation generated: | 2022-05-17T23:22:17.96711Z |

## Graphical Overview

![](data:image/svg+xml;base64;base64,)

Graphical Overview of the data model.  
Dashed lines mean implicit shortcut relations.

## Description

### Purpose of a Resource Catalogue

In our day-to-day work in projects, we are creating or re-using a large variety of digital resources. In order to keep them organized, to ensure that we are able to find them when needed, and to keep track of the work and responsibilities on them, we need a Resource Catalogue: This catalogue should represent all key characteristics of a digital resource to help us understand how it is structured, who is/was working on it with which tools, following which workflow and so on. As such, the entries should be the main administrative handle for representing a resource in related areas like project administration, archiving and infrastructure provisioning / server administration.

We envision resource catalogues to be relevant on three levels:

* On the basic level, each research project at ACDH-CH should have its own Project Resource Catalogue as part of its DMP. This catalogue should list all resources which are planned or actually created or re-used during the project, plus the processes and workflows which are needed to generate/curate/enrich … them. Being a structured side-kick of the project’s DMP, both the principle investigator and the project’s Data Officer are the main responsibles for ensuring the catalogue's completeness and accurateness.
* On a higher level, we aim to establish the ACDH-CH Resource Catalogue which aggregates all Project Resource Catalogues into a common space. Not only do we envision this to become a management tool which should help us improve data governance and resource planning, but we also hope that such a tool will foster internal collaboration and stimulate re-use of existing datasets, models and workflows by making similar formats and workflows more visible. Ideally, the institute’s catalogue should feed a future ACDH-CH Dashboard
* Finally, ideally, such a Resource Catalogue would be adopted universally within the Academy. and correspondingly also feed the Academy Knowledge Portal.

This relates to several other components or activities on the institutional/academy level

* The Data Inventory Survey (Datenbestandserhebung) has shown the need for structured digital asset management across the humanities institutes of the Academy. The results of Datenbestandserhebung will feed into the data catalogue specification and will be a first use case for it.
* Academy Knowledge Portal is a planned integrative/ed discovery solution for the research outputs of the Academy, for which a comprehensive overview of existing datasets on the institutional level is a necessary precondition
* AkademIS – the CRIS of the Academy. [To be explored: What is the current practice and coverage with respect to describing datasets as research outputs and what are the possibilities for integration and reuse.]

### Stakeholders and interests

* Data Curators and developers in a project: ease communication about / planning /exchange of data
* TF Preservation: streamline deposition process by collecting crucial aspects as structured information early on in a project – ideally source final ARCHE metadata from the Data Catalogue; help planning of human and technical resources needed for archiving (when will project x finish, how much material will have to be archived, which formats etc.)
* Technical Supervisors: make sure data and processes are properly documented; overview of formats / volumes; support infrastructure provisioning (server space)
* PI + ACDH-CH Project Management: structured planning of resources; keep track of the deliverables in a project / progress on the ways towards them; ensure that processes are properly documented
* DSIE: ease communication in the team; share knowledge; re-use of tools and processes
* BOD, institutional Quality Assurance: reporting, overview of data produced at the institute – visibility of data to the institution

### Example questions to be answered by the catalogue

* Which datasets will be / are being / have been produced in a project?
* Where, i.e. on which storage location(s), can they be found?
* Which are IPR-sensitive datasets?
* Who is working on a dataset with which tools?
* Who can work with / access a dataset?
* Who is responsible for a given dataset?
* What is the status of a given dataset? Is it actively developed or used or has development/use stalled? Can it be discarded?
* Can I reuse the data of project y for my project ?
* I also have data in format F and need it to convert to format V – how was this solved in this project and can I re-use the script?
* What's the overall size of data in project Y? What is it expected to be at the end?
* How do the files in a github repository relate to the data I see in the public web application at … ?
* How do the files in a dataset relate to each other? Which one is temporary, which one is a working copy, which one is authoritative output?
* Is that actual data valid against the specifications set out by the model and the format? (e.g. AAC: each text must have a corresponding directory with page images; XML schema conformance; CIDOC conformity in Open Atlas)
* How much additional space is needed in the coming year, by all projects combined?
* Which project is using how much storage space (and what type)?

### Implementation notes

Obviously, the usefulness of such a catalogue will depend on the amount and accuratness of the information in it. A hybrid curation strategy seems advisable to ensure that:

* Collecting information on the "lowest level" in the model is rather straightforward: A script can count files and their properties (file type, dimension, ownership etc.) on a network drive or in a git repository as well as numbers on tables/rows in a relational database.

## Classes

### Class: "Resource"

**Superclass of:** [<SOFTWARE>](#Software), [<DATA SET>](#Dataset), [<PUBLICATION>](#Publication)

#### Definition

The abstract notion of any kind of digital resource managed in the Resource Catalogue. This is an abstract helper class serving as the parent class for [<DATA SET>](#Dataset), [<SOFTWARE>](#Software) and [<PUBLICATION>](#Publication). meaning that only subclasses of it should be instantiated in the actual data.

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Access Restrictions** Cardinality: 1 | [Vocab Term](#term) in [Access Restrictions Vocabulary](#vocabs.accessRestrictions) | This property is global to all "incarnations" of a resource. |
| **Description** Cardinality: 1 | [Long Text](#longText) |  |
| **License** Cardinality: 1 | [Vocab Term](#term) in [ARCHE Licenses Vocabulary](#vocabs.licenses) | This property is the global default to all "incarnations" associated with that resource. In case a resource instance is licensed under different conditions, then it should have its own [property](#resourceInstance.license). |
| **Name** Cardinality: 1 | [Short Text](#shortText) |  |
| **PID** Cardinality: 0-n | [URI](#uri) | Any kind of persistent identifier |
| **Status** Cardinality: 1 | [Vocab Term](#term) in [ARCHE Lifecycle Status Vocabulary](#vocabs.resourceStatus) |  |
| **Subject** Cardinality: 0-n | [Vocab Term](#term) in [ARCHE Subjects Vocabulary](#vocabs.archeSubjects) |  |

#### Mapping to Open AIRE Research Graph Data Model 1.3

Version: 1.3

<RESULT>

CHECKME OARG defines a class "Other research product": *The class of Results Other Research Products includes any research output that is not literature, data, or software. Examples include, but are not limited to: algorithms, scientific workflows/pipelines, protocols, standard operating procedure (SOP), simulations, mathematical and statistical models, but also research packages*. It might be worth considering such a fourth subclass as well.

who: DS when: 2022-04-22 status: new

#### Mapping to OpenAIRE LOD Ontology 1.1

<OAV:RESULTENTITY>

#### Relations (outgoing)

* [<Resource>](#Resource) [is described in (describes)](#resourceHasMetadata) [<Metadata>](#Metadata)
* [<Resource>](#Resource) [is related to with role (is related to with role)](#resourceIsRelatedToResource) [<Resource>](#Resource)
* [<Resource>](#Resource) [re-uses (is re-used by)](#resourceReusesResource) [<Resource>](#Resource)

#### Relations (incoming)

* [<Person>](#Person) [is associated with (has associated person)](#personIsAssociatedWithResource) [<Resource>](#Resource)
* [<Resource Instance>](#ResourceInstance) [is authoritative instantiation of (has authoritative instantiation)](#X8c2e750a86f1ccbf63bc373d7e71d797a0ee2bc) [<Resource>](#Resource)
* [<Person>](#Person) [is creator of (created by)](#personCreatorOfResource) [<Resource>](#Resource)
* [<Resource Instance>](#ResourceInstance) [is instantiation of (is instantiated in)](#Xc7890e2fc9c1cc3dec538623414131e955df85e) [<Resource>](#Resource)
* [<Actor>](#Actor) [is owner of (is owned by)](#actorIsOwnerOfResource) [<Resource>](#Resource)
* [<Resource>](#Resource) [is related to with role (is related to with role)](#resourceIsRelatedToResource) [<Resource>](#Resource)
* [<Project>](#Project) [produces (is output of)](#projectProducesResource) [<Resource>](#Resource)
* [<Project>](#Project) [re-uses (is re-used by)](#projectReusesResource) [<Resource>](#Resource)
* [<Resource>](#Resource) [re-uses (is re-used by)](#resourceReusesResource) [<Resource>](#Resource)

### Class: "Collection"

**Subclass of:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

#### Definition

A group of [<RESOURCE INSTANCE PARTS>](#ResourceInstancePart) (e.g. files) which are in some way related one to another. A collection may or may not be serialized as a directory in a file system.

#### Examples

A corpus is made up of 5 texts, each encoded as one TEI document and serialized to a file in a directory. Each document is the result of three subsequent processing steps (e.g. tokenization, POS-Tagging, lemmatization) which all place their output in the same directory, making it a total of 5 x 4 files (source file + 3 process output files per text). In order to address the files output by each step as a group, they can be modelled as collection being the result of a process: [<PROCESS>](#Process) [*produces*](#processProducesInstancePart) a [<COLLECTION OF RESOURCE PARTS>](#Collection); [<UNKNOWN\_CLASS>](#resourceRelatesToResource.role).

Example #1

#### Mapping to ACDH-CH ARCHE Ontology 3.1

Version: 3.1

<ARCHE:COLLECTION>

#### Relations (outgoing)

No outgoing relations defined.

#### Relations (incoming)

* [<Resource Instance Part>](#ResourceInstancePart) [is part of (consists of)](#resourceInstancePartIsPartOfCollection) [<Collection>](#Collection)

### Class: "Format"

#### Definition

a defnied format or standard

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Mime Type** Cardinality: 0-n | [Vocab Term](#term) in [MIME Type Vocabulary](#vocabs.mimetypes) |  |
| **Name** Cardinality: 1 | [Short Text](#shortText) | A short denomination for the Format in question. |
| **PRONOM Unique Identifier** Cardinality: 0-1 | [Short Text](#shortText) | The unique identifier in the [PRONOM database of file formats](http://www.nationalarchives.gov.uk/pronom/). |

#### Relations (outgoing)

No outgoing relations defined.

#### Relations (incoming)

* [<Software>](#Software) [consumes data of format (is input format of)](#softwareHasInputFormat) [<Format>](#Format)
* [<Resource Instance Part>](#ResourceInstancePart) [has format (is used in)](#fileHasFormat) [<Format>](#Format)
* [<Software>](#Software) [produces data of format (is output format of)](#softwareHasOutputFormat) [<Format>](#Format)

### Class: "Software"

**Subclass of:** [<RESOURCE>](#Resource)

#### Definition

A software implementing a [<PROCESS>](#Process).

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Access Restrictions** Cardinality: 1 | [Vocab Term](#term) in [Access Restrictions Vocabulary](#vocabs.accessRestrictions) | This property is global to all "incarnations" of a resource.  Inherited from [<RESOURCE>](#Resource) |
| **Description** Cardinality: 1 | [Long Text](#longText) | Inherited from [<RESOURCE>](#Resource) |
| **License** Cardinality: 1 | [Vocab Term](#term) in [ARCHE Licenses Vocabulary](#vocabs.licenses) | This property is the global default to all "incarnations" associated with that resource. In case a resource instance is licensed under different conditions, then it should have its own [property](#resourceInstance.license).  Inherited from [<RESOURCE>](#Resource) |
| **Name** Cardinality: 1 | [Short Text](#shortText) | Inherited from [<RESOURCE>](#Resource) |
| **PID** Cardinality: 0-n | [URI](#uri) | Any kind of persistent identifier  Inherited from [<RESOURCE>](#Resource) |
| **Status** Cardinality: 1 | [Vocab Term](#term) in [ARCHE Lifecycle Status Vocabulary](#vocabs.resourceStatus) | Inherited from [<RESOURCE>](#Resource) |
| **Subject** Cardinality: 0-n | [Vocab Term](#term) in [ARCHE Subjects Vocabulary](#vocabs.archeSubjects) | Inherited from [<RESOURCE>](#Resource) |
| **Type** Cardinality: 1-n | [Vocab Term](#term) in [Software Type Vocabulary](#vocabs.softwareType) |  |

#### Mapping to Open AIRE Research Graph Data Model 1.3

Version: 1.3

<OAV:SOFTWARE>

#### Relations (outgoing)

* [<Software>](#Software) [consumes data of format (is input format of)](#softwareHasInputFormat) [<Format>](#Format)
* [<Software>](#Software) [has Maintainer (is maintainer of)](#softwareHasMaintainer) [<Person>](#Person)
* [<Software>](#Software) [has dependency (is dependency of)](#softwareHasDependencySoftware) [<Software>](#Software)
* [<Software>](#Software) [implements (is implemented in)](#softwareImplementsProcess) [<Process>](#Process)
* [<Software>](#Software) [is documented in (documents)](#softwareIsDocumentedInPublication) [<Publication>](#Publication)
* [<Software>](#Software) [is successor of (has successor)](#softwareIsSuccessorOfsoftware) [<Software>](#Software)
* [<Software>](#Software) [produces data of format (is output format of)](#softwareHasOutputFormat) [<Format>](#Format)

#### Relations (incoming)

* [<Software>](#Software) [has dependency (is dependency of)](#softwareHasDependencySoftware) [<Software>](#Software)
* [<Software>](#Software) [is successor of (has successor)](#softwareIsSuccessorOfsoftware) [<Software>](#Software)
* [<Service>](#Service) [provides instance of (is instantiated in)](#serviceProvidesInstanceOfSoftware) [<Software>](#Software)

### Class: "Computer"

#### Definition

Any computer, be it a physical server, a virtual machine in the cloud or a local PC.

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **IP Address** Cardinality: 0-n | [Short Text](#shortText) |  |
| **Server Name** Cardinality: 1 | [Short Text](#shortText) |  |

#### Relations (outgoing)

No outgoing relations defined.

#### Relations (incoming)

* [<Service>](#Service) [is running on (hosts)](#serviceIsRunningOnServer) [<Computer>](#Computer)

### Class: "Service"

**Subclass of:** [<ACTOR>](#Actor)

#### Definition

A a piece of [<SOFTWARE>](#Software) running on a [<COMPUTER>](#Computer), usually a server accessible over the internet.

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Access Restrictions** Cardinality: 1 | [Vocab Term](#term) in [Access Restrictions Vocabulary](#vocabs.accessRestrictions) | Defines who should have access to a resource as a whole or a specific manifestation of it as raw data or inside an application. Thus, this property occurs on several levels in the model which should be interpreted as a cascading rule: |
| **End of life** Cardinality: 1 | [Date](#date) | The date where the service is expected to be outphased. |
| **Name** Cardinality: 0-n | [Short Text](#shortText) | Inherited from [<ACTOR>](#Actor) |
| **Service Status** Cardinality: 1 | [Vocab Term](#term) in [Service Status Vocabulary](#vocabs.serviceStatus) |  |
| **URI** Cardinality: 0-n | [URI](#uri) | One or more URIs under which the service is addressable. In case this is running on a Server, this should be mandatory. |

#### Relations (outgoing)

* [<Service>](#Service) [has responsible (is responsible for)](#serviceHasResponsiblePerson) [<Person>](#Person)
* [<Service>](#Service) [is running on (hosts)](#serviceIsRunningOnServer) [<Computer>](#Computer)
* [<Service>](#Service) [provides (is provided by)](#serviceProvidesInstanceOfResource) [<Resource Instance>](#ResourceInstance)
* [<Service>](#Service) [provides instance of (is instantiated in)](#serviceProvidesInstanceOfSoftware) [<Software>](#Software)

#### Relations (incoming)

No outgoing relations defined.

### Class: "Data Set"

**Subclass of:** [<RESOURCE>](#Resource)

#### Definition

A set of structured or semistructured data.

#### Mapping to Open AIRE Research Graph Data Model 1.3

Version: 1.3

<DATASETS>

#### Relations (outgoing)

* [<Data Set>](#Dataset) [is documented in (documents)](#DatasetIsDocumentedInPublication) [<Publication>](#Publication)
* [<Data Set>](#Dataset) [is successor of (has successor)](#datasetIsSuccessorOfdataset) [<Data Set>](#Dataset)

#### Relations (incoming)

* [<Data Set>](#Dataset) [is successor of (has successor)](#datasetIsSuccessorOfdataset) [<Data Set>](#Dataset)

### Class: "Publication"

**Subclass of:** [<RESOURCE>](#Resource)

#### Definition

Any kind of written material, including informal documentation, including best practice, learning resources etc.

#### Mapping to Open AIRE Research Graph Data Model 1.3

Version: 1.3

<LITERATURE>

#### Relations (outgoing)

* [<Publication>](#Publication) [is successor of (has successor)](#publicationIsSuccessorOfPublication) [<Publication>](#Publication)

#### Relations (incoming)

* [<Software>](#Software) [is documented in (documents)](#softwareIsDocumentedInPublication) [<Publication>](#Publication)
* [<Project>](#Project) [is documented in (documents)](#projectIsDocumentedInPublication) [<Publication>](#Publication)
* [<Process>](#Process) [is documented in (documents)](#processIsDocumentedInPublication) [<Publication>](#Publication)
* [<Data Set>](#Dataset) [is documented in (documents)](#DatasetIsDocumentedInPublication) [<Publication>](#Publication)
* [<Publication>](#Publication) [is successor of (has successor)](#publicationIsSuccessorOfPublication) [<Publication>](#Publication)

### Class: "Project"

#### Definition

A research or software project.

NOTE This be equivalent to the Project entity class in the ARCHE ontology.

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Name** Cardinality: 1 | [Short Text](#shortText) |  |
| **external ID** Cardinality: 0-n | [Short Text](#shortText) |  |

#### Mapping to ACDH-CH ARCHE Ontology 3.1

<ARCHE:PROJECT>

#### Relations (outgoing)

* [<Project>](#Project) [is documented in (documents)](#projectIsDocumentedInPublication) [<Publication>](#Publication)
* [<Project>](#Project) [produces (is output of)](#projectProducesResource) [<Resource>](#Resource)
* [<Project>](#Project) [re-uses (is re-used by)](#projectReusesResource) [<Resource>](#Resource)

#### Relations (incoming)

* [<Institution>](#Institution) [is funder of (is funded by)](#institutionFundsProject) [<Project>](#Project)
* [<Institution>](#Institution) [is hosting institution of (has hosting institution)](#institutionHostsProject) [<Project>](#Project)
* [<Actor>](#Actor) [is member with role of (has member with role)](#actorIsMemberOfProject) [<Project>](#Project)

### Class: "Actor"

**Superclass of:** [<SERVICE>](#Service), [<PERSON>](#Person), [<INSTITUTION>](#Institution)

#### Definition

Abstract superclass of "Person" or "Institution"

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Name** Cardinality: 0-n | [Short Text](#shortText) |  |

#### Mapping to OpenAIRE LOD Ontology 1.1

CHECKME No explicit Actor class?

who: DS when: 2022-04-25 status: new

#### Relations (outgoing)

* [<Actor>](#Actor) [executes (executed by)](#actorExecutesProcess) [<Process>](#Process)
* [<Actor>](#Actor) [is member with role of (has member with role)](#actorIsMemberOfProject) [<Project>](#Project)
* [<Actor>](#Actor) [is owner of (is owned by)](#actorIsOwnerOfResource) [<Resource>](#Resource)

#### Relations (incoming)

* [<Institution>](#Institution) [has member (is member of)](#institutionHasMemberActor) [<Actor>](#Actor)

### Class: "Person"

**Subclass of:** [<ACTOR>](#Actor)

#### Definition

A natural person

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Name** Cardinality: 0-n | [Short Text](#shortText) | Inherited from [<ACTOR>](#Actor) |
| **ORCID** Cardinality: 1 | [URI](#uri) | A URI from the https://orcid.org/ namespace |

#### Mapping to Open AIRE Research Graph Data Model 1.3

Version: 1.3

<OAV:PERSON>

NOTE The model in version 1.3 "Person" is a so-called "Structured Type"; a Core entity "Person" is listed under "future extensions to the data model"

#### Relations (outgoing)

* [<Person>](#Person) [is associated with (has associated person)](#personIsAssociatedWithResource) [<Resource>](#Resource)
* [<Person>](#Person) [is creator of (created by)](#personCreatorOfResource) [<Resource>](#Resource)

#### Relations (incoming)

* [<Software>](#Software) [has Maintainer (is maintainer of)](#softwareHasMaintainer) [<Person>](#Person)
* [<Service>](#Service) [has responsible (is responsible for)](#serviceHasResponsiblePerson) [<Person>](#Person)

### Class: "Metadata"

**Subclass of:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

#### Definition

An external metadata record of the resource in question.

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Location** Cardinality: 0-n | [URI](#uri) |  |
| **Metadata name** Cardinality: 1 | [Short Text](#shortText) |  |
| **extent** Cardinality: 0-n | [Dimension](#dimension) | Inherited from [<RESOURCE INSTANCE PART>](#ResourceInstancePart) |
| **location** Cardinality: 0-1 | [URI](#uri) | the path to the resource part (e.g. file system path); see comment on [accessPoint](#resourceInstance.accessPoint) property of [<RESOURCE INSTANCE>](#ResourceInstance).  Inherited from [<RESOURCE INSTANCE PART>](#ResourceInstancePart) |

#### Relations (outgoing)

No outgoing relations defined.

#### Relations (incoming)

* [<Resource>](#Resource) [is described in (describes)](#resourceHasMetadata) [<Metadata>](#Metadata)

### Class: "Institution"

**Subclass of:** [<ACTOR>](#Actor)

#### Definition

An institution, corporate body or organization of any size or legal type.

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Name** Cardinality: 0-n | [Short Text](#shortText) | Inherited from [<ACTOR>](#Actor) |
| **ROR ID** Cardinality: 1 | [URI](#uri) | A URI from the https://ror.org/ namespace |

#### Mapping to Open AIRE Research Graph Data Model 1.3

Version: 1.3

<ORGANIZATION>

NOTE The OpenAIRE model also proposes a <FUNDER> class. Since an institution like ÖAW acts both as a funder and as a host it seems more advisable to keep the class generic and attach this kind of information to a property of each respective project.

NOTE In the OpenAIRE Research Graph Data Model this class does not have a superclass.

#### Relations (outgoing)

* [<Institution>](#Institution) [has member (is member of)](#institutionHasMemberActor) [<Actor>](#Actor)
* [<Institution>](#Institution) [is funder of (is funded by)](#institutionFundsProject) [<Project>](#Project)
* [<Institution>](#Institution) [is hosting institution of (has hosting institution)](#institutionHostsProject) [<Project>](#Project)
* [<Institution>](#Institution) [is part of (has part)](#institutionIsPartOfInstitution) [<Institution>](#Institution)

#### Relations (incoming)

* [<Institution>](#Institution) [is part of (has part)](#institutionIsPartOfInstitution) [<Institution>](#Institution)

### Class: "Resource Instance"

#### Definition

A container object describing a materialized [<RESOURCE>](#Resource) which is made of one or several [<RESOURCE INSTANCE PARTS>](#ResourceInstancePart) with actual data, representing the whole resource or only a fragment of it. In case of a publication resource, this might be one document file with embedded data (e.g. as PDF, EPUB) or a set of files making up a complete HTML website. in case of [<SOFTWARE>](#Software) this is a copy of the application code e.g. in a version control repository. In the case of a [<DATA SET>](#Dataset) resource this might be one or several files on a network drive or one table in a RDBMS.

NOTE

A resourceInstance is

In case a resource instance is stored as flat files use [<RESOURCE>](#Resource) [*is made up of (is part of)*](#instanceIsMadeUpOfResourceInstanceParts) <RESOURCE INSTANCE PARTS>.

In case a use [<RESOURCE>](#Resource) [*is provided by (provides)*](#serviceProvidesInstanceOfResource) [<SERVICE>](#Service)

If it is important to document the actual storage location of a resource instance even if it is accessed via a [<SERVICE>](#Service), use both. – E.g. in case of a http server, files are uploaded via SFTP to the local hard disk and served by the service ot the public.

If it is needed to express the inner logics of a [<RESOURCE INSTANCE>](#ResourceInstance), use [<COLLECTIONS>](#Collection) which group together files into a logical structure (irrespective of the way / location they are stored).

CHECKME How about introducing an entity class "AccessPoint" or the like which allows to provide some explicit context to the different URLs etc via which a resource can be accessed? (e.g. internally used url, upload via SFTP etc.)

who: DS when: 2022-04-23 status: new

#### Examples

These would count for three distinct resource instances of the *issue number 124 of the Wiener Zeitung from June 2nd, 1909* (a [<DATA SET>](#Dataset)):

* the [*representative*](#X8c2e750a86f1ccbf63bc373d7e71d797a0ee2bc) resource instance [*is made up of*](#instanceIsMadeUpOfResourceInstanceParts) *451 TIFF files* (a [<RESOURCE INSTANCE PART>](#ResourceInstancePart)) serving as the *master images of a scanned manuscript* (a [<COLLECTION>](#Collection)) and one *TEI document* (a [<RESOURCE INSTANCE PART>](#ResourceInstancePart)) containing the full-text transcription file stored on a network share. Since the TEI document references the image files, their functionality to the user dependents on their presence – thus both are parts of the same resource instance.
* The *JPEG2000 versions* (a [<RESOURCE INSTANCE>](#ResourceInstance)) of *the 451 images* (a [<RESOURCE INSTANCE PART>](#ResourceInstancePart)) residing on the *local hard disk drive* (a [<INFORMATION CARRIER>](#Carrier)) of a *IIIF server* (a [<SERVICE>](#Service)).
* A *copy of the TEI transriptions indexed as document nodes* (a [<RESOURCE INSTANCE PART>](#ResourceInstancePart)) in an eXistdb instance which are dynamically converted into IIIF manifests by XQuery script *tei2iif.xquery* (a [<SOFTWARE>](#Software)).

Example #1

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Completness** Cardinality: 1 | [Vocab Term](#term) in [Resource Instance completness Vocabulary](#vocabs.resourceInstanceCompleteness) |  |
| **Resource Instance Access Point** Cardinality: 1-n | [URI](#uri) | One or several URIs where a user can start accessing the actual data of the resource instance. In case of a resource instance organized in flat files, this might be a file:-scheme URI pointing to an index document or the directory containing the relevant files; in the case the resource instance is stored in some kind of database management system (RDBMS etc,) providing some kind of logical access path on top of the actual data storage layer, this might be a [custom URI scheme](https://github.com/libwww-perl/uri-db) pointing to a database or a query inside of it. |
| **Resource Instance License** Cardinality: 0-1 | [Vocab Term](#term) in [ARCHE Licenses Vocabulary](#vocabs.licenses) | This is overriding the parent's resource [license property](#resourceInstance.license). |
| **Type** Cardinality: 1 | [Vocab Term](#term) in [Instance Roles Vocabulary](#vocabs.instanceRoles) |  |

#### Mapping to ACDH-CH ARCHE Ontology 3.1

Version: 3.1

<ARCHE:COLLETION>

#### Relations (outgoing)

* [<Resource Instance>](#ResourceInstance) [is authoritative instantiation of (has authoritative instantiation)](#X8c2e750a86f1ccbf63bc373d7e71d797a0ee2bc) [<Resource>](#Resource)
* [<Resource Instance>](#ResourceInstance) [is instantiation of (is instantiated in)](#Xc7890e2fc9c1cc3dec538623414131e955df85e) [<Resource>](#Resource)
* [<Resource Instance>](#ResourceInstance) [is made up of (is part of)](#instanceIsMadeUpOfResourceInstanceParts) [<Resource Instance Part>](#ResourceInstancePart)

#### Relations (incoming)

* [<Service>](#Service) [provides (is provided by)](#serviceProvidesInstanceOfResource) [<Resource Instance>](#ResourceInstance)

### Class: "Information Carrier"

#### Definition

A physical storage (or some abstraction of it) transmitting as the medium of making a conceptual object.

#### Examples

The hard disk drive in my computer.

Example #1

A medieval codex

Example #2

A CD-ROM on a book shelf

Example #3

A network share

Example #4

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Carrier Type** Cardinality: 1 | [Vocab Term](#term) in [Carrier Types Vocabulary](#vocabs.carrierTypes) |  |
| **Carrier name** Cardinality: 1 | [Short Text](#shortText) |  |

#### Relations (outgoing)

No outgoing relations defined.

#### Relations (incoming)

* [<Resource Instance Part>](#ResourceInstancePart) [has carrier (carries)](#fileHasCarrier) [<Information Carrier>](#Carrier)

### Class: "Resource Instance Part"

**Superclass of:** [<COLLECTION>](#Collection), [<METADATA>](#Metadata)

#### Definition

Some binary or textual data which is treated as an atomic unit being a building block of a resource instance. Depending on the level of abstraction of the underlying architecture this can be:

* a file
* a table in a relational database (which in itself is "made up" by rows and fields but )
* a document node in case of a XML or JSON document stored in a NoSQL database

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **extent** Cardinality: 0-n | [Dimension](#dimension) |  |
| **location** Cardinality: 0-1 | [URI](#uri) | the path to the resource part (e.g. file system path); see comment on [accessPoint](#resourceInstance.accessPoint) property of [<RESOURCE INSTANCE>](#ResourceInstance). |

#### Relations (outgoing)

* [<Resource Instance Part>](#ResourceInstancePart) [has carrier (carries)](#fileHasCarrier) [<Information Carrier>](#Carrier)
* [<Resource Instance Part>](#ResourceInstancePart) [has format (is used in)](#fileHasFormat) [<Format>](#Format)
* [<Resource Instance Part>](#ResourceInstancePart) [is part of (consists of)](#resourceInstancePartIsPartOfCollection) [<Collection>](#Collection)

#### Relations (incoming)

* [<Resource Instance>](#ResourceInstance) [is made up of (is part of)](#instanceIsMadeUpOfResourceInstanceParts) [<Resource Instance Part>](#ResourceInstancePart)
* [<Process>](#Process) [produces (produced by)](#processProducesInstancePart) [<Resource Instance Part>](#ResourceInstancePart)
* [<Process>](#Process) [uses (is used by)](#processUsesInstancePart) [<Resource Instance Part>](#ResourceInstancePart)

### Class: "Process"

#### Definition

This class comprises any planned or actual activity which changes the state of a [<RESOURCE INSTANCE>](#ResourceInstancePart) or some of its parts, usually taking some input and transforming it to a output. Thus class comprises both the abstract notion of a process plan (heuristics, algorithms, instructions etc.) as well as their actual execution by an [<ACTOR>](#Actor).

NOTE

It is up to the user to decide to which granularity she wants to model the internals of a proces. Usually, a process will be something simple like "running script x" without providing information on its internal structure.

TODO Add concrete usage guidelines and examples.

who: DS when: 2022-04-22 status: new

CHECKME this is very vague, but probably nothing more specific is needed in the context of this model

who: DS when: 2022-04-22 status: new

#### Examples

Encoding a handwritten letter in TEI-XML.

Example #1

Processing the source data of this model with a script which produces a HTML view of it.

Example #2

Applying a chain of NLP tools on source files of a newspaper corpus.

Example #3

The Levenstein distance algorithm. Usually this be executed programmatically when running a software implementing it; it could theoretically also be executed manually by using it as a step-by-step instruction.

Example #4

#### Properties

| Property Name / Cardinality | Data Type | Remarks |
| --- | --- | --- |
| **Activity Type** Cardinality: 0-n | [Vocab Term](#term) in [TaDiRAH](#vocabs.tadirahActivities) |  |
| **Execution type** Cardinality: 0-1 | [Vocab Term](#term) in [Process Types Vocabulary](#vocabs.processExecutionTypes) |  |
| **Name** Cardinality: 1 | [Short Text](#shortText) |  |

#### Relations (outgoing)

* [<Process>](#Process) [is documented in (documents)](#processIsDocumentedInPublication) [<Publication>](#Publication)
* [<Process>](#Process) [is subprocess of (has subprocess)](#processIsSubprocessOfProcess) [<Process>](#Process)
* [<Process>](#Process) [is successor of (has successor)](#processIsSuccessorOfProcess) [<Process>](#Process)
* [<Process>](#Process) [produces (produced by)](#processProducesInstancePart) [<Resource Instance Part>](#ResourceInstancePart)
* [<Process>](#Process) [uses (is used by)](#processUsesInstancePart) [<Resource Instance Part>](#ResourceInstancePart)

#### Relations (incoming)

* [<Actor>](#Actor) [executes (executed by)](#actorExecutesProcess) [<Process>](#Process)
* [<Software>](#Software) [implements (is implemented in)](#softwareImplementsProcess) [<Process>](#Process)
* [<Process>](#Process) [is subprocess of (has subprocess)](#processIsSubprocessOfProcess) [<Process>](#Process)
* [<Process>](#Process) [is successor of (has successor)](#processIsSuccessorOfProcess) [<Process>](#Process)

## Groups

### Group: "CRIS"

Information on this classes are sourced from ÖAW's future CRIS / successor of "AkademIS".

**Group Members:** [<ACTOR>](#Actor), [<PROJECT>](#Project), [<INSTITUTION>](#Institution), [<PERSON>](#Person), [<PUBLICATION>](#Publication)

### Group: "Redmine"

Information on this classes are sourced from ACDH-CH's project management system "Redmine".

**Group Members:** [<SERVICE>](#Service), [<COMPUTER>](#Computer), [<RESOURCE>](#Resource)

## Relations

### has member

is member of

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<ACTOR>](#Actor)

### is hosting institution of

has hosting institution

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<PROJECT>](#Project)

### is associated with

has associated person

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PERSON>](#Person)  
**Target Class:** [<RESOURCE>](#Resource)  
**Properties:**

* 0-n **contribution role:** Datatype: [Vocab Term](#term)

NOTE

This is a catch-all relation between a resource and natural persons relevant for its creation, existance, maintenance in any way. The type of role a person plays for a resource is provided by the [contribution role](#personIsAssociatedWithResource.role) property.

More granular information on the nature of the contribution can be added via [<ACTOR>](#Actor) [*executes*](#actorExecutesProcess) [<PROCESS>](#Process) producing [<RESOURCE INSTANCE>](#ResourceInstancePart) which [*is instantiation*](#Xc7890e2fc9c1cc3dec538623414131e955df85e) of [<RESOURCE>](#Resource).

A note on the arity: While we *should* know at least a principal of a resource who is in charge of it, it might well be that we don't have this information.

#### Examples

*Charly Mörth* (a [<PERSON>](#Person)) [*is associated with*](#personIsAssociatedWithResource) *VICAV* (a [<RESOURCE>](#Resource)) as its [lead](#vocabs.contributorRoles).

Example #1

### is owner of

is owned by

**Cardinality:** 0-n / 1-n  
**Source Class:** [<ACTOR>](#Actor)  
**Target Class:** [<RESOURCE>](#Resource)

NOTE This is a mandatory shortcut for [<ACTOR>](#Actor) [*is associated with role*](#personIsAssociatedWithResource) [owner](#vocabs.contributorRoles) with [<RESOURCE>](#Resource).

### produces

is output of

**Cardinality:** 1 / 1-n  
**Source Class:** [<PROJECT>](#Project)  
**Target Class:** [<RESOURCE>](#Resource)

### re-uses

is re-used by

**Status:** needs\_review

**Cardinality:** 1-n / 1-n  
**Source Class:** [<PROJECT>](#Project)  
**Target Class:** [<RESOURCE>](#Resource)

CHECKME

Unclear whether we want to have this on the project level. We might assume that any relevant re-use in a project manifest itself in some kind of [<RESOURCE>](#Resource) so it would be better to make the relation on a more granular level. OTOH it might be difficult to gain such knowledge when external projects re-use data of the ACDH-CH.

who: DS when: 2022-04-25 status: new

### re-uses\*

is re-used by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<RESOURCE>](#Resource)  
**Target Class:** [<RESOURCE>](#Resource)  
**Properties:**

* 0-1 **Resource Re-use Description:**
* NOTE provides details on the re-use.
* Datatype: [Long Text](#longText)

NOTE

This is a general relation between two resource instances indicating that the target resource was used while creating the source resource. It is deliberately kept on the Resource level to allow to model cross-class re-use relations between its subclasses (e.g. publication re-uses a dataset, a dataset re-uses a software etc.).

This is a shortcut for Resource is related to resource with rrle "reuses"

### is related to with role

is related to with role

**Cardinality:** 1-n / 1-n  
**Source Class:** [<RESOURCE>](#Resource)  
**Target Class:** [<RESOURCE>](#Resource)  
**Properties:**

* 0-1 **:**
* NOTE provides details on the re-use.
* Datatype: [Long Text](#longText)

NOTE This is a general relation between two resource instances indicating that both resources are related in some way which is expressed via the property. It is deliberately kept on the Resource level to allow to model cross-class re-use relations between its subclasses (e.g. publication re-uses a dataset, a dataset re-uses a software etc.).

### is member with role of

has member with role

**Cardinality:** 1-n / 1-n  
**Source Class:** [<ACTOR>](#Actor)  
**Target Class:** [<PROJECT>](#Project)  
**Properties:**

* 1-n **role:** Datatype: [Vocab Term](#term)

### provides instance of

is instantiated in

**Cardinality:** 0-n / 1-n  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<SOFTWARE>](#Software)

#### Examples

Service *PMB* (a [<SERVICE>](#Service)) is [*running*](#serviceProvidesInstanceOfSoftware) *APIS* (a [<SOFTWARE>](#Software)) [*providing an instance of*](#serviceProvidesInstanceOfResource) the *APIS Dataset* (a [<RESOURCE>](#Resource)).

Example #1

### is made up of

is part of

**Cardinality:** 1 / 1-n  
**Source Class:** [<RESOURCE INSTANCE>](#ResourceInstance)  
**Target Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

### is part of

consists of

**Cardinality:** 1-n / 0-n  
**Source Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)  
**Target Class:** [<COLLECTION>](#Collection)

### provides

is provided by

**Cardinality:** 1-n / 0-n  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<RESOURCE INSTANCE>](#ResourceInstance)

### uses

is used by

**Cardinality:** 1-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

### produces

produced by

**Cardinality:** 1-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

### is subprocess of

has subprocess

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<PROCESS>](#Process)

#### Examples

*Cooking a soup* (a [<PROCESS>](#Process)), an [interactive](#vocabs.processExecutionTypes) [type of process](#process.executionType), [*has the parts*](#processIsSubprocessOfProcess)

* *washing vegetables* (a [<PROCESS>](#Process))
* *cutting vegetables* (a [<PROCESS>](#Process))
* *setting up water in a pot* (a [<PROCESS>](#Process))
* *turning on the oven* (a [<PROCESS>](#Process))
* *putting vegetables in the pot* (a [<PROCESS>](#Process))
* *letting vegetables cook in the boiling water* (a [<PROCESS>](#Process))
* *adding spices* (a [<PROCESS>](#Process))
* *blending the vegetables* (a [<PROCESS>](#Process))
* *filling the soup in a plate.* (a [<PROCESS>](#Process))

Example #1

### is successor of

has successor

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<PROCESS>](#Process)

#### Examples

*Transcribing* (a [<PROCESS>](#Process)) consists in

Example #1

### is instantiation of

is instantiated in

**Cardinality:** 1-n / 1  
**Source Class:** [<RESOURCE INSTANCE>](#ResourceInstance)  
**Target Class:** [<RESOURCE>](#Resource)

### is authoritative instantiation of

has authoritative instantiation

**Cardinality:** 1 / 1  
**Source Class:** [<RESOURCE INSTANCE>](#ResourceInstance)  
**Target Class:** [<RESOURCE>](#Resource)

NOTE This relation identifies exactly one resource instance which is the authoritative version of a [<RESOURCE>](#Resource). Depending on the [resource's lifecycle status](#resource.status) this can mean different things:

* if resource is "

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<PUBLICATION>](#Publication)

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROJECT>](#Project)  
**Target Class:** [<PUBLICATION>](#Publication)

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<PUBLICATION>](#Publication)

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<DATA SET>](#Dataset)  
**Target Class:** [<PUBLICATION>](#Publication)

### has format

is used in

**Cardinality:** 1 / 1  
**Source Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)  
**Target Class:** [<FORMAT>](#Format)

### is funder of\*

is funded by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<PROJECT>](#Project)

NOTE This is a shortcut for [<INSTITUTION>](#Institution) [*is member of*](#actorIsMemberOfProject) [<PROJECT>](#Project) with [role="funder"](#vocabs.projectRoles)

### is part of

has part

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<INSTITUTION>](#Institution)

### executes

executed by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<ACTOR>](#Actor)  
**Target Class:** [<PROCESS>](#Process)

#### Examples

*Tesseract* (a [<SOFTWARE>](#Software)) [*running on*](#actorExecutesProcess) *my local PC* (a [<COMPUTER>](#Computer)) [*outputting*](#processProducesInstancePart) the file *output.txt* (a [<RESOURCE INSTANCE PART>](#ResourceInstancePart)) [*of type*](#fileHasFormat) *plain text* (a [<FORMAT>](#Format)) residing on *my local PC's SDD* (a [<INFORMATION CARRIER>](#Carrier)) is a [<PROCESS>](#Process) [of type](#process.executionType) ["programmatic"](#vocabs.processExecutionTypes).

Example #1

### is successor of

has successor

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<SOFTWARE>](#Software)

### has dependency

is dependency of

**Cardinality:** 1-n / 1-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<SOFTWARE>](#Software)

NOTE used to model software dependency graphs

#### Examples

*TEI\_publisher* (a [<SOFTWARE>](#Software)) *has dependency* *eXistdb* (a [<SOFTWARE>](#Software)) *has dependency* *Java* (a [<SOFTWARE>](#Software))

Example #1

### is successor of

has successor

**Cardinality:** 1-n / 1-n  
**Source Class:** [<PUBLICATION>](#Publication)  
**Target Class:** [<PUBLICATION>](#Publication)

NOTE establishes a chain of versions

### is successor of

has successor

**Cardinality:** 1-n / 1-n  
**Source Class:** [<DATA SET>](#Dataset)  
**Target Class:** [<DATA SET>](#Dataset)

NOTE establishes a chain of versions

### has carrier

carries

**Cardinality:** 1-n / 1  
**Source Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)  
**Target Class:** [<INFORMATION CARRIER>](#Carrier)

### implements

is implemented in

**Cardinality:** 1-n / 1-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<PROCESS>](#Process)

### has Maintainer

is maintainer of

**Cardinality:** 1-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<PERSON>](#Person)

### is creator of

created by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<PERSON>](#Person)  
**Target Class:** [<RESOURCE>](#Resource)

NOTE This is the most general

### is described in

describes

**Cardinality:** 1 / 0-n  
**Source Class:** [<RESOURCE>](#Resource)  
**Target Class:** [<METADATA>](#Metadata)

### has responsible\*

is responsible for

**Cardinality:** 1-n / 0-n  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<PERSON>](#Person)

NOTE This is a shortcut for Service hasContributor with Role Person

### is running on

hosts

**Cardinality:** 1-n / 0-1  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<COMPUTER>](#Computer)

### consumes data of format

is input format of

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<FORMAT>](#Format)

### produces data of format

is output format of

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<FORMAT>](#Format)

## Vocabularies

| Vocabulary Name | Example Values | Remarks |
| --- | --- | --- |
| MIME Type Vocabulary |  |  |
| Project Roles Vocabulary | PI / project partner / senior consultant / team member / SAB member / funder |  |
| Access Restrictions Vocabulary | public access / acadmic access / external access / institution-wide access / institute-wide access / group-wide access / user-specific access | Defines who should have access to the resource. Depending on the location of the resource a script should be able to |
| ARCHE Licenses Vocabulary | unlicensed / CC BY 4.0 / CC BY SA 4.0 / CC BY SA NC 4.0 / CC 0 / Public Domain | The ARCHE vocabulary will need to be extended by some terms. |
| Process Types Vocabulary | manual / programmatic / interactive |  |
| ARCHE Lifecycle Status Vocabulary | planned / in preparation / in use internally / published / to be deprecated / deprecated | The [ARCHE life cycle status vocabulary](https://vocabs.acdh.oeaw.ac.at/arche_lifecycle_status) will need to be extended by some terms.  In case a resource has status="published", there should be at least one resourceInstance with role "authoritative copy". |
| ARCHE Subjects Vocabulary |  |  |
| ISO Language Codes |  | [ISO 639-1 Language Codes on ACDH-CH Vocabs](https://vocabs.acdh.oeaw.ac.at/iso639-1/en/) |
| Software Type Vocabulary | Tagger / NE Recognizer / Editor / Server / Relational Database / … | Specifies the type of the software being described. |
| Service Status Vocabulary |  |  |
| Instance Roles Vocabulary | working copy / temporary copy / authoritative copy / SIP / AIP / backup / unknown | The role an instance plays in the lifecycle of a resource. |
| Instance Part Types Vocabulary | file / document / database / database table / database row / database record |  |
| Vontributor Role Vocabulary |  | Values should be taken from [MARC Code List for Relators](https://www.loc.gov/marc/relators/relaterm.html). |
| Dimension Precision Vocabulary | exact / estimated |  |
| Units Vocabulary | cm / mm / in / lines of text / folio / characters / tokens / kilobyte / gigabyte |  |
| Resource Instance completness Vocabulary | complete / fragmentary / supernumerary / component |  |
| TaDiRAH |  |  |
| Carrier Types Vocabulary | Network drive / local HDD / local SSD / external HDD / external SSD / CD-ROM / DVD-ROM / Tape |  |

## Datatypes

| Name | Used by | Remarks |
| --- | --- | --- |
| Date | [Service.End of life](#service.eol) | a iso date |
| Dimension | [Metadata.extent](#metadata.resourceInstancePart.extent), [ResourceInstancePart.extent](#resourceInstancePart.extent) | a complex datatype describing the extent of a [<RESOURCE INSTANCE PART>](#ResourceInstancePart) by three dimensions:   * unit: the [unit of measurement](#vocabs.units) (e.g. kilobytes, files, rows etc.) * quantity: the amount of measurement units as an integer value * precision: the precision of measurement expressed as a [controlled vocabulary](#vocabs.dimensionPrecision) |
| Integer |  |  |
| Long Text | [resourceIsRelatedToResource.](#resourceRelatesToResource.role), [resourceReusesResource.Resource Re-use Description](#resourceReusesResource.description), [Resource.Description](#resource.description), [Software.Description](#software.resource.description) | A string with more than 256 characters |
| Short Text | [Actor.Name](#actor.name), [Computer.IP Address](#server.ip), [Computer.Server Name](#server.name), [Format.Name](#format.shortName), [Format.PRONOM Unique Identifier](#format.pronomID), [Carrier.Carrier name](#carrier.name), [Institution.Name](#institution.actor.name), [Metadata.Metadata name](#metadata.name), [Person.Name](#person.actor.name), [Process.Name](#process.name), [Project.Name](#project.name), [Project.external ID](#project.externalIdentifier), [Resource.Name](#resource.name), [Service.Name](#service.actor.name), [Software.Name](#software.resource.name) | A string with less than 256 characters |
| URI | [Institution.ROR ID](#institution.rorid), [Metadata.Location](#metadata.location), [Metadata.location](#metadata.ResourceInstancePart.location), [Person.ORCID](#person.orcid), [ResourceInstancePart.location](#ResourceInstancePart.location), [ResourceInstance.Resource Instance Access Point](#resourceInstance.accessPoint), [Resource.PID](#resource.pid), [Service.URI](#service.uri), [Software.PID](#software.resource.pid) | any URI |
| Vocab Term | [personIsAssociatedWithResource.contribution role](#personIsAssociatedWithResource.role), [actorIsMemberOfProject.role](#projectHasMember.role), [Format.Mime Type](#format.mimetype), [Carrier.Carrier Type](#carrier.type), [Process.Activity Type](#process.activityType), [Process.Execution type](#process.executionType), [ResourceInstance.Completness](#resourceInstance.completeness), [ResourceInstance.Resource Instance License](#resourceInstance.license), [ResourceInstance.Type](#resourceInstance.function), [Resource.Access Restrictions](#resource.accessRestrictions), [Resource.License](#resource.license), [Resource.Status](#resource.status), [Resource.Subject](#resource.subject), [Service.Access Restrictions](#service.accessRestrictions), [Service.Service Status](#service.status), [Software.Access Restrictions](#software.resource.accessRestrictions), [Software.License](#software.resource.license), [Software.Status](#software.resource.status), [Software.Subject](#software.resource.subject), [Software.Type](#software.type) | A term from a controlled vocabulary. |

## References

Open AIRE Research Graph Data Model 1.3Manghi, Paolo, Bardi, Alessia, Atzori, Claudio, Baglioni, Miriam, Manola, Natalia, Schirrwagen, Jochen, & Principe, Pedro. (2019). The OpenAIRE Research Graph Data Model (1.3). Zenodo. <https://doi.org/10.5281/zenodo.2643199>

OpenAIRE LOD Ontology 1.1ATHENA RC, UBONN: OpenAIRE LOD Ontology 1.1 (2021-02-20). <http://lod.openaire.eu/vocab>.

ACDH-CH ARCHE Ontology 3.1Trogniz, Martina, Zoltak, Matteusz, Carloni, Massimiliano, Czeitschner, Ulrike: ACDH-CH ARCHE Ontology. Version 3.1<https://github.com/acdh-oeaw/arche-schema/releases/tag/v3.1>

## Relations

### has member

is member of

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<ACTOR>](#Actor)

### is hosting institution of

has hosting institution

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<PROJECT>](#Project)

### is associated with

has associated person

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PERSON>](#Person)  
**Target Class:** [<RESOURCE>](#Resource)  
**Properties:**

* 0-n **contribution role:** Datatype: [Vocab Term](#term)

NOTE

This is a catch-all relation between a resource and natural persons relevant for its creation, existance, maintenance in any way. The type of role a person plays for a resource is provided by the [contribution role](#personIsAssociatedWithResource.role) property.

More granular information on the nature of the contribution can be added via [<ACTOR>](#Actor) [*executes*](#actorExecutesProcess) [<PROCESS>](#Process) producing [<RESOURCE INSTANCE>](#ResourceInstancePart) which [*is instantiation*](#Xc7890e2fc9c1cc3dec538623414131e955df85e) of [<RESOURCE>](#Resource).

A note on the arity: While we *should* know at least a principal of a resource who is in charge of it, it might well be that we don't have this information.

#### Examples

*Charly Mörth* (a [<PERSON>](#Person)) [*is associated with*](#personIsAssociatedWithResource) *VICAV* (a [<RESOURCE>](#Resource)) as its [lead](#vocabs.contributorRoles).

Example #1

### is owner of

is owned by

**Cardinality:** 0-n / 1-n  
**Source Class:** [<ACTOR>](#Actor)  
**Target Class:** [<RESOURCE>](#Resource)

NOTE This is a mandatory shortcut for [<ACTOR>](#Actor) [*is associated with role*](#personIsAssociatedWithResource) [owner](#vocabs.contributorRoles) with [<RESOURCE>](#Resource).

### produces

is output of

**Cardinality:** 1 / 1-n  
**Source Class:** [<PROJECT>](#Project)  
**Target Class:** [<RESOURCE>](#Resource)

### re-uses

is re-used by

**Status:** needs\_review

**Cardinality:** 1-n / 1-n  
**Source Class:** [<PROJECT>](#Project)  
**Target Class:** [<RESOURCE>](#Resource)

CHECKME

Unclear whether we want to have this on the project level. We might assume that any relevant re-use in a project manifest itself in some kind of [<RESOURCE>](#Resource) so it would be better to make the relation on a more granular level. OTOH it might be difficult to gain such knowledge when external projects re-use data of the ACDH-CH.

who: DS when: 2022-04-25 status: new

### re-uses\*

is re-used by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<RESOURCE>](#Resource)  
**Target Class:** [<RESOURCE>](#Resource)  
**Properties:**

* 0-1 **Resource Re-use Description:**
* NOTE provides details on the re-use.
* Datatype: [Long Text](#longText)

NOTE

This is a general relation between two resource instances indicating that the target resource was used while creating the source resource. It is deliberately kept on the Resource level to allow to model cross-class re-use relations between its subclasses (e.g. publication re-uses a dataset, a dataset re-uses a software etc.).

This is a shortcut for Resource is related to resource with rrle "reuses"

### is related to with role

is related to with role

**Cardinality:** 1-n / 1-n  
**Source Class:** [<RESOURCE>](#Resource)  
**Target Class:** [<RESOURCE>](#Resource)  
**Properties:**

* 0-1 **:**
* NOTE provides details on the re-use.
* Datatype: [Long Text](#longText)

NOTE This is a general relation between two resource instances indicating that both resources are related in some way which is expressed via the property. It is deliberately kept on the Resource level to allow to model cross-class re-use relations between its subclasses (e.g. publication re-uses a dataset, a dataset re-uses a software etc.).

### is member with role of

has member with role

**Cardinality:** 1-n / 1-n  
**Source Class:** [<ACTOR>](#Actor)  
**Target Class:** [<PROJECT>](#Project)  
**Properties:**

* 1-n **role:** Datatype: [Vocab Term](#term)

### provides instance of

is instantiated in

**Cardinality:** 0-n / 1-n  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<SOFTWARE>](#Software)

#### Examples

Service *PMB* (a [<SERVICE>](#Service)) is [*running*](#serviceProvidesInstanceOfSoftware) *APIS* (a [<SOFTWARE>](#Software)) [*providing an instance of*](#serviceProvidesInstanceOfResource) the *APIS Dataset* (a [<RESOURCE>](#Resource)).

Example #1

### is made up of

is part of

**Cardinality:** 1 / 1-n  
**Source Class:** [<RESOURCE INSTANCE>](#ResourceInstance)  
**Target Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

### is part of

consists of

**Cardinality:** 1-n / 0-n  
**Source Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)  
**Target Class:** [<COLLECTION>](#Collection)

### provides

is provided by

**Cardinality:** 1-n / 0-n  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<RESOURCE INSTANCE>](#ResourceInstance)

### uses

is used by

**Cardinality:** 1-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

### produces

produced by

**Cardinality:** 1-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)

### is subprocess of

has subprocess

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<PROCESS>](#Process)

#### Examples

*Cooking a soup* (a [<PROCESS>](#Process)), an [interactive](#vocabs.processExecutionTypes) [type of process](#process.executionType), [*has the parts*](#processIsSubprocessOfProcess)

* *washing vegetables* (a [<PROCESS>](#Process))
* *cutting vegetables* (a [<PROCESS>](#Process))
* *setting up water in a pot* (a [<PROCESS>](#Process))
* *turning on the oven* (a [<PROCESS>](#Process))
* *putting vegetables in the pot* (a [<PROCESS>](#Process))
* *letting vegetables cook in the boiling water* (a [<PROCESS>](#Process))
* *adding spices* (a [<PROCESS>](#Process))
* *blending the vegetables* (a [<PROCESS>](#Process))
* *filling the soup in a plate.* (a [<PROCESS>](#Process))

Example #1

### is successor of

has successor

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<PROCESS>](#Process)

#### Examples

*Transcribing* (a [<PROCESS>](#Process)) consists in

Example #1

### is instantiation of

is instantiated in

**Cardinality:** 1-n / 1  
**Source Class:** [<RESOURCE INSTANCE>](#ResourceInstance)  
**Target Class:** [<RESOURCE>](#Resource)

### is authoritative instantiation of

has authoritative instantiation

**Cardinality:** 1 / 1  
**Source Class:** [<RESOURCE INSTANCE>](#ResourceInstance)  
**Target Class:** [<RESOURCE>](#Resource)

NOTE This relation identifies exactly one resource instance which is the authoritative version of a [<RESOURCE>](#Resource). Depending on the [resource's lifecycle status](#resource.status) this can mean different things:

* if resource is "

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<PUBLICATION>](#Publication)

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROJECT>](#Project)  
**Target Class:** [<PUBLICATION>](#Publication)

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<PROCESS>](#Process)  
**Target Class:** [<PUBLICATION>](#Publication)

### is documented in

documents

**Cardinality:** 0-n / 0-n  
**Source Class:** [<DATA SET>](#Dataset)  
**Target Class:** [<PUBLICATION>](#Publication)

### has format

is used in

**Cardinality:** 1 / 1  
**Source Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)  
**Target Class:** [<FORMAT>](#Format)

### is funder of\*

is funded by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<PROJECT>](#Project)

NOTE This is a shortcut for [<INSTITUTION>](#Institution) [*is member of*](#actorIsMemberOfProject) [<PROJECT>](#Project) with [role="funder"](#vocabs.projectRoles)

### is part of

has part

**Cardinality:** 1-n / 1-n  
**Source Class:** [<INSTITUTION>](#Institution)  
**Target Class:** [<INSTITUTION>](#Institution)

### executes

executed by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<ACTOR>](#Actor)  
**Target Class:** [<PROCESS>](#Process)

#### Examples

*Tesseract* (a [<SOFTWARE>](#Software)) [*running on*](#actorExecutesProcess) *my local PC* (a [<COMPUTER>](#Computer)) [*outputting*](#processProducesInstancePart) the file *output.txt* (a [<RESOURCE INSTANCE PART>](#ResourceInstancePart)) [*of type*](#fileHasFormat) *plain text* (a [<FORMAT>](#Format)) residing on *my local PC's SDD* (a [<INFORMATION CARRIER>](#Carrier)) is a [<PROCESS>](#Process) [of type](#process.executionType) ["programmatic"](#vocabs.processExecutionTypes).

Example #1

### is successor of

has successor

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<SOFTWARE>](#Software)

### has dependency

is dependency of

**Cardinality:** 1-n / 1-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<SOFTWARE>](#Software)

NOTE used to model software dependency graphs

#### Examples

*TEI\_publisher* (a [<SOFTWARE>](#Software)) *has dependency* *eXistdb* (a [<SOFTWARE>](#Software)) *has dependency* *Java* (a [<SOFTWARE>](#Software))

Example #1

### is successor of

has successor

**Cardinality:** 1-n / 1-n  
**Source Class:** [<PUBLICATION>](#Publication)  
**Target Class:** [<PUBLICATION>](#Publication)

NOTE establishes a chain of versions

### is successor of

has successor

**Cardinality:** 1-n / 1-n  
**Source Class:** [<DATA SET>](#Dataset)  
**Target Class:** [<DATA SET>](#Dataset)

NOTE establishes a chain of versions

### has carrier

carries

**Cardinality:** 1-n / 1  
**Source Class:** [<RESOURCE INSTANCE PART>](#ResourceInstancePart)  
**Target Class:** [<INFORMATION CARRIER>](#Carrier)

### implements

is implemented in

**Cardinality:** 1-n / 1-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<PROCESS>](#Process)

### has Maintainer

is maintainer of

**Cardinality:** 1-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<PERSON>](#Person)

### is creator of

created by

**Cardinality:** 1-n / 1-n  
**Source Class:** [<PERSON>](#Person)  
**Target Class:** [<RESOURCE>](#Resource)

NOTE This is the most general

### is described in

describes

**Cardinality:** 1 / 0-n  
**Source Class:** [<RESOURCE>](#Resource)  
**Target Class:** [<METADATA>](#Metadata)

### has responsible\*

is responsible for

**Cardinality:** 1-n / 0-n  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<PERSON>](#Person)

NOTE This is a shortcut for Service hasContributor with Role Person

### is running on

hosts

**Cardinality:** 1-n / 0-1  
**Source Class:** [<SERVICE>](#Service)  
**Target Class:** [<COMPUTER>](#Computer)

### consumes data of format

is input format of

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<FORMAT>](#Format)

### produces data of format

is output format of

**Cardinality:** 0-n / 0-n  
**Source Class:** [<SOFTWARE>](#Software)  
**Target Class:** [<FORMAT>](#Format)