Leibniz-Institut für
Astrophysik Potsdam

SimDM and ProvenanceDM

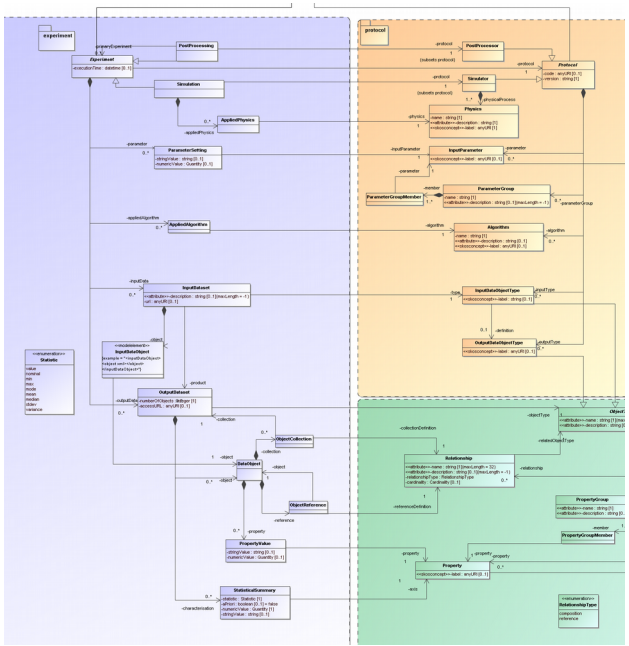
Asterics Technology Forum, 22-23. March 2017

Provenance Discussions

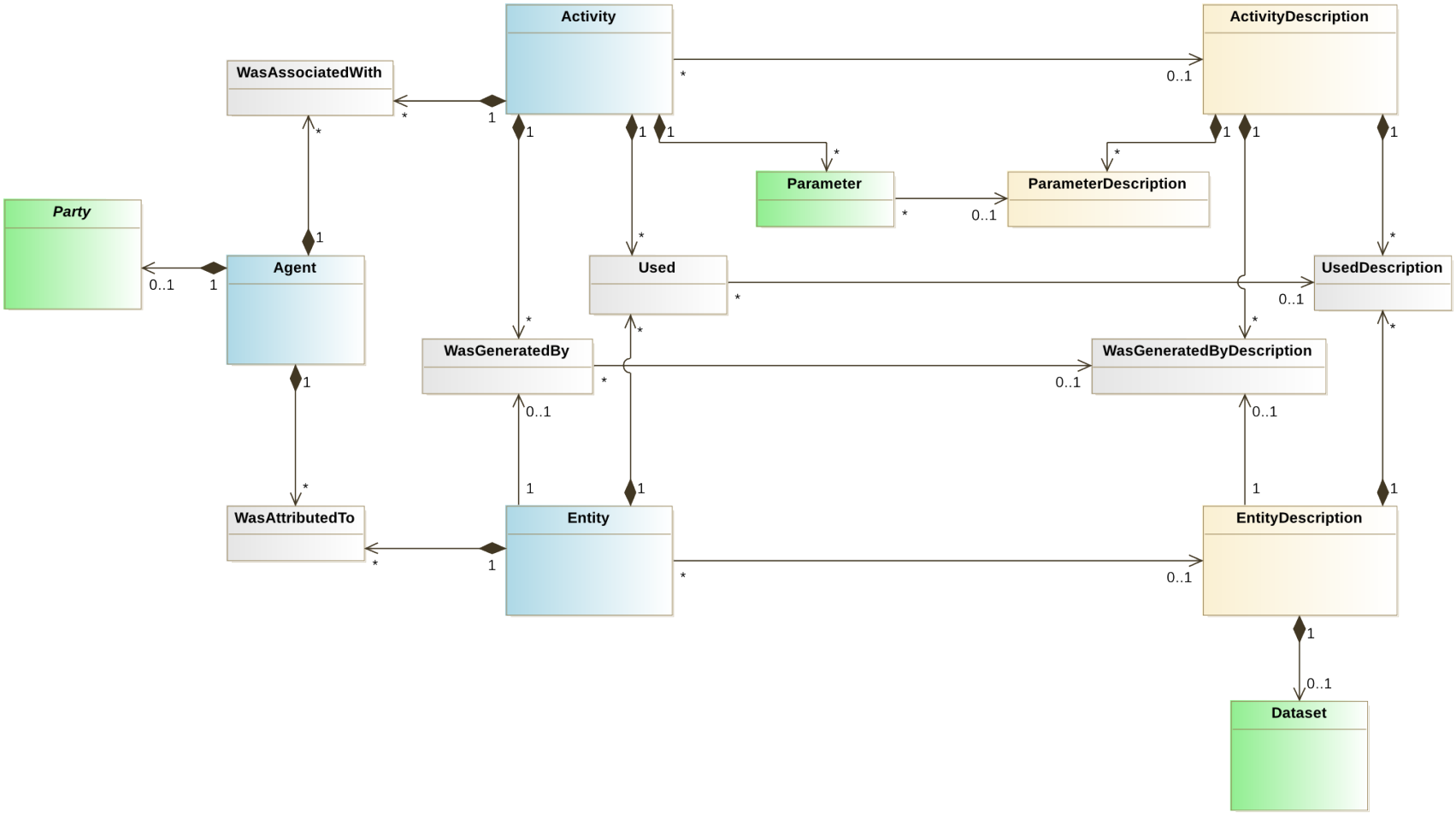
Kristin Riebe

Simulation Data Model (SimDM)

- Data model for structuring metadata for simulations
- Contains some provenance aspects
- See also talk by David Languignon in Dec. 2016
- Extract from SimDM class diagram:



ProvencenDM - overview



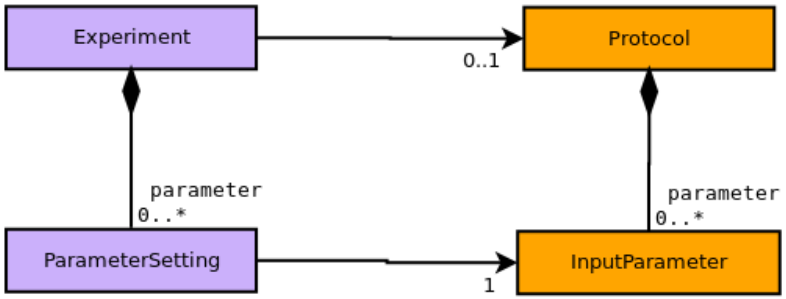
Links between SimDM and ProvenanceDM

- also see latest Provenance Working Draft for some examples (Table 12):

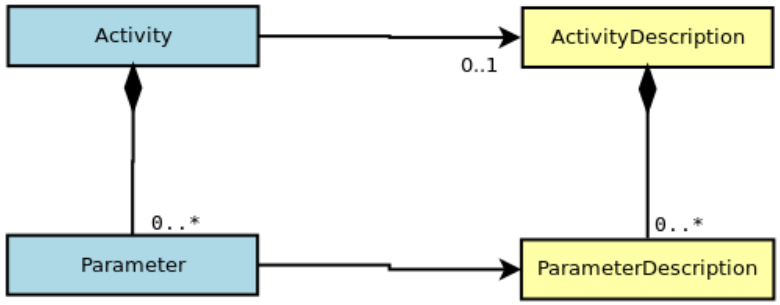
Simulation DM	Provenance DM	Comment
Experiment	Activity	
Experiment.name	Activity.label	human readable name; name attribute in SimDM is inherited from Resource-class
Experiment.executionTime	Activity.endTime	end time of the execution of an experiment/activity
Experiment.protocol	Activity.description_ref	reference to the protocol or description class
Protocol	ActivityDescription	
Protocol.name	ActivityDescription.label	human readable name
Protocol.referenceURL	ActivityDescription.doculink	reference to a webpage describing it
ParameterSetting	Parameter	value of an (input) parameter
InputParameter	ParameterDescription	description of an (input) parameter
Party	Agent	responsible person or organization
Party.name	Agent.label	name of the agent
Contact	WasAssociatedWith	
Contact.role	WasAssociatedWith.role	role which the agent/-party had for a certain experiment (activity); SimDM roles contain: `owner`, `creator`, `publisher`, `contributor`
Contact.party	WasAssociatedWith.agent	reference to the agent/-party

Experiment and Activity

SimDM



ProvenanceDM



- Experiment, Protocol → Activity, ActivityDescription
- ParameterSetting, InputParameter → Parameter, ParameterDescription
- SimDM offers additionally:
  - (Applied)Algorithm, (Applied)Physics

  → used for characterising the datasets/experiments
  → not expressed in ProvenanceDM
  → could be included as parameters

Experiment and Activity

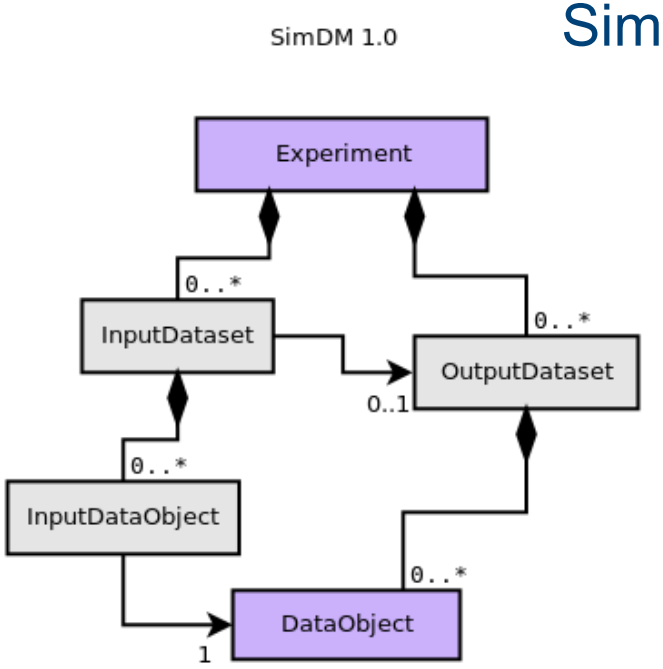
- Attributes:
  - Experiment.name → Activity.label
    Protocol.name → ActivityDescription.label
  - in SimDM a label is a SKOS concept (e.g. Experiment.label)
    → useful for Provenance as well?
  - Experiment.executionTime → Activity.endTime
  - Experiment, Activity etc. are *Resources*
    → inherited attributes:
    name, description, created, updated datetime, status

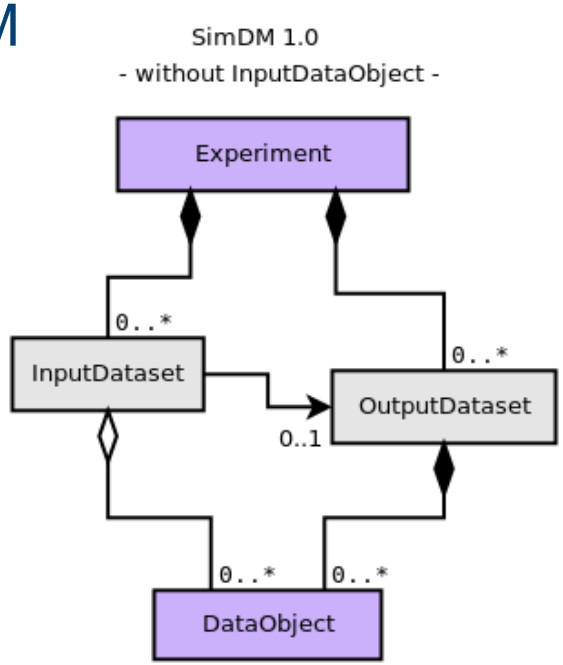
Parameters and Protocols

- very similar in SimDM and ProvenanceDM
- in SimDM: InputParameter inherits properties from abstract class „Field“; SKOS label as UCD
- ParameterValue can be int, char, float etc. => not known beforehand
- SimDM: define different protocol for each code (and code version; especially if there are different parameters for each code)
  - e.g. different simulation codes may require different parameters
  - each parameter is attached to exactly 1 protocol (even if the same meaning, e.g. boxsize)
- Protocol.type can be used to group all simulation codes of the same kind, (should use vocabulary list for this)

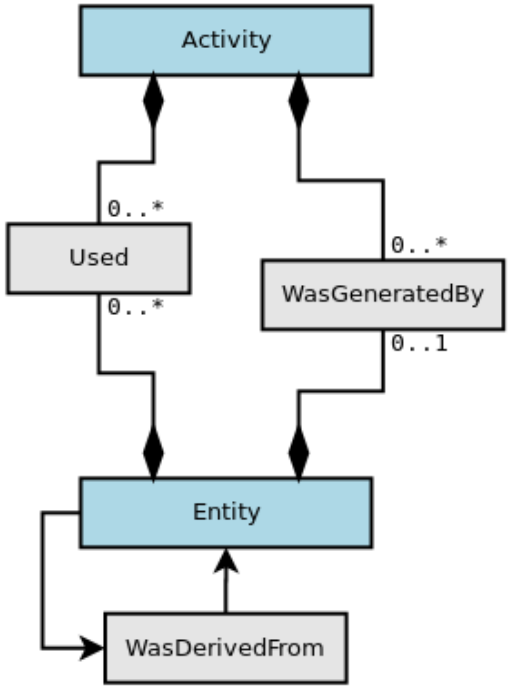
Datasets

SimDM





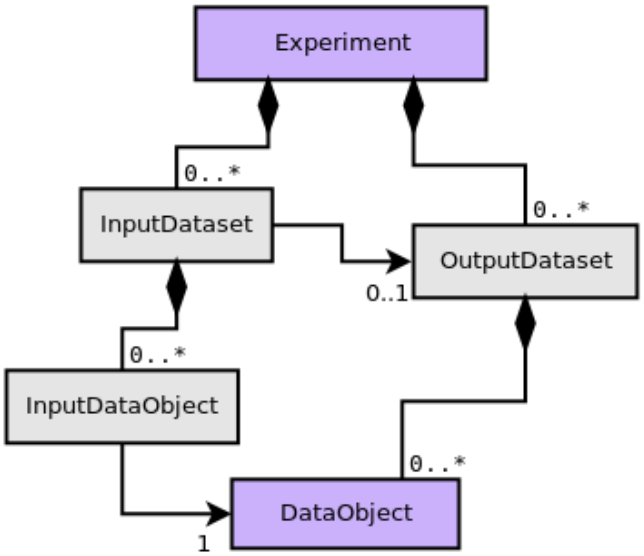
ProvenanceDM

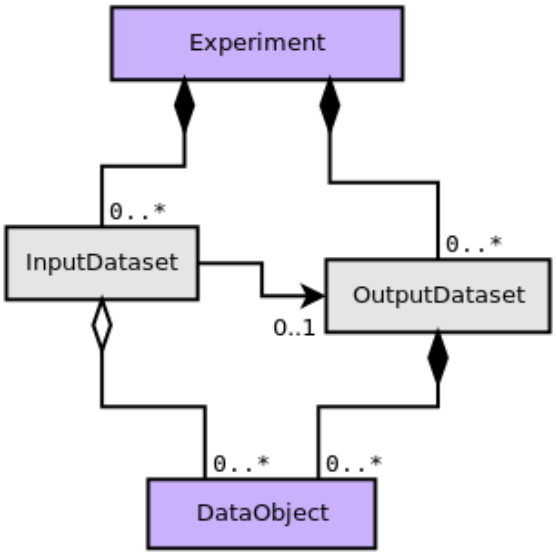


- input/output modelling is very different
- DataObjects are **physical objects** with properties (e.g. halos from halo catalog, particles from a simulation snapshot)
- in Provenance: entity can be file, database table etc., physical content not so important

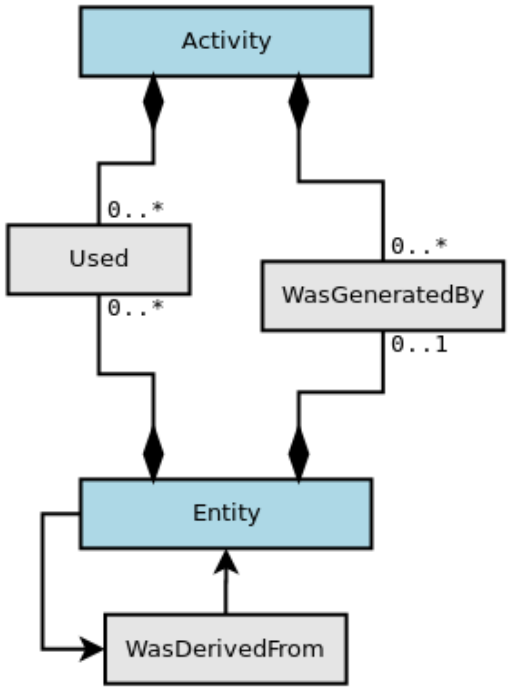
Datasets

SimDM



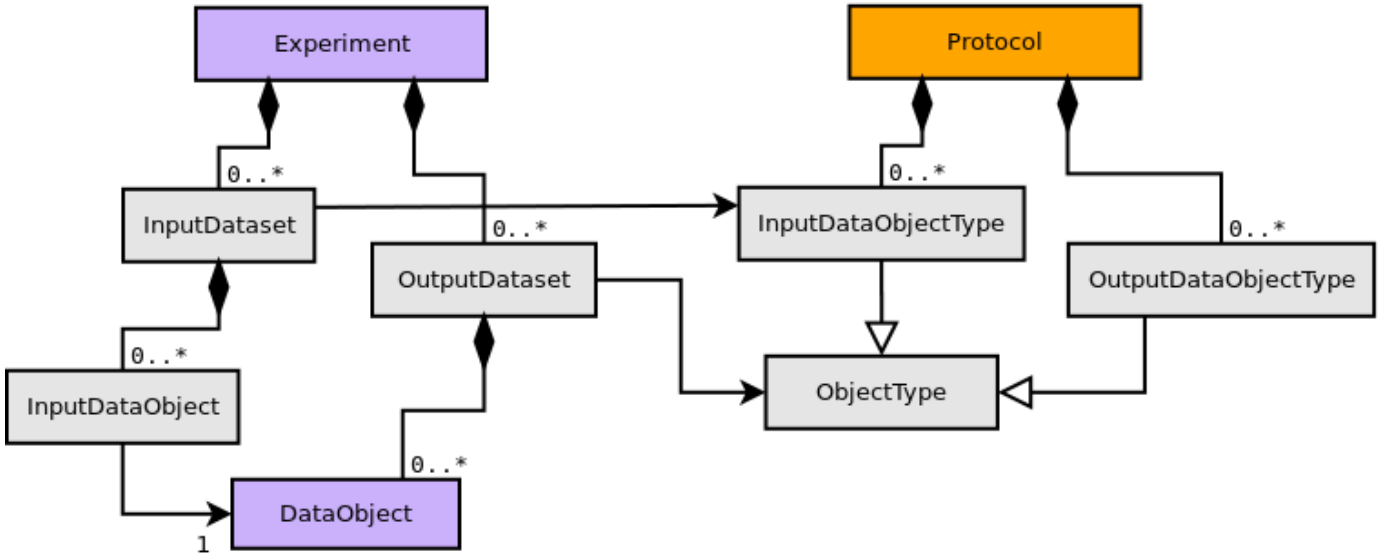


ProvenanceDM



- used.role is hidden in InputDataset.description
- no role for wasGeneratedBy in SimDM

DataObject and -Type



- each InputDataset has an InputDataObjectType (which is linked to Protocol)
- OutputDataset points to ObjectType, can be Target or OutputDataObjectType
- SimDM offers Property and PropertyValue classes
  - similar to parameters, but for datasets, not experiments

Party, Contact and Agent

- SimDM/Party -> ProvenanceDM/Agent
- SimDM/Contact is used for relations between:
  - Party -> Experiment
    - = wasAssociantedWith in ProvenanceDM
  - Party -> Protocol
    - = people responsible for a code (version),
    - not (yet) included in ProvenanceDM

SimDM implementation

- Expanded django web application with prov_simdm app, see https://github.com/kristinriebe/provenance-cosmosim/
- Many parts of SimDM implemented for CosmoSim metadata, (cosmological simulations with post-processing steps)
- Enable querying by SimDM metadata (search form)

- Provenance endpoints (entities, activities) using SimDM metadata
- a few SimDAL endpoints (datasets, protocols) implemented as well, returning simple VOTABLE

Provenance from SimDM

- Provenance from SimDM
  - export provenance information from SimDM implementation
  - need mapping of classes, attributes
  - make provenance-related classes and attributes ***required***?

- Which „endpoints“ etc. should a provenance service provide?
  - Full „workflow“ information, also in Prov-N or similar, for using tools from W3C/Southampton suite
  - Search for version, dataset history, but also: derived data

Open for discussions