A Facet-based Open and Extensible Resource Model for Research Data Infrastructures *

Luca Frosini and Pasquale Pagano (Italy)

Abstract
Research Data represent a valuable assets in science as well as in our society. Their management requires the development and operation of Research Data Infrastructures, i.e. complex and distributed systems specifically conceived to address the needs arising in Research Data collection, collation, processing and publishing. The development of such systems require a shared model for describing the existing "resources", namely the datasets as well as the rest of services and entities worth being considered to properly deal with the datasets. In this paper it is presented an open and extensible model based on two basic notions: Resource to describe the entities, Facet to characterize a feature of a Resource. The model enables its users to instantiate these basic concepts and define context-specific relationships among the typologies of defined resources.
Keywords: *Research Data Infrastructure, Resource Model, Resource, Facet.*

1. Introduction

Research Data play a key role in our society. They include both "primary dataset", i.e. data genuinely produced, as well as "derived datasets", i.e. datasets resulting by processing existing datasets. Their management requires dedicated Research Infrastructure and a description of the entire set of "resources" surrounding each dataset (e.g. other datasets, services the dataset has been produced with or suitable for "consuming" the dataset, entities responsible for the dataset).

Research infrastructure (RI) are "facilities, resources and services used by the science community to conduct research and foster innovation"[1]. Grid and Cloud computing Infrastructure provides an excellent support to address Data/Computation intensive paradigm but in a certain sense they can be seen as facilities to implement such a paradigm.

Bardi and Frosini [1] highlighted the needs of researcher for digital services to realize Digital Research Infrastructures (henceforth e-Infrastructure) and highlighting the Data e-Infrastructure as one of the relevant category (in this paper we will use Date e-Infrastructure and Research Data Infrastructure interchangeably). According to Candela et al. [2] Data e-Infrastructures "integrates several technologies, including Grid and Cloud, and promises to offer the necessary management and usage capabilities required to implement the 'Big Data' enabled scientific paradigm" and promoting data sharing and consumption.

To realise this, it is key to implement an Information System (IS) capable to represent datasets as well as the rest of resources associated with it such as Services, Hardware, Actors, Facilities, and Policies. Such an IS acts as a registry offering global and partial view of the Infrastructure resources, their current status, their relationships with other resources, and the policies governing their exploitation. The "descriptions" attached to such "entities" should not be prescribed a priori, rather they should be open and extensible thus to enable diverse actors (being them publishers or consumers) to annotate each entity with specific features.

In particular, we need a model to deal with heterogeneity with respect to:

- Open-ended set of manageable resources;
- Open-ended model for describing resources;
- Diverse workflows governing registration and update of resources;

Due to such an heterogeneity the IS should have the ability to evolve with the evolving needs of the infrastructure at no cost for its clients by (a) supporting new resource types, (b) supporting evolution in the way a resource is described, (c) supporting the same resource type described by using different models.

In Section 2 it is introduced a core model (Information System Model) defining the building blocks for developing a resource model with the envisaged characteristics. In Section 3 it is presented the resource model obtained by relying on the core model to capture the entities of interest identified in D4Science.org[2], a Research Data Infrastructure conceived to support several communities and data management scenarios arising in fields including biological sciences, earth and environmental sciences, agricultural sciences, social sciences and humanities.

---

* First published in the GL19 Conference Proceedings, February 2018.
[1] https://ec.europa.eu/research/infrastructures/index_en.cfm?pg=about
[2] www.d4science.org

2. The Core Model

The proposed "core model" is conceived to provide its users with the building blocks needed to develop an information system suitable for Data e-Infrastructure. This model is based on a graph model having ***Entities*** as nodes and ***Relations*** as edges.

Two typologies of **Entities** are envisaged (cf. Fig. 1): **Resources**, i.e. entities representing a description of a *"thing"* to be managed; **Facets**, i.e. entities contributing to "build" a description of a Resource. Every Resource is characterised by a number of Facets. Every facet, once attached to a Resource profile captures a certain aspect / characterization of the resource. Every facet is characterised by a number of properties.

Two typologies of **Relations** are envisaged: **isRelatedTo**, i.e. a relation linking any two Resources; **consistsOf**, i.e. a relation connecting each Resource with each one of the Facets characterizing it.

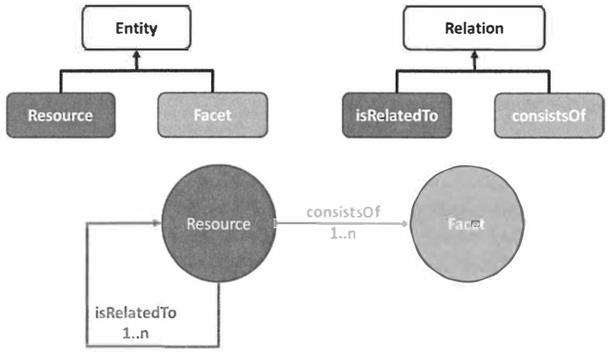

Fig. 1 represent the main concepts of the model.
Fig. 1a evidences the inheritance model of entities and relations.
Fig. 1b evidences the conceptual graph model that is realized by using entities and relations.

Each Entity and Relation: has an **Header** automatically generated for the sake of identification and provenance of the specific information and can be **specialized**. A number of Entity and Relation are expected to be defined when defining a specific information model (cf. Sec. 3). Facet and Relation instances can have additional properties which are not defined in the schema (henceforth schema-mixed mode).

Every Relation relation has - apart the Header - zero or more properties. One of these properties is the **Propagation Constraint** (see dedicated section). Any Relations has a direction, i.e. a "source" and a "target". When inspecting the graph (e.g. at query time) relations can be navigated in both directions, i.e. from source to target and from target to source.

Facet describe a characteristic of a Resource definition, for such a reason, it is not permitted to define a Relation having a Facet as "source". In other words: it is not permitted to define a Relation connecting a Facet with another one or Relation connecting a Facet with a Resource (as target).

Property

As stated any entity and relation is characterised by some properties. A property can be described by the attributes in Table 1.

Table 1. Property attributes

Attribute name	Attribute description
Name	The name of the property
Type	The type of the property
Description	A textual description of the property.
Mandatory	A boolean flag describing the mandatory nature of the property.
ReadOnly	A boolean flag describing the alter-ability nature of the property.
NotNull	A boolean flag describing whether the value of the property must be filled with values other than null or not.
Max	The maximum acceptable value. It is significant for Numbers, Strings (intended as maximum length of the String) and Dates.
Min	The minimum acceptable value. It is significant for Numbers, Strings (intended as minimum length of the String) and Dates.
Regexpr	A Regular Expression used to validate the property value, i.e. precisely characterising the allowed values.

Types can vary from *Basic Type* (typical language programming types i.e. Boolean, Integer, Short, Long, Float, Double, Date, String, Byte and Binary (any values as byte array)); to *Embedded Types* (i.e. complex objects defined from clients and composed of two or more properties belonging to one of the Basic Types). Moreover List, Set (a list with no duplicates) and Map (a key-value pairs having a String as key and an Embedded instance as value) of Embedded can be used.

Defined Embedded Types

Embedded types are mainly composed by two or more properties belonging to *Basic Types* or to another *Embedded Type* (for clear reason recursion is not allowed).

Header

As already stated, every Entity and Relation has an header automatically created and updated by the System. The header is composed by the following properties:

- **uuid** (*String*) [Mandatory=true, NotNull=true, ReadOnly=true] *Regex=^([a-fA-F0-9]{8}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{4}-[a-fA-F0-9]{12}){1}$;)*: this uuid can be used to univocally identify the Entity or the Relation;
- **creator** (*String*) [Mandatory=true, NotNull=true, ReadOnly=true] : the individual or service which create the Entity of Relation;
- **modifiedBy** (*String*) [Mandatory=true, NotNull=true] : the individual or service which modified the last time the Entity of Relation. At creation time it assumes the same value of creator;
- **creationTime** (*Date*) [Mandatory=true, NotNull=true, ReadOnly=true] : creation time in milliseconds. Represent the difference, measured in milliseconds, between the creation time and midnight, January 1, 1970 UTC;
- **lastUpdateTime** (*Date*) [Mandatory=true, NotNull=true] : last Update time in milliseconds. Represent the difference, measured in milliseconds, between the last update time and midnight, January 1, 1970 UTC.

PropagationConstraint

As already stated, each Relation has a propagation constraint which indicates the behavior to be held on a target entity when an event occur in the source entity (please note that the source entity of a relation is alway a Resource by Relation definition). The following two are envisaged:

- **remove** (*Enum) Regex=(cascadeWhenOrphan|cascade|keep*) : i.e. indicate the behaviour to implement for the target Entity when a remove action is performed on the source Resource.
- **add** (*Enum*) *Regex=(propagate|unpropagate*) : i.e. indicate the behaviour to implement for the target Entity when a add action is performed on the source Resource. The default values of the propagation constraints for the basic relations are the following:
- **consistsOf:** remove=cascadeWhenOrphan, add=propagate;
- **isRelatedTo:** remove=keep, add=unpropagate.

isIdentifiedBy
This **consistsOf** specialization is a relation connecting each Resource with one of the Facet which can be used to identify the Resource. Each Resource must be have at least one **isIdentifiedBy** relation. Moreover every Resource can decide to define the type of target Facet for such a relation.

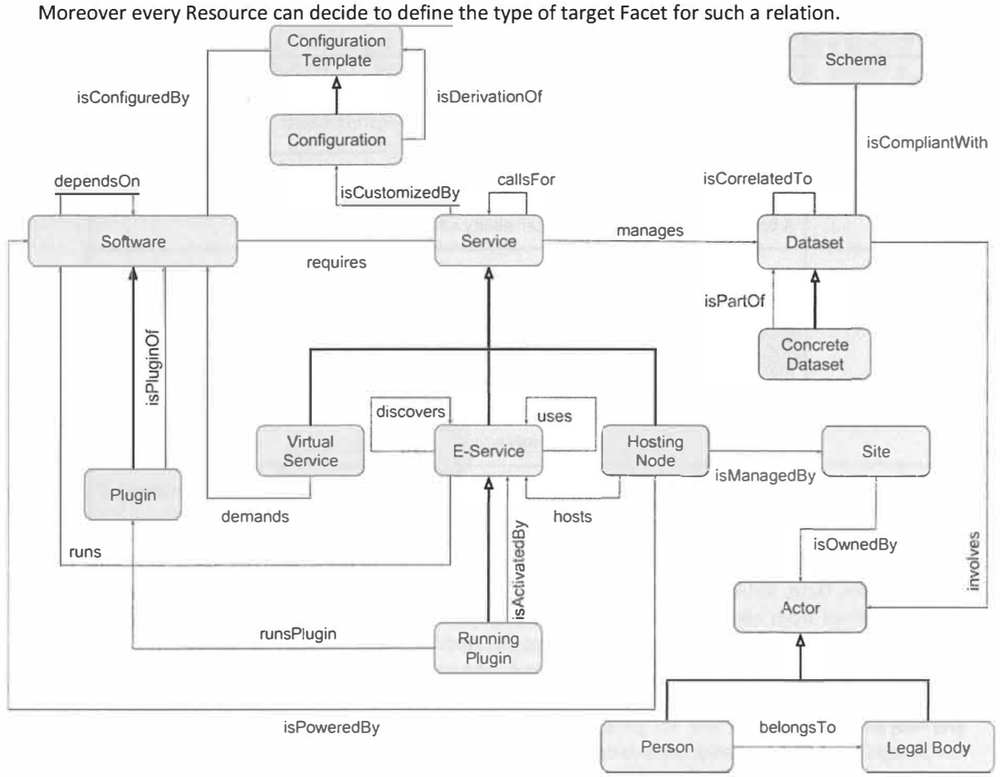

Fig. 2 Overview of gCube Model Resources and (isRelatedTo) Relations

3. The Resource Model

An extended resource model built on top of IS Model which aim to capture the different aspect of the most common resources needed in a Research Data Infrastructure has been defined. The model get the name of gCube Model. gCube is an open-source software toolkit used for building and operating data infrastructures [2] enabling the data sharing and reuse [3]. gCube is developed and maintained by CNR-ISTI.

3.1 Resources

A Resource Type can be identified as Abstract. This means that cannot be instantiated. It is expected that one of its specializations are instantiated. It is not required that an Abstract class establishes an **isIdentifiedBy** relation with a Facet.
Seven resource typologies have been identified and defined (cf. Fig. 2), namely *Dataset*, *Actor*, *Schema*, *Configuration Template*, *Site*, *Service*, and *Software*. In some cases these typologies have been further specialized to capture specific entities, e.g. Concrete Dataset is a sub-type of Dataset, E-Servcie is a subtype of Service. In the reminder of this section the defined Resources are described.

Dataset
A **DataSet** is any set of digital objects representing data and treated collectively as a unit. It is not only the key resource of a Research Data Infrastructure but we could affirm that is the reason why a Research Data Infrastructure is created. It is characterized by the following facets:

- **Identifier Facet** (associated with ***isIdentifiedBy***) : this facet captures information on Identifiers (other than the ones automatically generated by the system) that can be attached to the dataset, e.g. for discovery purpose;
- **Contact Facet** : this facet captures information on a point of contact for the dataset, i.e. a person or a department serving as the coordinator or focal point of information concerning the dataset. There are diverse points of contact that can be associated to the dataset and the role of the association is

captured by using a specific *consistsOf* relation e.g. to represent the owner, the responsible, the creator, the curator, the maintainers and any contributors of the dataset;

- **Access Point Facet** : this facet captures information on an "*access point*" of a dataset, i.e. any web-based endpoint to programmatically interact with the resource via a known protocol. It represent the access point to use for having access to the dataset. The embargo state can be modeled through access policy defined in the *consistsOf* relation;
- **License Facet** : this facet captures information on any license associated with the dataset to capture the policies governing its exploitation and use. The duration of license (if any) can be captured expiry date property defined in the *consistsOf* relation;
- **Provenance Facet** : this facet captures information on provenance/lineage of associated with the dataset;
- **Coverage Facet** this facet captures information on the *extent* of the dataset, i.e. any aspect aiming at capturing an indicator of the amount/area the resource covers be it a geospatial area, a temporal area, or any other "area". Any temporal coverage information characterising the content of the dataset, e.g. the time-span covered by the dataset can be described with this facet associated to the dataset with a specific *consistsOf* relation. Any geospatial coverage information characterising the content of the dataset, e.g. the area covered by the dataset can be described with this facet associated to the dataset with a specific *consistsOf* relation;
- **Descriptive Metadata Facet** : this facet captures information on descriptive metadata to be associated with the dataset, e.g. for discovery purposes;
- **Subject Facet :** this facet captures information on subjects associated with the dataset for descriptive and discovery purposes.

A **Dataset** can be correlated to another **Dataset** (see ***isCorrelatedTo*** relation Fig. 2).

Concrete Dataset
Concrete Dataset (specialization of *Dataset*) is any incarnation/manifestation of a dataset or part of it. It is characterized by the following facets:

- **Identifier Facet** (associated with ***isIdentifiedBy***) : the set of Identifiers associated with the concrete dataset instance;
- **Contact Facet** : the contact information of the entity responsible for the maintenance of the concrete dataset;
- **Access Point Facet** : the access point to use for having access to the concrete dataset.

A **Concrete Dataset** is part of a **Dataset** (see ***isPartOf*** relation Fig. 2).

Actor
Actor (*Abstract*) is any entity (human or machine) playing an active role in a Research Data Infrastructure. It is characterized by the following facets:

- **Contact Facet** (associated with ***isIdentifiedBy***) : an Actor has at least a Contact Facet which permit to identify the Actor per se. An Actor can have other Contact Facets which provide secondary contact information;
- **Contact Reference Facet** : this facet captures information on the primary and authoritative contact for the resource it is associated with.

This Resource is used from **Dataset** to indicates the involved **Actors** by using specialization (see ***involves*** relation in Fig. 2).

Legal Body
A **Legal Body** (specialization of **Actor**) is any legal entity playing the role of an Actor.

Person
A **Person** (specialization of **Actor**) is any human playing the role of Actor.
Please note that a person can belongs to a legal body (see ***belongsTo*** relation Fig. 2).

Schema
Schema any reference schema to be used to specify values compliant with it. Examples include controlled vocabularies, ontologies, etc. It is characterized by the following facets:

- **Schema Facet** (associated with **isIdentifiedBy**) : this facet captures information on any schema associated with a resource. There are diverse type of schema that can be associated to the schema each one is capture by a dedicated schema facet specialization i.e. **JSON Schema Facet**, **XML Schema Facet**;
- **Contact Facet** : this facet captures information on a *point of contact* for the Schema;

- **Descriptive Metadata Facet** : this facet captures information on descriptive metadata to be associated with the schema, e.g. for discovery purposes;
- **Subject Facet** : this facet captures information on subjects associated with the schema for descriptive and discovery purposes.

This resource is mainly used by **Dataset** to evidence that is compliant with a **Schema** (see ***isCompliantWith*** relation Fig. 2).

Configuration Template

Configuration Template represents a template for a configuration. It describe how a configuration has to be realized. E.g. Used to define the accounting configuration parameters template. It is characterized by the following facets:

- **Identifier Facet** (associated with **isIdentifiedBy**) : the set of Identifiers associated with the configuration template instance;
- **Simple Property Facet** : This facet captures information on any property by a simple name-value pair.

Configuration

Configuration (specialization of **Configuration Template**) an instance of a configuration template characterising the behaviour and shape of the resource it is attached to.
The Configuration can be related to the template it derives to (see ***isDerivationOf*** relation Fig. 2).

Site

Site is an entity representing the location (physical or virtual) hosting and providing the resources associated with it. It is characterized by the following facets:

- **Identifier Facet** (associated with ***isIdentifiedBy***) : the set of Identifiers associated with the site instance;
- **Contact Facet** : There are diverse points of contact that can be associated to the site and the role of the association is captured by using a specific *consistsOf* relation e.g. to represent the manager and the maintainers of the site;
- **Location Facet** : this facet captures information on a physical area characterising the resource it is associated with e.g. the gps coordinates of the site or the geographical address of the site. This should not be confused with Coverage Facet.
- **Networking Facet** : this facet captures information on any (computer) network interface/access point associated with the resource. A site has one or more ip subnet to address the machines in the site.

Any Site is owned by an Actor (see ***isOwnedBy*** relation Fig. 2).

Service

Service (*Abstract*) represents any typology of Service worth registering in the infrastructure. It is characterized by the following facets:

- **Descriptive Metadata Facet** : any descriptive information associated with the service, e.g. for discovery purposes;
- **Subject Facet** : any subject / tag associated with the service for descriptive, cataloguing and discovery purposes;
- **Capability Facet** : this facet captures a defined facility for performing a specified task supported/offered by a given Service.

Any specializations of **Service** can: manage a dataset or its specialization such as concrete dataset (see ***manages*** relation Fig. 2); be customized from a **Configuration** (see ***isCustomizedBy*** relation Fig. 2); require another **Service** to properly operates (see ***callsFor*** relation Fig. 2);

E-Service

E-Service (specialization of **Service**) is any working service that is registered in the infrastructure and made available by an Access Point. It is characterized by the following facets:

- **Software Facet** (associated with **isIdentifiedBy**) : this facet captures information on any software associated with the resource. The one associated associated with isIdenfiedBy represent the main software enabling the E-Service capabilities (this facet is the one identifying the E-Service);
- **Software Facet** : software available in the E-Service environment that characterizes the specific E-Service instance;
- **Access Point Facet** : identify the endpoints of the E-Service;
- **Event Facet** : this facet captures information on a certain event / happening characterising the current status and the life cycle of the E-Service events (e.g. Activation Time, Deployment Time);

- **Service State Facet** : this facet captures information on the current operational state of the E-Service it is associated with (e.g. started, ready, down, failed);
- **License Facet** : this facet captures information on any license associated with the E-Service to capture the policies governing its exploitation and use.

Any E-Service or its specializations can be related with other E-Service because it: discovers another E-Service for example to check the availability (see ***discovers*** relation Fig. 2); uses another E-Service to accomplish its tasks (see ***uses*** relation Fig. 2).
Please note that both relations are specializations of ***callsFor*** relation.

Running Plugin

Running Plugin (specialization of **E-Service**) is any instance of a Plugin deployed and running by an E-Service. This knowledge is expressed by ***isActivatedBy*** relation (see Fig. 2).

Hosting Node

Hosting Node (specialization of **Service**) is any server\machine playing the role of "Hosting Node", i.e., being capable to host and operate an E-Service. This knowledge is expressed by ***hosts*** relations (see Fig. 2). Hosting Node is characterized by the following facets:

- **Networking Facet** (associated with **isIdentifiedBy**) : this facet captures information on any (computer) network interface/access point associated with the resource. It define the Network ID characterising the Hosting Node;
- **CPU Facet** : this facet captures information on the Central Processing Unit of the resource it is associated;
- **Memory Facet** : this facet captures information on computer memory equipping the resource and its usage such as the persistent memory (i.e. the Disk Space Capacity of the Hosting Node) or the volatile memory (the RAM Capacity of the Hosting Node);
- **Event Facet** : this facet captures information on a certain event / happening characterising the life cycle of the Hosting Node, e.g. the activation time;
- **Container State Facet** : this facet captures information on the operational status of the Hosting Node (e.g. started, ready, certified, down, failed);
- **Simple Property Facet** : this facet captures information by a simple <key, value> pair property worth associating with the Hosting Node, e.g. Environment Variables;
- **Software Facet** : this facet captures information on any software associated with the Hosting Node. Useful to report the hosted software such as the operating system.

Any hosting node is located in a site which provides the management facilities to create, maintains and dismiss it. This knowledge is expressed by ***isManagedBy*** relation (see Fig. 2).

Virtual Service

Virtual Service (specialization of **Service**) is an abstract service (non physically existing service) worth being represented as an existing Service for management purposes. Examples of usage include cases where classes or set of services are to be managed like an existing unit.

Software

Any **Software** entity worth being represented for management purposes. It is characterized by the following facets:

- **Software Facet** (associated with **isIdentifiedBy**) : Software coordinates which identify the Software per se;
- **Software Facet** : Apart the one connected by the *isIdentifiedBy* relation the others identify the sw in other way e.g. (Maven coordinates);
- **Access Point Facet** : identify endpoint useful for software download, documentation, source code etc e.g. links to maven artifact on public maven repositories, javadoc, wiki, svn;
- **License Facet** : the Software License characterizing its possible exploitation and use eg EUPL, LGPL, GPL, Apache2;
- **State Facet** : This facet captures information on state to be associated with the resource. State is captured by any controlled vocabulary which is an integral part of the facet e.g. Deprecated, Active, Obsolete;
- **Capability Facet** : any facility supported/offered by the Software.

Any Service or its specializations can : depends on other software (see ***dependsOn*** relation Fig. 2); be configured by a configuration template (see ***isConfiguredBy*** relation Fig. 2);
Moreover: Any E-Service runs a certain software (see ***runs*** relation Fig. 2); Any hosting hode provides its capabilities thank to a certain software (see ***isPoweredBy*** relation Fig. 2).

Plugin
A piece of Software extending the capabilities of another Software (main) and requiring the main Software to be executed. The relation between main software and plugin is expressed by ***isPluginOf*** relation (see Fig. 2).

4. Conclusion

Research Data Infrastructures are complex systems called to offer services for Research Data Management. In order to meet this goal their developers and managers as well as their constituents (systems on its own) need to be provided with a constantly update and comprehensive description of the datasets to be managed and the associated resources (e.g. other datasets, services, people, machines) that are "available" at a given point in time. Capturing this information need poses a number of challenges including to deal with the heterogeneous and evolving nature of both the typologies and descriptions of the resources of interest. This paper described both a core model and a comprehensive model to capture the information needs arising in a Research Data Infrastructure when managing Research Data.
Such a model has been used by D4Science infrastructure in the context of BlueBRIDGE and PARTHENOS european projects.

Acknowledgments This work has received funding from the European Union's Horizon 2020 research and innovation programme under the BlueBRIDGE project (Grant agreement No. 675680) and the PARTHENOS project (Grant agreement No. 654119).

References

[1] A. Bardi, L. Frosini : Building a Federation of Digital Humanities Infrastructures. ERCIM News 111. October 2017.
[2] L. Candela et al.: "Managing Big Data through Hybrid Data Infrastructures", in ERCIM News, Issue 89, April 2012.
[3] L. Candela, P. Pagano: "Cross-disciplinary data sharing and reuse via gCube", in: ERCIM News, Issue 100, January 2015.
https://kwz.me/hO7