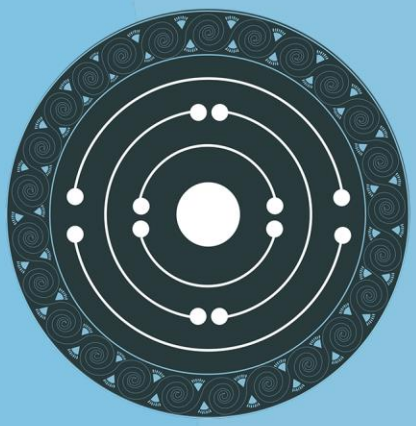

PARTHENOS

Pooling Activities, Resources and Tools for Heritage E-research Networking, Optimization and Synergies

Report on the Common Semantic Framework (D5.1)

AUTHORS (George Bruseker, Martin Doerr, Maria Theodoridou)

DATE 30/04/2017

PARTHENOS is a Horizon 2020 project funded by the European Commission. The views and opinions expressed in this publication are the sole responsibility of the author and do not necessarily reflect the views of the European Commission.

HORIZON 2020 - INFRADEV-4-2014/2015:

Grant Agreement No. 654119

PARTHENOS

Pooling Activities, Resources and Tools for Heritage E-research Networking, Optimization and Synergies

Report on the Common Semantic Framework

Deliverable Number	D5.1
Dissemination Level	PUBLIC
Delivery date	15 April 2017
Status	Final
Author(s)	George Bruseker Martin Doerr Maria Theodoridou

Project Acronym	PARTHENOS
Project Full title	Pooling Activities, Resources and Tools for Heritage E-research Networking, Optimization and Synergies
Grant Agreement nr.	654119

Deliverable/Document Information

Deliverable nr/title	5.1
Document title	Report on the Common Semantic Framework
Author(s)	George Bruseker, Martin Doerr, Maria Theodoridou
Dissemination level/distribution	

Document History

Version/date	Changes/approval	Author/Approved by
V 0.1 15.02.2017	First draft	George Bruseker
V 0.2 01.04.2017	Draft revision	Nicola Aloia
V 0.3 12.04.2017	Final Draft Revision	Martin Doerr, Maria Theodoridou
V 0.4 20.04.2017	Review of Final Draft	Sheena Bassett

Table of content

Table of Figures

Table of Tables

Executive Summary

This deliverable describes the first results of the design of the common semantic framework for the PARTHENOS project. The common semantic framework aims to build interoperability between the resources made available by participating Research Infrastructures (RI) and ERICs in the PARTHENOS project infrastructure. Beyond the technical aspects of creating interoperability, a common semantic framework implies a translation of our state of knowledge with regards to the resources offered in a cross-disciplinary environment into a common semantic representation and, consequently, entails the elaboration of an architecture for the management of the supporting common semantic framework, including work flow for the management of the maintenance and update of this common picture. The approach undertaken in this work package has been to avoid a one-off, resource market strategy for the aggregation of knowledge about resources that would inevitably fail to support a sustainable continuation of the resource pooling activity and, which furthermore, would not solve the key problem in such an effort. The key problem identified to support a useful and sustainable cross-disciplinary semantic graph of available resources is not the simple aggregation of metadata, but rather is the aggregation of metadata in such a way that the resources described in the resultant graph maintain a provenance of knowledge with regards to their epistemic status and origin. This tracing of the origin of knowledge back to its source is crucial both from a research and practical perspective. From the research perspective, only data with proper provenance can be used reliably for further research. From the practical perspective, maintaining a consistent link to the providers and curators of resources ensures that the source or target for changes in our state of knowledge is known and be contacted either for technical or manual updating of resources. The following document describes the strategy and semantic model elaborated in order to meet this challenge

1. Introduction

The PARTHENOS project was conceived in order to support the cohesion of research across a wide range of disciplines currently represented by RIs and ERICs at the European level including inter alia linguistic studies, humanities, cultural heritage, history, and archaeology. The opportunities of such an effort to create cross-disciplinary cohesion across research efforts in Europe are many. RIs already form an important plank in the strategy to support cooperation and efficiency in research across Europe. Successful RIs are a proven means of creating coordination of research efforts across borders and in helping create common policies and solutions for the elaboration of research in a specific domain. RIs play a role as connectors of people, multipliers of results and host/curators of current and past research results for their respective communities. The establishment of RIs quite naturally has begun at a disciplinary level, aiming to bring together the efforts, activities and outputs of common research communities, where the immediate result of pooled resources would be evident to participating researchers and institutions. But the maturation of RIs begins to make evident that, while they are quite rightly oriented towards and nurtured from individual disciplinary research communities in order to get off the ground and produce tangible results for their community, the natural evolution of this process is to seek synergies across research communities. This is the case for at least two fundamental reasons. On the one hand, RIs looking to pool the resources of a particular community at the European level face similar intellectual and technical challenges in terms of bringing about this consolidation of information. On the other hand, cutting edge research in many disciplines can no longer be limited to that disciplines itself. Researchers must keep abreast of and respond to development in fields outside their main research topic which nevertheless have effects on their research results and outcomes. As a result there is significant benefit to be realized from an effort such as the PARTHENOS Project, to act as a sort of RI of RIs, which will interlink these different efforts and not only find efficiencies and best practice solutions for the role of connecting, multiplying and host/curating data but will also create a new layer of interactivity not only intra-disciplinary but also inter-disciplinary in nature among the RIs and the researchers that they support.

The common semantic framework has been developed to face just this challenge: how to create a layer of interoperability between RIs which will offer an information model that will allow the integration of records from highly diverse individual RIs into a common view that

will a) allow for the exploitation of resources across the participating member RIs in a way that maintains the epistemic validity of those resources and b) create an environment for a sustainable, expanding integration of these resources according to the needs of researchers and RIs.

This document describes the framework thus proposed which includes: an information management architecture and process, a semantic model for representing RI data, and a minimal metadata proposal for tracking the identity of resources in a common semantic environment. Furthermore, working from examples arising during the mapping activities undertaken in 5.2, it provides examples and application scenarios for the proposed framework.

2. Architecture Design Concepts

This section describes the proposed architecture for managing a long-term sustainable data integration process that is domain independent and allows on-demand integration of sources according to needs of researchers while building a library of integration tools and resources enabling future researchers to benefit from previous research work. The architecture proposed in this section depends fundamentally on the elaboration of a semantic model for information integration and minimal metadata for the practical activity of establishing the identities of tracked entities in the information system. The proposed architecture, therefore, motivates the extended elaboration of a semantic model for research infrastructures outlined in section 3 and the minimal metadata for managing entity identities outlined in section 4. In contradistinction to other research infrastructure proposals, the PARTHENOS project aims to provide a generic model for the long term integration of data to any format of interest to researchers. For this reason, its central feature is not a rich data catalogue, but rather a registry (expressed in a common semantic model with specific minimal metadata) which gives a picture of resources and their keepers within the environment. It allows the tracking of past transformations of data, existing maps of data and, by extension, potential transformations of data to new standards and indices. The end result is a system which does not impose a one size fits all solution to its participants, but rather allows the generic representation of the key relationships among data, actors and services, in order to allow an organic growth of interoperability.

2.1 The Challenge to be Met

The challenge driving the design of the architecture for the PARTHENOS project lies in trying to create a generic technology for adoption by RIs. This effectively entails the design of a sufficiently abstract yet pragmatic infrastructure architecture that could be adopted by and become an attractive service for RIs in general.

It is important, then, to identify from the beginning what is meant by research infrastructure as a term and thence what we would mean by supporting one. We can differentiate three meanings adopted when referring to the term RI:

- A consortium with a business goal,
- An activity pursuing this open goal by offering a set of services (an IT supported research ecosystem),
- A centralized information service.

It is these three senses of RI then that must be supported in order to provide a complete architecture offer. That is to say that the design must be informed not by a restricted meaning of the term, which is a particular possibility of limiting the problem. The most common restriction would be to limit the support to providing a centralized information space, providing an aggregation service for making all data available in one place. A more full interpretation of the challenge would be to address also the project as such, indexing and correlating services with the information space. The maximal interpretation, however, to be covered includes both these necessary functions, but links them to active consortium and its goals, such that there is a link from the information space and services indexed to the living actors engaged in research activities, holding and curating digital resources on behalf of a research community. This entails enabling a communication space where centralized information and services can be accessed and used and where the results of these actions in terms of new results, discovery of error, amelioration or expansion of resources is progressively managed and brought up-to-date. The activity of scientific and scholarly research is a progressive project of advancing programmes and updating theories and results which depends fundamentally on a continued communication and informing of actors with regards to advances and changes. Therefore, the information architecture should aim to meet these needs.

The challenge of developing a generic infrastructure for research infrastructures can thus be maximally interpreted as the enabling of a communication space for the on-going pursuit of research goals within an agreed and explicit framework.

2.2 The Strategy Developed

The challenge to develop a generic infrastructure as described in the previous section demarks what strategies will be incomplete with regards to the needs identified. Particularly, the common minimal interpretation of a research infrastructure architecture as consisting of an aggregation service to provide a centralized information service is insufficient. This interpretation is insufficient to the challenge because it under defines the challenge leading to missing key elements of the problems to be tackled. Specifically, the aggregation approach which misses the element of supporting an RI considered as a consortium working towards research/business goals, therefore treats the problem as overly static. Typically, a one time effort is envisioned to converge information into a common space which, in order to support operationality, creates an immediate demand for an integrated rich metadata space. Such an approach has runs into a number of fundamental difficulties.

- Resources are heterogeneous by default,
- Resources are under constant update by providers,
- Developing core metadata for a large information space means either overly watering down requirements or putting too large a requirement on providers,
- The initial expense for harmonization is great and the on-going upkeep to stay up to date with sources is onerous.

The problems associated with such an under interpretation of the challenge, however, highlight the principles that should go into defining the strategy for our architecture.

At the heart of the strategy for the design of the architecture must be the recognition of the need for a mechanism to support the ongoing curation activity of an active research consortium. What is needed is not another aggregator but a centralized information service which is able to document, relate and improve the centralized information services that it supports.

Elements of this curation that the architecture must be able to support are:

- A plan to integrate complementary information (subject completeness & relatedness, gapless lifecycle/research process),
- A plan for how to start an attractive, sustainable service,
- A strategy to maintain/ improve content and scientific quality,
- Recognize that people know what data means and if that data is authentic.

That is to say that the proposed architecture should support as fundamental these aspects of the known on-going curation activities of RIs and provide a service whereby these activities can be done better. Resources should be integrated by this architecture but they should be integrated knowing that they are necessarily incomplete and heterogeneous and therefore must be connected to the living research process which aims to improve them. This means enabling these processes of curation explicitly, making space for and structuring the information aggregated such that an ongoing plan for its integration, at the level where it can usefully be integrated by active researchers, is supported. Because RIs function as structures serving a community, it must support the ability to integrate information and tools in services that will be interconnected and viable on the longer term in order to become an attractive offer to the researchers served. Data within RIs has a specifically scientific aim and therefore a main aspect of curation to be supported are the communication cycles which enable the maintenance and improvement of the quality of content of this information by linking researchers to researchers through the integrated environment. Because data and tools are the products of human actions, it is a core principle that aggregated information on resources must link these resources back to the people involved in them, in the capacity in which they are involved. Such an architecture, therefore, should aim to create a true virtual research environment in which the natural communication cycles between researchers are natively supported and facilitated in terms of efficiency and clarity.

To enable the above, we can think in terms of Quality of Service agreements (QoS) that document and make explicit the relation between resources integrated into the common information space and the participating research infrastructures that stand behind them in the research community. RIs entering into a common information environment such as the PARTHENOS project do not simply offer resources but they offer particular services of provision of data and tools. The offer is only understood and manageable if it is explicit in what the nature of the offer is in terms of the commitment to maintenance/curation of

offered resources. Therefore, at the top most level of the architecture, we must enable the documentation of these service offers themselves (not just the data and tools they offer) and the elements that make up the plan to support them. The new centralized information space provisioned, then, is not one of simply more resources but resources connected to known active sources of resource curation who make explicit and commit to their curation plans. This facility then aims to illuminate the state of resources at some point across multiple RIs offering services but also to provide the tools to understand the status of those resources in terms of how they are managed, in what format they are available, where they are indexed, what level of integration they are and thus make clear potential forward actions for augmenting integration and reducing duplication and error.

To support such an explicit, common, integrated curation process entails a high level analysis of the types of curation activity that need to be tracked to create an effective information exchange amongst actors in the common information space. As part of the initial planning for the architecture the following list of key features of curation processes was devised:

- Registering people, services, data, metadata and software,
- Deduplication of all identities,
- Copying data between hosts (including PARTHENOS cloud),
- Inviting curation of resources,
- Transforming data/metadata to standards,
- Aggregating and indexing data/metadata,
- Communicating QoS request to service providers and knowledge creators,
- Data cleaning, bug fixing, accessibility, access conditions, metadata enrichment.

This list aims to spell out the actions that RIs and their community can take within a common information space that must be traceable and supported by the proffered architecture. This list provisionally identifies the basic actions and relations that are established by such actions which have an effect on the information space and which must be made explicit in order to meet the requirements for a dynamic, human action centered information space where aggregated resources are not simply dumped into a common space but are represented in such a way that the participating actors (RIs and their communities) understand the provenance of resources and the complex relations that hold

between them which affect their usability and reliability and further offer the ability to take action to improve and update the contributed resources.

The strategic dead end of requiring a one off rich metadata integration is avoided by limiting the requirements for the central information space to a representation not of the complete contents of resources provided but rather to the entities and relations necessary to represent the above curation situation. The common information space provides a picture of the state of curation at the present moment as well as the intentions moving forward for further integration. The common information space is a picture of all the resources that are available and just the data required to track their interrelations. Rich metadata integration for specific purposes is done at a second stage after having analyzed the resources available and understanding their completeness, interrelation and relevance. From the central information space, areas of compatibility can be identified and targeted integrations expanding and enriching areas of interrelation can be developed.

2.3 Architecture Proposed

The architecture proposed then provides a set of components that will enable the cross-domain registration of and curation of central information services and the resources they provide.

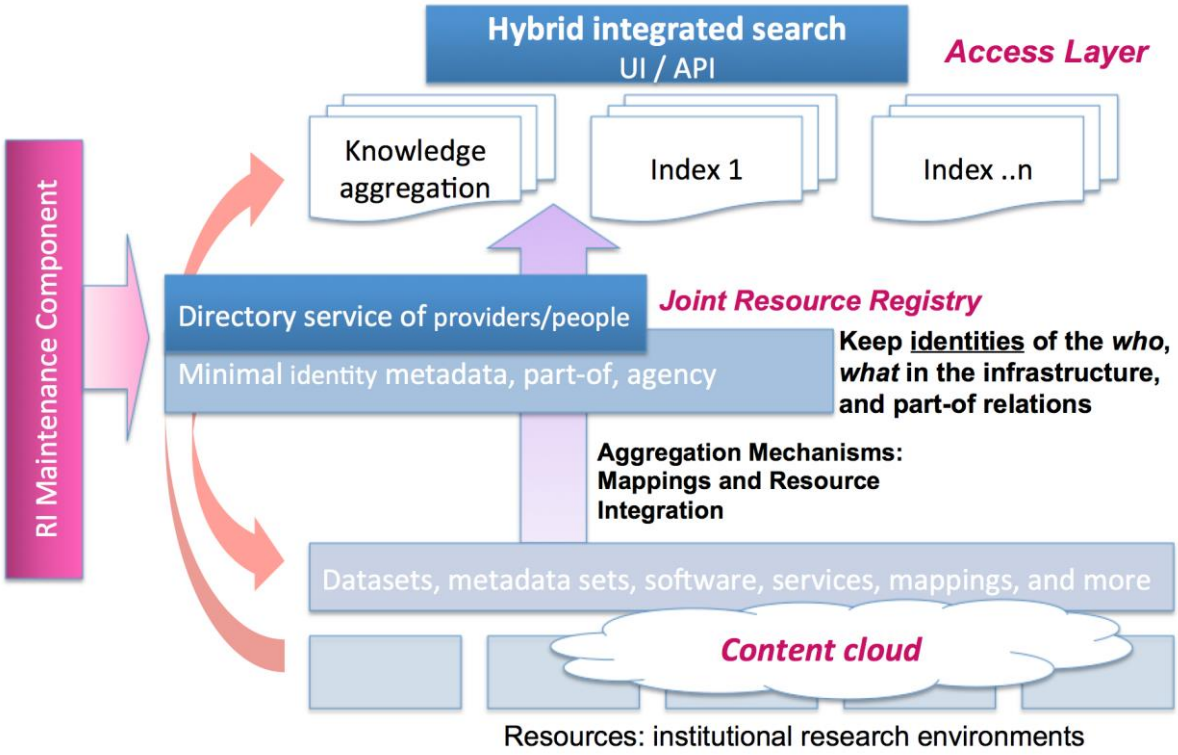

Fig. 1 Illustration of Conceptual Architecture for PARTHENOS

The components envisioned for this architecture are:

- a **Registry**: the common information space that keeps track of metadata relating to central information services and the resources they provision within the common information space using a common information model.
- a **Cloud/Content Layer**: a layer in which data and original metadata provisioned by the central information services is made available and related to the registry. This layer gives access to original data and metadata that forms the source of the registry.
- **Indices**: an open set of potential indices on the information available in the content layer which provides specific aggregation, indexing and transformation of relevant datasets.
- A **Query System**: a component working on top of the indices and registry layer to provide hybrid search across one or many indices according to user aims/needs
- An **RI Management System**: a component for the management of the common information space, that allows for the tracking of RI maintenance actions, enables the maintenance of the Registry, Content Layer and Indices in consistency, while improving QOS of integrated central information services.

2.3.1 The Registry

The registry is the common information space into which metadata from contributing RIs with regards to their central information services are transformed, in order to give the overall picture of the state of curation of resources at any one time. The function of the registry is just to provide the present state of affairs regarding the ongoing curation process and products of participating RIs in order to aid this process. Specifically, the registry is a semantic graph describing the essential relations between actors, services and resources (data and software). The knowledge that it should support are the identities of who participates, what is offered and the part-whole relations between objects in the information space.

In section 3 we will go into detail on the semantic model proposed for elaborating these relationships. Here we can indicate that the semantically relevant relations and entities for representing this state of affairs were identified at a high level as: services (with subdivisions of hosting, curating and e-services), data vs. software and persistent vs. volatile digital objects. This high level picture has the advantage of sufficient generality to be able to represent highly diverse situations in terms of information curation and curation states while providing enough precision that semantically relevant differences for understanding the content and state of resources represented is provided. By adopting such a high level model, the registry need only require a light set of minimal metadata, detailed in section 4, for the registered entities which focusses only on being able to support the identification of a given resource (as being the same or different than another, and where it can be found). The business of rich metadata integration is left to more specific, goal oriented activities which can be elaborated in various types of indices. The resultant “meta metadata” that forms the semantic network generated in the registry is maintained always in relation to its source which it does not aim to replace, but to provide an up-to-date pointer to along with vital information about the present state of that resource.

The registry purposefully contains only the present state of knowledge of resource management. Historical provenance data such as registration actions, adding, copying, removing and other knowledge creation events) is paradata which is linked to by the registry alongside the data/metadata that it refers to, without ingesting it. Thus, the registry

is maintained as a clear, separate picture of present knowledge, but allows investigation of the actions which led to this state through search in the content layer to the rich metadata.

2.3.2 The Cloud/Content Layer

The content layer of the system contains the distributed resources that are described by and registered to the PARTHENOS Registry. The content layer feeds the registry picture and the meta metadata elements in the registry are always linked back to the source from which they derive and which they describe.

The content layer is the source for all changes to the registry. The registry itself is never directly edited, but rather changes in information are executed at source level and transformations to the registry are re-run, documenting change actions in order to update the present state of knowledge. This follows from the principle described above that knowledge provenance is guaranteed by the relation between human actors and information sources they declare themselves responsible for.

All content in the content layer has at least one mapping to the registry which is documented and linked to the resource in the semantic graph. This element is crucial in order to allow the on-going update of the registry. This allows for continuously updating the registry with the present state of knowledge as it is changed over time at source. Furthermore, if the metadata structure at source changes, the saved mapping to the registry can be modified accordingly and a new transform run to properly represent the resource according to the information structure of the new schema.

The content layer contains both data and their metadata treated as data without distinction. Metadata is just another kind of data ‘about’ other data. Authority files (gazetteers, thesauri, person lists) used in harmonization processes are (volatile) data as well. Mapping files used to transform metadata to the registry are also kept in the content layer.

Each content object can be multiply related to different metadata objects. Mapping files providing transformation rules between metadata formats enable the possibility of generating additional metadata formats for known, mapped metadata formats. Thus, compatibility and integration is enabled in an on-demand fashion, by making available both

formats presently available for use in services while maintaining alternative metadata formats and the open potential for generating additional formats.

Data is maintained in a distributed fashion where partners maintain their own data hosting facilities. The content layer can also contain a cloud service for ingesting and hosting data resources as well in the case where hosting facilities are not available on the side of the provider.

2.3.3 The Indices

An open set of indices can be made for targeted types of retrieval purposes and/or information spaces which can be made available to the user through the query system. These indices are generated making use of the registry in order to identify data and metadata suitable for aggregation and indexing activities and generating/updating the specific index. Types of index that could be made available at this layer include full text retrieval, image/sound, CBRI, triple store, GIS systems, PostGIS amongst others. They could run over all or part of the data space described by the registry. Triple stores containing detailed mappings of highly related resources are an important form of index that allow highly accurate semantic retrieval across selected heterogeneous sources.

Each index is the result of a curating service provided by PARTHENOS which decides a domain of indexing and indexing type. The index is thus itself registered in the registry along with the dataset it generates.

The indexing services are considered open ended and can be generated on demand according to the needs of the research community and made accessible as possible retrieval tools for end users.

2.3.4 The Query System

The query system proposed serves to dynamically meet the needs of end user communities by enabling hybrid research across the registry and one or more indices generated by the community. The function of this component is to provide end users discovery and querying facilities by providing to them the correct mixture of overall resources view (registry) with specialized indices (found in the registry) in order to support

their research goals. A general query system accessing the registry and certain generic index types across the information space such as GIS and image search could provide basic access to the aggregated resources for general information discovery needs. Specialists searching specific topics could load up theme related indices, especially deeply integrated semantic networks in triple store indices to discover information related to a certain problem space. The specific question of querying provenance of knowledge can be addressed by the integration of a specialized index for such paradata actions and linking this to registry results.

2.3.5 The RI Management System

The final component envisioned for this architecture is a control mechanism by which to keep the system up-to-date This RI Management component would have the function of allowing the supervision, planning and tracing of workflows and progress with regards to the RI maintenance actions described above. This component is intended for use by the maintainers of the integration common information service. It works on top of the registry reading the documented relations, and allowing for tracking these against the agreed QoS plans for integrated services. The dashboard would read these relations and identify areas in which to take action as well as generate and read metadata on what actions have been taken and to which resources they relate. By reading the contents of the registry against information on services available from the content layer, it allows for tracking when updating of the semantic network needs to be undertaken. It also allows for maintaining the consistency of indices dependent on resources mapped in the semantic framework by keeping track of changes and highlighting when re-indexing or other information updates are required.

3. PARTHENOS Entities Model

A semantic framework exploiting an ontological modelling of data structures considered as propositional statements about the world is proposed as the means for creating the data expression which is capable of supporting an agile framework for the representation and management of cross-RI metadata about resources. Such a model is necessitated by the nature of the data integration challenge inherent to the PARTHENOS Project. There is a

double basic requirement to create a common expression of data which by its nature will necessarily be heterogeneous.

For many reasons it must be taken as a foundational condition that the data gathered from RIs are and will remain as heterogeneous in form. These reasons range from the theoretical to the historical, the technological and the practical. The theoretic focus of particular RIs on particular disciplinary complexes means that the data produced will be tailored towards that community. Historical and technological reasons will also militate against any assumption of data standardization at source. Different RIs at different times will necessarily have had to have implemented particular data systems at particular time to meet their unique criteria at that date, according to the technology available, in order to meet their data management needs. These will necessarily be multiple. This leads to a final, practical reason why source data from RIs has to be assumed to be heterogeneous by nature. Even assuming standards that would hold across all RIs, legacy systems and traditions of research will necessarily exist, while new systems will come online continuously in line with the direction of their individual research. Therefore, from a practical perspective, it cannot be imagined to harmonize all data to a single standard, giving the limits of resources in terms of time, money and available human resources.

The development of a semantic model does not aim to initiate a process with the aim of replacing the source data structures, but rather has the goal of providing them a common expression, within some bounds, that will allow for data that is produced and will continue to be produced in heterogeneous forms to have a well defined semantic re-expression which gives them a certain level of interoperability with other datasets with the same semantic re-expression. A semantic model will be valid for some domain provided it is able to offer a sufficiently accurate re-expression of the source data structure, that the original propositional meaning encoded in the source is not lost when re-expressed in the new representation. A semantic model adds value for discovery when it is expressed in an ontological model either newly developed or pre-existing which offers higher level general classes and relations which generalize on the specific data structures and allow for querying specific data with both highly general and subject specific reasoning patterns. The semantic model does not propose or prescribe a documentation pattern, but rather should ‘simply’ have the capacity to re-represent its target domain in an accurate fashion. This marks a crucial difference between a semantic model proposal for integrating data

and the idea of a standardized data model often proposed in order to create a common core catalogue across heterogeneous data sources. The semantic model does not require any particular expression in the source data to be mapped to it, but simply provides a means for re-expressing it. This means that the semantic model poses no practical barrier to integration from the point of view of demanding a complex and rich data structure equally filled out for all participating data providers before data can be meaningfully integrated. Rather, data should be re-expressible at the level of granularity that it is available with the possibility for its continuous improvement as further knowledge is gained either by source data providers at source or through the integration process which may reveal through the cross correlation of information, further particulars about any particular given data point.

Key to the elaboration of a model to support the PARTHENOS Project is an understanding of what information RIs are interested in and do collect about what real world objects and to understand precisely the relations that they are interested in tracking and following amongst these objects. It is first to the definition that the elaboration of the PARTHENOS Model turned, in order to then carry out the work of conceptual modelling over the relevant data structures and identify the key entities and their relations about which RIs gather information and which they would want to be able to advertise, share and integrate in a cross-RI environment.

3.1 Model Scope

Investigation and reasoning into the function of metadata gathering for resources within RIs, poses a fundamental question as to the function of an RI, and what it aims to support. Considered from an information management perspective, the easy answer is that the RI provides a sort of supermarket stock of information which might be about datasets or software or anything else, which is made available in a digital environment for use by researchers. Such a picture, however, fails to tie together the informatics mission of an RI with its overall research goal and practical brokering function within a research community. The representation of information in a RI goes well beyond providing stand-alone resources, and rather aims to describe and make accessible a whole ecosystem of relations that come to be within the scope of research initiatives. Research Infrastructures deal fundamentally with scientific data and scientific tools. This changes entirely the scope

of relations to be addressed in conceptualizing a data model to represent the data available in RIs. Scientific data, in contrast to final, self-contained publications, are components of a dynamic process aimed at generating, evolving and consolidating knowledge. RIs are facilitators of the scientific process. That is to say, they stand as facilitators of an on-going process of knowledge generation, evaluation and correction/extension. It is not simply the results of such a process or the individual elements of this process that an RI attempts to describe, but rather they attempt, within the limits allowable, to create an efficient representation and relationship between data discovery, creation, revision and deposition inter alia. For this reason, the scope envisioned by the semantic model needs to account for the ability to connect and represent these diverse processes of knowledge generation. Data cannot be divorced and abstracted from the processes where they are generated, and from the researchers that execute those processes in their daily activity. The RI provides support for a knowledge generation trajectory that manifests a tight relation between data, software and services. Therefore, the challenge is to move the scope of the representation of RI data from an “open to the public” (lib-centric view) where objects are described in an isolated, flat and static way, to a lab-centric view that constitutes a priori an environment of human collaboration with the collective goal of advancement of knowledge. This is already the function of RIs but the challenge lies in the thorough and consequent conceptual analysis that is required to lay the conceptual and architectural foundation to semantically represent and support this paradigm.

3.2 Model Development Method

In order to arrive at a semantic structure to support the rich relations which are created in the elaboration of research processes, the semantic model was developed following the analysis of the data structures of the central registries of the participating research infrastructures of the PARTHENOS Project. The empirical development principle is crucial to the development of a semantic structure that will be true to the actual world of discourse held in the domain to be modelled. The initial evidence base for the elaboration of the PARTHENOS Entities model was gathered by the activities of T5.4 and reused in the mapping activity.

The summary of these sources is outlined in the following table:

Providing RI Name	Providing RI Dataset	Dataset Type
CLARIN	VLO Observatory	Linguistic
Meta-Share	Meta-share Registry	Linguistic
LRE	LRE Map	Linguistic
Dariah GR	Dyas Organizations and Collections Registry	Cultural Heritage
ARIADNE	ARIADNE Catalogue	Archaeology
CENDARI	CENDARI Registry	History
Lifewatch Greece	Metadata Catalogue Registry	Biology
Cultura Italia	CulturaItalia Registry	Cultural Heritage
Cultura Italia	Muse-D	Cultural Heritage
EHRI	EHRI Registry	History
Huma-Num	Nakala	Humanities
Huma-Num	Isidore	Humanities
DASISH	TERESAH	Social Science, Arts and Humanities
National Science Foundation	Open Metadata Registry	Linguistic

Table 1. List of Initial Datasets Considered in Generating PE Model

To have a sufficiently representative evidence base from which to model research processes and their outputs at a level of generality that would be adequate in cross-domain research, it is necessary to begin from a sufficiently heterogeneous overall set of RIs in order to identify the common patterns across disciplines. The data gathered in T5.4 provided the coverage of data structures from diverse fields in order to support a generalized modelling practice.

In addition to empirical evidence from data structures, the conceptual modelling process requires input from researchers and practitioners in the domain of concern in order to understand the types of questions that they need to ask of their data in order to maximize their use of it. In this regard, the conceptual modelling task was strongly fed by T2.1 which was concerned with user requirements gathering across infrastructures. Using the Cockburn method, D2.1 gathered hundreds of use cases of researchers wanting to

interact with, operate on, annotate and store data which served as an input to conceiving of the types of relations that needed to be modelled in order to support such queries.

The model development is predicated on a cycle of research, modelling, validation and further generalization and elaboration of the model.

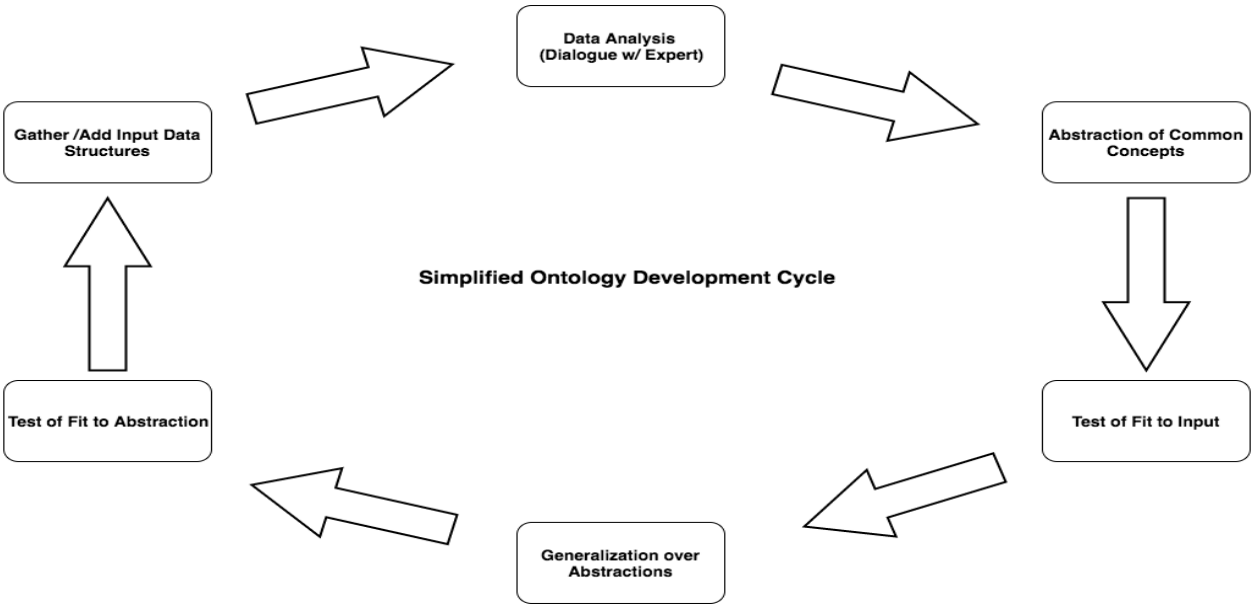

Fig. 2 Simplified Ontology Development Cycle Diagram

For the purposes of research into the provided data structures, analysis focused not on the particular semantics of disciplinary fields, but rather on the high level structures that support the discovery, use/reuse and publication of existing resources in research environments. This process was undertaken, as per the methodology described above, in close collaboration with the experts involved. This entailed dialogue with and feedback from managers of RI infrastructures as well as taking on board the input of the user needs described in D2.1. The resulting analysis produced an abstraction of a series of high level entities for describing and identifying the relations between a set of resources identified as: projects, services, datasets, software, actors and knowledge generation processes.

3.3 Identification of High Level Entities and Relations in RI Discourse

In this section we provide a brief overview of the initial identification of the high level entities and relations of interest in an RI. These high level entities were first detected

through the analysis of existing data structures. This section provides initial specifications of the kinds of entities and relations identified and an initial outline of their ontological nature. This is the groundwork from which to proceed to a formal modelling process as described in section 3.4.

- A **project** is defined as an ongoing effort maintained by some group specifically formed to pursue a certain aim over some period;
- A **service** is defined as the continued, declared willingness and ability of an actor to execute on demand by a client certain activities of specific benefit to the client;
- A **digital object** is an information object represented as sets of bit sequences, which can have either a persistent or volatile nature;
- A **dataset** is a set or collection of data, records or information that is kept as a persistent unit of information in the knowledge generation process;
- A **software** is an artefact that can be executed on a computer to perform specific operations;
- An **actor** is either an institution, a team or an individual person that participates in the research infrastructure as partner providing data and/or services;
- A **knowledge generation process** represents the workflow of the processes used to produce specific datasets.

3.3.1 Project

We define a project as an intentional activity undertaken by a group formed for the specific purpose of carrying out the stated goals of the project. The identity of a project relies on its connection to the specific group that declares and maintains its will to carry out the specific project aims. A project comes into existence with the formation of the group which intends to carry out its aims and its declaration of commitment to achieve them. A project continues to exist, regardless of the internal composition of the particular group so long as the group remains constituted with the aim and the will to attain its aims. The existence of a project does not depend on continuous activity but rather on the continuous will to support the project. A project entails the support of any number of activities that are initiated in order to realize its aims. These activities form part of the overall project.

A special case of project is that of the Research Infrastructure Project. We specifically identify this case for two basic reasons. First, there is a polysemy in reference to Research Infrastructure which must be clarified in the information structure in order to accurately describe the present state of affairs. A Research Infrastructure can mean either a) the actor/group dedicated to support and maintain a project, b) the overall research project that is supported by this actor/group or, more vaguely, c) the collection of resources made available by a research infrastructure. The correct entity of reference must be identified in documentation to correctly describe the situation. Second, the nature of a Research Infrastructure Project is more specific than a project in general, entailing the specific organization of a Research Infrastructure Consortium to maintain this activity. Further, the specific nature of the Research Infrastructure project is to support not just general activities but to offer services to some research community.

3.3.2 Service

We define a service as the continued, declared willingness and ability of an actor to execute on demand by a client certain activities of specific benefit to the client. The identity of a service, therefore, depends on the individual actor, the type of activity and/or the type of product of such an activity. An instance of a service begins to exist with the ability and willingness of provision by the actor and ends when either one permanently ends, i.e., the ability may temporarily be interrupted, such as the actor being on vacation or a machine is on repair, without meaning that the service as such has ended. The service includes all auxiliary abilities of the same actor to execute the respective activities, but not services provided by third parties in the course of his service provisioning.

Services considered within the context of an RI project have a very broad potential range of forms. Working from the provided data structures we identified the following repeating high level forms of service.

Hosting services are instances of service whose specific offer is to hold and provide access to some object(s). Hosting services do not necessarily entail an e-environment of service provisioning. Hosting can simply be the case of providing holding and access facilities under terms and conditions for any given type of object. Therefore, defining a high level notion of hosting service is requisite in order to be able to handle cases of hosting that are not made available through some e-environment.

Curation services are instances of service whose specific offer is to manage and organize a collection of object(s) according to some stated plan. As with hosting service, curation services need entail no e-service provision. A typical example of a curation activity could be the management of a physical collection of objects, organized according to a particular plan.

E-services are instances of service that entail the provisioning of an e-environment for accessing and using the proffered service. An e-service is a sequence of states of activation of a software system installed on some particular machines offering facilities on the Web that reacts on mechanical requests through the Web or a similar electronic communication network with receiving data, manipulating it and sending it back. The term e-service describes a form of communication with an installed and running software system of whatever kind.

Looking at the empirical use cases of service, we can further distinguish between services according to the types of objects handled and/or the kind of tools employed, such as provisioning access to software, or datasets, enabling their running and of course the management of physical resources. Of particular valence and importance in the world of RIs is the provision of service related to datasets and software.

The identity of a software service depends on the particular processing software it holds/offers or runs, the actor maintaining the service active, and the logical communication address for issuing requests to it. These require clear ontological distinction in the proposed model and documentation in an information system.

The identity of a data service is given by its offer to provide or curate a dataset. The particular combination of the offer with regards to a dataset requires clear ontological distinction and minimal documentation in an information system for tracking this activity.

Services are not necessarily directly expressed in local data structures but are implied by the data representation in registries. The representation and documentation of services, is nevertheless crucial to represent when creating an integrated cross-disciplinary semantic network which follows the knowledge generation process. The identity and documentation

of a service gives crucial provenance of knowledge information not only with regards to the setup of the particular service and its potential effects on the objects it handles, under what goal/plan, but also in order to connect researchers to creators, holders and curators of data. This connection is what makes possible not a static, disconnected collection of research products, but enables a living research community to interact through an integrated information space, to follow research results to their grounds, to re-run questions with new evidence, and update other researchers with regards to needs for emendation or reconsideration of previous results based on new evidence.

3.3.3 Digital Object

Within the context of a RI environment, scholarly outputs and tools are often digital in nature. It is, therefore, necessary to understand the identity of the digital object as such in order to elaborate on the different types of digital objects that are used in the research process.

Previous research into digitization processes for the CRMdig ontology had defined instances of digital objects as being, “identifiable immaterial items that can be represented as sets of bit sequences, such as data sets, e-texts, images, audio or video items, software, etc., and are documented as single units.” CRMdig 3.2

Missing from this definition and made clear from our input data structures, however, lies an important distinction between the digital object considered as an established object and the digital object considered during its development and ongoing maintenance. This is of particular concern in the research process where both datasets and software may be referenced either as stable, complete objects or as on-going works which are continually changing due to on-going curatorial and development work on the object.

We, therefore, introduce a general distinction between persistent and volatile digital objects in order to facilitate the accurate referencing of these two different cases. The overall definition of a digital object remains correctly described by the definition provided in CRMdig 3.2 but must be supplemented by the following distinction.

A persistent digital object is the result of a distinct creation moment which allows the object to be documented as a whole and verified by its bit-level encoding identity. Characteristic

examples of such an object include particular software releases and published, complete datasets.

A volatile digital object, on the other hand, is in a process of continual potential change. Such an object could be a live database in which curators make updates to the data, the outputs of sensors that feed a continuous stream of data to a data store, or a software programme that is under development and not some particular version. In this case, the digital object has no bit-level encoding identity at any point by which to verify reference to the same entity. In the case of the volatile digital object, its identity can be ascertained by reference to a curation plan which indicates the overall goal of the process of management of this object and can be referenced by the snapshots of the data stream which give an identity to the volatile object for some time span.

3.3.4 Dataset

In the context of the scientific/scholarly process, datasets are sets or collections of data, records or information that constitute distinct units of information in the knowledge generation process. They contain propositional information about the empirical world susceptible to critical analysis and potential falsification by an empirical scientific/scholarly process. The dataset, as a digital object, can be considered either as a persistent or volatile digital object.

3.3.4.1 Persistent Dataset

A dataset is any set or collection of data, records or information kept as a persistent unit of information in the knowledge generation process from primary records up to any level of aggregation or integration.

The identity of a dataset is given by its content on the bit-level of encoding and its provenance. Since large datasets have a very small chance to be “reinvented” with another meaning, it is often practical to base the identity of a dataset on the content only, and apply a respective disambiguation of provenance only in case of obviously accidental identity. Different versions of a dataset are regarded as different datasets. Their relation

should be defined by metadata describing the derivation process, rather than by version numbers.

In general, a dataset may be integrated from different sources of provenance, such as a corpus of inscriptions compiled from different publication or a snapshot of a complete digital library. The integrated dataset may preserve the units of information of the source from which it has taken components. The content of knowledge organization systems, such as gazetteers, author lists, thesauri and formal ontologies of terms at a particular point in time, fall under datasets.

3.3.4.2 Volatile Dataset

A volatile dataset is a dataset that is changed without notice or necessarily archiving of the intermediate states. That is, a volatile dataset is a linear sequence in time of persistent datasets replacing each other, bound together by a common information goal, subject coverage and curation. Volatile datasets are typically whole databases or mash-ups with active data feeds. A volatile dataset is only identified by the persistent identifier and can only be verified by the responsible actor. Reference to its content is by snapshots. Snapshots are persistent datasets that are uniquely related to the volatile resource by time-stamp. In case more than one snapshot of the same volatile dataset at exactly the same time exists, all except one must be false. The responsible actor may be able to identify the correct snapshot.

3.3.5 Software

We define software as an artefact that can be executed on a computer to perform specific operations. In particular, software is the necessary information to process datasets algorithmically and to transform or integrate datasets in a collaborative infrastructure. The identity of a software instance depends on its content on the bit-level of encoding and its provenance. The validity of the results produced by the software's application depends categorically on its algorithmic correctness. We also include in this category all data structures and formal ontologies that are used to configure the behavior of the software at an infrastructure component level. As a digital object, software can be further considered in its persistent and volatile nature.

3.3.5.1 Persistent Software

Persistent software is just some artefact that can be executed on a computer to perform specific operations which was created at a unique moment, can be documented as a whole and verified as the same object by its bit-wise identity. A software release is characteristically defined as an instance of persistent software. The software release begins to exist with its provision by the actor who is responsible for producing it.

2.3.5.2 Volatile Software

Volatile software is an instance of an artefact intended to be executed on a computer to perform specific operations which is presently being developed and expanded. Not having a bit-level encoding identity, it can be confirmed as the same object by reference to the curation action upon it that gives it its intended function and referenced by its releases or backup/snapshots.

2.3.6 Actor

An actor is an individual or a group that exercises agency in the knowledge generation process, for which they are responsible. With regards to the specific case of research infrastructure activity, typical examples of such agency might include the offering of some type of service, the maintenance of a project or the creation, modification of datasets/software or their use in some research context.

In the case of RI environments, of particular importance to document and single out are teams, considered as groups of actors formed for a specific purpose, and RI consortiums as a further refinement of the notion of teams, formed for the specific purpose of running RI projects.

2.3.7 Knowledge Generation Process

One of the key challenges stated in the scope of the proposed model is to enable the tracking of the knowledge generation process considered as an ecosystem of actors using services that provide datasets and software so as to discover, generate and publish research results back into the network.

A knowledge generation process, then, represents the workflow of the processes that are used to produce a specific research result. Such processes are complex and may involve both automatic and principled intellectual and manual procedures for dealing with research objects and documenting them.

The knowledge generation process is, therefore, not identifiable as a single class or relation but must be broken down into its components. To enable the deep modelling and then tracking of the knowledge generation process, a further analysis of the specific actions that characterize the research processes of different disciplines would have to be analyzed to provide a general model. This goal is set for the second iteration of the PARTHENOS Entities modelling.

That being said, the constellation of projects, services, datasets, software, and actors and their interrelations can be considered already a high level model of the current state of who holds datasets/software/objects and who curates, manages and modifies them. It therefore provides a high level entry point by which to trace the provenance of knowledge by identifying the network of relations between research and the communities that produce them. It should enable the connection to persons responsible for datasets and software either as creators, keepers or managers and give a picture of the kinds of resources available in the cross-domain environment and their compatibility with services, especially those providing specialized software for automated workflows. This already provides a high level entry point to the knowledge generation process by providing the top level analysis of where to find and who to communicate with about data.

3.4 Model Description

On the basis of the initial analysis carried out above, a stage of formalization was initiated in order to declare a formal semantic model for the objects of discourse in RI environments. The initial formalization of the model was carried out in two steps. First, the entities and relations identified above were represented as classes and properties in an independent ontology. The different basic relations required to represent the realm of discourse were outlined and then classes added and expanded in order to handle the specific potential situations for documentation. Second, the proposed model was re-analyzed in relation to the CIDOC Conceptual Reference Model. The model was

harmonized with CIDOC CRM after an analysis of the scope of CIDOC CRM, which could safely stand as a super-model for the proposed ontology given its scope to support research into the human past. The adoption of CIDOC CRM brought two crucial added values to the proposed model. First, it offered an already standardized ontology with a robust system of reasoning over the relation of objects/concepts with people through encounters in events traceable through space and time. Second, as an ISO standard it provides a general, recognized framework for data integration which guarantees the compatibility of data mapped to PARTHENOS Entities with the broader CIDOC CRM encoded data world. In the sections that follow, we will describe the formalization of the definitions adduced above for the top level entities in research infrastructures as well as the harmonization that was made between these proposed classes and relations and the existing CIDOC CRM model. What is proposed at the end is a CIDOC CRM compatible extension called CRMpe. The full specifications for this model in its latest version are provided in an annex to this document. Below, the logic behind the modelling decisions is laid out and can be used as a reference resource for learning the model and understanding how to use it in mapping activities.

3.4.1 Project

The data outputs, software tools and services used in the knowledge generation process can generally be sourced to a more general activity of some group of actors who initiated and carried out a research project. It is therefore a priority to enable the documentation of such projects and their relation to the other entities within the RI world of discourse.

Three basic relations of interest in RI environments motivated the declaration of a project class.

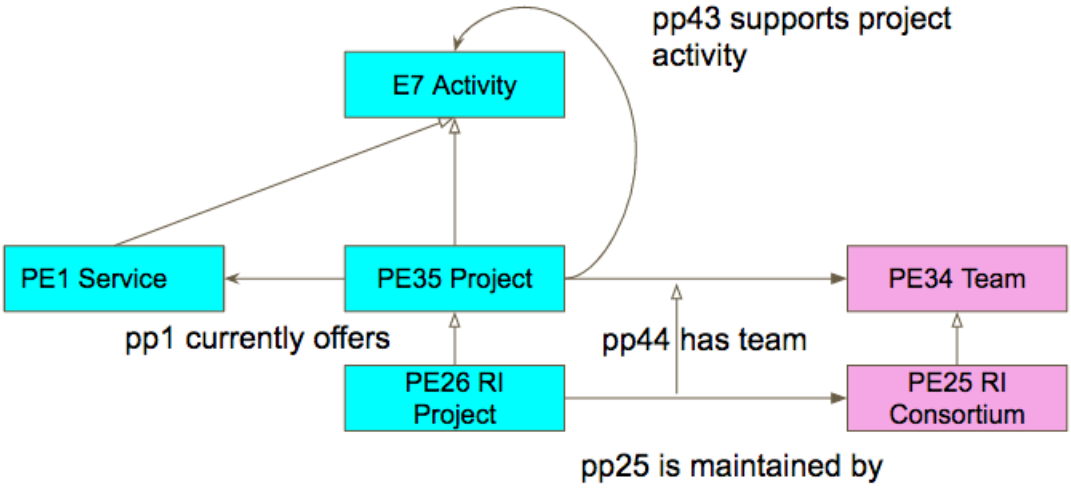

Fig. 3 Model Illustration of basic relations of Project Entity

First, there is the interest to find who runs and supports a project. This is a relation of agency between some group and some set of activities considered as a whole. We need to document the connection between a team formed for a dedicated project purpose towards the support and achievement of that project's aim. Therefore, classes were declared for documenting a project (**PE35**) and for the team (**PE34**) that supports this activity joined by the relation *PP44 has team*.

Within the realm of research infrastructures, the interest of researchers is first to discover what services are made available by the RI in order to understand if they offer capacity to the meet their needs. We therefore declared a relation *PP1 offers* holding between **PE35 Project** and a new class (described in detail below) **PE1 Service**, destined for the documentation of instances of services.

Lastly, a relation was required to enable the open ended sub-division of the project into finite activities undertaken in support of the overall project. For this a relation had to be declared between a project and its parts. This relation required relating the **PE35 Project** to a more general documentation unit for activities, which was taken from CIDOC CRM (**E7 Activity**).

A more specific relation holds between an RI Project (**PE26**) and the RI Consortium (**PE25**) that supports, specifically documenting the relation of maintenance held by the

consortium (*PP25*). That being said, these classes and relations can be considered special cases of the general pattern of relation between project and team and are therefore declared subclasses of the latter.

Harmonization to the CIDOC CRM involved considering the nature of a project. Generally, as an action in time it fits to the temporal classes of the CRM. Further analysis, places a project as a sub-class of **E7 Activity** which are intentional, defined efforts in time at some place by some actors that result in changes of state in the world.

Examples of the use of this modelling pattern would be to document the CLARIN Consortium as a whole as an instance of **PE25** in relation of maintenance (*PP25*) to the CLARIN Project (**PE26**). The CLARIN Project (**PE26**) would have the relation *PP1 offers* to instances of service that it offers such as the Virtual Language Observatory and the CMDI.

3.4.2 Service

As environments for the advancement and management of resources, it is a central, if often indirectly documented feature of RIs, that they provide services to their communities.

A service class, **PE1 Service**, was declared in order to be able to document a specific type of activity which is a willingness and ability to carry out some action for another. The character of service is differentiated because its unity is given to it by this character of willingness and ability. Temporary stoppages of service do not invalidate a service but are inevitable parts of service offer.

The basic relations that we wish to enable the researcher to discover through the declaration of such a service class are its relation to projects, its state of existence, by whom it is offered, and the competency of the service to do something. The latter two relations required the declaration of new relations

PP2 offered by: this relation enables the tracing of which agent is to be contacted and understood as the support of this service

PP45 has competence: this relation enables the user to understand what the service is for and thereby find appropriate offers. In order to differentiate the kinds of competence in a controlled manner, this relation motivates a new class **PE36 Competency Type**.

With regards to harmonization with the CIDOC CRM model, the nature of service is temporal and brings out a change of states affairs in the world, so is properly modelled as a subclass of **E7 Activity**. The resultant general service modelling pattern therefore is as follows.

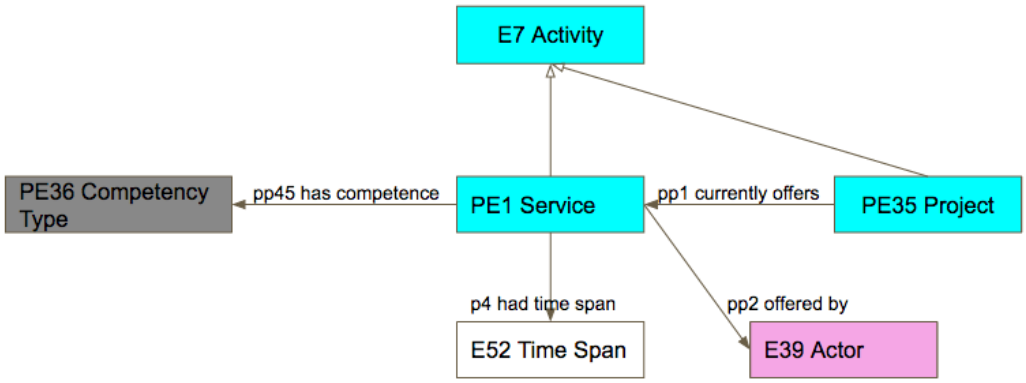

Fig. 4 Model Illustration of basic relations of Service Entity

As indicated above, research into the kinds of services provided differentiated the service offers of agents into three major types, which are declared as sub-classes of **PE1 Service** in order to support searching the relationships of interest to researchers in terms of kind and effect of service activity. Specifically, the high level categories were differentiated as:

- **Hosting** as characterized by the service provider's willingness to hold and return on request an object.
- **Curation** as characterized by the service provider's willingness to prepare and maintain a plan of selection over a group of objects
- **E-Service** as characterized by the service provider's willingness to provide an e-environment for the provisioning of services

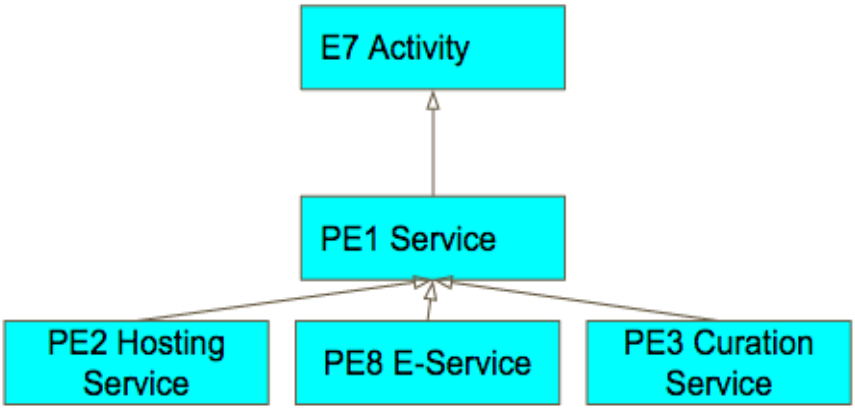

Fig. 5 Breakdown of Basic Service classes in CRMpe and relation to CRM

Each of these sub-classes is declared in order to support the distinct pattern of relations that it entails which are further discussed in the immediate following sections.

3.4.2.1 Hosting Service

Instances of hosting services have the additional specification of helping trace who holds some object, when they hold it and enables the tracing of different holdings of the same object where they exist. The identity of a hosting service is further characterized by the type of object that it holds. Differentiating the type of object held helps identify the precise kind of service that a researcher wants to find.

The key relation then to declare is one of hosting:

PP4 hosts object: this relation is declared in order to enable the tracing of some instance of **PE2 Hosting Service** and any object in general (using the CRM class **E70 Thing**)

This allows a most general specification of hosting, when we know the characteristic of the activity is to hold and provide access to an object, but we cannot further specify how. It further provides a general relation to query on to find all specific instances of hosting.

We then declare a series of sub-relations of the relation *pp4 hosts object* which allow the identification of more specialized forms of hosting. These are:

PP6 hosts digital object: this relation is declared in order to enable the tracing of some instance of hosting with any digital object in general. Since this is a distinct form of hosting, it motivates the declaration of **PE5 Digital Hosting Service** as a subclass of **PE2 Hosting** and relates this to the appropriate class for any digital object in general (adopting the CRMdig class **D1 Digital Object**).

The functionality of this relationship is to enable the expression of instances of holding of digital objects in general, where we are not able to specify what type of digital object may be hosted. Furthermore, it allows for individuated searching for all digital objects regardless of subtype. The holding of digital objects significantly changes the nature of the hosting activity since a digital object as a conceptual object can be repeated and hosted by many services, while the hosting of physical holdings is limited to a single instance.

It is important to note that the identity that is defined here of a service hosting some digital object does not imply that an e-environment is provisioned giving an online access to the hosted objects. Digital hosting only entails the will and ability to hold and provide access to a digital object. The nature of the storage is not indicated in this relation. The hosted object may or may not be provisioned with an e-access environment, which would have to be determined by other relations. Digital hosting should not therefore be confused with e-provisioning, which will be a function of the relations related to **PE8 E-service**.

PP8 hosts dataset: this relation is declared in order to enable the tracing of some instance of hosting with any dataset in general. Since this is a distinct form of hosting, it motivates the declaration of **PE7 Data Hosting Service** as a sub-class of **PE5 Digital Hosting Service**. It motivates the declaration of a new subclass in CRMpe for the documentation of a special subclass of **D1 Digital Object**, used for the identification of dataset.

The functionality of this relation is to allow the tracing of hosting of all and only datasets.

PP7 hosts software object: this relation is declared in order to enable the tracing of some instance of hosting with any software in general. Since this is a distinct form of hosting, it motivates the declaration of **PE6 Software Hosting Service** as a subclass of **PE5 Digital Hosting Service** and relates this to the appropriate class for a software object in general (adopting the CRMdig class **D14 Software**).

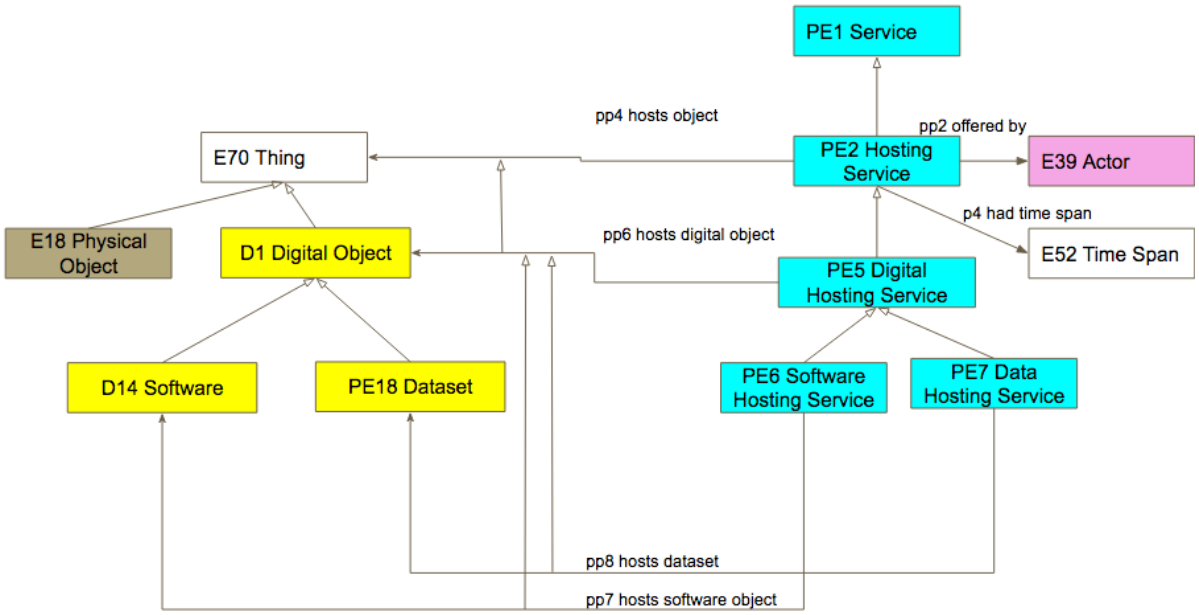

Fig. 6 Illustration of basic hosting service patterns

The resultant modelling pattern is robust for the documentation of who holds what and when, allowing the specification down to a detailed level of kinds of objects hosted, but also offering general relations and classes for documenting hosting services. The general relations and classes for documenting hosting serve two important functions. First, they allow a high level search over all specialized forms of hosting. Second, they allow the documentation of currently un-modelled forms of hosting. In particular, the **PE2 Hosting Service** allows for the documentation of all types of 'non-IT' hosting, such as the custody of a physical CH collection.

3.4.2.2 Curating Service

An entirely different set of relations hold around the service of curating.

Instances of curating service have the function of helping trace who holds a curatorial responsibility for some set of things, for what time they have curated it, and what plan they have in place for this curation. There is a dependent relation between the identity of a curating activity and the set of things that it curates. The curating activity comes to be through the declaration of the will to curate and the declaration of a plan to do so and this in turn brings about a new entity which is the unit that circumscribes the collective set of things curated. We are not able to identify a curated thing through its parts which are at any moment potentially different. We identify a curated thing as the same thing, rather, through the constancy of the curatorial will and plan which maintains it as a whole. It is,

therefore, crucial to provide the ability to trace this relation of curation in distinction to other services, as it affects our ability to know the identity or difference of curated objects.

As with hosting services, the identity of a curatorial service is also differentiated by the kind of object that it is a curation of. Differentiating the type of object curated helps identify the precise kind of curating service a researcher wants to find.

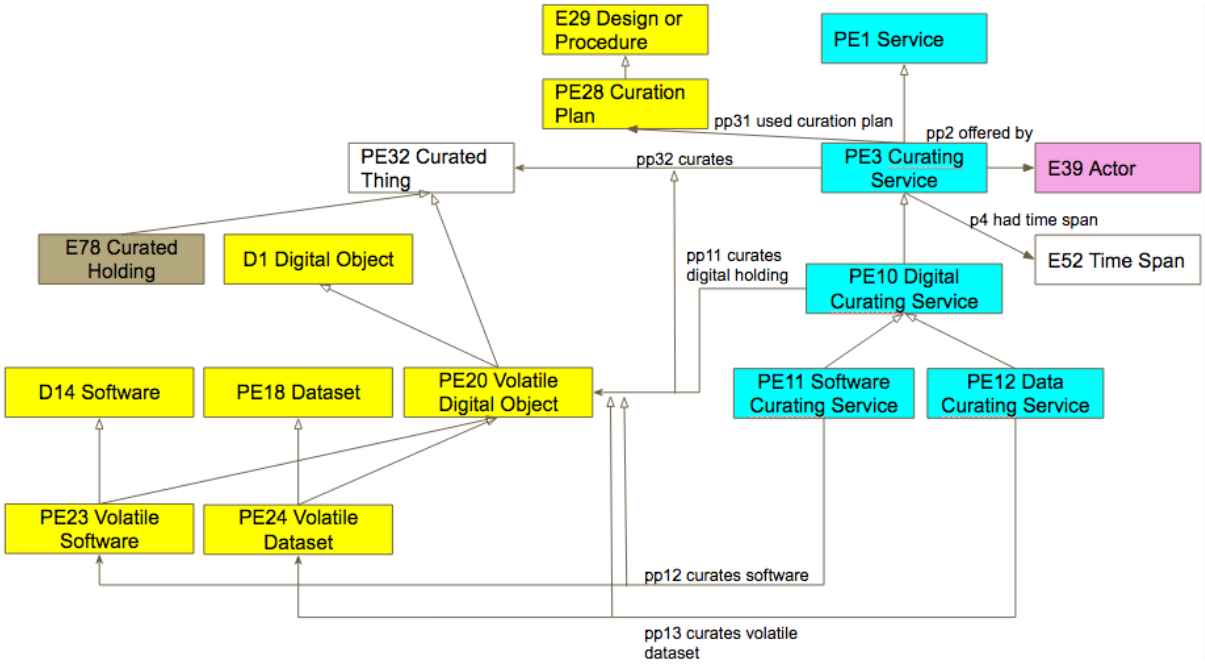

Fig. 7 Illustration of basic curating service patterns

The key relations that must be expressed then are:

PP32 curates: this relation is declared in order to enable the tracing of some instance of curating in general and any curated thing in general. The need to document such a relation motivates the declaration of a new class for this type of service, **PE3 Curating Service** as a sub-class of **PE1 Service** as well as a new class **PE32 Curated Thing** to document the thing curated. The declaration of the class **PE32 Curated Thing** is necessitated in order to create a generalization over curated things whether they be physical or conceptual. The nature of the curating activity is definitively changed depending on its object being physical or digital.

PP31 used curation plan: this relation is declared in order to enable the tracing of the plan which was used an activity of curating over some curated thing. It is the declared intention

of the curating process that enables the identification of the curated thing as one over time and is therefore an essential relation for documentation. This relation connects the instance of **PE3 Curating Service** to an instance of a new class **PE28 Curation Plan** which is harmonized to the CIDOC CRM as a subclass of **E29 Design or Procedure**.

The following subrelations of *PP32 curates* allow the more granular identification of types of curating service and motivate the co-implied declaration of new subclasses of **PE3 Curating Service** as well as motivating the declaration of a subclass of **D1 Digital Object** to account for non-static, dynamic digital objects.

PP11 curates digital holding: this relation is declared as a subclass of *PP32 curates* in order to enable the tracing of some instance of curating of a digital object in general. Since this is a distinct form of curating, it motivates the declaration of **PE10 Digital Curating Service** as a subclass of **PE3 Curating Service**. The nature of such a curating service is different since its object is not physical but digital and can therefore have multiple instances. Likewise, the declaration of this class entails the declaration of the class **PE20 Volatile Digital Object** in order to document a digital object which is not constant in nature but is subject to change at any moment and can therefore only be known through the identity of the curating activity that is responsible for it, the plan that guides its curation and persistent snapshots of the volatile resource. **PE20 Volatile Digital Object** is declared as a subclass of **D1 Digital Object**, in order to allow this general level of reference to a digital object in the case where its more precise nature is not known. For more on these relations see the Digital Object section below.

PP13 curates volatile dataset: this relation is declared as a subclass of *PP11 curates digital holding* in order to enable the tracing of the curating activity over some instance of a dataset. Since the curation of datasets is a distinct form of curating, from the nature of datasets as expressions of propositions about the empirical world, it motivates the declaration of **PE11 Software Curating Service** as a subclass of **PE10 Digital Curating Service**. The object of such a curation is a dataset that is not stable in nature but under constant change according to the plan of the curator. Therefore, we declare a new class **PE24 Volatile Dataset** as a sub-class of **PE20 Volatile Digital Object**. For more on the relations of this class see the Digital Object section below.

An example of such a documentation scenario would be to indicate the active use of a database by some researchers to generated tabular data on their research topic. The dataset is not fixed but under constant modification, but can be identified by the curating activity and backups of the database.

PP12 curates software: this relation is declared as a subclass of *PP11 curates digital holding* in order to enable the tracing of the curating activity over some instance of a software object. Since the curation of software is a distinct form of curating, from the nature of the software object as sets of instructions for running algorithms over datasets, it motivates the declaration of **PE12 Data Curating Service** as a subclass of **PE10 Digital Curating Service**. The object of such a curation is a software object that is not stable in nature but under constant change according to the plan of the curator. Therefore, we declare a new class **PE23 Volatile Software** as a sub-class of **PE20 Volatile Digital Object**. For more on the relations of this class see the Digital Object section below.

An example of such a documentation scenario would be to indicate the active development of a software program by some team. The software is not fixed but under constant modification, but can be identified by the curating activity and releases of the software.

3.4.2.3 E-Service

The third top level service identified as requiring unique documentation in an RI environment is the actual provisioning of e-services in order to access and use or run digital objects. E-services are automated in nature and react to mechanical requests through an information network, receiving commands and sending back appropriate output. E-services are a central part of the offer of many RIs. E-Services provide a point of access in an information network to many other types of services.

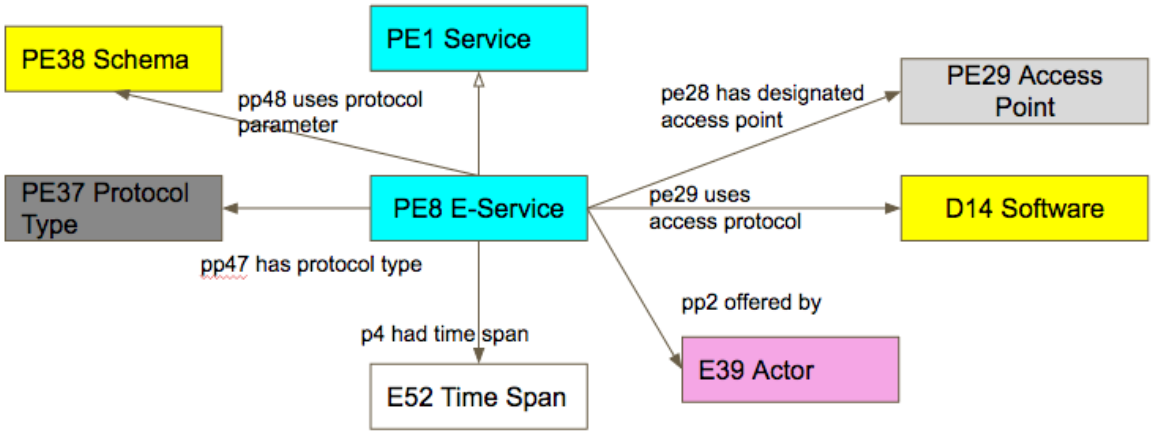

Fig. 8 Model Illustration of basic relations of E-Service Entity

With regards to an E-service itself, there are a number of fundamental relations that must be documented in order for the e-service to be usable in automated requests. The following relationships are therefore declared:

PP28 has designated access point: this relation is declared in order to be able to indicate the address at which an E-Service can be found. **PE8 E-Service** is declared as a new sub-class of **PE1 Service** which, as an automated service, is able to be found through a machine address, documented in the new class **PE29 Access Point**. The intention of this relation is to enable the resolution of an instance of **PE8 E-Service** in an information network.

PP29 used access protocol: this relation is declared in order to document the particular protocol running on an instance of **PE8 E-service** which is required information in order to know how to access the service. A protocol, as a set of instructions for executing some commands, is documented through an instance of **D14 Software**.

PP47 has protocol type: this relation is declared in order to document the protocol type used by an instance of **PE8 E-Service**, where the particular instance of protocol software running is not known, but the type is. The protocol type is documented by a new subclass of **E55 Type** from CIDOC CRM, **PE37 Protocol Type**.

PP48 uses protocol parameter: this relation is declared in order to document the particular protocol parameters that must be passed to an instance of **PE8 E-Service** in order to

access it. Protocol parameters as a particular type of codified instructions are documented through an instance of **PE38 Schema**, a sub-class in turn of **D14 Software**.

E-Service in itself is simply a provisioning of environments and not yet any specific provisioning of something. Therefore, this class will most commonly be used in combination with other types of service, hosting or curating and in respect to their specific identities with regards to the kinds of object they work with, in order to document different service types.

E-Service: Dataset

Instances of e-service in combination with curating and holding data have the function of helping provide access to materials available online, trace who runs the e-service in case of problems and indicate curatorial and hosting responsibilities for some dataset.

In order to provide convenience of documentation, we, therefore, declare specialized sub-classes to indicate services that combine on the one hand, both data hosting and e-services facilities and, on the other, data curating and e-services activities.

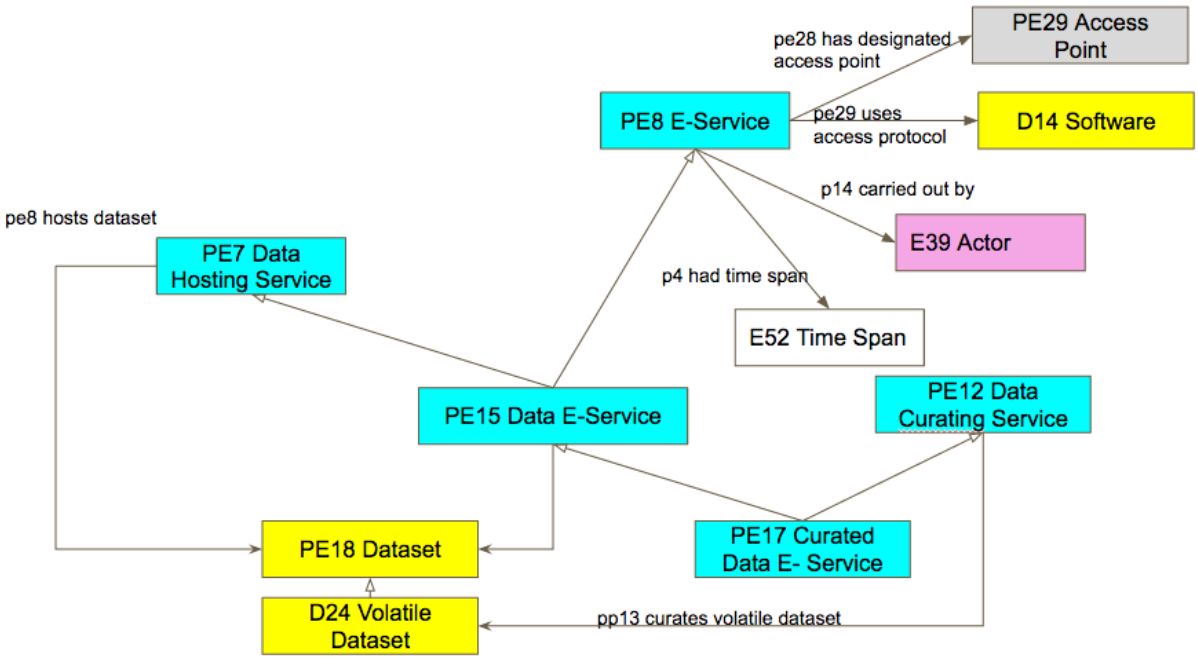

Fig. 9 Illustration of basic modelling pattern of Data E-Services

PE15 Data E-Service: is declared as a subclass of **PE7 Data Hosting Service** and **PE8 E-Service** in order to be able to document both the hosting relation to some dataset and the means to access it in an information network.

PE17 Curated Data E-Service is declared as a subclass of **PE12 Data Curating Service** and **PE15 Data E-Service** in order to be able to document both the act of curation over some volatile dataset and the means to access it in information network. It is specifically declared as a sub-class of **PE15** and not **PE8** because the provision of access already entails an act of hosting of the data curated.

E-Service: Software

Instances of e-service in combination with curating and holding of software have the function of helping provide access to software available online and by what method, trace who runs the e-service in case of problems and indicate curatorial and hosting responsibilities for some software.

In order to provide convenience of documentation, we therefore declare specialized sub-classes to indicate services that combine on the one hand, both software hosting and e-services facilities and, on the other, software curating and e-services facilities.

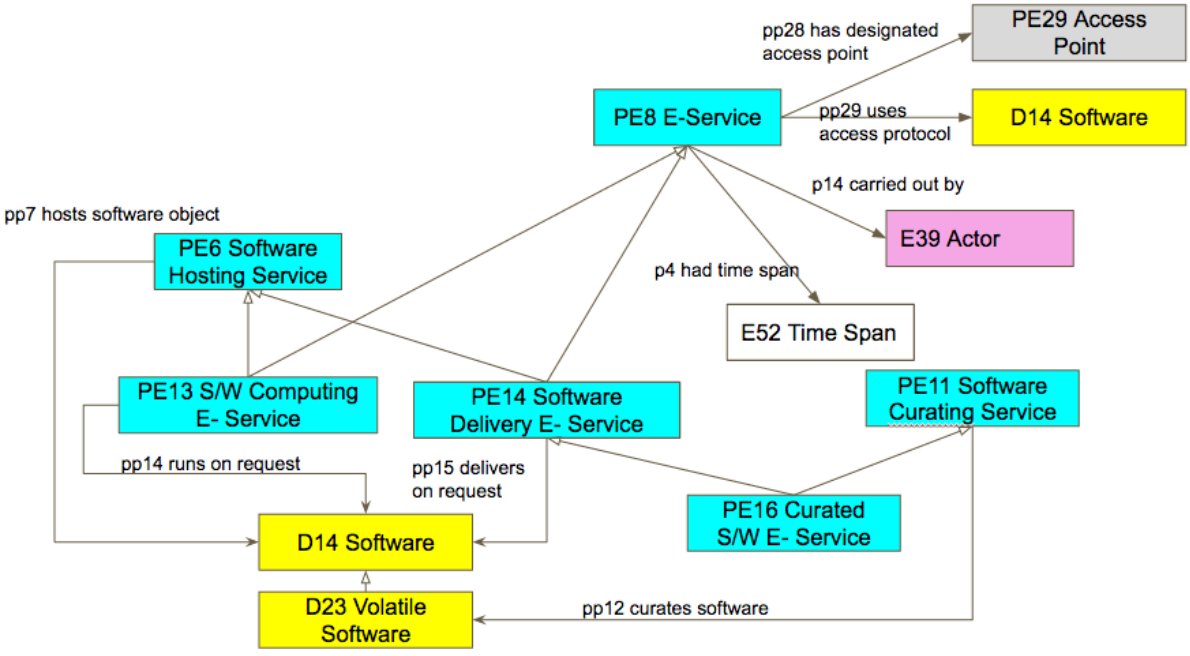

Fig. 10 Illustration of basic modelling pattern of Software E-Services

PE13 Software Computing E-Service: is declared as a subclass of **PE6 Software Hosting Service** and **PE8 E-Service** in order to be able to document both the hosting relation to some software and the means to access it in an information network. It is further motivated by the need to trace the means by which the software is made available. In this case, it documents an instance of a service able to run specific software.

PP14 runs on request: documents the ability of an instance of **PE13 S/W Computing E-Service** to run some instance of **D14 Software**.

PE14 Software Delivery E-Service: is declared as a subclass of **PE6 Software Hosting Service** and **PE8 E-Service** in order to be able to document both the hosting relation to some software and the means to access it in an information network. It is further motivated by the need to trace the means by which the software is made available. In this case, it documents an instance of a service that provides specific software for download.

PP15 delivers on request: documents the ability of an instance of **PE14 Software Delivery E-Service** to make available for download some instance of **D14 Software**.

PE16 Curated Software E-Service: is declared as a subclass of **PE11 Software Curating Service** and **PE14 Software Delivery E-Service** in order to be able to document both the curating relation to some software and the means to access it in an information network. It is declared as a sub-class of **PE14 Software Delivery E-Service** since the curation of the software and making it accessible in an e-environment entails a hosting activity.

E-Service Hosting: Software and Dataset

In querying for and accessing hosting services, researchers are concerned with access to resources. It is a crucial relation, then, to describe where a resource can be accessed in an information network environment. Likewise, if it is made available by multiple providers or providers change, this information is important to ensure continued access to the sought for resources.

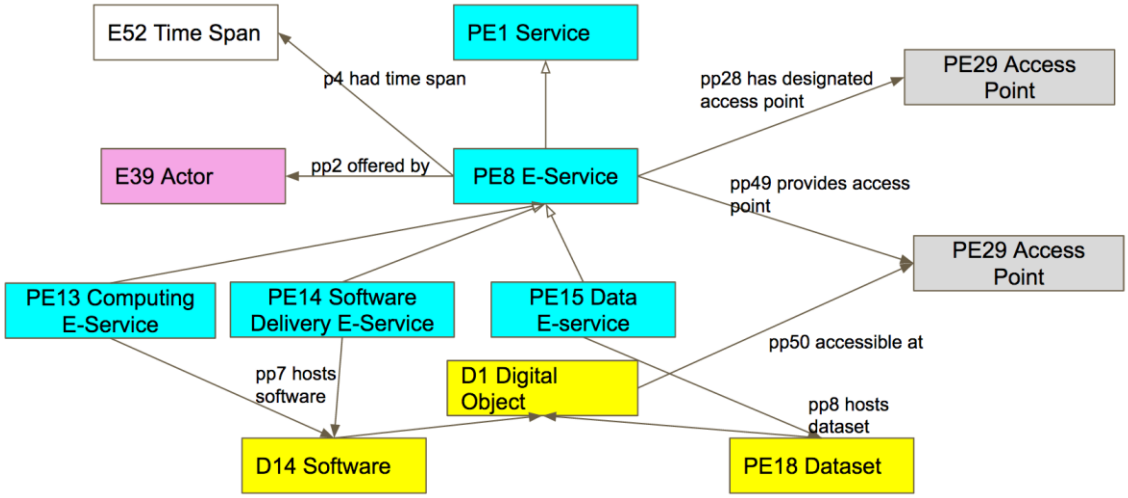

Fig. 11 Illustration of basic modelling pattern of E-Service Access Provision for Resources

To support this functionality, therefore, it is necessary to declare relations that indicate where a resource is to be found, which service provides this address and which particular resource offered by that service has this particular address. In order to make this information available in the model, we declare the following relations:
PP49 provides access point: this relation is declared to indicate which instance of **PE8 E-Service** has provided a particular address (instance of **PE29 Access Point**) for making available some digital object. The instance of **PE8 E-Service** will, as a rule, be one of its hosting sub-classes (PE13, 14, 15 and their subclasses)

PP50 accessible at: this relation is declared to indicate for an instance of **D1 Digital Object** the address at which it is accessible, documented as an instance of **PE29 Access Point**.

Together with the hosting relation that holds between an instance of **PE2 Hosting Service** and its object, this forms a relation triad which is complete to describe: where the digital object is hosted, at what address it can be found and from where it has been provided this address. This makes it possible to document an instance of PE13,14,15 and their subclasses as hosting and making accessible an open number of instances of **D1 Digital Object**.

3.4.3 Digital Object

Within a contemporary RI environment, the end goal of researchers is often access to and use of digital objects. For this reason their proper characterization as well as the relations that hold between them is necessary. Of particular importance is establishing the quality of data and the relation of the data amongst itself in terms of parts and wholes. While provenance and access information is provided by relation to instances of PE1 Service as described above, the proper modeling of these types of relations is equally of important in order to facilitate the researcher's access to the right object.

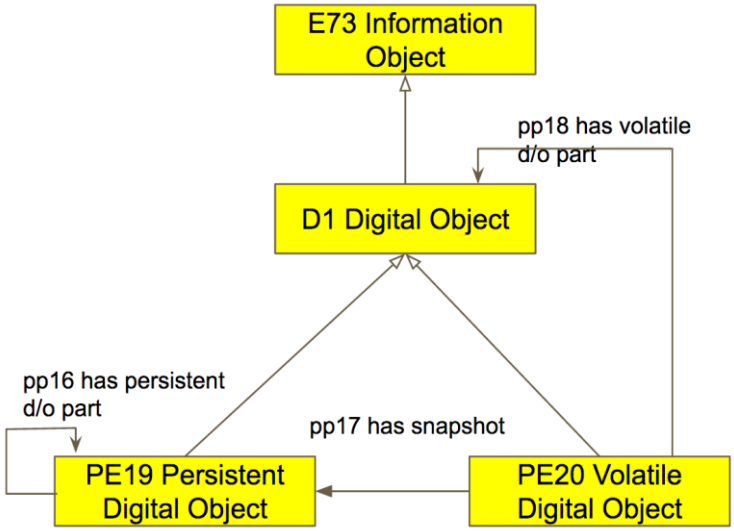

Fig. 12 Illustration of basic modelling pattern for Digital Objects

As described above in the analysis of services, acts of curating and holding imply the need for a fundamental distinction in digital objects between those that can be characterized as persistent and those that are best characterized as volatile.

For this reason, two subclasses of **D1 Digital Object**, as adopted from CRMdig are declared in CRMpe:

PE19 Persistent Digital Object: this class is declared in order to identify instances of digital object that are identifiable at the bit level enabling their repeated identification over time.

PE20 Volatile Digital Object: this class is declared in order to identify instances of digital object that are undergoing a situation of potential, continuous change. There is no immediate identity ascribable to this object, but it can nonetheless be identified by proxy through instances of persistent dataset that store an instance of it, as well as the curation plan associated to the instance of **PE3 Curating Service** which is responsible for it.

Declaring these two classes allows a more accurate statement of the relations between these types of digital object in terms of parts and wholes.

PP16 has persistent d/o part: this relation is declared in order to allow the documentation of part-whole relations amongst instances of **PE19 Persistent Digital Object**. Due to their nature as persistent, these objects can only be made up of other persistent objects. This has an effect on the evaluation of the data on the part of the researcher.

PP18 has d/o part: this relation is declared in order to allow the documentation of part-whole relations amongst an instance of **PE20 Volatile Digital Object** and some other instance of digital object. A volatile digital object may be made up of parts that are themselves volatile, a data stream within a data stream, or persistent, a particular record collected into the overall data stream.

PP17 has snapshot: this relation is declared in order to enable to describe the relation between a changing digital object and its persistent manifestations. Whether in the form of backups or releases, such manifestations provide an official reference point for a state of the volatile digital object at some point in time.

3.4.4 Dataset

In order to provide the ability to search for relations amongst digital objects considered as collections of propositions about the world, the digital object class was further specialized into a dataset class. Following the pattern of reasoning that holds for digital objects, as such, it is necessary to model the relations between datasets considered as persistent or volatile in an analogous manner. By declaring all the new relations subrelations of those modelled for digital object as such, we gain a generalization over this specific subset of investigation which will allow general or specific query and retrieval of data with regards to part-whole relations on datasets.

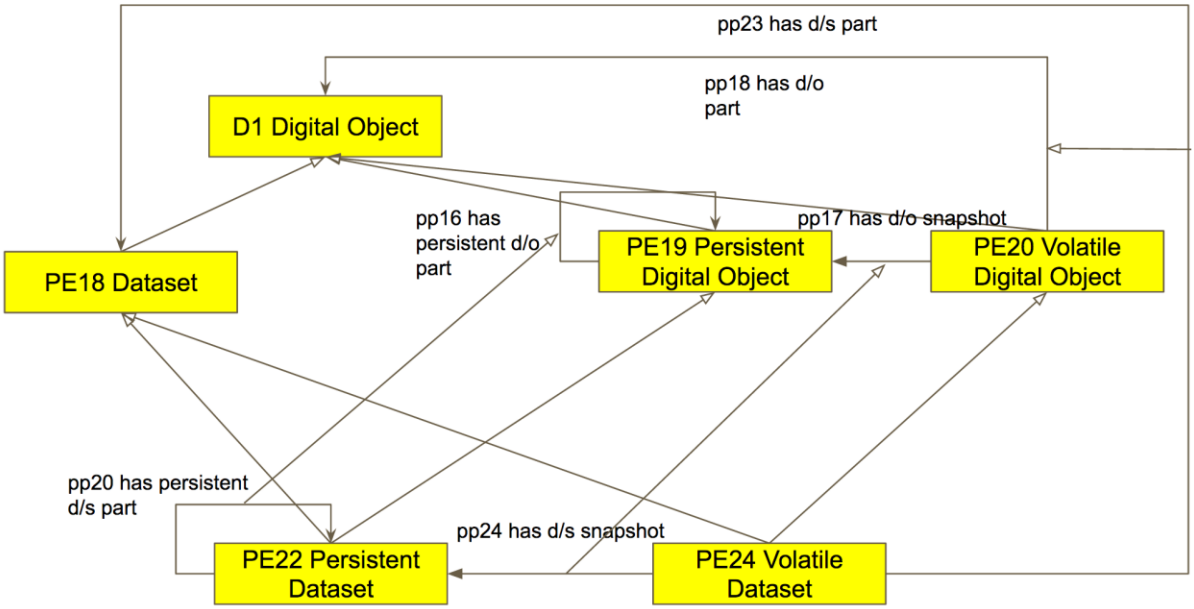

Fig. 13 Illustration of basic modelling pattern for Dataset and subclasses

PE18 Dataset: this class is declared to allow the documentation of instances of digital object that have as their primary function the encoding of propositions made about the empirical world.

In order to track the potential relations between datasets the following relations and associated classes were declared:

PP20 has persistent d/s part: this relation is declared in order to allow the documentation of part-whole relations amongst instances of **PE22 Persistent Dataset**. Due to their nature as persistent, these objects can only be made up of other persistent datasets. This has an effect on the evaluation of the data on the part of the researcher. This relation is declared as a subrelation of *PP16 has persistent d/o part*.

PP23 has d/s part: this relation is declared in order to allow the documentation of part-whole relations amongst an instance of **PE24 Volatile Dataset** and some other instance of dataset. A volatile dataset may be made up of parts that are themselves volatile, a data stream within a data stream, or persistent, a particular record collected into the overall data stream. This relation is declared as a subrelation of *PP18 has volatile d/o part*.

PP24 has d/s snapshot: this relation is declared in order to enable to describe the relation between a changing dataset and its persistent manifestations. In the form of backups or releases, such manifestations provide an official reference point for a state of the volatile dataset at some point in time. This relation is declared as a subrelation of *PP17 has d/o snapshot*.

3.4.5 Software

In order to provide the ability to search for relations amongst digital objects considered as instructions to process datasets algorithmically, the digital object class was further specialized into a software class. This class already having been declared in CRMdig (D14 Software), could be borrowed from this standard. Following the pattern of reasoning that holds for digital objects, as such, it is necessary to model the relations between software considered as persistent or volatile in an analogous manner. By declaring all the new relations subrelations of those modelled for digital object as such, we gain a generalization over this specific subset of investigation which will allow general or specific query and retrieval of data with regards to part-whole relations on software objects.

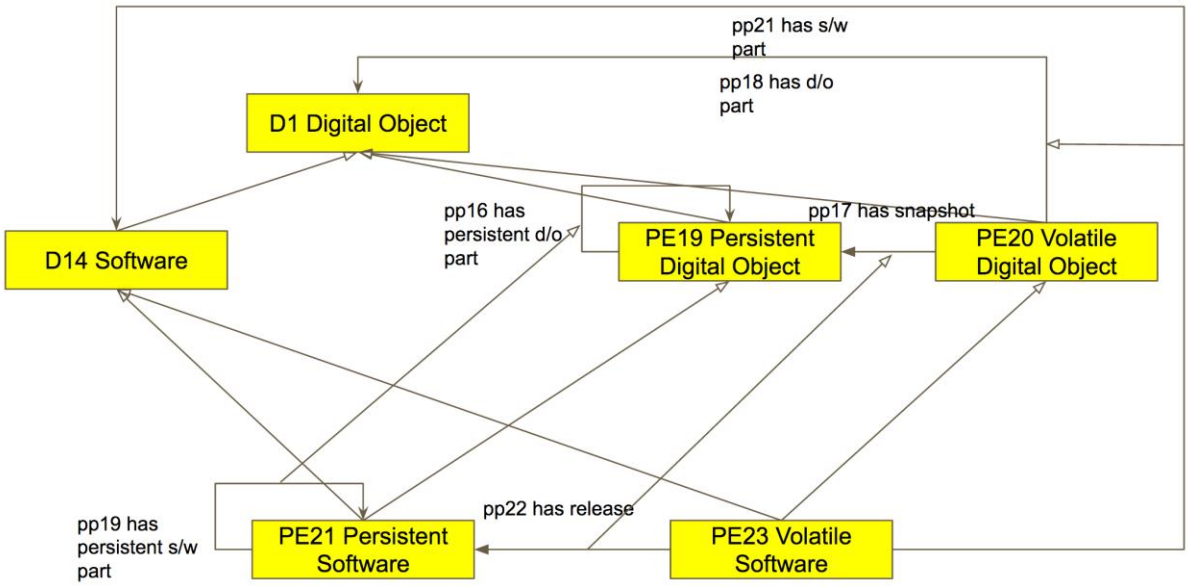

Fig. 14 Illustration of basic modelling pattern for Software and subclasses

D14 Software: this class is adopted from CRMdig and is declared to allow the documentation of instances of digital object that have as their primary function the encoding of instructions to process datasets algorithmically.

In order to track the potential relations between software the following relations and associated classes were declared:

PP19 has persistent s/w part: this relation is declared in order to allow the documentation of part-whole relations amongst instances of **PE21 Persistent Software**. Due to their nature as persistent, these objects can only be made up of other persistent software objects. This has an effect on the evaluation of the programme on the part of the researcher. This relation is declared as a subrelation of *PP16 has persistent d/o part*.

PP21 has s/w part: this relation is declared in order to allow the documentation of part-whole relations amongst an instance of **PE23 Volatile Software** and some other instance of software. A volatile software object may be made up of parts that are themselves volatile, two distinct streams of code development, or persistent software objects (e.g. a particular release of a library) within the overall development code. This relation is declared as a subrelation of *PP18 has volatile d/o part*.

PP22 has release: this relation is declared in order to describe the relation between a changing software object and its persistent manifestations. In the case of releases, such manifestations provide an official reference point for a state of the volatile software at some point in time. This relation is declared as a subrelation of *PP17 has d/o snapshot*.

3.4.6 Actor

Maintaining knowledge of the interaction between resources and their holders/curators, necessitates, of course, the ability to represent the key relations between different kinds of actors and these resources, as well as to track the evolution of relations of actors amongst themselves. Within the context of a RI environment, it is important to know who holds/curates what and their specific responsibility, under what rights they do so, as well as the how to contact/reach these actors

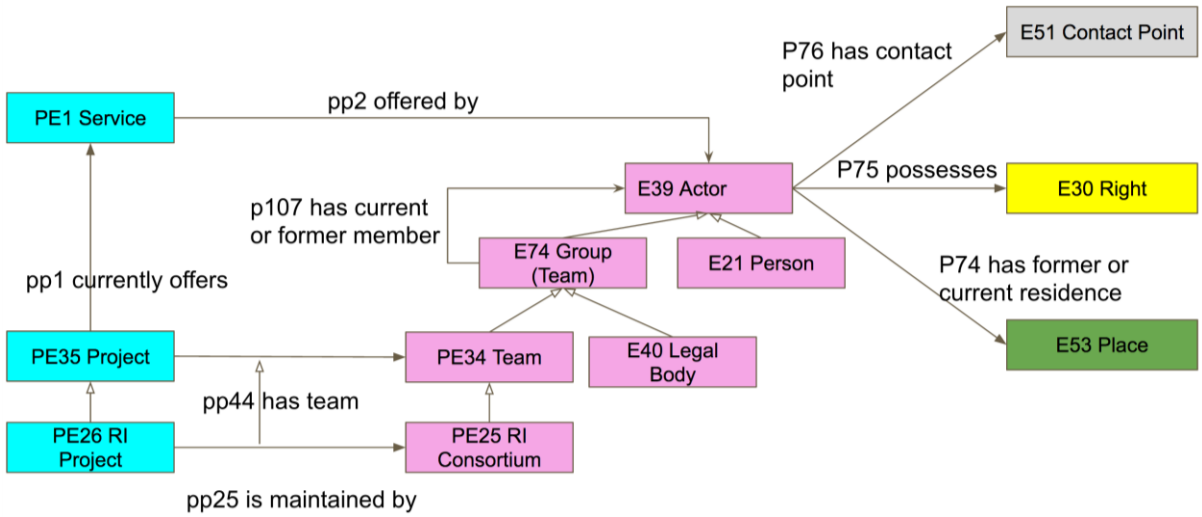

Fig. 15 Illustration of basic modelling pattern for Actor subclasses

Adopting the modelling already given in CIDOC CRM, relations for tracking actors either as individuals or groups, legal body or not and basic information on how to contact these individuals is already covered.

Unique relations to be tracked in RIs that provoked the declaration of new classes for the CRMpe model were:

PP44 has maintaining team: this relation is declared in order to track the relation between some group formed especially towards the accomplishment of some goal. The relation motivated the declaration of a new sub-class of **E74 Group**, **PE34 Team**. It also motivated the declaration of a new sub-class of **E7 Activity** to document especially projects, considered as activity runs by a particular group for a particular end, **PE35 Project**.

PP25 is maintained by RI: this relation is declared in order to track the relation between some RI formed especially towards the accomplishment of some goal. The form of this relation is a derivation of the *pp44 relation*, specialized for more accurate tracking just of this one type of case of particular import with the domain of our modelling. Therefore, a new actor class for RIs was declared under the **PE34 Team** class, **PE25 RI Consortium**. Likewise, a new RI project class was declared under **PE35 Project**, **PE25 RI Project**.

PP2 offered by: this relation did not provoke the declaration of a new class, but forms the necessary link between an instance of **E1 Service** and the particular instance of **E39 Actor** that supports it.

3.5 Model Validation Process

The adequacy of the model was tested by checking the proposed classes against the modelled data structures and finding if an adequate representation could be made of the latter using the former. In particular, this checking process happened in two different model validation moments. The first moment came in constructing a mapping of example data and the schema of participating RIs to the PEs. It is envisioned that the model will be further refined when large scale aggregation is carried out on the basis of these maps and further exceptions are detected and marked.

4. PARTHENOS Minimal Metadata

In order to support an active registry running a semantic graph tracing the relationships and entities modelled above, it is necessary to develop a functional set of minimal metadata requirements for the data to be represented by the graph. Such minimal metadata requirements are not meant as a ‘core’ standard but are rather specified precisely to demand the least possible data from aggregated sources while still being able to guarantee a functional management of the data being aggregated. The minimal metadata requirements are specified in order to be able to establish for any entity mapped in the graph what its identity is and, by extension, if it is the same or different from another aggregated entity. This identification function is the basic requirement in order to manage a semantic graph as envisioned in the architecture design proposed in section 2 of this document.

This minimal specification is adopted in order to work together with the rich source metadata in whatever format it may be that is not covered or translated by the PARTHENOS entities model. Together, the registry identifies what exists in the world covered by its aggregation and the rich metadata at source covers exactly what this resource is. The two work in combination and with mappings of clusters of resources to common formats in order to provide as accurate a picture of the aggregated world of resources as is presently possible. By adopting only a minimal metadata requirement, this strategy also avoids the manpower and financial bottleneck entailed by the specification of a ‘core’ standard with a rich central metadata catalogue. A minimal metadata standard begins from the assumption that resources are imperfectly and incompletely and differently

documented meaning that any strategy which relies on initially well formed and rich metadata sets will either fail or, if reaching its goal, constantly struggle to maintain the imposed order over time.

With the above reasoning in mind, the minimal metadata standard that follows is specified with the ontological considerations of the identity of the main entities discovered in the modelling process and with feedback from the participating communities and their infrastructure managers with regards to the information without which a resource cannot be managed. What is defined, then, as minimal metadata is divided between mandatory and suggested minimal metadata. The mandatory minimal metadata is specified precisely in relation to the identity conditions laid out in the original analysis. The non-mandatory data provides useful additional contextual information for data managers and end users to understand better the content and context of the resource even at this general level without reference to its originating rich metadata, if necessary.

In the following section we provide the logic behind the declaration of the minimal metadata facets for each of the identified relation/class clusters modelled. In the appendix to this document, this minimal metadata is specified in its own functional requirements document for practical use.

4.1 All Entities Minimal Metadata

Minimal metadata specifications can be drawn up hierarchically based on those fields required for all resources, and then those fields required for documentation of different subclasses of resources.

For all resources described in the registry, we declare mandatory that they have:

Label	Man. ?	Description	CRM Translation
ID	Yes	An identifier used to indicate the resource.	E1->P1->E42
Type	Yes	The place of the resource in the overall ontological hierarchy	-
Specific Type	Yes	The classification of the resource according to a taxonomy appropriate to the type of resource.	E1->P2->E55
Title	Yes	The name by which the resource is known or referred to by.	E1->P1->E41

Table 2. Minimal Metadata Requirements for Any Entity

It holds for all and any resource specified within the joint resource registry graph that we must hold this information in order to be able to manage the aggregated resource effectively. This set of minimal metadata assures that we have means to reference the object (ID, Title) and that we know what type of object it is in our universe of discourse (Type) as well as the more specific taxonomic association that would be given to it by specialists in the area (Specific Type).

4.2 Project Minimal Metadata

The identity of a project is directly linked to its being supported by some team that maintains it. The nature of the project can be deduced from the composition of activities it supports. Therefore, we mandatorily require the documentation of the team that maintains the project, while we suggest the enumeration of the related project activities through reverse links.

Label	Man. ?	Description	CRM Translation
Has Team	Yes	Indicates the team that is committed to running the project and upholding its stated aims.	PE35->PP44->PE34
Supported activity	No	Indicates the set of activities including services that make up the activities supported by the project.	PE35->PP43->E7

Table 3. Minimal Metadata Requirements for Projects

4.3 Service Minimal Metadata

As defined in the analysis above, a service is the willingness and ability of an actor to carry out certain activities for the benefit of some requesting actor. To identify a service, then, requires knowing the actor, the type of activity they are involved in and the kind of product that will come from such a service. A service begins when the willingness to carry it out and the ability are established and ends when this will and ability is permanently interrupted.

The above identity conditions allow for both a general requirement for minimal metadata related to a service and then, according to the differentiations of actor, type of activity and kind of product, entails further specifications for sub-types of service.

Label	**Man. ?**	**Description**	**CRM Translation**
Competence	Yes	The ability of a service to do something successfully: is a relation that connects a *service* with an *activity type*	PE1->P2->E55
Provided by	Yes	The actor that provides the service.	PE1->PP2->E39->P1->E41
Declared Begin/End of Operation	No	The date that the service providers indicates as the beginning and/or ending of the offer of the service.	PE1->PP42->E61
Last Confirmation	Yes	The date that it is confirmed that the service still exists.	PE1->P4->E52->P81->E61
Date of Registration	Yes	The date when registered with PARTHENOS (acts as at least first confirmation of existence of service… must be running when added).	
Availability	No	The kind of service provision with regards to the periods in which it is provisioned. E.g.: 24/7, 24/5, on request, unknown, periodic, business hours	PE1->P2->E55
Condition of use/Rights Type	Yes	The type of conditions that the use of this service are subject to (Open Access, Open Access - required registration, license-based, on request, embargo)	

Condition of Use / Rights Text	Yes	The actual text outlining conditions of use.	
Contact Person	No	The contact person representing the actor (usually an institution or group) that provides the service.	PE1->PP2->E39-> p107 -> E21
Communication Address	Yes	The contact method for this particular service (regardless of providers address)	PE1->PP2->E39-> p107 -> E21 -> p76 -> E51

Table 4. Minimal Metadata Requirements for Services

The minimal metadata for an instance of service, then, is required in order to show what it does (Competence, Availability), who offers it (Provided by), if and when it is in operation (Declared Begin/End of Operation, Last Confirmation, Date of Registration), the conditions under which it can be made use of (Conditions of Use) and how to contact someone with regards to the service (Contact Person, Communication Address).

As entailed by the definition of service, we require additional more specific minimal metadata in function of the type of service offered, and the manner in which this differentiates the identity of the service itself.

Following the distinctions in the conceptual model, we differentiate hosting, curating and e-service as high level types of service requiring their own metadata.

4.3.1 Hosting Service Minimal Metadata

In addition to minimal metadata required for all services, instances of hosting service should additionally document precisely the objects that they host, as this is important to determining their identity. Therefore we require:
(+Service)

Label	**Man. ?**	**Description**	**CRM Translation**
Hosts Object	No	A list of objects that are hosted by the service.	PE2->PP4->E70

Table 5. Minimal Metadata Requirements for Hosting Services

Note that it is not possible to require such a list of hosted objects necessarily but it is put in as a suggestion to document if possible. This information can be acquired from the reverse relationship of the hosted objects to the service, rather than putting it as a documentation element of the service itself.

With regards to hosting, the kind of hosting can be further specialized to track the specific relations of digital hosting, software or dataset hosting. In this case, we require the further relevant specialized minimal metadata:

Digital Hosting Service Minimal Metadata
(+Service)

Label	Man. ?	Description	CRM Translation
Hosts Digital Object	No	A list of digital objects that are hosted by the service.	PE5->PP6->D1

Table 6. Minimal Metadata Requirements for Digital Hosting Services

Software Hosting Service Minimal Metadata
(+Service +Digital Hosting Service)

Label	Man. ?	Description	CRM Translation
Hosts Software Object	No	A list of software objects that are hosted by the service.	PE6->PP7->D14

Table 7. Minimal Metadata Requirements for Software Hosting Services

Data Hosting Service Minimal Metadata
(+Service +Digital Hosting Service)

Label	Man. ?	Description	CRM Translation
Hosts Dataset	No	A list of datasets that are hosted by the service.	PE6->PP8->PE18

Table 8. Minimal Metadata Requirements for Data Hosting Services

4.3.2 Curating Service Minimal Metadata

In addition to minimal metadata required for all services, instances of curating service should additionally document precisely the objects that they curate, as this is crucial to determining their identity. The volatile object that is curated and the curating service give identity to each other with the curation plan set by the instance of curating service giving the ability to better understand the nature of the curated object. Therefore we require:

Label	**Man.?**	**Description**	**CRM Translation**
Curates Object	No	A link of the service to the objects that it curates	PE3->PP32->PE32

Table 9. Minimal Metadata Requirements for Curating Services

Digital Curating Service

(+Service)

Label	**Man.?**	**Description**	**CRM Translation**
Curates Digital Object	No	Link the curation service to the volatile digital object that it manages	PE10->PP11->PE20

Table 10. Minimal Metadata Requirements for Digital Curating Services

Data Curating Service

(+Service +Digital Curating Service)

Label	**Man.?**	**Description**	**CRM Translation**
Curates Dataset	No	Link the curation service to the volatile dataset that it manages	PE12->PP13->PE24

Table 11. Minimal Metadata Requirements for Dataset Curating Services

Software Curating Service

(+Service +Digital Curating Service)

Label	Man.?	Description	CRM Translation
Curates Software Object	No	Link the curation service to the volatile software that it manages	PE11->PP12->D14

Table 12. Minimal Metadata Requirements for Software Curating Services

4.3.3 E-Service Minimal Metadata

An e-service is a sequence of states of activation of a software system installed on some particular machines offering facilities on the Web or an information network that reacts to mechanical requests by receiving data, manipulating it and sending it back. The term e-service describes a form of communication with an installed and running software system of whatever kind. The identity of an e-service depends on the particular processing software it runs, the actor maintaining the e-service active, and the logical communication address for issuing requests to it.

The above definition motivates the minimal metadata required to identify unique instances of an e-service which are:

(+Service)

Label	Man.?	Description	CRM Translation
Access Point	Yes	The information network address whereat the instance of e-service is to be accessed.	PE8->PP28->PE29
Authorization	Yes	A link to the authentication and authorization policy types which are in place to access the e-service. (*e.g., OAUTH, SAML)*	
Protocol Type	Yes	A link to the type of access protocol which this e-service invokes in order to be used. (E.g. SOAP/REST)	PE8->PP47->PE37

Protocol	No	A link, where available, to the particular instance of the running software protocol which is running in order for the e-service to be accessed.	PE8->PP29->D14
Protocol Parameters	No	A link, where available, to the particular protocol parameters instance necessary to be invoked in order to make use of the protocol instance presently run by the e-service.	PE8->PP48->D14
Provides Access Point	No	A link to an information network address (e.g. URL) that the e-service supports to give	PE8->PP49->PE29

Table 13. Minimal Metadata Requirements for E-Services

The required minimal metadata aims to ensure the central information to understand the identity of a particular e-service which entails where it is to be accessed (Access Point) and how it can be accessed (Authorization and Protocol Type). Two levels of documentation are made available. The first as described above is at the type level. Where further information is available the specific instances of protocol software and the parameter files required to run these particular instances can and should be documented (Protocol, Protocol Parameters). An additional parameter of the e-service that is documented are access point addresses at which objects which are hosted by instances of E-Service that are also hosting services is made available. This connection allows the user to trace back to the appropriate e-service where to access a digital object and, in case it is not accessible for some reason, to come into contact with the appropriate authority to restore the functionality of the e-service.

4.4 Digital Object Minimal Metadata

The digital object is an encoded piece of information stored in a digital environment. Establishing its identity then relies particularly one where it is held and if it is or is not part of other digital objects.

(+Object)

Label	Man. ?	Description	CRM Translation
Hosted by	Yes	Here we indicate the digital hosting service responsible for the hosting of this digital object.	D1->PP6i->PE5
Is / Was Part of	Yes	Here we indicate digital objects of which this digital object has formed part.	D1->P106i->D1

Table 13. Minimal Metadata Requirements for Digital Object

4.4.1 Persistent Digital Object Minimal Metadata

The persistent digital object can be identified at the bit level. Contextually, however, we wish to gather also its part of relation to other digital objects, if it is a snapshot for some volatile digital object and other identity confirming attributes such as size, date of creation and checksum.

(+ Object + Digital Object)

Label	Man. ?	Description	CRM Translation
Is Part Of	No	Here we indicate the persistent data object that forms a distinct part of the overall persistent data object in question. N.B. a persistent data object can have as part any other type of persistent digital object. It cannot have a volatile data object as part.	PE19->PP16->PE19
Is Snapshot of	No	If the persistent data object stands as the identifying snapshot for some volatile data object, this can be indicated here.	PE19->PP17i->PE20

Same as	No		
Compilation Date	Yes	Here we indicate the date when the current encoding was fixed.	PE19->L11i-D7->P4->E52->P81->E61
File Size	Yes	Here we indicate file size in bytes	PE19->P43->**E54** + PE19->P43->**E54**->2->E55
Checksum	Yes	Here we indicate the checksum of the persistent dataset.	PE22->P39i->E16->P40->E54 + PE19->P39i->E16->P40->E54 ->P2->E55

Table 14. Minimal Metadata Requirements for Persistent Digital Object

4.4.2 Volatile Digital Object Minimal Metadata

In contrast to the persistent digital object, a voluntary digital object cannot be known directly but only through its surrogates and its maintainer. In order to facilitate making this identification, then, we require the documentation of the curating service responsible for some volatile digital object, its last snapshot and a link to the curation plan which can help identify what should be encoded in this object.

(+ Object + Digital Object)

Label	**Man. ?**	**Description**	**CRM Translation**
Curated by	Yes	Here we indicate the digital curating service responsible for the curation of this object.	PE20->PP11i->PE10
Has Snapshot	No	Here we indicate the snapshot that gives the identity to a volatile data object. In order for a volatile data object to have proper provenance it must at any time have one official	PE20->PP17->PE19

		snapshot that is known to the curator of the object.	
Is Part Of	No	Here we can indicate the parts of a volatile data object. A volatile data object can be made up of volatile as much as persistent data objects. If it has as component as volatile data object, this object in turn, in order to have proper provenance must have its own snapshot.	PE20->PP18->D1
Has Curation Plan	Yes	Link the curation service to the curation plan which it implements	PE3->PP31->PE28

Table 15. Minimal Metadata Requirements for Volatile Digital Object

4.5 Dataset Minimal Metadata

Datasets are additionally qualified as digital objects that encode propositions that make statements about the world. For this object type no particular new mandatory descriptors are required to aid in the identification of a particular instance of a dataset. We do, however, list a number of optional descriptors for breaking down the coverage of the dataset in question in order to aid users of the system to understand the actual content of the dataset and whether it is a useful candidate for further investigation and integration.

(+ Object + Digital Object)

Label	**Man.?**	**Description**	**CRM Translation**
Hosted by	Yes	Here we indicate the data hosting service responsible for the hosting of dataset	PE18->PP8i->PE7
Encoding Type	Yes	Here we indicate the encoding(s) of the dataset in question	PE18->L11i->D7->P33->E29->P2-E55
Schema/Format	No	Here indicate the schema used to structure the dataset.	PE18->L11i->D7->L23->D14
Subject	No	Here we indicate the role that the dataset can play in research	PE18->P129->E55

Spatiotemporal Coverage	No	Here we indicate the geographic scope for which the dataset has relevance.	PE18->E2
Created by	Yes	Here we link the dataset to its creator	PE18->L11i->D7->P14->E39

Table 16. Minimal Metadata Requirements for Datasets

4.6 Software Minimal Metadata

The substance of a software object is to provide algorithms to systematically execute processes. These are encoded in digital objects. The specific different relations which we are interested in facilitating in tracing with regards to software objects in the registry are if they are made available by some service for use, what they can be used for and what language and or parameters must be invoked to use the software object. This guides the selection of the minimal metadata prescribed here.

(+ Object + Digital Object)

Label	**Man. ?**	**Description**	**CRM Translation**
Hosted by	No	Here we indicate the software hosting service responsible for the hosting of the software object.	D14->PP7i->PE6
Delivered on request by	No	Here we indicate the software delivery e-service capable of delivering the software to a client.	D14->PP15i->PE14
Run on Request by	No	Here we indicate the software computing e-service capable of delivering the software to a client.	D14->PP14i->PE13
Configurations	No	Here we indicate all configurations that define the behaviour of the software (RDF schema, #include[GB1])	
Programming language	No	Here we indicate the programming language used in creating the software	D14->L11i->D7->P33->E29

Executes processes of type	Yes	Here we indicate the kind of process types that the software (typically an algorithm) can execute	D14->P103-> E55

Table 17. Minimal Metadata Requirements for Software

4.7 Actor Minimal Metadata

The tracking of actors within the network of relations of interest in a research infrastructure environment is done in order to facilitate agent-to-agent communication with regards to resources within the network, activating changes, seeking access, initiating contracts. Therefore, the primary minimal metadata that is suggested with regards to the actor are various forms of address at which the actor can be reached. None of these characteristics specifically characterize the identity of the actor so they are no stipulated as mandatory. In the full specification of the minimal metadata set, additional non mandatory fields for tracking the change of relations between actors over time are also suggested.

Label	**Man.?**	**Description**	**CRM Translation**
Legal Address	No	Here we give the legal address for the actor	E39->P76->E45 + E39->P76->E45->P2->E55
Mailing Address	No	Here we give the mailing address for the actor	E39->P76->E45 + E39->P76->E45->P2->E55
Contact Person	No	Here we link to the designated contact person for this actor.	
Phone	No		E39->P76->E51 + E39->P76->E51->P2->E55

Email	No		E39->p76->E51 + E39->p76->E51->p2->E55
Provides Service	No	Here we indicate the services the actor provides	E39->PP2i->PE1
Requests Service	No	Here we indicate the services the actor requests.	E39->PP3i->PE1

Table 18. Minimal Metadata Requirements for Actors

4.8 Minimal Metadata Validation Process

The minimal metadata will be checked for validity and functionality at multiple stages of the data integration process into the registry. First it will be tested with the initial registration of top level entities into the registry. Second it will be tested with the first validated mappings and sample data in the 3M mapping tool. Finally, it will be evaluated a third time through the aggregation activity. The possibility for the strict application of this minimal set of data along with the additional usability that it offers will be tested and serve as the basis for making adjustments to the suggestions.

5. Examples and Application Scenarios

The use scenario of the proposed architecture is as follows. A cohort of RIs decide to make use of a common research infrastructure proposition (PARTHENOS) in order to derive the benefits of cross-domain research and availability of tools.

Participating RIs register their services directly to the common registry using provisioned metadata forms for gathering required minimal metadata and expressing it according to the common PARTHENOS Entities Model. This initial registration provisions the registry

with the top level entities which will support an understanding of who is in the community, what they offer and the part whole relations amongst the resources offered at the highest level. This manual metadata entry process for the provisioned resources is envisioned since structured data on an organization itself of itself is normally not available. That being said, it is precisely this information which is needed in the first instance in order to ground the common information service. It provides the initial map of the present state of relations amongst resources and actors. At this time QoS agreements are registered in relation to services in order to have explicit documentation of the update and maintenance scenarios envisioned for the services provided by participating actors.

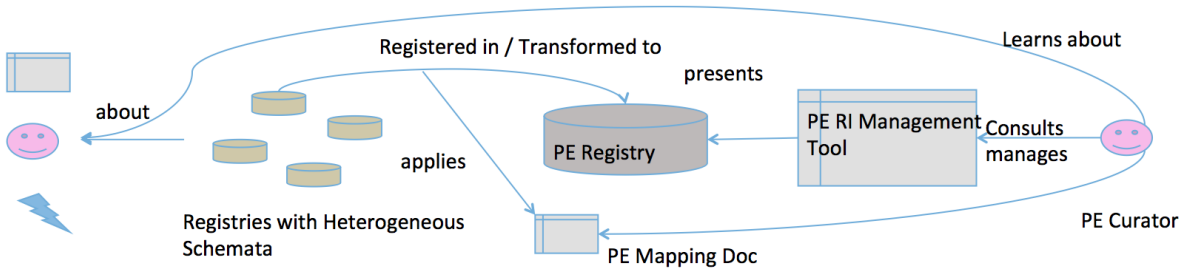

Fig. 16 illustration of the initial registration and high level registration process

Amongst the resources offered by participating RIs, some will be offers of hosting or curation of datasets that, in turn, will hold information with regards to the identities of resources, who hold/curate them and the part whole relations amongst them. These data sources will inevitably be expressed in a variety of formats and documented to different degrees of detail. The information contained within them provides the picture of the wider network of actors and information that is supported/known by the participating RI. Therefore these high level datasets of the registry/aggregation services of participating RIs require a mapping to the PARTHENOS Entities Model and verification against the minimal metadata model, in order to enable a transformation into the common information space of the registry.

Once loaded with the manually entered top level information and fed by the mapped high level resources such as registries held/curated by participating RIs, the registry will provide a semantic network that will track the relations amongst entities that will allow for undertaking the RI maintenance actions of: registering resources, deduplication identities, copying data between hosts, inviting curation, transforming data/metadata to standards, aggregating and indexing data/metadata and communicating QoS request to service

providers and knowledge creators. It will provide the additional benefit of being able to do this across research infrastructures which entails possibilities of efficiencies in terms of identifying redundancies or already existing resources. Central (PARTHENOS) curators will be able to use the loaded registry in order to create plans for generating specialized integrations of data for projects and specific research communities.

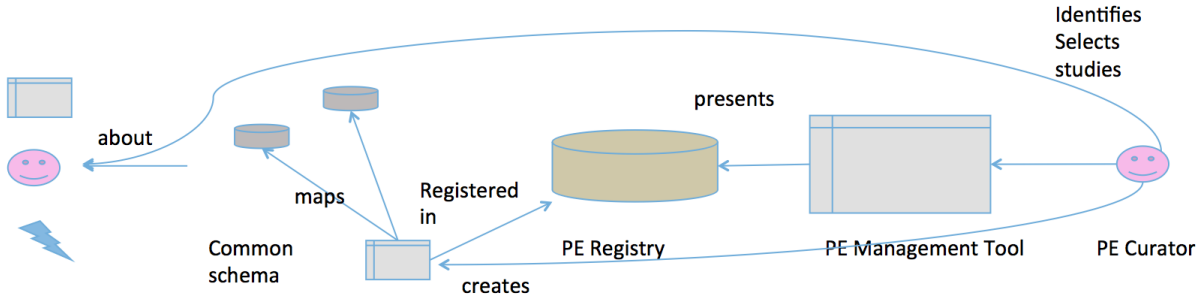

Fig. 17 Illustration of targeted mappings based on integration potentials identified from analysis of registry

PARTHENOS Curators will be able to assess the state of the registry making use of the RI Management console which will read from the semantic network in order to identify states of resources and take actions with regards to them. This could range from identifying services that have become unavailable and seeking the team responsible for bringing it back online, to identifying orphaned datasets that match to actor specializations in the network and inviting the adoption/curation of data, to determining appropriate subsets of data for creating indices of different types.

End users will not access directly to the registry but rather approach it through hybrid query systems, potentially embedded in individual, task oriented virtual research environments.

The general user will arrive at resources registered with PARTHENOS through a hybrid query system that will search both the registry and common indices in order to drive the user towards resources of interest to their research. The interface will follow the architectural vision of PARTHENOS insofar as it will always link resources back to the actors responsible for them and allow for the user to initiate requests for action to the resource managers such as requests for data cleaning, bug fixing, metadata enrichment etc. Actions taken on these requests will be done at source and potentially initiate an update to the overall semantic network.

Specialized users provisioned with virtual research environment services will have access to query services that include specialized indices. One important use case scenario is the provisioning of triple stores with deep integrations of mutually complementary and relevant data for some research projects. These users will be able to search to a highly granular level across relations in semantically mapped datasets as to integrate other datasets using mapping tools according to their research needs. Such environments will be the engines for the on-demand driven integration processes that the PARTHENOS architecture wishes to facilitate. Rather than providing a highly generic but specifically useless top level data structure, the registry and its minimal semantic model, guides researchers to relevant resources in their original formats. Once identified as relevant, researchers can undertake their own mappings processes in order to meet their specific needs. These activities, however, in turn generate new mappings documented in the registry which creates wider circles of compatibility across datasets now at an intra-domain level.

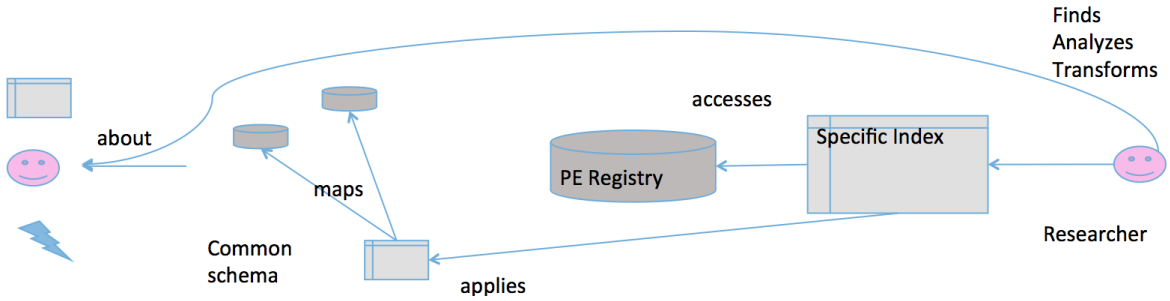

Fig. 18 Illustration of the researchers use of VREs to identify resources for specific integration and create on demand integrations and register these to the Registry

The architecture, therefore, provides a progressively expandable information management structure by creating a high level registry that represents resources and their relations at a generic level and provides the tools to create granular level integrations of datasets in an on-demand but traced environment. The architecture does not impose a common standard to all data and therefore leaves open ended the standards and information formats to which data transformations will be made.

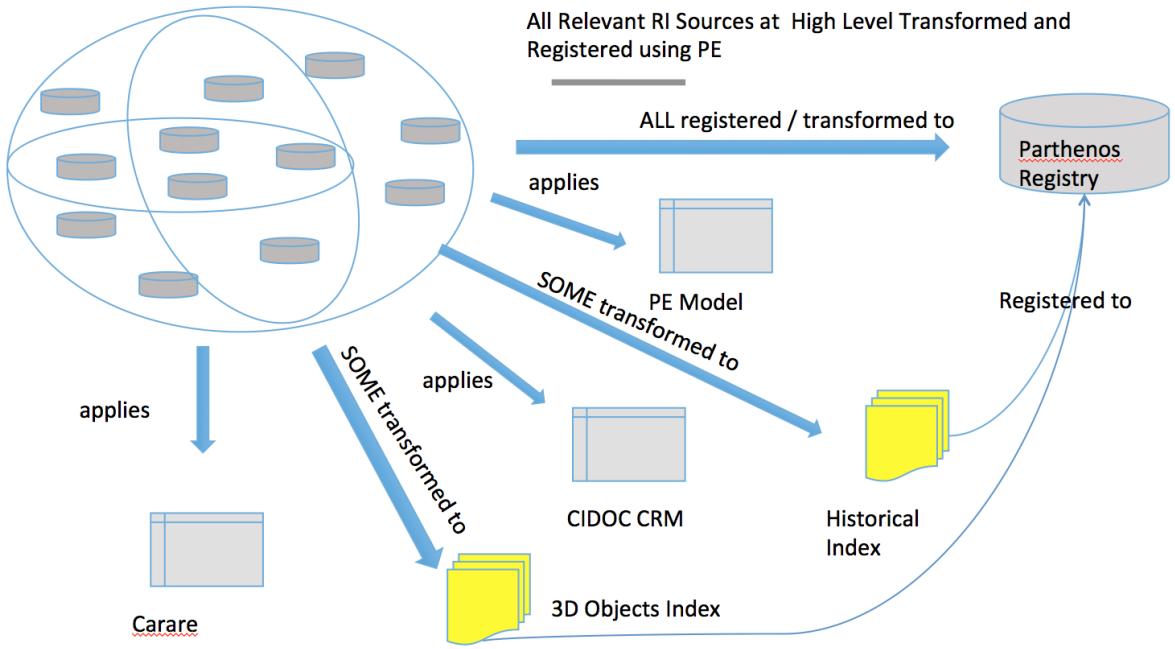

Fig. 19 Simplified Representation of multiply mapped data scenarios enabled by PARTHENOS architecture model

6. Conclusions

The PARTHENOS Project undertakes to face the problem of providing long term sustainable management of data integration across heterogeneous resources. The semantic model proposed in this deliverable consists of an architecture, ontology and minimal metadata registry that aims to support a strategy for tackling this problem. The architecture proposes a registry that tracks the highest level entities of a domain with minimal level commitment at the content level, but a clear awareness of the actors, services and the relations these facilitate to resources such as datasets and software. The architectural proposal takes as given that datasets are and will continue to be heterogeneous. It proposes to identify where regular resources are available in a content cloud and provide tracked mappings of these resources which can be used to generate a state of the art central resource at any time to support specific research purposes including but not limited to the registry itself. The proposed solution has been developed using a bottom up modelling strategy analyzing data inputs from the PARTHENOS projects various partners as gathered in T5.4. This modelled data is then tested by content provider users and implementation partners in WPs 5 and 6 directly. This feedback provides the necessary critique and testing in order to generate new iterations of the model and minimal metadata set at any one time. At the time of writing of this report the proposed model and

minimal metadata have been validated to the point of a first complete semantic mapping to the PARTHENOS Entities including generators to produce RDF data. The full test of the architecture and model will begin with the aggregation of full batches of data based on these initial mappings. In order to create proper integration on the data level, WP5/6 will require input from WP4 on standards in order to know against which official thesauri and name lists to normalize data.

7. Bibliography

Doerr, Martin. (2003). "The CIDOC conceptual reference module: an ontological approach to semantic interoperability of metadata." AI Magazine 24, no. 3: 75.

Doerr, M. (2005). “The CIDOC CRM, an Ontological Approach to Schema Heterogeneity. Semantic Interoperability and Integration” Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany (1862-4405).

Doerr, M., & Crofts, N. (1999). Electronic Esperanto: The Role of the Object Oriented CIDOC Reference Model. ICHIM, (pp. 157-173).

Doerr, M., & Hiebel, G.H (2013). “Where did the Varus battle take place? - A spatial refinement for the CIDOC CRM ontology.” Seventh World Archaeological Congress, The Dead Sea, Jordan, January 13th - 18th 2013.

Doerr, M., & Le Boeuf, P. (2007). “Modelling Intellectual Processes: The FRBR - CRM Harmonization.” DELOS Conference, Pisa, Italy, February 13-14 (pp. 114-123).

Doerr, M., & Theodoridou, M. (2011). “CRMdig: A generic digital provenance model for scientific observation.” TaPP’11, 3rd USENIX Workshop on the Theory and Practice of Provenance, Heraklion, Crete, Greece, June 20-21, 2011.

Niccolucci, F., & Doerr, M. (2013). CIDOC-CRM: the universal glue to connect diverse archaeological datasets. “The contributions of 3D-ICONS and ARIADNE.” Seventh World Archaeological Congress, The Dead Sea, Jordan, January 13th - 18th 2013.

Oldman, D., de Doerr, M., de Jong, G., Norton, B., & Wikman, T. (2014). “Realizing Lessons of the Last 20 Years: A Manifesto for Data Provisioning and Aggregation Services for the Digital Humanities (A Position Paper) System.” D-Lib Magazine, 20(7/8). https://doi.org/10.1045/july2014-oldman

Theodoridou, M., Bruseker, G., Daskalaki, M., & Doerr, M. (2016). “Methodological tips for mappings to CIDOC CRM.” 44th Computer Applications and Quantitative Methods in Archaeology Conference (CAA 2016) "Exploring Oceans of Data", Oslo, Norway, March 29 - April 2, 2016.

Tzompanaki, K., Doerr, M., Theodoridou, M., & Fundulaki, I. (2013). “Reasoning based on property propagation on CIDOC-CRM and CRMdig based repositories.” Proceedings of the Workshop Practical Experiences with CIDOC CRM and its Extensions co-located with the 17th International Conference on Theory and Practice of Digital Libraries (TPDL 2013), Valetta, Malta, September 26, 2013.

Appendix I: PARTHENOS Entities Model

Appendix II: PARTHENOS Entities Minimal Metadata

Appendix III: PARTHENOS Entities Model RDFS encoding

PARTHENOS Entities: Research Infrastructure Model

V2.0

FORTH-ICS

First Created: 25/5/2016
Update: 12/4/2017

Document History

Version/date	Date	Changes/approval	Author/Approved by
V 1.0	25/5/2016	Initial version	George, Martin
V 1.1	25/5/2016	Minor Editing	George
V 1.2	2/6/2016	Relation pe27 add	George
V 1.3	7/6/2016	Physical Curation and Hosting Classes removed, changed property names to ‘pp’ format from ‘pe’, added class curated thing	George
V 1.4	7/6/2016	Minor Editing	George
V 1.5	10/6/2016	Alignment with discussions from WP5/6 Joint Meeting in Crete	George
V1.6	11/7/2016	Corrections to document based on feedback from CNR. Corrections and feedback on model from Athina. Check of IsA relations on classes and properties. Fixed where necessary. Added names to all relations and classes in relation description tables. Added shortcut links to all relations for easier use of doc. Added more referred classes and relations from CIDOC CRM and CRMdig. Made extended names for repeated relation names like ‘has part’ in order to conform with GCube. Added class and relation hierarchy table for ease of	George, Leonardo Candela, Athina

		navigation of doc + better overview of model.	
V1.7	18/8/2016	Added three new properties to the model pp39 is metadata for and pp40 created successor of, pp41 is index of. These three properties are added in order to allow tracking and management of changes in metadata.	Doerr, Bruseker
V1.8	30/8/2016	Added new property pp42 'has declarative time'	Bruseker
V1.8.1	3/2/2017	Correction to document, indicating declaration of PE20 Volatile Digital Object as subclass of PE32 Curated Thing; Part of general alignment to RDFS 1.8.1 after corrections from WP6 team	Bruseker;Frosini
V1.9	13/2/2017	Introduction of new classes: PE33 E-Access Brokering Service PE34 Team PE35 Project PE36 Competency Type PE37 Protocol Type PE38 Schema And new relations: PP43 supported project activity (was project activity supported by) PP45 was competence (had competence of) PP46 brokered access to (had access brokered by) PP47 had protocol type (was protocol type of) PP48 used protocol parameter (was protocol parameter of) Updated:	Bruseker

		PE25 RI Consortium PP25 is maintained by (label) PE25 RI Consortium, now subclass of E34 Team, not E40 Legal Body PE26 RI Project now subclass of PE35 Project and not E7 Activity (directly)	
V1.10	10/3/2017	Harmonization with RDFS, updating all relation classes declared in PE to present tense format to reflect 'current state of knowledge' position.	Theodoridou; Bruseker
V1.11	14/3/2017	Changed PP45 has competence (is competence of) to PP45 has competency (is competency of)	Theodoridou; Bruseker
V2.0	12/4/2017	Removed draft classes judged unnecessary for PE model (related to provenance, to be expressed elsewhere)	Bruseker

PARTHENOS Entities: Research Infrastructure Model DRAFT

The PARTHENOS Entities (PE) propose an ontological model and RDF schema to encode data of use in supporting the activities and aims of research infrastructures to pool and connect services, software, datasets and to enable users of such services to reach the actors and understand the knowledge generation processes which generated the offered datasets. Research infrastructures integrate highly heterogeneous resources for an often equally heterogeneous audience. A central component of the activity of and RI in a digital environment involves building a data model that will support intuitive and accurate recall of information produced within the domain supported. It is the implicit or explicit belief of communities that organize into RIs that the integration of data from different members of the community offers not only the possibility of more efficient research and knowledge sharing but also the asking and answering of new questions by the crossing of data by sections of the community that normally would not consider their data in relation. Within this frame, PE proposes an ontological model that tries to capture the general basic entities deployed in building RI registries which is offered both as an intellectual tool for the checking and generation of such models and also as a means to create a common expression by which data could be shared across research communities, thus creating an RI of RIs. Such an effort is a logical extension of the belief inherent to individual research communities but broadened to an interdisciplinary scale.

PE is modelled as an extension of CIDOC CRM, the ISO standard ontology for cultural heritage data, and CRMdig, an extension of the latter which models provenance information in digitization processes. In this way, the modelling of a minimal metadata set for use in a registry as proposed above can be complimented by full modelling of detailed datasets in order to provide a rich web of data that can be accessed from the starting point of an RI registry. CIDOC CRM with its open extension policy and support of analytic data generated by empirical sciences with regards to the human past provides a suitably general ontology to allow for the integration of data across a wide spread of humanities and scientific disciplines.

PE is being developed in the context of the PARTHENOS Project, a European funded project.

Table of Contents

Hierarchies

Class Hierarchy

#	IsA Hierarchy	Orig.	Reg?
E7	Activity	CRM	N
PE35	Project	PE	Y
PE26	RI Project	PE	Y
PE27	Service Action [Draft]	PE	Y
PE1	Service	PE	Y
PE2	Hosting Service	PE	Y
PE5	Digital Hosting Service	PE	Y
PE6	Software Hosting Service	PE	Y
PE13	Software Computing E-Service	PE	Y
PE16	Curated Software E-Service	PE	Y
PE14	Software Delivery E-Service	PE	Y
PE16	Curated Software E-Service	PE	Y
PE7	Data Hosting Service	PE	Y
PE15	Data E-Service	PE	Y
PE17	Curated Data E-Service	PE	Y
PE3	Curating Service	PE	Y
PE10	Digital Curating Service	PE	Y
PE11	Software Curating Service	PE	Y
PE16	Curated Software E-Service	PE	Y
PE12	Data Curating Service	PE	Y
PE17	Curated Data E-Service	PE	Y
PE8	E-Service	PE	Y
PE33	E-Access Brokering Service	PE	
PE13	Software Computing E-Service	PE	Y
PE16	Curated Software E-Service	PE	Y
PE14	Software Delivery E-Service	PE	Y
PE16	Curated Software E-Service	PE	Y
PE15	Data E-Service	PE	Y
PE17	Curated Data E-Service	PE	Y
E65	Creation	CRM	N
E77	Permanent Item	CRM	N
E39	Actor	CRM	Y
E74	Group		
E40	Legal Body	CRM	Y
PE34	Team		
PE25	RI Consortium	PE	Y
E70	Thing	CRM	Y
PE32	Curated Thing	PE	Y
E78	Curated Holding	CRM	Y
PE20	Volatile Digital Object	PE	Y
E71	Man Made Thing	CRM	N
E24	Physical Man Made Thing	CRM	N
E78	Curated Holding	CRM	Y
E28	Conceptual Object	CRM	N
E55	Type		
PE36	Competency Type		
PE37	Protocol Type		
E89	Propositional Object	CRM	N
E73	Information Object	CRM	N
E29	Design or Procedure	CRM	N

PE28	Curation Plan	PE	Y
D1	Digital Object	dig	Y
PE19	Persistent Digital Object	PE	Y
PE21	Persistent Software	PE	Y
PE22	Persistent Dataset	PE	Y
PE20	Volatile Digital Object	PE	Y
PE23	Volatile Software	PE	Y
PE24	Volatile Dataset	PE	Y
PE18	Dataset	PE	Y
PE22	Persistent Dataset	PE	Y
PE24	Volatile Dataset	PE	Y
D14	Software	dig	Y
PE21	Persistent Software	PE	Y
PE38	Schema		
PE23	Volatile Software	PE	Y

Relations Hierarchy

#	Hierarchy	Domain	Range	Origin
P1	is identified by	E1 CRM Entity	E41 Appelation	CRM
PP28	has designated access point (is designated access point of)	PE8 E-Service	PE29 Access Point	PE
PP50	Accessible at (provides access to)	D1 Digital Object	PE29 Access Point	
P9	consists of (forms part of)	E4 Period	E4 Period	CRM
PP1	currently offers (currently offered by)	PE26 RI Project	PE1 Service	PE
PP43	supports project activity (is project activity supported by)	PE35 Project	E7 Activity	
PP38	Executes (is executed by)	PE1 Service	PE27 Service Action	PE
P14	carried out by (performed)	E7 Activity	E39 Actor	CRM
PP2	Provided by (provides)	PE1 Service	E39 Actor	PE
PP3	Requested by (requests)	PE27 Service Action	E39 Actor	PE
P15	was influenced by (influenced)	E7 Activity	E1 CRM Entity	CRM
PP44	has maintaining team (is maintaining team of)	PE35 Project	PE34 Team	
PP25	has maintaining RI (is maintaining RI of)	PE26 RI Project	PE25 RI Consortium	PE
P16	used specific object (was used for)	E7 Activity	E70 Thing	CRM
PP4	hosts object (is object hosted by)	PE2 Hosting Service	E70 Thing	PE
PP6	hosts digital object (is digital object hosted by)	PE5 Digital Hosting Service	D1 Digital Object	PE
PP7	hosts software object (is software object hosted by)	PE6 Software Hosting Service	D14 Software	PE
PP8	hosts dataset (is dataset hosted by)	PE7 Data Hosting Service	PE18 Dataset	PE
PP14	runs on request (is run by)	PE13 Software Computing E-Service	D14 Software	PE
PP15	delivers on request (is delivered by)	PE14 Software Delivery E-Service	D14 Software	PE
PP29	uses access protocol (is access protocol used by)	PE8 E-Service	D14 Software	PE
PP48	uses protocol parameter (is protocol parameter of)	PE8 E-Service	PE38 Schema	
PP49	provides access point (is access point provided by)	PE8 E-Service	E29 Access Point	
PP40	created successor of (is deprecated by)	E65 Creation	PE22 Persistent Dataset	PE
P21	had general purpose (was purpose of)	E7 Activity	E55 Type	
PP45	has competency (is competency of)	PE1 Service	PE36 Competency Type	
PP32	curates (is curated by)	PE3 Curating Service	PE32 Curated Thing	PE
PP11	curates volatile digital object (is volatile digital object curated by)	PE10 Digital Curating Service	PE20 Volatile Digital Object	PE
PP12	curates volatile software (is volatile software curated by)	PE11 Software Curating Service	PE23 Volatile Software	PE

PP13	curates volatile dataset (is volatile dataset curated by)	PE12 Data Curating Service	PE24 Volatile Dataset	PE
P147	curated (was curated by)	E87 Curation Activity	E78 Curated Holding	CRM
P33	used specific technique (was used by)	E7 Activity	E29 Design or Procedure	CRM
PP31	uses curation plan (is curation plan used by)	PE3 Curating Service	PE28 Curation Plan	PE
P106	is composed of (forms part of)	E90 Symbolic Object	E90 Symbolic Object	CRM
PP16	has persistent digital object part (is persistent digital object part of)	PE19 Persistent Digital Object	PE19 Persistent Digital Object	PE
PP19	has persistent software part (is persistent software part of)	PE21 Persistent Software	PE21 Persistent Software	PE
PP20	has persistent dataset part (is persistent dataset part of)	PE22 Persistent Dataset	PE22 Persistent Dataset	PE
PP18	has digital object part (is digital object part of)	PE20 Volatile Digital Object	D1 Digital Object	PE
PP21	has software part (is software part of)	PE23 Volatile Software	D14 Software	PE
PP23	has dataset part (is dataset part of)	PE24 Volatile Dataset	PE18 Dataset	PE
P125	Used object of type (was type of object used in)	E7 Activity	E55 Type	
PP47	has protocol type (is protocol type of)	PE8 E-Service	PE37 Protocol Type	
P129	is about (is subject of)	E89 Propositional Object	E1 CRM Entity	CRM
PP39	is metadata for (has metadata)	PE22 Persistent Dataset	D1 Digital Object	PE
P130	shows features of (features also found on)	E70 Thing	E70 Thing	CRM
PP17	has snapshot (is snapshot of)	PE20 Volatile Digital Object	PE19 Persistent Digital Object	PE
PP22	has release (is release of)	PE23 Volatile Software	PE21 Persistent Software	PE
PP24	has dataset snapshot (is dataset snapshot of)	PE24 Volatile Dataset	PE22 Persistent Dataset	PE
PP46	brokers access to (access brokered by)	PE33 E-Acces Brokering Service	PE8 E-Service	

Classes

PE1 Service

Class Label	PE1 Service
Subclass of	E7 Activity
Superclass of	PE2 Hosting Service PE3 Curating Service PE8 E-Service
Scope Note	This class comprises declared offers by some instance of E39Actor of their willingness and ability to execute an activity or series of activities at the request of another instance of E39 Actor for the specific benefit of the latter. The identity of a service therefore depends on the individual instance of E39 actor making the offer, the type of activity(ies) offered and/or the type of product resultant from such an activity(ies). An instance of a PE1 Service begins to exist with the declaration of the ability and willingness of an instance of E39 actor to perform the particular set of activities for the benefit of another actor. The instance of PE1 Service ends when either the declared willingness or ability permanently ends. N.B.: this means that the ability may temporarily be interrupted, such as when an actor is on vacation or where the machine on which the service relies is being repaired, without meaning that the service as such has ended. A service need not continually be running in order for it be considered to be continuous, for example a service may be defined to fall within certain working hours. The instance of PE1 Service includes all auxiliary abilities of the same actor to execute the respective activities, but not services provided by third parties in the course of the service provisioning.
Examples	The local car repair shop's car repair services.

New Direct Properties

Label	Domain	Range	Scope Note
PP38 executes	PE1	PE27	Links to the Service Action carried out in fulfilment of the Service Offer
PP45 has competency	PE1	PE36	Links a Service to the instance of competency type which it is intended to fulfill.

PE2 Hosting Service

Class Label	PE2 Hosting Service
Subclass of	PE1 Service
Superclass of	PE 5 Digital Hosting Service
Scope Note	This class comprises declared offers by some instance of E39Actorto hold, protect and provide access to one or more objects in a generic sense, either physical or conceptual, at the request of an instance of E39 Actor, where the latter may be the initial party or a second party. An instance of PE2 Hosting Service begins from the moment of agreement between the contracting parties that the host will carry out these holding and protection activities in order to provide access, upon request, to some instance or instances of E70 Thing for the sake of the client. The hosting services continue so long as the hosting actor retains the ability to provide access to the object(s) to the client. The instance of hosting service ends when the host is either no longer willing or able to provide access to the objects that they undertook to hold and protect for the client.
Examples	Amazon cloud hosting of a user's files [PE5]

New Direct Properties

Label	Domain	Range	Scope Note
PP4 hosts object	PE2	E70	Link to the object(s) (generic) of which the service is a host

PE3 Curating Service

Class Label	PE3 Curating Service
Subclass of	PE1 Service
Superclass of	PE10 Digital Curating Service
Scope Note	This class comprises declared offers by some instance of E39 Actor of their willingness and ability to engage in a series of selection and organization activities on a collection of objects according to a specified plan. The identity of the curation service is tied to the collection of which it is the curator. A curation service comes into existence for the curation of some determinate collection taken as a whole, and is further determined in its identity by provider of the service and the plan which is adopted in order to carry out the curation. It is, in particular, the nature of the object of curation to be a collection in the sense of a plurality of objects from which parts can be added or removed without the overall identity of that collection being changed. An instance of PE3 Curating Service begins when the curator initiates the selection and organization of a collection of objects under the declared

	curation plan. The curating service may take over the curation of an existing collection or begin the curation of a new collection. So as long as the curator maintains these selecting and organizing activities of these objects according to the declared plan, the curation activity is considered on-going, regardless of any particular activities or lack thereof at any one time. Should the actor no longer be willing to engage in these activities or the objects be unavailable in a permanent manner, then the instance of PE3 Curating Service is to be considered ended. While curated objects may need to be hosted, this service may or may not be undertaken by the same actor. Therefore hosting can be documented separately and attributed to the appropriate third party actor.
Examples	Curation of the Collection of Ancient Greek Art by Nikolas Papadimitriou at the Museum of Cycladic Art

Inherited / Mapped Properties

Label	Domain	Range	Scope Note
PP31 uses curation plan	PE3	PE28	Link to a document indicating the curation plan
PP32 curates	PE3	PE32	Indicates the collection of objects which this service curates

PE5 Digital Hosting Service

Class Label	PE5 Digital Hosting Service
Subclass of	PE2 Hosting Service
Superclass of	PE6 Software Hosting Service PE7 Data Hosting Service
Scope Note	This class comprises declared offers by some instance of E39Actorto hold, protect and provide access to one or more digital objects at the request of an instance of E39 Actor. The identity of digital hosting is determined by the type of object that the host undertakes to keep and provide access to. The hosting is digital in the sense that the object being held and protected is of a digital nature. Digital hosting does not entail the running of machines and software. An instance of PE5Digital Hosting Service begins from the moment of agreement between the contracting parties that the host will carry out these holding and protection activities in order to provide access, upon request, to some instance or instances of D1Digital Object for the sake of the client. Digital hosting services continue so long as the hosting actor retains the ability to provide access to the hosted object(s) to the client. The instance of

	hosting service ends when the host is either no longer willing or able to provide access to the object or collection of objects that they undertook to hold and protect for the client.
Examples	Google Art hosting of the digital images of the collections of Mathaf: the Arab Museum of Modern Art

New Direct Properties

Label	Domain	Range	Scope Note
PP6 hosts digital object	PE5	D1	Link to the digital object(s) (generic) of which the service is a host

PE6 Software Hosting Service

Class Label	PE6 Software Hosting Service
Subclass of	PE5 Digital Hosting Service
Superclass of	PE13 Software Computing E-Service PE14 Software Delivery E-Service
Scope Note	This class comprises declared offers by some instance of E39Actor to hold and protect one or more software objects at the request of an instance of E39 Actor. The identity of software hosting is determined by the type of object that the host undertakes to keep and provide access to. The hosting is an instance of PE6 Software Hosting Service, just in case the object or objects which are held and protected are software. Software hosting does not entail the running of machines and software. An instance of PE6 Software Hosting Service begins from the moment of agreement between the contracting parties that the host will carry out these holding and protection activities in order to provide access, upon request, to some instance or instances of D14 Software for the sake of the client. Digital hosting services continue so long as the hosting actor retains the ability to provide access to the hosted object(s) to the client. The instance of hosting service ends when the host is either no longer willing or able to provide access to the object or collection of objects that they undertook to hold and protect for the client.
Examples	Hosting of the “Historical Software Collection” by archive.org

New Direct Properties

Label	Domain	Range	Scope Note
PP7 hosts software object	PE6	D14	Link to the software(s) of which the service is a host

PE7 Data Hosting Service

Class Label	PE7 Data Hosting Service
Subclass of	PE5 Digital Hosting Service
Superclass of	PE15 Data E-Service
Scope Note	This class comprises declared offers by some instance of E39Actor to hold and protect one or more datasets at the request of an instance of E39 Actor. The identity of data hosting is determined by the type of object that the host undertakes to keep and provide access to. The hosting is an instance of PE7 Data Hosting Service, just in case the object or objects which are held and protected are dataset. Data hosting does not entail the running of machines and software. An instance of PE7 Data Hosting Service begins from the moment of agreement between the contracting parties that the host will carry out these holding and protection activities in order to provide access, upon request, to some instance or instances of PE18 Dataset for the sake of the client. Digital hosting services continue so long as the hosting actor retains the ability to provide access to the hosted object(s) to the client. The instance of hosting service ends when the host is either no longer willing or able to provide access to the object or collection of objects that they undertook to hold and protect for the client.
Examples	Archaeological Data Service's Hosting of project data for the "Church Wilne Deserted Medieval Settlement, Derbyshire"

New Direct Properties

Label	Domain	Range	Scope Note
PP8 hosts dataset	PE7	PE18	Link to the dataset(s) of which the service is a host

PE8 E-Service

Class Label	PE8 E-Service
Subclass of	PE1 Service
Superclass of	PE13 Software Computing E-Service

	PE14 Software Delivery E-Service PE15 Data E-Service PE33 E-Access Brokering Service
Scope Note	This class comprises declared offers to provide computing facilities by some instance of an E39 Actor who provisions a hardware/software setup that is able to respond to the use requests of some E39 Actor through automated receipt, manipulation and sending of data. The identity of an instance of PE8 E-Service depends on the particular communication software it runs, the actor maintaining the service active, and the logical communication address for issuing requests to it. An instance of PE8 E-Service comes into existence on the declaration of its offer and the making available of the service through some access point. It ceases to exist just in case the instance of E39 Actor is no longer willing or able to maintain the e-service when, for example an organization ceases to operation entirely, cancels the particular service, or is no longer able to support the software/hardware entailed.
Examples	IBM quantum computing service to quantum computing researchers Offer of community user to SETI project of computational power as part of its grid computing effort

New Direct Properties

Label	Domain	Range	Scope Note
PP28 has designated access point	PE8	PE29	Link to the web address at which the e-service can be accessed
PP29 uses access protocol	PE8	D14	Links the service to the access protocol, considered as a form of software, which it invokes
PP47 has protocol type	PE8	PE37	Link the E-Service to a type of protocol used for accessing it
PP48 uses protocol parameter	PE8	PE38	Link the E-Service to an instance of PE38 Schema that must be invoked during the access protocol
PP49 provides access point	PE8	PE29	Links the E-Service to an instance of PE29 Access Point that provides access to an instance of D1 Digital Object in an information network.

PE10 Digital Curating Service

Class Label	PE10 Digital Curating Service
Subclass of	PE3 Curating Service

Superclass of	PE11 Software Curating Service PE12 Data Curating Service
Scope Note	This class comprises declared offers by some instance of E39 Actor of their willingness and ability to engage in a series of selection and organization activities on an instance of PE20 Volatile Digital Object according to a specified plan. The identity of the instance of PE10 Digital Curation Service is tied to the instance of PE20 Volatile Digital Object of which it is the curation. Instances of PE20 Volatile Digital Object are by their nature composites of different data sources. The curation activity on the volatile digital object in executing its plan for the volatile digital object - some functional goal - ensures the unity of the one volatile digital object and provides it an identity. Thus again, as with physical curation of a collection, it is normal for parts to be added or removed from the volatile digital object without its overall identity changing. It is precisely having this one object of the digital curation service that in turn allows the identification of the service itself, alongside knowledge of the curator and the plan. An instance of PE10 Digital Curating Service begins when the curator initiates the selection and organization of a volatile digital object under the declared curation plan. The curating service may take over the curation of an existing volatile digital object or begin the curation of an entirely new volatile digital object. As long as the curator maintains the will and ability to carry out these selecting and organizing activities according to the declared plan, the curation activity is considered on-going, regardless of any particular activities or lack thereof at any one time. Should the actor no longer be willing to engage in these activities or the volatile digital object be unavailable in a permanent manner, then the instance of PE10 Digital Curating Service is to be considered ended. While curated objects may need to be hosted, this service may or may not be undertaken by the same actor. Therefore hosting can be documented separately and attributed to the appropriate third party actor.
Examples	

Properties

Label	Domain	Range	Scope Note
PP11 curates volatile digital object	PE10	PE20	Link to the digital objects which the service curates

PE11Software Curating Service

Class Label	PE11 Software Curating Service
Subclass of	PE10 Digital Curating Service

Superclass of	PE16 Curated Software E-Service
Scope Note	This class comprises declared offers by some instance of E39 Actor of their willingness and ability to engage in a series of selection and organization activities on an instance of PE23 Volatile Software according to a specified plan. The identity of the instance of PE11 Software Curation Service is tied to the instance of PE23 Volatile Software of which it is the curation. Instances of PE23 Volatile Software are by their nature composites of different data sources. The curation activity on the volatile software in executing its plan for the volatile software - some functional goal - ensures its unity and provides it an identity. Thus again, as with physical curation of a collection, it is normal for parts to be added or removed from the volatile software object without its overall identity changing. It is precisely having this one object of the software curation service that, in turn, allows the identification of the service itself, alongside knowledge of the curator and the plan. An instance of PE11Software Curating Service begins when the curator initiates the selection and organization of a volatile software object under the declared curation plan. The curating service may take over the curation of an existing volatile software object or begin the curation of an entirely new volatile software object. As long as the curator maintains the will and ability to carry out these selecting and organizing activities according to the declared plan, the curation activity is considered on-going, regardless of any particular activities or lack thereof at any one time. Should the actor no longer be willing to engage in these activities or the volatile digital object be unavailable in a permanent manner, then the instance of PE11Software Curating Service is to be considered ended. While curated objects may need to be hosted, this service may or may not be undertaken by the same actor. Therefore hosting can be documented separately and attributed to the appropriate third party actor.
Examples	

Properties

Label	Domain	Range	Scope Note
PP12 curates volatile software object	PE11	PE23	Link to the volatile software object that is curated by this activity

PE12 Data Curating Service

Class Label	PE12 Data Curating Service
Subclass of	PE10 Digital Curating Service
Superclass of	PE17 Curated Data E-Service

Scope Note	This class comprises declared offers by some instance of E39 Actor of their willingness and ability to engage in a series of selection and organization activities on an instance of PE24 Volatile Dataset according to a specified plan. The identity of the instance of PE12 Data Curating Service is tied to the instance of PE24 Volatile Dataset of which it is the curation. Instances of PE24 Volatile Dataset are by their nature composites of different data sources. The curation activity on the volatile dataset in executing its plan for the volatile software - some functional goal - ensures its unity and provides it an identity. Thus again, as with physical curation of a collection, it is normal for parts to be added or removed from the volatile software object without its overall identity changing. It is precisely having this one object of the software curation service that, in turn, allows the identification of the service itself, alongside knowledge of the curator and the plan. An instance of Data Curating Service begins when the curator initiates the selection and organization of a volatile dataset under the declared curation plan. The curating service may take over the curation of an existing volatile dataset or begin the curation of an entirely new volatile dataset. As long as the curator maintains the will and ability to carry out these selecting and organizing activities according to the declared plan, the curation activity is considered on-going, regardless of any particular activities or lack thereof at any one time. Should the actor no longer be willing to engage in these activities or the volatile digital object be unavailable in a permanent manner, then the instance of Data Curating Service is to be considered ended. While curated objects may need to be hosted, this service may or may not be undertaken by the same actor. Therefore hosting can be documented separately and attributed to the appropriate third party actor.
Examples	

Properties

Label	Domain	Range	Scope Note
PP13 curates volatile dataset	PE12	PE24	Link to the volatile software object that is curated by this activity

PE13 Software Computing E-Service

Class Label	PE13 Software Computing E-Service
Subclass of	PE6 Software Hosting Service PE8 E-Service
Superclass of	PE16 Curated Software E-Service
Scope Note	This class comprises instances of offers that are made up of both instances of PE6 Software Hosting and PE8 E-Service while additionally offering the

	ability and willingness to run certain software for the requesting instance of E39 Actor. That is to say, the service provider takes on duties of hosting software, running the equipment to provide it, and delivering computing power to run it on request. The identity of this service is likewise composite depending on those factors relevant to instances of PE6 Software Hosting Service and PE8 E-Service, while additionally requiring that we have a clear identity of the software. The software release that the service runs may change without affecting the identity of the overall service, but to retain its identity this change would need to be documented in the access protocol, and to be archived in a log file. If an E39 Actor provides software computing e-services that run more than one software release at the same time, each of these should be documented as a separate instance of PE13 Software Computing E-Service. The processing software is not regarded as part of the service, but as being used by the service. An instance of PE13 Software Computing E-Service comes into existence on the declaration of its offer and the making available of the service along with the software it offers to run through some access point. It ceases to exist just in case the instance of E39 Actor is no longer willing or able to maintain the service when, for example if an organization ceases operation entirely, or the particular service is abandoned, if the software provisioned is permanently unavailable, or the host is no longer able to support the software/hardware entailed in providing the computing service.
Examples	The provisioning of Google Doc Service to clients by Google

Properties

Label	Domain	Range	Scope Note
PP14 runs on request	PE13	D14	This property associates an instance of software computing e-service with the software that it runs when requested.

PE14 Software Delivery E-Service

Class Label	PE14 Software Delivery E-Service
Subclass of	PE6 Software Hosting Service PE8 E-Service
Superclass of	PE16 Curated Software E-Service
Scope Note	This class comprises instances of offers that are made up of both instances of PE6 Software Hosting and PE8 E-Service while additionally offering the

	ability and willingness to deliver a particular piece of software to the requesting instance of E39 Actor. That is to say, the service provider takes on duties of hosting software, running the equipment to provide it, and delivering software on demand to a client. The identity of this service is likewise composite depending on those factors relevant to instances of PE6 Software Hosting Service and PE8 E-Service, while additionally requiring that we have a clear identity of the software to be delivered. The software release that the service delivers may change without affecting the identity of the overall service, but to retain its identity this change would need to be documented in the access protocol, and to be archived in a log file. If an E39 Actor provides e-services that deliver more than one software release at the same time, each of these should be documented as a separate instance of PE13 Software Computing E-Service. The processing software is not regarded as part of the service, but as being used by the service. An instance of PE14 Software Delivery E-Service comes into existence on the declaration of its offer and the making available of the service along with the software it offers to deliver through some access point. It ceases to exist just in case the instance of E39 Actor is no longer willing or able to maintain the service when, for example if an organization ceases operation entirely, or the particular service is abandoned, if the software provisioned is permanently unavailable, or the host is no longer able to support the software/hardware entailed in providing the computing service.
Examples	The offer of Github to a client to store his/her software and deliver it to other users

Properties

Label	Domain	Range	Scope Note
PP15 delivers on request	PE14	D14	This property associates an instance of software delivery e-service with the software that it delivers when requested.

PE15 Data E-Service

Class Label	PE15 Data E-Service
Subclass of	PE7 Data Hosting Service PE8 E-Service
Superclass of	PE17 Curated Data E-Service

Scope Note	This class comprises instances of offers that are made up of both instances of PE7 Data Hosting and PE8 E-Service while additionally offering the ability and willingness to offer electronic access to one or more datasets to the requesting instance of E39 Actor. That is to say, the service provider takes on duties of both hosting dataset(s) while running the equipment to provide access to the same. The identity of this service is a composite of those factors relevant to instances of PE7Data Hosting Service and PE8 E-Service. An instance of PE15Data E-Service comes into existence on the declaration of its offer and the making available of the service along with the dataset it aims to provide access to through some access point. It ceases to exist just in case the instance of E39 Actor is no longer willing or able to maintain the service when, for example if an organization ceases operation entirely, or the particular dataset is permanently unavailable, or the host is no longer able to support the software/hardware entailed in providing the computing service.
Examples	Offer of the British School at Athens of e-access to the digitized collection of the Byzantine Research Fund

PE16 Curated Software E-Service

Class Label	**PE16 Curated Software E-Service**
Subclass of	PE11 Software Curating Service PE14 Software Delivery E-Service PE13 Software Computing E-Service
Superclass of	-
Scope Note	This class comprises instances of offers that are made up of both instances of PE11 Software Curating Service and PE14S/W Delivery E-Service or PE13 Software Computing E-Service. Here then we speak of an offer both to curate some software(s), host it and running the equipment enabling its delivery to or running for clients. The identity of an instance of PE16Curated Software E-Service depends thus on the actor providing the service, the software hosted and curated, as well as the particular processing software its E-service component runs, as well as the logical communication address for issuing requests to it. The software release the service delivers or runs may change without affecting the identity of the overall service, but to retain its identity this change would need to be documented in the access protocol, and to be archived in a log file. An instance of PE16 Curated Software E-Service comes into existence on the declaration of its offer and the making available of the service along with

	the software it curates and delivers/runs through some access point. It ceases to exist just in case the instance of E39 Actor is no longer willing or able to maintain the service when, for example if an organization ceases operation entirely, or the particular service is abandoned, if the software to be hosted and curated is lost, or the host/curator is no longer able to support the software/hardware entailed in providing the delivery service.
Examples	Lyrasis offer of online hosted "Collection Space" collection management software of which they are also the developers/curators

PE17 Curated Data E-Service

Class Label	PE17 Curated Data E-Service
Subclass of	PE12 Data Curating Service PE15 Data E-Service
Superclass of	-
Scope Note	This class comprises instances of offers that are made up of both instances of PE12 Data Curating Service and PE15 Data E-Service. Here then we speak of an offer to curate some volatile dataset, host it and run the equipment necessary in order for clients to be able to access it electronically on demand. The identity of an instance of PE17 Curated Data E-Service depends thus on the actor providing the service, the dataset hosted and curated, the particular processing software its E-service component runs, as well as the logical communication address for issuing requests to it. An instance of PE17 Curated Data E-Service comes into existence on the declaration of its offer and the making available of the service along with the data it curates and provides access to through some access point. It ceases to exist just in case the instance of E39 Actor is no longer willing or able to maintain the service when, for example if an organization ceases operation entirely, or the particular service is abandoned, if the dataset to be hosted and curated is lost, or the host/curator is no longer able to support the software/hardware entailed in providing the delivery service.
Examples	Spotify custom crafted playlist for Spotify user

PE18 Dataset

Class Label	PE18 Dataset
Subclass of	D1 Digital Object
Superclass of	PE22 Persistent Dataset PE24 Volatile Dataset
Scope Note	This class comprises identifiable immaterial items that can be represented

	as sets of bit sequences and whose content contains propositions about the objective world. The identity of an instance of PE18 is determined by its content in bit level encoding alongside its provenance. Any instance of a dataset may be composed of many distinct parts of other identifiable datasets. An aggregate of instances of PE18 dataset is treated as one instance and its parts can be documented as having a part of relation (p106). Datasets in practice are either volatile or persistent.
Examples	The collections database of the Qatar Museum Authority A 3D model of the Asinou Church in Crete

PE19 Persistent Digital Object

Class Label	**PE19 Persistent Digital Object**
Subclass of	D1 Digital Object
Superclass of	PE21 Persistent Software PE22 Persistent Dataset
Scope Note	This class comprises instances of D1 digital object which are the result of a distinct creation moment in which the whole of the content of the digital object as a propositional set was established and encoded at a bit level, whether this creation moment is known or not. Persistent digital objects are thus identified by their content, bit level encoding and the moment of production as a whole unit of information. An instance of persistent digital object continues to exist so long as one copy of it remains on one carrier which has been maintained without change to its internal content, thus propagating the original condition of the instance.
Examples	Version 5.2 of Microsoft DOS Backup file of the shared drive at FORTH Submitted copy of deliverable 5.1 in word format

Properties

Label	**Domain**	**Range**	**Scope Note**
PP16 has persistent	PE19	PE19	Indicates the persistent digital object part which is a part of the persistent digital object, used as abstract relation

digital object part	to form restriction for persistent software and dataset relations

PE20 Volatile Digital Object

Class Label	PE20 Volatile Digital Object
Subclass of	PE32 Curated Thing D1 Digital Object
Superclass of	PE23 Volatile Software PE24 Volatile Dataset
Scope Note	This class comprises instances of digital objects whose content is subject to continuous change without notice or necessary archiving of intermediate state but which can be considered as one with regards to its provenance in some curation plan that determines its information, goal and subject coverage. At any one point, an instance of PE20 Volatile Digital can be identified by an official snapshot of the actual data stream, an instance of PE19 Persistent Digital Object, taken by the responsible curating authority which has as ancestors any previous snapshots taken of the data stream. The curator assigns a persistent identifier to the official snapshot and is the only individual who can identify the true representative snapshot. Reference to the content of an instance of PE20 Volatile Digital Object is down by way of the official snapshot.
Examples	The catalogue of iTunes Store music offerings The Archive of Archaeological Data Service UK

Properties

Label	Domain	Range	Scope Note
PP17 has snapshot	PE20	PE19	Indicates the snapshot that is representative of the volatile data object according to a curator.
PP18 has volatile digital object part	PE20	D1	Indicates that a volatile data object can have as part some other volatile dataset

PE21 Persistent Software

Class Label	PE21 Persistent Software
Subclass of	D14 Software PE19 Persistent Digital Object
Superclass of	

Scope Note	This class compromises instances of digital objects that that can be executed on a computer to perform specific operations. In particular, an instance of PE21 Persistent software is the necessary information to process datasets algorithmically and to transform or integrate datasets in a collaborative infrastructure. The identity of software depends on its content on the bit-level of encoding. The validity of the results produced by the software's application depends categorically on its algorithmic correctness. A software release is defined as an instance of software. The software release begins to exist with its provision by the actor who is responsible for producing it. We also include in this category all data structures and formal ontologies that are used to configure the behavior of the software at an infrastructure component level.
Examples	Sketchup

Properties

Label	Domain	Range	Scope Note
PP19 has persistent software part	P21	P21	Indicates the persistent software of which the software instance may be composed

PE22 Persistent Dataset

Class Label	PE22 Persistent Dataset
Subclass of	PE18 Dataset PE19 Persistent Digital Object
Superclass of	
Scope Note	This class compromises datasets that contain collections of data, records or information kept as a persistent unit of information in the knowledge generation process from primary records up to any level of aggregation or integration. The identity of a dataset is given by its content on the bit-level of encoding and its provenance. Since large datasets have a very small chance to be "reinvented" with another meaning, it is often practical to base the identity of a dataset on the content only, and apply a respective disambiguation of provenance only in case of obviously accidental identity. Different versions of a dataset are regarded as different datasets. Their relation should be defined by metadata describing the derivation process, rather than by version numbers. In general, a dataset may be integrated from different sources of provenance, such as a corpus of inscriptions compiled from different publication or a snapshot of a complete digital library. The integrated

	dataset may preserve the units of information of the source from which it has taken components. The content of knowledge organization systems, such as gazetteers, author lists, thesauri and formal ontologies of terms at a particular point in time, fall under datasets.
Examples	Records of the Excavations at 198 High Street, Exeter (Exeter archive site 55)

Properties

Label	Domain	Range	Scope Note
PP20 has persistent dataset part	PE22	PE22	Indicates the persistent dataset of which the dataset may be composed
PP39 is metadata for	PE22	D1	Indicates that this instance of PE22 Dataset acts as a metadata set for the range instance of D1 Digital Object.

PE23 Volatile Software

Class Label	**PE23 Volatile Software**
Subclass of	D14 Software PE20 Volatile Digital Object
Superclass of	
Scope Note	This class comprises software that is in the process of active development volatile software class is comprised of instances of the working copy of some software in development. The software in development is the necessary information to perform specific operations. The identity of an instance of PE23 Volatile Software depends on the unity provided it by the instance of PE11 Software Curating Service responsible for it, that provides it its unity of purpose. The PE11 Software Curating Service is responsible for the creation of instances of PE21 Persistent Software which are the official release of this development stream and the ability to find and run its instructions at some time.
Examples	Source code of development of Sketchup

Properties

Label	Domain	Range	Scope Note
PP21 has software part	PE23	D14	Indicates the software, persistent or volatile, that forms a part of this volatile software
PP22 has release	PE23	PE21	Indicates the persistent software version released at some point which stands as an identifier and usable product from the volatile software production

PE24 Volatile Dataset

Class Label	**PE24 Volatile Dataset**

Subclass of	PE18 Dataset PE20 Volatile Digital Object
Superclass of	
Scope Note	This class comprises datasets that are changed without notice or archiving of intermediate states but maintained by an instance of PE12 Data Curating Service. The identity of a volatile dataset is enabled by the unity given to it by curation programme that operates on it, that bequeaths the volatile dataset common information goal and subject coverage. In order for an instance of PE24 Volatile Dataset to be referenceable it is necessary for the official curator to take snapshots, creating instances of PE22 Persistent Data Set which can be assigned and official identifier and referenced. Volatile datasets are typically whole databases or mash-ups with active data feeds.
Examples	Ancient World Online Blogspot curated by Charles Jones

Properties

Label	Domain	Range	Scope Note
PP23 has dataset part	PE24	PE18	Indicates the datasets, volatile or persistent, that form part of the volatile dataset
PP24 has dataset snapshot	PE24	PE22	Indicates the representative snapshot of the volatile dataset created at some point to stand as an identifier for the whole volatile dataset

PE25 RI Consortium

Class Label	PE25 RI Consortium
Subclass of	PE34_Team
Superclass of	
Scope Note	This class comprises special groups of actors who come together for the purpose of supporting a research infrastructure project. An RI Consortium can be composed of all other types of actors including other RI Consortiums. An RI Consortium is identified by its commonality of purpose and not by its membership at any one time. The group comes into existence with the agreement to maintain some collective project. So long as the group continues to support the common RI project and is non-empty the consortium continues to exist.
Examples	PARTHENOS Consortium

	ARIADNE Consortium CLARIN Consortium

Properties

Label	Domain	Range	Scope Note
PP25 is maintained by	PE26	PE25	Indicates the RI project of which the consortium is the maintainer and chief supporter

PE26 RI Project

Class Label	PE26 RI Project
Subclass of	PE35_Project
Superclass of	
Scope Note	This class comprises instances of collaborative enterprise undertaken over a period of time by an instance of PE25 RI Consortium with the intention of supporting research activities by providing a number of services to instances of E39 Actor. The project's existence depends on the continued maintenance by some consortium. It ends when there is no consortium left to maintain it.
Examples	PARTHENOS Project ARIADNE Project CLARIN Project

PE28 Curation Plan

Class Label	PE28 Curation Plan
Subclass of	E29 Design or Procedure
Superclass of	
Scope Note	This class comprises instances of plans that guide curation projects and which provide the information necessary to understand the intention and overall aim of an actor in carrying out some instances of PE3 Curating Service.
Examples	

PE29 Access Point

Class Label	PE29 Access Point
Subclass of	E51 Contact Point
Superclass of	
Scope Note	This class comprises instances of web addresses and network addresses by which e-services can be accessed.

Examples

PE32 Curated Thing

Class Label	PE32 Curated Thing
Subclass of	E70 Thing
Superclass of	E78 Curated Holding PE20 Volatile Digital Object
Scope Note	This class comprises aggregations of instances of either E18 Physical Thing or of PE20 Volatile Digital Object that are assembled and maintained by one or more instances of E39 Actor over time for a specific purpose and audience, and according to a particular collection development plan. Items may be added or removed from an instance of P32 Curated Thing in pursuit of this plan. The instance of PE32 Curated Thing gets identity not through a physical togetherness of things, nor through a concatenation of information objects, but rather through the deliberate management of the curated thing according to a plan.
Examples	

PE33 E-Access Brokering Service

Class Label	PE33 E-Access Brokering Service
Subclass of	PE8 E-Service
Superclass of	-
Scope Note	This class comprises declared offers by some instance of E39 Actor of their willingness and ability to provide electronic access brokering services for another instance of E39 Actor. E-Access brokering services offer mediation between the user of this instance of PE33 and some instance of PE8 E-Service, providing the means for the user to access the specified service. The actual E-Access brokering service function as an automatic process, and is indicated by the link to an instance of PE8 E-Service which is the service to which it gives access. An instance of a PE33 Access Brokering Service begins to exist with the declaration of the ability and willingness of an instance of E39 actor to broker access to some instance of PE8 E-Service. The instance of PE3 Access Brokering Service ends when either the declared willingness or ability to effectuate the mediation between the user and the target service permanently ends.
Examples	

New Direct Properties

Label	Domain	Range	Scope Note
PP46 brokers access to	PE33	PE8	Documents the relation of the access brokering service to the e-service for which it is a mediator

PE34 Team

Class Label	PE34 Team
Subclass of	E74 Group
Superclass of	PE25 RI Consortium
Scope Note	This class comprises groups of actors who come together for some defined project. The identity of the team is given by the collective will to achieve and support some project/aim. Membership in the group is determined by official association to the team for the purpose of contributing to the achievement of its aim. Membership need not be mediated by institutional association. An instance of PE34 Team is identified by its commonality of purpose and not by its membership at any one time. A PE34 Team instance comes into existence with the agreement to maintain its collective project. So long as the will to maintain the project is upheld by a minimal membership of the team (1), the team can be said to exist, although any or all of its members may change over time.
Examples	

PE35 Project

Class Label	PE35 Project
Subclass of	E7 Activity
Superclass of	PE26 RI Project
Scope Note	This class comprises instances of collaborative enterprise undertaken over a period of time by an instance of PE35 Team with the intention of effectuating some defined programme entailing the support of a number of instances of E7 Activity. An instance of PE35 Project comes into being with the formation of an instance of PE34 Team whose aim it is to carry out and maintain the project. The project continues to exist so long as the team both exists and continues to exercise its agency towards the maintenance of this project. A project ends either when it has reached its declared end point, attained its goal or the team constituted to support it is dissolved with no successor specified.
Examples	

New Direct Properties

Label	Domain	Range	Scope Note
PP43 supports project activities	PE35	E7	Documents the relation of a project to the activities that it supports and enables through its existence.

PE36 Competency Type

Class Label	PE36 Competency Type
Subclass of	E55 Type
Superclass of	
Scope Note	This class comprises concepts that are used to classify the processes or actions that a service is supposed to be capable of carrying out.
Examples	

PE37 Protocol Type

Class Label	PE37 Protocol Type
Subclass of	E55 Type
Superclass of	
Scope Note	This class comprises concepts that are used to classify the protocols that are used to access an instance of PE8 E-Service.
Examples	

PE38 Schema

Class Label	PE38 Schema
Subclass of	D14 Software
Superclass of	
Scope Note	This class is used to document instances of data structures, including formal ontologies, that are used to configure the behavior of software.
Examples	• The Pico XML schema • The SQL Schema for a relational database

Relations

PP1 currently offers (currently offered by)

Relation Label	PP1 currently offers (is currently offered by)
Subrelation of	P9 consists of (forms part of)
Superrelation of	-
Domain	PE26 RI Project
Range	PE1 Service
Scope	Allows research infrastructure project to be linked to the services it presently offers
Examples	

PP2 provided by (provides)

Relation Label	PP2 provided by (provides)
Subrelation of	P14 carried out by (performed)
Superrelation of	-
Domain	PE1 Service
Range	E39 Actor
Scope	Indicates the intention and willingness of an actor to carry out some service
Examples	

object (is object hosted by)

Relation Label	PP4 hosts object (is object hosted by)
Subrelation of	P16 used specific object (was used for)
Superrelation of	PP6 hosts digital object (is digital object hosted by)
Domain	PE2 Hosting Service
Range	E70 Thing
Scope	Indicates the generic relation of provision of some hosting service of an object of any kind.
Examples	

PP6 hosts digital object (is digital object hosted by)

Relation Label	PP6 hosts digital object (is digital object hosted by)
Subrelation of	PP4 hosts object (is object hosted by)
Superrelation of	PP7 hosts software object (is software object hosted by) PP8 hosts dataset (is dataset hosted by)
Domain	PE5 Digital Hosting Service
Range	D1 Digital Object
Scope	Indicates the relation of provision of a hosting service of a digital object of any kind.
Examples	

PP7 hosts software object (is software object hosted by)

Relation Label	PP7 hosts software object (is software object hosted by)
Subrelation of	PP6 hosts digital object (is digital object hosted by)
Superrelation of	-
Domain	PE6 Software Hosting Service
Range	D14 Software
Scope	Indicates the relation of provision of some hosting service of a software object.
Examples	

PP8 hosts dataset (is dataset hosted by)

Relation Label	PP8 hosts dataset (is dataset hosted by)
Subrelation of	PP6 hosts digital object (is digital object hosted by)
Superrelation of	-
Domain	PE7 Data Hosting Service
Range	PE18 Dataset
Scope	Indicates the relation of provision of some hosting service of a dataset object.
Examples	

PP11 curates volatile digital object (is volatile digital object curated by)

Relation Label	PP11 curates volatile digital object (is volatile digital object curated by)
Subrelation of	PP32 curates (was curated by)
Superrelation of	PP12 curates volatile software (is volatile software curated by) PP13 curates volatile dataset (is volatile dataset curated by)
Domain	PE10 Digital Curating Service
Range	PE20 Volatile Digital Object
Scope	This property associates an instance of digital curating service with the digital object of which it is the curation activity.
Examples	

PP12 curates volatile software (is volatile software curated by)

Relation Label	PP12 curates volatile software (is volatile software curated by)
Subrelation of	PP11 curates volatile digital object (is volatile digital object curated by)
Superrelation of	-
Domain	PE11 Software Curating Service
Range	PE23 Volatile Software
Scope	This property associates an instance of software curating service with the software of which it is the curation activity.
Examples	

PP13 curates volatile dataset (is volatile dataset curated by)

Relation Label	PP13 curates volatile dataset (is volatile dataset curated by)
Subrelation of	PP11 curates volatile digital object (is volatile digital object curated by)
Superrelation of	-
Domain	PE12 Data Curating Service
Range	PE24 Volatile Dataset
Scope	This property associates an instance of data curating service with the volatile dataset of which it is the curation activity.
Examples	

PP14 runs on request (is run by)

Relation Label	PP14 runs on request (is run by)
Subrelation of	P16 used specific object (was used for)
Superrelation of	-
Domain	PE13 Software Computing E-Service
Range	D14 Software
Scope	This property associates an instance of software computing e-service with the software that it runs when requested.
Examples	

PP15 delivers on request (is delivered by)

Relation Label	PP15 delivers on request (is delivered by)
Subrelation of	P16 used specific object (was used for)
Superrelation of	-
Domain	PE14 Software Delivery E-Service
Range	D14 Software
Scope	This property associates an instance of software delivery e-service with the software that it delivers when requested.
Examples	

PP16 has persistent digital object part (is persistent digital object part of)

Relation Label	PP16 has persistent digital object part (is persistent digital object part of)
Subrelation of	P106 is composed of (forms part of)
Superrelation of	PP19 has persistent software part (is persistent software part of) PP20 has persistent dataset part (is persistent dataset part of)
Domain	PE19 Persistent Digital Object
Range	PE19 Persistent Digital Object
Scope	This property associates an instance of PE19 Persistent Digital Object with a structural part of that instance which is, in turn, also an instance of PE19 Persistent Object. An instance of PE19 Persistent Digital Object can only have parts which are themselves also instances of PE19. This is in juxtaposition to PE20 Volatile Digital Object which may have parts which are themselves either instances of P20 Volatile Digital Object or P19 Persistent Digital Object.

Examples

PP17 has snapshot (is snapshot of)

Relation Label	PP17 has snapshot (is snapshot of)
Subrelation of	P130 shows features of (features are also found on)
Superrelation of	PP22 has release (is release of) PP24 has dataset snapshot (is dataset snapshot of)
Domain	PE20 Volatile Digital Object
Range	P19 Persistent Digital Object
Scope	This property associates an instance of PE20 Volatile Digital Object with an instance of PE19 Persistent Object which at any one point stands as an official version of the overall data stream.
Examples	

PP18 has digital object part (is digital object part of)

Relation Label	PP18 has digital object part (is digital object part of)
Subrelation of	P106 is composed of (forms part of)
Superrelation of	PP21 has software part (is software part of) PP23 has dataset part (is dataset part of)
Domain	PE20 Volatile Digital Object
Range	D1 Digital Object
Scope	This property associates an instance of PE20 Volatile Digital Object with a structural part of that instance. This structural part may be another instance of D1 Digital object, be it also a PE20 Volatile Digital Object or in fact be an instance of PE19 Persistent Object.
Examples	

PP19 has persistent software part (is persistent software part of)

Relation Label	PP19 has persistent software part (is persistent software part of)
Subrelation of	PP16 has persistent digital object part (is persistent digital object part of)
Superrelation of	-
Domain	PE21 Persistent Software
Range	PE21 Persistent Software
Scope	This property associates an instance of PE21 Persistent Software with a structural part of that instance which is, in turn, also an instance of PE21 Persistent Software.
Examples	

PP20 has persistent dataset part (is persistent dataset part of)

Relation Label	PP20 has persistent dataset part (is persistent dataset part of)
Subrelation of	PP16 has persistent digital object part (is persistent digital object part of)
Superrelation of	-
Domain	PE22 Persistent Dataset

Range	PE22 Persistent Dataset
Scope	This property associates an instance of PE22 Persistent Dataset with a structural part of that instance which is, in turn, also an instance of PE22 Persistent Dataset.
Examples	

PP21 has software part (is software part of)

Relation Label	**PP21 has software part (is software part of)**
Subrelation of	PP18 has digital object part (is digital object part of)
Superrelation of	-
Domain	PE23 Volatile Software
Range	D14 Software
Scope	This property associates an instance of PE23 Volatile Software with a structural part of that instance. This structural part will be an instance of D14 Software and can be either of its subclasses, PE21 Persistent Software of PE23 Volatile Software.
Examples	

PP22 has release (is release of)

Relation Label	**PP22 has release (is release of)**
Subrelation of	PP17 has snapshot (is snapshot of)
Superrelation of	-
Domain	PE23 Volatile Software
Range	PE21 Persistent Software
Scope	This property associates an instance of PE23 Volatile Software with an instance of PE21 Persistent Software which at any one point stands as an official version of that software development stream.
Examples	

PP23 has dataset part (is dataset part of)

Relation Label	**PP23 has dataset part (is dataset part of)**
Subrelation of	PP18 has digital object part (is digital object part of)
Superrelation of	-
Domain	PE24 Volatile Dataset
Range	PE18 Dataset
Scope	This property associates an instance of PE24 Volatile Dataset with a structural part of that instance. This structural part will be an instance of PE18 Dataset and can be either of its subclasses, PE22 Persistent Dataset of PE24 Persistent Dataset.
Examples	

PP24 has dataset snapshot (is dataset snapshot of)

Relation Label	**PP24 has dataset snapshot (is dataset snapshot of)**

Subrelation of	PP17 has snapshot (is snapshot of)
Superrelation of	-
Domain	PE24 Volatile Dataset
Range	PE22 Persistent Dataset
Scope	This property associates an instance of PE24Volatile Dataset with an instance of PE22 Persistent Dataset which at any one point stands as an official version of that dataset.
Examples	

PP25 has maintaining RI (is maintaining RI of)

Relation Label	**PP25 has maintaining RI (is maintaining RI of)**
Subrelation of	PP44 has maintaining team (is maintaining team of)
Superrelation of	-
Domain	PE26 RI Project
Range	PE25 RI Consortium
Scope	This property indicates the relation that exists between an instance of PE25 RI Consortium and some instance of PE26 RI Project, where the instance of PE25 is the responsible group of actors who maintain and support the instance of PE26.
Examples	

PP28 has designated access point (is designated access point of)

Relation Label	**PP28 has designated access point (is designated access point of)**
Subrelation of	P1 is identified by (identifies)
Superrelation of	-
Domain	PE8 E-Service
Range	PE29 Access Point
Scope	Links an instance of a PE8 E-Service to the web address at which the e-service can be accessed.
Examples	

PP29 uses access protocol (is access protocol used by)

Relation Label	**PP29 uses access protocol (is access protocol used by)**
Subrelation of	P16 used specific object (was used for)
Superrelation of	-
Domain	PE8 E-Service
Range	D14 Software
Scope	Links an instance of PE8 E-Service with the instance of D14 software which encodes the access protocol by which the e-service is to be accessed.
Examples	

PP31 uses curation plan (is curation plan used by)

Relation Label	PP31 uses curation plan (is curation plan used by)
Subrelation of	P33 used specific technique (was used by)
Superrelation of	-
Domain	PE3 Curating Service
Range	PE28 Curation Plan
Scope	Links an instance of PE3 Curation Service with the plan that organizes this activity
Examples	

PP32 curates (is curated by)

Relation Label	PP32 curates (is curated by)
Subrelation of	
Superrelation of	P147 curated (was curated by) PP11 curates volatile digital object (was volatile D/O curated by)
Domain	PE3 Curating Service
Range	PE32 Curated Thing
Scope	Links an instance of PE3 Curation Service with the object or objects for which it provides curation services.
Examples	

PP39 is metadata for (has metadata)

Relation Label	PP39 is metadata for (has metadata)
Subrelation of	P129 is about (is subject of)
Superrelation of	-
Domain	PE22 Persistent Dataset
Range	D1 Digital Object
Scope	Relates an instance of PE22 Persistent Dataset to some other instance of D1 Digital Object for which it plays the role of metadata. This relation establishes that the function of the information contained in the domain instance of PE22 is to describe the information contained in the range instance of D1.
Examples	

PP40 created successor of (is deprecated by)

Relation Label	PP40 created successor of (is deprecated by)
Subrelation of	P16 used specific object (was used for)
Superrelation of	-
Domain	E65 Creation
Range	PE22 Persistent Dataset
Scope	Relates an instance of E65 Creation to an instance of E22 Persistent Dataset that is acting as a metadata set. The latter E22 Persistent Dataset is referred to in the act of creation, specifically as the object of some correction. It is thus deprecated in the act of creation of some new instance of E22 Persistent Dataset. The new instance can be considered the successor of this deprecated dataset. The most recent successor, all things being equal, represents the present state of knowledge.
Examples	

PP41 is index of (is indexed by)

Relation Label	PP41 is index of (is indexed by)
Subrelation of	
Superrelation of	-
Domain	PE24 Volatile Dataset
Range	D1 Digital Object
Scope	Relates an instance of PE24 to an instance of D1 Digital object in the capacity of being an index for the latter.
Examples	

PP42 has declarative time

Relation Label	PP42 has declarative time
Subrelation of	
Superrelation of	-
Domain	PE1 Service
Range	E61 Time Primitive (XSD:Date)
Scope	Relates an instance of PE1 Service to a time span during which the service provider declares the service is, will be, has been in effect.
Examples	

PP43 supports project activity (is project activity supported by)

Relation Label	PP43 supported project activity (is project activity supported by)
Subrelation of	P9 consists of (forms part of)
Superrelation of	-
Domain	PE35 Project
Range	E7 Activity
Scope	Relates an instance of PE35 Project to an instance of E7 Activity which it supports as part of its overall program.
Examples	

PP44 has maintaining team (is maintaining team of)

Relation Label	PP44 has maintaining team (is maintaining team of)
Subrelation of	P17 was motivated by (motivated)
Superrelation of	PP26 has maintaining RI (is maintaining RI of)
Domain	PE35 Project
Range	PE34 Team
Scope	Relates an instance of PE35 Project to an instance of E34 Team which is the supporting agency that facilitates it.
Examples	

PP45 has competency (is competency of)

Relation Label	PP45 has competency (is competency of)
Subrelation of	P21 had general purpose (was purpose of)
Superrelation of	-
Domain	PE1 Service
Range	PE36 Competency Type
Scope	Relates an instance of PE1 Service to an instance of E36 Competency Type which it is competent to perform.

Examples

PP46 brokers access to (access brokered by)

Relation Label	PP46 brokers access to (has access brokered by)
Subrelation of	
Superrelation of	-
Domain	PE33 E-Access Brokering Service
Range	PE8 E-Service
Scope	Relates an instance of PE33 E-Access Brokering Service to instances an instance of PE8 E-Service which is a service to which it brokers access.
Examples	

PP47 has protocol type (is protocol type of)

Relation Label	PP47 has protocol type (is protocol type of)
Subrelation of	P125 used object of type (was type of object used in)
Superrelation of	-
Domain	PE8 E-Service
Range	PE37 Protocol Type
Scope	Relates an instance of PE8 E-Service to instances of PE34 Protocol that classify the protocols used to access the service.
Examples	

PP48 uses protocol parameter (is protocol parameter of)

Relation Label	PP48 uses protocol parameter (is protocol parameter of)
Subrelation of	P16 used specific object (was used for)
Superrelation of	-
Domain	PE8 E-Service
Range	PE38 Schema
Scope	Relates an instance of PE8 E-Service to instances of PE35 Schema that this service requires in order to run.
Examples	

PP49 provides access point (is access point provided by)

Relation Label	PP49 provides access point (is access point provided by)
Subrelation of	
Superrelation of	-
Domain	PE8 E-Service

Range	PE29 Access Point
Scope	Relates an instance of PE8 E-Service to an instance of PE29 Access Point which the service provides for an instance of D1 Digital Object.
Examples	

PP50 accessible at (provides access to)

Relation Label	PP50 accessible at (provides access to)
Subrelation of	
Superrelation of	-
Domain	D1 Digital Object
Range	PE29 Access Point
Scope	Relates an instance of D1 Digital Object to an instance of PE29 Access Point which has been provided to it by some PE8 E-Service.
Examples	

Referred Classes

D1 Digital Object

Class Label	D1 Digital Object
Subclass of	E73 Information Object
Superclass of	PE19 Persistent Digital Object PE20 Volatile Digital Object D14 Software PE18 Dataset
Scope Note	This class comprises identifiable immaterial items that can be represented as sets of bit sequences, such as data sets, e-texts, images, audio or video items, software, etc., and are documented as single units. Any aggregation of instances of D1 Digital Object into a whole treated as single unit is also regarded as an instance of D1 Digital Object. This means that for instance, the content of a DVD, an XML file on it, and an element of this file, are regarded as distinct instances of D1 Digital Object, mutually related by the P106 is composed of (forms part of) property. A D1 Digital Object does not depend on a specific physical carrier, and it can exist on one or more carriers simultaneously.
Examples	
External Ontology Origin	CRMdig 3.2.1

Properties

Label	Domain	Range	Scope Note
PP50 accessible at	D1	PE29	Relates an instance of D1 Digital Object to an instance of PE29 Access Point which has been provided to it by some PE8 E-Service.

D14 Software

Class Label	D14 Software
Subclass of	D1 Digital Object
Superclass of	PE21 Persistent Software PE23 Volatile Software
Scope Note	This class comprises software codes, computer programs, procedures and functions that are used to operate a system of digital objects.
Examples	
External	CRMdig 3.2.1

Ontology Origin	

E7 Activity

Class Label	E7 Activity
Subclass of	E5 Event
Superclass of	PE1 Service PE26 RI Project PE27 Service Action
Scope Note	This class comprises actions intentionally carried out by instances of E39 Actor that result in changes of state in the cultural, social, or physical systems documented. This notion includes complex, composite and long-lasting actions such as the building of a settlement or a war, as well as simple, short-lived actions such as the opening of a door.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

E21 Person

Class Label	E21 Person
Subclass of	E39 Actor E20 Biological Object
Superclass of	
Scope Note	This class comprises real persons who live or are assumed to have lived. Legendary figures that may have existed, such as Ulysses and King Arthur, fall into this class if the documentation refers to them as historical figures. In cases where doubt exists as to whether several persons are in fact identical, multiple instances can be created and linked to indicate their relationship. The CRM does not propose a specific form to support reasoning about possible identity.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

E39Actor

Class Label	E39 Actor
Subclass of	E77 Persistent Item
Superclass of	E21 Person E74 Group
Scope Note	This class comprises people, either individually or in groups, who have the potential to perform intentional actions of kinds for which someone may be held responsible. The CRM does not attempt to model the inadvertent actions of such actors. Individual people should be documented as instances of E21 Person, whereas groups should be documented as instances of either E74 Group or its subclass E40 Legal Body.

Examples	
External Ontology Origin	CIDOC CRM 6.2.1

PP27 has designated contact	**PE39**	**E39**	**Link to the actor whom is designated as the contact point for this service (may or may not be the actor who offers the service, may or may not form a member of the group or institution that provides a service)**

E40 Legal Body

Class Label	**E40 Legal Body**
Subclass of	E74 Group
Superclass of	PE25 RI Consortium
Scope Note	This class comprises institutions or groups of people that have obtained a legal recognition as a group and can act collectively as agents. This means that they can perform actions, own property, create or destroy things and can be held collectively responsible for their actions like individual people. The term 'personne morale' is often used for this in French.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

E65 Creation

Class Label	**E65 Creation**
Subclass of	E7 Activity
Superclass of	
Scope Note	This class comprises events that result in the creation of conceptual items or immaterial products, such as legends, poems, texts, music, images, movies, laws, types etc.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

Properties

Label	**Domain**	**Range**	**Scope Note**
PP40 created successor of	E65	PE22	Indicates the relation between the act of creation and a metadata set that is corrected in the act of creation.

E70 Thing

Class Label	E70 Thing
Subclass of	E77 Persistent Item
Superclass of	PE32 Curated Thing
Scope Note	This general class comprises discrete, identifiable, instances of E77 Persistent Item that are documented as single units, that either consist of matter or depend on being carried by matter and are characterized by relative stability. They may be intellectual products or physical things. They may for instance have a solid physical form, an electronic encoding, or they may be a logical concept or structure.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

E71 Man Made Thing

Class Label	E71 Man Made Thing
Subclass of	E70 Thing
Superclass of	E24 Physical Man-Made Thing E28 Conceptual Object
Scope Note	This class comprises discrete, identifiable man-made items that are documented as single units. These items are either intellectual products or man-made physical things, and are characterized by relative stability. They may for instance have a solid physical form, an electronic encoding, or they may be logical concepts or structures
Examples	
External Ontology Origin	CIDOC CRM 6.2.2

E74 Group

Class Label	E74 Group
Subclass of	E39 Actor
Superclass of	E40 Legal Body PE34 Team
Scope Note	This class comprises any gatherings or organizations of E39 Actors that act collectively or in a similar way due to any form of unifying relationship. In the wider sense this class also comprises official positions which used to be regarded in certain contexts as one actor, independent of the current holder

	of the office, such as the president of a country. In such cases, it may happen that the Group never had more than one member. A joint pseudonym (i.e., a name that seems indicative of an individual but that is actually used as a persona by two or more people) is a particular case of E74 Group. A gathering of people becomes an E74 Group when it exhibits organizational characteristics usually typified by a set of ideas or beliefs held in common, or actions performed together. These might be communication, creating some common artifact, a common purpose such as study, worship, business, sports, etc. Nationality can be modelled as membership in an E74 Group (cf. HumanML markup). Married couples and other concepts of family are regarded as particular examples of E74 Group.
Examples	
External Ontology Origin	CIDOC CRM 6.2.2

E77 Persistent Item

Class Label	**E77 Persistent Item**
Subclass of	E1 CRM Entity
Superclass of	E39 Actor E70 Thing
Scope Note	This class comprises items that have a persistent identity, sometimes known as “endurants” in philosophy. They can be repeatedly recognized within the duration of their existence by identity criteria rather than by continuity or observation. Persistent Items can be either physical entities, such as people, animals or things, or conceptual entities such as ideas, concepts, products of the imagination or common names. The criteria that determine the identity of an item are often difficult to establish -; the decision depends largely on the judgement of the observer. For example, a building is regarded as no longer existing if it is dismantled and the materials reused in a different configuration. On the other hand, human beings go through radical and profound changes during their life-span, affecting both material composition and form, yet preserve their identity by other criteria. Similarly, inanimate objects may be subject to exchange of parts and matter. The class E77 Persistent Item does not take any position about the nature of the applicable identity criteria and if actual knowledge about identity of an instance of this class exists. There may be cases, where the identity of an E77 Persistent Item is not decidable by a certain state of knowledge. The main classes of objects that fall outside the scope the E77 Persistent Item class are temporal objects such as periods, events and acts, and

	descriptive properties.
Examples	
External Ontology Origin	CIDOC CRM 6.2.2

E78 Curated Holding

Class Label	**E78 Curated Holding**
Subclass of	E24 Physical Man Made Thing PE32 Curated Thing
Superclass of	
Scope Note	This class comprises aggregations of instances of E18 Physical Thing that are assembled and maintained ("curated" and "preserved," in museological terminology) by one or more instances of E39 Actor over time for a specific purpose and audience, and according to a particular collection development plan. Typical instances of curated holdings are museum collections, archives, library holdings and digital libraries. A digital library is regarded as an instance of E18 Physical Thing because it requires keeping physical carriers of the electronic content. Items may be added or removed from an E78 Curated Holding in pursuit of this plan. This class should not be confused with the E39 Actor maintaining the E78 Curated Holding often referred to with the name of the E78 Curated Holding (e.g. "The Wallace Collection decided…"). Collective objects in the general sense, like a tomb full of gifts, a folder with stamps or a set of chessmen, should be documented as instances of E19 Physical Object, and not as instances of E78 Curated Holding. This is because they form wholes either because they are physically bound together or because they are kept together for their functionality.
Examples	
External Ontology Origin	CIDOC CRM 6.2.2

Referred Relations

P1 is identified by (identifies)

Relation Label	P1 is identified by (identifies)
Subrelation of	-
Superrelation of	PP28 has designated access point (is designated access point of)
Domain	E1 CRM Entity
Range	E41 Appellation
Scope	This property describes the naming or identification of any real world item by a name or any other identifier. This property is intended for identifiers in general use, which form part of the world the model intends to describe, and not merely for internal database identifiers which are specific to a technical system, unless these latter also have a more general use outside the technical context. This property includes in particular identification by mathematical expressions such as coordinate systems used for the identification of instances of E53 Place. The property does not reveal anything about when, where and by whom this identifier was used. A more detailed representation can be made using the fully developed (i.e. indirect) path through E15 Identifier Assignment.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P9 consists of (forms part of)

Relation Label	P9 consists of (forms part of)
Subrelation of	-
Superrelation of	PP1 currently offers (currently offered by)
Domain	E4 Period
Range	E4 Period
Scope	This property associates an instance of E4 Period with another instance of E4 Period that is defined by a subset of the phenomena that define the former. Therefore the space time volume of the latter must fall within the space time volume of the former. This property is transitive.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P14 carried out by (performed)

Relation Label	P14 carried out by (performed)
Subrelation of	-

Superrelation of	PP2 provided by (provides) PP3 requested by (requests)
Domain	E7 Activity
Range	E39 Actor
Scope	This property describes the active participation of an E39 Actor in an E7 Activity. It implies causal or legal responsibility. The P14.1 in the role of property of the property allows the nature of an Actor's participation to be specified.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P15 was influenced by (influenced)

Relation Label	**P15 was influenced by (influenced)**
Subrelation of	-
Superrelation of	PP25 is maintained by (maintains)
Domain	E7 Activity
Range	E1 CRM Activity
Scope	This is a high level property, which captures the relationship between an E7 Activity and anything that may have had some bearing upon it. The property has more specific sub properties.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P16 used specific object (was used for)

Relation Label	**P16 used specific object (was used for)**
Subrelation of	-
Superrelation of	PP4 hosts object (is object hosted by) PP14 runs on request (is run by) PP15 delivers on request (is delivered by) PP29 uses access protocol (is access protocol used by)
Domain	E7 Activity
Range	E70 Thing
Scope	This property describes the use of material or immaterial things in a way essential to the performance or the outcome of an E7 Activity. This property typically applies to tools, instruments, moulds, raw materials and items embedded in a product. It implies that the presence of the object in question was a necessary condition for the action. For example, the activity of writing this text required the use of a

	computer. An immaterial thing can be used if at least one of its carriers is present. For example, the software tools on a computer. Another example is the use of a particular name by a particular group of people over some span to identify a thing, such as a settlement. In this case, the physical carriers of this name are at least the people understanding its use.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P33 used specific technique (was used by)

Relation Label	**P33 used specific technique (was used by)**
Subrelation of	-
Superrelation of	PP31 used curation plan (was curation plan used by)
Domain	E7 Activity
Range	E29 Design or Procedure
Scope	This property identifies a specific instance of E29 Design or Procedure in order to carry out an instance of E7 Activity or parts of it. The property differs from P32 used general technique (was technique of) in that P33 refers to an instance of E29 Design or Procedure, which is a concrete information object in its own right rather than simply being a term or a method known by tradition. Typical examples would include intervention plans for conservation or the construction plans of a building.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P106 is composed of (forms part of)

Relation Label	**P106 is composed of (forms part of)**
Subrelation of	-
Superrelation of	PP16 has persistent digital object part (is persistent digital object part of) PP18 has digital object part (is digital object part of)
Domain	E90 Symbolic Object
Range	E90 Symbolic Object
Scope	This property associates an instance of E90 Symbolic Object with a part of it that is by itself an instance of E90 Symbolic Object, such as fragments of texts or clippings from an image. This property is transitive

Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P129 is about (is subject of)

Relation Label	**P129 is about (is subject of)**
Subrelation of	-
Superrelation of	PP39 is metadata about (has metadata)
Domain	E89 Propositional Object
Range	E1 CRM Entity
Scope	This property documents that an E89 Propositional Object has as subject an instance of E1 CRM Entity. This differs from P67 refers to (is referred to by), which refers to an E1 CRM Entity, in that it describes the primary subject or subjects of an E89 Propositional Object.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P130 shows features of (features are also found on)

Relation Label	**P130 shows features of (features are also found on)**
Subrelation of	-
Superrelation of	PP17 has snapshot (is snapshot of)
Domain	E70 Thing
Range	E70 Thing
Scope	This property generalises the notions of "copy of" and "similar to" into a directed relationship, where the domain expresses the derivative, if such a direction can be established. Otherwise, the relationship is symmetric. If the reason for similarity is a sort of derivation process, i.e., that the creator has used or had in mind the form of a particular thing during the creation or production, this process should be explicitly modelled. Moreover it expresses similarity in cases that can be stated between two objects only, without historical knowledge about its reasons.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

P147curated (was curated by)

Relation Label	P147 curated (was curated by)
Subrelation of	PP32 curates (is curated by)
Superrelation of	-
Domain	E87 Curation Activity
Range	E78 Collection
Scope	This property associates an instance of E87 Curation Activity with the instance of E78 Collection or collections with that is subject of that curation activity following some implicit or explicit curation plan.
Examples	
External Ontology Origin	CIDOC CRM 6.2.1

PARTHENOS Entities Minimal ‘Metadata’ Document

This document offers a specification of the minimal ‘metadata’ that should be collected with regards to the entities referred to in the PARTHENOS Entities model for the provisioning of the PARTHENOS Registry. The function of the PARTHENOS registry is to gather the minimal ‘metadata’ information in order to allow the identification of distinct datasets, software and services and to connect these to the appropriate actors responsible for the generation of maintenance of these objects. The metadata set then is kept as light as possible with an emphasis that all objects represented in the register are traceable.

Precise provenance information would form part of the richer, actual metadata(s) about a data object. This would form part of the general content cloud supported by the PARTHENOS Architecture and would be accessed by indices generated for specific ends.

First Created: 25/5/2016
Last Updated: 12/4/2017

Document History

Version/date	Date	Changes/approval	Author/Approved by
V 1.0	25/5/2016	Initial version	George, Martin
V 1.1	2/6/2016	Identification of potential mandatory elements in certain services	George, Martin
V 1.2	3/6/2016	Continued discussion on mandatory elements, add of mandatory column	George, Martin
V 1.3	7/6/2016	Edits and considerations concerning curation	George, Martin
V 1.5	10/6/2016	Update to align with discussions of Joint WP5/6 meeting in Crete	George
V 1.6	N/A		
V 1.7	N/A		
V 1.8	30/8/2106	Aligned to V1.8 of Parthenos Entities Model, Basic editing of mistakes	George
V 1.11	12/4/2017	Aligned to V1.11 of Parthenos Entities Model	George
V 2.0	12/4/2017	Aligned to V2.0 of PE Model	George

Services

Label	Man.?	Field Type	Description	CRM Translation	Comment
ID	Yes	String	The identifier used to indicate the service	PE1->P1->E42	
Typology	Yes	Controlled Vocabulary (Determines Constraints)	• Hosting service - Digital Hosting Service Software Hosting Service Data Hosting Service Data E-Services S/W Computing E-Service S/W Delivery E-Service - Physical Hosting Service • E Service • Curating Service - Digital Curating Service Software Curating Service Curated Software E-Service Data Curating Service Curated Data E-Service - Physical Curating Service	Place in Service IsA hierarchy	N.B. in fact the class will vary depending on the type Controlled Vocabulary, but the attributes below hold for all instances of PE1 Service class except where if conditions have been introduced.
Title	Yes	String	The name by which the service is known or referred to	PE1->P1->E41	
Description	No	Long Text	A textual description of the service, brief history, and intended usage. This textual attribute could also describe the community/users of the service, and its popularity within a community.	PE1->P3->E62	

Competence	Yes	Controlled Vocabulary	The ability of a service to do something successfully: is a relation that connects a *service* with an *activity type*	PE1->pp45->PE36	
Provided by	Yes	Link (Actor)	The actor that provides the service, e.g., for a curating service we keep the curator	PE1->PP2->E39	N.B. the semantic path will differ based upon our level of knowledge
Declared Begin/End of operation	No	Date-Date	The date that the service providers indicates as the beginning and/or ending of the offer of the service	PE1->PP42->E61	
Last confirmation	Yes	Date	The date that it is confirmed that the service still exists	PE1->P4->E52->P81->E61	
Date of Registration (w/Parthenos)	Conditionally		The date when registered with Parthenos (acts as at least first confirmation of existence of service… must be running when added).		This would actually be meta meta data on the act of registering this very service. Could be implemented as named graph or by rules.
Availability	No	Controlled Vocabulary	E.g., 24/7, 24/5, on request, unknown, periodic (.e.g. business hours)	PE1->p2->E55	
Conditions of	Yes	Controlled	Indicate the type of conditions that the use	PE1->p33-	Double

use/rights Type		Vocabulary	of this service are subject to (Open Access, Open Access - required registration, license-based, on request, embargo)	>E29/E30->E55	Instantiate the middle node as both design and procedure and right
Conditions of Use / Rights Text	Yes (Conditionally)	Link (Document)	Link to the actual text outlining conditions of use	PE1->p33->E29/E30	If there are restriction conditions then how to manage must be documented.
Contact Person	No	Link (Actor)	E.g., the contact person of the actor that provides the service	Follow path of service ‘Provided by’ and switch E39 for E21: E21->p76->E51	
Communication address	Yes	String	E.g., the contact method for this particular service (regardless of providers address)	Follow path of service provider and then add from E39: E39->p76->E51	N.B. E39 is the service provider

If Hosting Service

(+Service)

Hosts Object	No	Reverse Link (Object - open)	Indicate the object hosted by the hosting service	PE2->PP4->E70	If hosting service has objects, display these under hosting service, hierarchically.

If Digital Hosting Service

(+Service + Hosting Service)

Hosts Digital Object	No	Reverse Link (Object – restrict – digital object)	Indicate the digital object hosted	PE5->PP6->D1	If hosting service has objects, display these under hosting service, hierarchically.
Preservation Activity Type	No	Controlled Vocabulary	Indicate the type of preservation activity undertaken on hosted digital object	PE5->P9->D12->P2->E55	Snapshot, Backup, Give Copy

If Software Hosting Service

(+Service + Hosting Service + Digital Hosting Service)

Hosts Software Object	No	Reverse Link (Object – restrict - Software)	Indicate the software object hosted	PE6->PP7->D14	If hosting service has objects, display these under hosting service, hierarchically.

If Data Hosting Service

(+Service + Hosting Service + Digital Hosting Service)

Hosts Dataset	No	Reverse Link (Object – restrict - Dataset)	Indicate the dataset hosted	PE7->PP8->PE18	If hosting service has objects, display these under hosting service, hierarchically.

If E-Service

(+Service)

Online Access Point	Yes	String	*URL where the service can be accessed by a client application*	PE8->PP28->PE29	
Authorization	Yes (Conditionally)	Controlled Vocabulary	Authentication and authorisation policies (*e.g., OAUTH, SAML)*	PE8->P70i->**E31**->P2->E55"Authorization Policies'	N.B. end point is E30 Document. But since these documents will all be of same type, this can be recorded this way
Protocol	Yes	Controlled Vocabulary	Links the service to the access protocol, considered as a form of software, which it invokes e.g. SOAP/REST	PE8->PP29->D14	
Protocol Type	No	Controlled Vocabulary	Links the service to the access protocol type is particular software instance is not referenced	PE8->pp47->PE37	
Protocol Parameters	Yes (Conditionally)	Link	Links to the documentation of parameters fo the protocol invoked	PE8->pp48->PE38	

If Curation Service

(+Service)

Curates	No	Reverse Link (Object)	Link the curation service to the general object it curates	PE3->PP32->PE32	If curation service is service for some curated holding, display it.

If Digital Curation Service

(+Service + Curation Service)

Curates Digital Holding	No	Reverse Link (Object – Physical Holding)	Link the curation service to the volatile digital object that it manages	PE10->PP11->PE20	If curation service is service for some curated holding, display it.

If Software Curation Service

(+Service + Curation Service + Digital Curation Service)

Curates Software	No	Reverse Link (Object – Volatile Software)	Link the curation service to the volatile software that it manages	PE11->pp12->D14	If curation service is service for some curated holding, display it.

If Data Curation Service

(+Service + Curation Service + Digital Curation Service)

Curates Volatile Dataset	No	Reverse Link (Object – Volatile Dataset)	Link the curation service to the volatile dataset that it manages	PE12->pp13->PE24	If curation service is service for some curated holding, display it.

If Software Computing E-Service

(+ E-Services + Software Hosting Services)

Runs on Request	Yes	Reverse Link (Object – restrict - Software)	Indicate the software object the service runs on request	PE13->PP14->D14	

If Software Delivery E-Service

(+E-Services + Software Hosting Services)

Delivers on Request	Yes	Reverse Link (Object – restrict - Software)	Indicate the software object the service delivers on request	PE14->PP15->D14	

If Data E-Services

(+ E-Services)

If Curated Software E-Service

(+ Data E-Services + Software curation Services)

If Curated Data E-Service

(+ Data E-services + Data curation Services)

Object

Label	Mandatory	Field Type	Description	CRM Translation	Comment
ID	Yes	String	The identifier used to indicate the object	E70->P1->E42	N.B. in fact the class will vary depending on the type selected, but the attributes below hold for all instances of PE1 Service class except where if conditions have been introduced.
Other IDs & Type & Attributor	No	String	Additional identifiers given to the object. Type can be recorded in order to indicate kind of ID. The actor who bestowed the ID can also be recorded.	E70->p140i->E13->p141->E42 + E70->p140i->E13->p141->E42->p2-	

				>E55 +E70-p140i->E13->p14->E39	
Typology	Yes	Controlled Vocabulary (Determines Constraints)	- Physical Object - Digital Object Dataset Volatile Dataset Persistent Dataset Software Volatile Software Persistent Software	Place in IsA starting from E70 Thing	
Title	Yes	String	The name by which the object is known or referred to	E70->P1->E41	
Description	No	Long Text	A textual description of the object	E70->P3->E62	

If Physical Object

(+ Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Is or was part of	No	Reverse Link (Object – Restrict Curated Holding)	Here we indicate the physical collection of which this physical object had been a part	E70->p46i->E70	

If Curated Holding

(+Object + Physical Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Curated by	Yes	Link (Service – Restrict Physical Curation Service)	Here we indicate the curation service that is responsible for the maintenance of this physical collection	E78->p147i->E87	
Had Curation Plan	Yes	Link (Curation Plan)	Here we indicate the curation plan associated to this curated holding.	E78->P147i->E87->PP31->PE28	

If Digital Object

(+Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Hosted by	Yes	Link (Service – Restrict to Digital Hosting Service)	Here we indicate the digital hosting service responsible for the hosting of this digital object.	D1->PP6i->PE5	
Is / Was Part of	Yes	Link (Object – Restrict Digital Object	Here we indicate digital objects of which this digital object has formed part.	D1->P106i->D1	

If Dataset

(+ Object + Digital Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Hosted by	Yes	Link (Service – Restrict Data Hosting Service)	Here we indicate the data hosting service responsible for the hosting of dataset	PE18->PP8i->PE7	
Encoding Type	Yes	Controlled Vocabulary	Here we indicate the encoding(s) of the dataset in question	PE18->L11i->D7->P33->E29->P2->E55	
Schema/Format	No	Link (Object – Restrict Persistent Software)	Here indicate the schema used to structure the dataset.	PE18->l11i->D7->L23->PE38	
Subject	No	Controlled Vocabulary	Here we indicate the role that the dataset can play in research	PE18->P129->E55	
Spatiotemporal Coverage	No	Controlled Vocabulary	Here we indicate the geographic scope for which the dataset has relevance.	PE18-> P129->E2	
Created by	Yes	Link (Actor)	Here we link the dataset to its creator	PE18->L11i->D7->P14->E39	

If Software

(+ Object + Digital Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Hosted by	No	Link (Service – Restrict Software Hosting Service)	Here we indicate the software hosting service responsible for the hosting of the software object.	D14->PP7i->PE6	
Delivered on request by	No	Reverse Link (Service – Restrict S/W Delivery E-Service)	Here we indicate the software delivery e-service capable of delivering the software to a client.	D14->PP15i->PE14	
Run on Request by	No	Reverse Link (Service – Restrict S/W Computing E-Service)	Here we indicate the software computing e-service capable of delivering the software to a client.	D14->PP14i->PE13	
Programming language	No	Controlled Vocabulary	Here we indicate the programming language used in creating the software	D14->L11i->D7->P33->E29	
Executes processes of type	Yes	Controlled Vocabulary	Here we indicate the kind of process types that the software (typically an algorithm) can execute	D14->P103-> E55	

If Volatile Digital Object

(+ Object + Digital Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Curated by	Yes	Link (Service – Restrict to Digital Curating Service)	Here we indicate the digital curating service responsible for the curation of this object.	PE20->PP11i->PE10	
Has Snapshot	No	Reverse Link (Object – Restrict Persistent Digital Object)	Here we indicate the snapshot that gives the identity to a volatile data object. In order for a volatile data object to have proper provenance it must at any time have one official snapshot that is known to the curator of the object.	PE20->PP17->PE19	
Is Part Of	No	Link (Object – Restrict Digital Object)	Here we can indicate the parts of a volatile data object. A volatile data object can be made up of volatile as much as persistent data objects. If it has as component as volatile data object, this object in turn, in order to have proper provenance must have its own snapshot.	PE20->PP18->D1	
Has Curation Plan	Yes	Link	Link the curation service to the curation plan which it implements	PE3-> PP11i->PE10->PP31->PE28	

If Persistent Digital Object

(+ Object + Digital Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Is Part Of	No	Link (Object – Restrict Persistent Data Object)	Here we indicate the persistent data object that forms a distinct part of the overall persistent data object in question. N.B. a persistent data object can have as part any other type of persistent digital object. It cannot have a volatile data object as part.	PE19->PP16->PE19	
Is Snapshot of	No	Link (Object – Restrict Volatile Data Object)	If the persistent data object stands as the identifying snapshot for some volatile data object, this can be indicated here.	PE19->PP17i->PE20	
Same as	No	Link (Object – Restrict Persistent Data Object)			
Compilation Date	Yes	Date	Here we indicate the date when the current encoding was fixed.	PE19->L11i-D7->P4->E52->P81->E61	
File Size	Yes	Integer	Here we indicate file size in bytes	PE19->p43->**E54** + PE19->p43->**E54**-	Where dimension type is

				>2->E55	constant as ‘byte’
Checksum	Yes	Integer	Here we indicate the checksum of the persistent dataset.	PE22->p39i->E16->p40->E54 + PE19->p39i->E16->p40->E54 ->p2->E55	Where dimension type is constant as ‘checksum’

If Volatile Dataset

(+Object + Digital Object + Dataset+ Volatile Digital Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Curated by	Yes	Link (Service – Restrict Data Curating Service)	A link between the volatile dataset object and the data curation service that is responsible for its curation.	PE24->PP13i->PE12	
Has Snapshot	No	Link (Object – Restrict Persistent Dataset)	Here we link to the dataset which is the snapshot of this volatile dataset.	PE24->PP24->PE22	
Is Part Of	No	Link (Object – Restrict Digital Object)	Here we link to the parts of this volatile dataset. These parts can be persistent or volatile, dataset or software.	PE24->PP23->PE18	

If Persistent Dataset

(+Object + Digital Object + Dataset+ Persistent Digital Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Is Part Of	No	Link (Object – Restrict Persistent Dataset)	Here we indicate all distinct persistent datasets that form part of this dataset (all of which in turn can be documented in their own right).	PE22->PP20->PE22	
Is Snapshot of	No	Link (Object – Restrict Volatile Dataset)	Here we indicate the volatile dataset of which this persistent dataset was or is a snapshot.	PE22->PP24i->PE24	

If Volatile Software

(+Object + Digital Object + Software + Volatile Digital Object)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Curated by	Yes	Link (Service – Restrict to Software Curator	A link between the volatile software object and the software curation service that is responsible for its curation.	PE23->PP12i->PE11	
Is Part Of	No	Link (Object – Restrict	Here we link to the distinct parts of the software that can be identified whether	PE23->PP21->D14	

		Software)	also volatile or persistent.		
Has Release	No	Link (Object – Restrict Persistent Software)	Here we link to the official release of the volatile software.	PE23->PP22->PE21	

If Persistent Software

(+Object + Digital Object + Software + Persistent Digital Object)

Label		Field Type	Description	CRM Translation	Comment
Is Part Of	No	Link (Object – Restrict Persistent Software)	Here we link the persistent software to its component parts.	PE21->PP19->PE21	
Is Release of	No	Link (Restrict – Volatile Software)	Here we link to the volatile software of which this persistent software is a release.	PE21->PP22i->PE23	

Actor

Label	Mandatory	Field Type	Description	CRM Translation	Comment
ID	Yes	String	The identifier used to indicate the actor	E39->P1->E42	
Typology	Yes	Controlled Vocabulary (Determines Constraints)	Person Institution Team Research	Place in IsA hierarchy from E39	
Appellation	Yes	String	The name by which the actor is known or referred to	E39>P1->E41	
Description	No	Long Text	A textual description of the actor	E39->P3->E62	
Legal Address	No	String	Here we give the legal address for the actor	E39->p76->E45 + E39->p76->E45->p2->E55	Where type is a constant "Legal Address"
Mailing Address	No	String	Here we give the mailing address for the actor	E39->p76->E45 + E39->p76->E45->p2->E55	Where type is a constant "Legal Address"
Contact Person	No	Link (Actor – Restrict Person)	Here we link to the designated contact person for this actor.	E39->PP27->E39	
Phone	No	String		E39->p76->E51 +	Where type is constant

				E39->p76->E51-p2->E55	'Phone'
Email	No	String		E39->P76->E51 + E39->P76->E51-P2->E55	Where type is constant 'email'
Provides Service	No	Reverse Link (Service)	Here we indicate the services the actor provides	E39->PP2i->PE1	
Requests Service	No	Reverse Link (Service)	Here we indicate the services the actor requests.	E39->PP3i->PE1	

If Team

(+Actor)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Legal Statuses	Yes	Controlled Vocabulary	Team or Department	PE34->p2->E55	
Beginning of Existence	No	Date	Here we indicate when the team/department /institution came into existence	PE34->P95i->E66->P4->E52	
End of Existence	No	Date	Here we indicate when the team/department /institution ceased existing	PE34->P99i->E68->P4->E52	
Merged with	No	Link (Actor – Restrict	Here we indicate a team/department /institution with which an team/department /institution		

		Institution)	merged.		
Merged Date	No	Date	Here we indicate the date of the merge event		
Is Member of	No	Link (Actor – Restrict Team)	Here we indicate any membership which the team/department /institution might have with a team	PE34->P143->E85->P144->E74	
Date of Joining Team	No	Date	Here we indicate when a team/department /institution joined a team/department /institution.	PE34->P143->E85->p4->E52	
Is no longer member of	No	Link (Actor – Restrict Team)	Here we indicate what team/department /institution was left.	PE34->p145i->E86->P146->E74	
Date of Leaving	No	Date	Here we indicate when an team/department /institution left a team/department /institution.	PE34->p145i->E86->PP->E52	
Has Individual Member	No	Link (Actor – Restrict Person)	Here we indicate individual members of a team/department/institution.	PE34->p107->E39	

If Institution

(+actor+team)

Label	**Mandatory**	**Field Type**	**Description**	**CRM Translation**	**Comment**
Legal Statuses	Yes	Controlled Vocabulary	Choose from Public Body Non-Profit International Organization Research Organization Legal Person	E40->p2->E55	

If Research Infrastructure

(+actor+team)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Maintains	Yes	Link (Services – Restrict Project)	Here we indicate the project that the RI is responsible for maintaining.	PE25->PP25->PE26	

If Person

(+Actor)

Label	Mandatory	Field Type	Description	CRM Translation	Comment
Is Member of team	Yes	Link (Actor – Restrict Team)	Here we indicate the team of which an individual person is a member	E21->P143->E85->P144->E74	
Member Since	No	Date	Here we indicate when the individual joined the team as a date.	E21->P143->E85->p4->E52	
Left Team	No	Link (Actor – Restrict Team)	Here we indicate the team from which an individual left.	E21->p145i->E86->P146->E74	
Left Team on Date	No		Here we indicate the date of the departure from the team.	E21->P145i->E86->PP->E52	

Is Member of institution	No	Link (Actor – Restrict Institution)	Here we indicate the institution of which an individual person is an employee.	E21->P143->E85->P144->E40	
Member Since	No	Date	Here we indicate the date from which the individual is a member of the institution.	E21->P143->E85->P4->E52	
Left Institution	No	Link (Actor – Restrict Institution)	Here we indicate the institution from which an individual ceased to be a member	E21->P145i->E86->P146->E40	
Left Institution on Date	No	Date	Here we indicate the date upon which the individual ceased to be a member of the institution.	E21->p145i->E86->P4->E52	

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!--
PARTHENOS Entities CRMpe 2.0 Encoded in RDFS

RDFS updated by FORTH April 13, 2017
corrected typos that by Luca Frosini pointed out

RDFS updated by FORTH March 9, 2017

RDFS updated by CNR February 2, 2017
added PE20_Volatile_Digital_Object as subclass of PE32_Curated_Thing

RDFS updated by CNR January 30, 2017
corrected PE20, PE21 and PP28i (they contained blanks)

RDFS updated by FORTH-ICS September 19, 2016
corrected some typos, added PE26, removed PE27, added PP41 and PP42

RDFS created by FORTH-ICS August 19, 2016

Encoding Rules:
1. The RDF spelling rules do not allow blanks. Hence we have replaced them by underscores.
  The blank between the concept identifier and concept name is replaced by underscore too.
  For instance "PE1_Service" or "PP1_currently_offers".

2. RDF does not allow to instantiate properties beginning from a range value.
  Therefore, each CRMpe property is represented as two RDFS properties.
  For instance "PP1 currently offers(is currently offered by)" is represented as:
  "PP1_currently offers" for the domain to range direction and "PP1i_is_currently_offered_by" for the range to domain direction.

-->
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.ics.forth.gr/isl/CRMpe/" xml:lang="en">

<rdfs:Class rdf:about="PE1_Service">
	<rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E7_Activity"/>
</rdfs:Class>

<rdfs:Class rdf:about="PE2_Hosting_Service">
	<rdfs:subClassOf rdf:resource="PE1_Service"/>
</rdfs:Class>

<rdfs:Class rdf:about="PE3_Curating_Service">
	<rdfs:subClassOf rdf:resource="PE1_Service"/>
</rdfs:Class>

<rdfs:Class rdf:about="PE5_Digital_Hosting_Service">
	<rdfs:subClassOf rdf:resource="PE2_Hosting_Service"/>

```
</rdfs:Class>

<rdfs:Class  rdf:about="PE6_Software_Hosting_Service">
        <rdfs:subClassOf rdf:resource="PE5_Digital_Hosting_Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE7_Data_Hosting_Service">
        <rdfs:subClassOf rdf:resource="PE5_Digital_Hosting_Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE8_E-Service">
        <rdfs:subClassOf rdf:resource="PE1_Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE10_Digital_Curating_Service">
        <rdfs:subClassOf rdf:resource="PE3_Curating_Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE11_Software_Curating_Service">
        <rdfs:subClassOf rdf:resource="PE10_Digital_Curating_Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE12_Data_Curating_Service">
        <rdfs:subClassOf rdf:resource="PE10_Digital_Curating_Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE13_Software_Computing_E-Service">
        <rdfs:subClassOf rdf:resource="PE6_Software_Hosting_Service"/>
        <rdfs:subClassOf rdf:resource="PE8_E-Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE14_Software_Delivery_E-Service">
        <rdfs:subClassOf rdf:resource="PE6_Software_Hosting_Service"/>
        <rdfs:subClassOf rdf:resource="PE8_E-Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE15_Data_E-Service">
        <rdfs:subClassOf rdf:resource="PE7_Data_Hosting_Service"/>
        <rdfs:subClassOf rdf:resource="PE8_E-Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE16_Curated_Software_E-Service">
        <rdfs:subClassOf rdf:resource="PE11_Software_Curating_Service"/>
        <rdfs:subClassOf rdf:resource="PE14_Software_Delivery_E-Service"/>
        <rdfs:subClassOf rdf:resource="PE13_Software_Computing_E-Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE17_Curated_Data_E-Service">
        <rdfs:subClassOf rdf:resource="PE12_Data_Curating_Service"/>
        <rdfs:subClassOf rdf:resource="PE15_Data_E-Service"/>
</rdfs:Class>
```

```
<rdfs:Class  rdf:about="PE18_Dataset">
        <rdfs:subClassOf rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE19_Persistent_Digital_Object">
        <rdfs:subClassOf rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE20_Volatile_Digital_Object">
        <rdfs:subClassOf rdf:resource="PE32_Curated_Thing"/>
        <rdfs:subClassOf rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE21_Persistent_Software">
        <rdfs:subClassOf rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:subClassOf rdf:resource="PE19_Persistent_Digital_Object"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE22_Persistent_Dataset">
        <rdfs:subClassOf rdf:resource="PE18_Dataset"/>
        <rdfs:subClassOf rdf:resource="PE19_Persistent_Digital_Object"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE23_Volatile_Software">
        <rdfs:subClassOf rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:subClassOf rdf:resource="PE20_Volatile_Digital_Object"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE24_Volatile_Dataset">
        <rdfs:subClassOf rdf:resource="PE18_Dataset"/>
        <rdfs:subClassOf rdf:resource="PE20_Volatile_Digital_Object"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE25_RI_Consortium">
   <rdfs:subClassOf rdf:resource="PE34_Team"/>
</rdfs:Class>


<rdfs:Class rdf:about="PE26_RI_Project">
        <rdfs:subClassOf rdf:resource="PE35_Project"/>
</rdfs:Class>

<rdfs:Class rdf:about="PE28_Curation_Plan">
        <rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/E29_Design_or_Procedure"/>
</rdfs:Class>

<rdfs:Class rdf:about="PE29_Access_Point">
        <rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E51_Contact_Point"/>
</rdfs:Class>
```

```
<rdfs:Class  rdf:about="PE32_Curated_Thing">
        <rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E70_Thing"/>
</rdfs:Class>

<rdfs:Class  rdf:about="http://www.cidoc-crm.org/cidoc-crm/E78_Curated_Holding">
        <rdfs:subClassOf rdf:resource="PE32_Curated_Thing"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE33_E-Access_Brokering_Service">
        <rdfs:subClassOf rdf:resource="PE8_E-Service"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE34_Team">
        <rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E74_Group"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE35_Project">
        <rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E7_Activity"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE36_Competency_Type">
        <rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E55_Type"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE37_Protocol_Type">
        <rdfs:subClassOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E55_Type"/>
</rdfs:Class>

<rdfs:Class  rdf:about="PE38_Schema">
        <rdfs:subClassOf rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
</rdfs:Class>

<rdf:Property rdf:about="PP1_currently_offers">
        <rdfs:domain rdf:resource="PE26_RI_Project"/>
        <rdfs:range rdf:resource="PE1_Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/P9_consists_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP1i_is_currently_offered_by">
        <rdfs:domain rdf:resource="PE1_Service"/>
        <rdfs:range rdf:resource="PE26_RI_Project"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P9i_forms_part_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP2_provided_by">
        <rdfs:domain rdf:resource="PE1_Service"/>
        <rdfs:range rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E39_Actor"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P14_carried_out_by"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP2i_provides">
        <rdfs:domain rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E39_Actor"/>
        <rdfs:range rdf:resource="PE1_Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P14i_performed"/>
</rdf:Property>

<rdf:Property rdf:about="PP4_hosts_object">
        <rdfs:domain rdf:resource="PE2_Hosting_Service"/>
        <rdfs:range rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E70_Thing"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16_used_specific_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP4i_is_object_hosted_by">
        <rdfs:domain rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E70_Thing"/>
        <rdfs:range rdf:resource="PE2_Hosting_Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16i_was_used_for"/>
</rdf:Property>

<rdf:Property rdf:about="PP6_hosts_digital_object">
        <rdfs:domain rdf:resource="PE5_Digital_Hosting_Service"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="PP4_hosts_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP6i_is_digital_object_hosted_by">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:range rdf:resource="PE5_Digital_Hosting_Service"/>
        <rdfs:subPropertyOf rdf:resource="PP4i_is_object_hosted_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP7_hosts_software_object">
        <rdfs:domain rdf:resource="PE6_Software_Hosting_Service"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP6_hosts_digital_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP7i_is_software_object_hosted_by">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:range rdf:resource="PE6_Software_Hosting_Service"/>
        <rdfs:subPropertyOf rdf:resource="PP6i_is_digital_object_hosted_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP8_hosts_dataset">
        <rdfs:domain rdf:resource="PE7_Data_Hosting_Service"/>
        <rdfs:range rdf:resource="PE18_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP6_hosts_digital_object"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP8i_is_dataset_hosted_by">
        <rdfs:domain rdf:resource="PE18_Dataset"/>
        <rdfs:range rdf:resource="PE7_Data_Hosting_Service"/>
        <rdfs:subPropertyOf rdf:resource="PP6i_is_digital_object_hosted_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP11_curates_volatile_digital_object">
        <rdfs:domain rdf:resource="PE10_Digital_Curating_Service"/>
        <rdfs:range rdf:resource="PE20_Volatile_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="PP32_curates"/>
</rdf:Property>

<rdf:Property rdf:about="PP11i_is_volatile_digital_object_curated_by">
        <rdfs:domain rdf:resource="PE20_Volatile_Digital_Object"/>
        <rdfs:range rdf:resource="PE10_Digital_Curating_Service"/>
        <rdfs:subPropertyOf rdf:resource="PP32i_is_curated_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP12_curates_volatile_software">
        <rdfs:domain rdf:resource="PE11_Software_Curating_Service"/>
        <rdfs:range rdf:resource="PE23_Volatile_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP11_curates_volatile_digital_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP12i_is_volatile_software_curated_by">
        <rdfs:domain rdf:resource="PE23_Volatile_Software"/>
        <rdfs:range rdf:resource="PE11_Software_Curating_Service"/>
        <rdfs:subPropertyOf rdf:resource="PP11i_is_volatile_digital_object_curated_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP13_curates_volatile_dataset">
        <rdfs:domain rdf:resource="PE12_Data_Curating_Service"/>
        <rdfs:range rdf:resource="PE24_Volatile_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP11_curates_volatile_digital_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP13i_is_volatile_dataset_curated_by">
        <rdfs:domain rdf:resource="PE24_Volatile_Dataset"/>
        <rdfs:range rdf:resource="PE12_Data_Curating_Service"/>
        <rdfs:subPropertyOf rdf:resource="PP11i_is_volatile_digital_object_curated_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP14_runs_on_request">
        <rdfs:domain rdf:resource="PE13_Software_Computing_E-Service"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16_used_specific_object"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP14i_is_run_by">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:range rdf:resource="PE13_Software_Computing_E-Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16i_was_used_for"/>
</rdf:Property>

<rdf:Property rdf:about="PP15_delivers_on_request">
        <rdfs:domain rdf:resource="PE14_Software_Delivery_E-Service"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16_used_specific_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP15i_is_delivered_by">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:range rdf:resource="PE14_Software_Delivery_E-Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16i_was_used_for"/>
</rdf:Property>

<rdf:Property rdf:about="PP16_has_persistent_digital_object_part">
        <rdfs:domain rdf:resource="PE19_Persistent_Digital_Object"/>
        <rdfs:range rdf:resource="PE19_Persistent_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P106_is_composed_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP16i_is_persistent_digital_object_part_of">
        <rdfs:domain rdf:resource="PE19_Persistent_Digital_Object"/>
        <rdfs:range rdf:resource="PE19_Persistent_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P106i_forms_part_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP17_has_snapshot">
        <rdfs:domain rdf:resource="PE20_Volatile_Digital_Object"/>
        <rdfs:range rdf:resource="PE19_Persistent_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P130_shows_features_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP17i_is_snapshot_of">
        <rdfs:domain rdf:resource="PE19_Persistent_Digital_Object"/>
        <rdfs:range rdf:resource="PE20_Volatile_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P130i_features_are_also_found_on"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP18_has_digital_object_part">
        <rdfs:domain rdf:resource="PE20_Volatile_Digital_Object"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P106_is_composed_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP18i_is_digital_object_part_of">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:range rdf:resource="PE20_Volatile_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P106i_forms_part_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP19_has_persistent_software_part">
        <rdfs:domain rdf:resource="PE21_Persistent_Software"/>
        <rdfs:range rdf:resource="PE21_Persistent_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP16_has_persistent_digital_object_part"/>
</rdf:Property>

<rdf:Property rdf:about="PP19i_is_persistent_software_part_of">
        <rdfs:domain rdf:resource="PE21_Persistent_Software"/>
        <rdfs:range rdf:resource="PE21_Persistent_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP16i_is_persistent_digital_object_part_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP20_has_persistent_dataset_part">
        <rdfs:domain rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:range rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP16_has_persistent_digital_object_part"/>
</rdf:Property>

<rdf:Property rdf:about="PP20i_is_persistent_dataset_part_of">
        <rdfs:domain rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:range rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP16i_is_persistent_digital_object_part_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP21_has_software_part">
        <rdfs:domain rdf:resource="PE23_Volatile_Software"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP18_has_digital_object_part"/>
</rdf:Property>

<rdf:Property rdf:about="PP21i_is_software_part_of">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:range rdf:resource="PE23_Volatile_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP18i_is_digital_object_part_of"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP22_has_release">
        <rdfs:domain rdf:resource="PE23_Volatile_Software"/>
        <rdfs:range rdf:resource="PE21_Persistent_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP17_has_snapshot"/>
</rdf:Property>

<rdf:Property rdf:about="PP22i_is_release_of">
        <rdfs:domain rdf:resource="PE21_Persistent_Software"/>
        <rdfs:range rdf:resource="PE23_Volatile_Software"/>
        <rdfs:subPropertyOf rdf:resource="PP17i_is_snapshot_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP23_has_dataset_part">
        <rdfs:domain rdf:resource="PE24_Volatile_Dataset"/>
        <rdfs:range rdf:resource="PE18_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP18_has_digital_object_part"/>
</rdf:Property>

<rdf:Property rdf:about="PP23i_is_dataset_part_of">
        <rdfs:domain rdf:resource="PE18_Dataset"/>
        <rdfs:range rdf:resource="PE24_Volatile_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP18i_is_digital_object_part_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP24_has_dataset_snapshot">
        <rdfs:domain rdf:resource="PE24_Volatile_Dataset"/>
        <rdfs:range rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP17_has_snapshot"/>
</rdf:Property>

<rdf:Property rdf:about="PP24i_is_dataset_snapshot_of">
        <rdfs:domain rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:range rdf:resource="PE24_Volatile_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="PP17i_is_snapshot_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP25_has_maintaining_RI">
        <rdfs:domain rdf:resource="PE26_RI_Project"/>
        <rdfs:range rdf:resource="PE25_RI_Consortium"/>
        <rdfs:subPropertyOf rdf:resource="PP44_has_maintaining_team"/>
</rdf:Property>

<rdf:Property rdf:about="PP25i_is_maintaining_RI_of">
        <rdfs:domain rdf:resource="PE25_RI_Consortium"/>
        <rdfs:range rdf:resource="PE26_RI_Project"/>
        <rdfs:subPropertyOf rdf:resource="PP44i_is_maintaining_team_of"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP28_has_designated_access_point">
        <rdfs:domain rdf:resource="PE8_E-Service"/>
        <rdfs:range rdf:resource="PE29_Access_Point"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P1_is_identified_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP28i_is_designated_access_point_of">
        <rdfs:domain rdf:resource="PE29_Access_Point"/>
        <rdfs:range rdf:resource="PE8_E-Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/P1i_identifies"/>
</rdf:Property>

<rdf:Property rdf:about="PP29_uses_access_protocol">
        <rdfs:domain rdf:resource="PE8_E-Service"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16_used_specific_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP29i_is_access_protocol_used_by">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D14_Software"/>
        <rdfs:range rdf:resource="PE8_E-Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16i_was_used_for"/>
</rdf:Property>

<rdf:Property rdf:about="PP31_uses_curation_plan">
        <rdfs:domain rdf:resource="PE3_Curating_Service"/>
        <rdfs:range rdf:resource="PE28_Curation_Plan"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P33_used_specific_technique"/>
</rdf:Property>

<rdf:Property rdf:about="PP31i_is_curation_plan_used_by">
        <rdfs:domain rdf:resource="PE28_Curation_Plan"/>
        <rdfs:range rdf:resource="PE3_Curating_Service"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P33i_was_used_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP32_curates">
        <rdfs:domain rdf:resource="PE3_Curating_Service"/>
        <rdfs:range rdf:resource="PE32_Curated_Thing"/>
</rdf:Property>

<rdf:Property rdf:about="PP32i_is_curated_by">
        <rdfs:domain rdf:resource="PE32_Curated_Thing"/>
        <rdfs:range rdf:resource="PE3_Curating_Service"/>
</rdf:Property>
```

```
<!-- the next two declarations affect the CIDOC CRM P147 curated (was curated by)
which is declared as subproperty of PP32 curates (is curated by) -->
<rdf:Property rdf:about="http://www.cidoc-crm.org/cidoc-crm/P147_curated">
        <rdfs:subPropertyOf rdf:resource="PP32_curates"/>
</rdf:Property>

<rdf:Property rdf:about="http://www.cidoc-crm.org/cidoc-crm/P147i_was_curated_by">
        <rdfs:subPropertyOf rdf:resource="PP32i_is_curated_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP39_is_metadata_for">
        <rdfs:domain rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/P129_is_about"/>
</rdf:Property>

<rdf:Property rdf:about="PP39i_has_metadata">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:range rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P129i_is_subject_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP40_created_successor_of">
        <rdfs:domain rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E65_Creation"/>
        <rdfs:range rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16_used_specific_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP40i_is_deprecated_by">
        <rdfs:domain rdf:resource="PE22_Persistent_Dataset"/>
        <rdfs:range rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E65_Creation"/>
        <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16i_was_used_for"/>
</rdf:Property>

<rdf:Property rdf:about="PP41_is_index_of">
        <rdfs:domain rdf:resource="PE24_Volatile_Dataset"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
</rdf:Property>

<rdf:Property rdf:about="PP41i_is_indexed_by">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:range rdf:resource="PE24_Volatile_Dataset"/>
</rdf:Property>

<rdf:Property rdf:about="PP42_has_declarative_time">
        <rdfs:domain rdf:resource="PE1_Service"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#dateTime"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP43_supported_project_activity">
        <rdfs:domain rdf:resource="PE35_Project"/>
        <rdfs:range rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E7_Activity"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/P9_consists_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP43i_is_project_activity_supported_by">
        <rdfs:domain rdf:resource="http://www.cidoc-crm.org/cidoc-crm/E7_Activity"/>
        <rdfs:range rdf:resource="PE35_Project"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/P9i_forms_part_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP44_has_maintaining_team">
        <rdfs:domain rdf:resource="PE35_Project"/>
        <rdfs:range rdf:resource="PE34_Team"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P17_was_motivated_by"/>
</rdf:Property>

<rdf:Property rdf:about="PP44i_is_maintaining_team_of">
        <rdfs:domain rdf:resource="PE34_Team"/>
        <rdfs:range rdf:resource="PE35_Project"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/P17i_motivated"/>
</rdf:Property>

<rdf:Property rdf:about="PP45_has_competency">
        <rdfs:domain rdf:resource="PE1_Service"/>
        <rdfs:range rdf:resource="PE36_Competency_Type"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P21_had_general_purpose"/>
</rdf:Property>

<rdf:Property rdf:about="PP45i_is_competency_of">
        <rdfs:domain rdf:resource="PE36_Competency_Type"/>
        <rdfs:range rdf:resource="PE1_Service"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P21i_was_purpose_of"/>
</rdf:Property>

<rdf:Property rdf:about="PP46_brokers_access_to">
        <rdfs:domain rdf:resource="PE33_E-Access_Brokering_Service"/>
        <rdfs:range rdf:resource="PE8_E-Service"/>
</rdf:Property>

<rdf:Property rdf:about="PP46i_has_access_brokered_by">
        <rdfs:domain rdf:resource="PE8_E-Service"/>
        <rdfs:range rdf:resource="PE33_E-Access_Brokering_Service"/>
</rdf:Property>
```

```
<rdf:Property rdf:about="PP47_has_protocol_type">
        <rdfs:domain rdf:resource="PE8_E-Service"/>
        <rdfs:range rdf:resource="PE37_Protocol_Type"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P125_used_object_of_type"/>
</rdf:Property>

<rdf:Property rdf:about="PP47i_is_protocol_type_of">
        <rdfs:domain rdf:resource="PE37_Protocol_Type"/>
        <rdfs:range rdf:resource="PE8_E-Service"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P125i_was_type_of_object_used_in"/>
</rdf:Property>

<rdf:Property rdf:about="PP48_uses_protocol_parameter">
        <rdfs:domain rdf:resource="PE8_E-Service"/>
        <rdfs:range rdf:resource="PE38_Schema"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-
crm/P16_used_specific_object"/>
</rdf:Property>

<rdf:Property rdf:about="PP48i_is_protocol_parameter_of">
        <rdfs:domain rdf:resource="PE38_Schema"/>
        <rdfs:range rdf:resource="PE8_E-Service"/>
  <rdfs:subPropertyOf rdf:resource="http://www.cidoc-crm.org/cidoc-crm/P16i_was_used_for"/>
</rdf:Property>

<rdf:Property rdf:about="PP49_provides_access_point">
        <rdfs:domain rdf:resource="PE8_E-Service"/>
        <rdfs:range rdf:resource="PE29_Access_Point"/>
</rdf:Property>

<rdf:Property rdf:about="PP49i_is_access_point_provided_by">
        <rdfs:domain rdf:resource="PE29_Access_Point"/>
        <rdfs:range rdf:resource="PE8_E-Service"/>
</rdf:Property>

<rdf:Property rdf:about="PP50_accessible_at">
        <rdfs:domain rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
        <rdfs:range rdf:resource="PE29_Access_Point"/>
</rdf:Property>

<rdf:Property rdf:about="PP50i_provides_access_to">
        <rdfs:domain rdf:resource="PE29_Access_Point"/>
        <rdfs:range rdf:resource="http://www.ics.forth.gr/isl/CRMdig/D1_Digital_Object"/>
</rdf:Property>

</rdf:RDF>
```