PROMISE

Participative Research labOratory for Multimedia and Multilingual Information Systems Evaluation

FP7 ICT 2009.4.3, Intelligent Information Management

Deliverable 3.2
Specification of the evaluation infrastructure based on user requirements

Version 1.0, 30th August 2011

Network of Excellence co-funded by the 7th Framework Program of the European Commission, grant agreement no. 258191

Document Information

Deliverable number:	3.2
Deliverable title:	Specification of the evaluation infrastructure based on user requirements
Delivery date:	30/08/2011
Lead contractor for this deliverable	UNIPD
Author(s):	Maristella Agosti, Martin Braschler, Emanuele Di Buccio, Marco Dussin, Nicola Ferro, Guido Lorenzo Granato, Ivano Masiero, Emanuele Pianta, Giuseppe Santucci, Gianmaria Silvello, and Giuseppe Tino
Participant(s):	UNIPD, ROMA1, ZHAW
Workpackage:	3
Workpackage title:	Evaluation Infrastructure
Workpackage leader:	UNIPD
Dissemination Level:	PU – Public
Version:	1.0
Keywords:	Experimental evaluation, software infrastructure, scientific data, conceptual schema, architecture, RESTful Web service, API

History of Versions

Version	Date	Status	Author (Partner)	Description/Approval Level
0.1	07/07/2011	Draft	Marco Dussin, UNIPD	Circulated to UNIPD members
0.2	18/07/2011	Draft	Emanuele Di Buccio, Marco Dussin, Ivano Masiero, Gianmaria Silvello, UNIPD	Circulated to UNIPD members
0.3	20/07/2011	Draft	Emanuele Di Buccio, Gianmaria Silvello, UNIPD	Circulated to UNIPD members
0.4	02/08/2011	Draft	Emanuele Di Buccio, Marco Dussin, Ivano Masiero, Gianmaria Silvello UNIPD	Circulated to UNIPD members
0.5	06/08/2011	Draft	Nicola Ferro UNIPD	First draft circulated to reviewers and partners

Version	Date	Status	Author (Partner)	Description/Approval Level
0.6	29/08/2011	Draft	UNIPD Team	Revision of the First Draft which includes: English revisions, comments and suggestions from partners
1.0	30/08/2011	Final	UNIPD Team	Revised after partners' comments

Abstract

This deliverable provides the specification of the PROMISE evaluation infrastructure, based on the requirements emerging from the adoption of an early prototype of it in CLEF 2011 (see D3.1), the initial specification of the evaluation tasks (see D2.1), the requirements for collaboration (see D5.1), and the requirements for visual analytics (see D5.2).

The deliverable describes: (i) the conceptual schema for the entities and relationships involved in the experimental evaluation; (ii) the architecture of the evaluation infrastructure; (iii) the RESTful Web API for accessing it; (iv) some examples of its use related to performance measures and visualizations.

Table of Contents

Executive Summary

PROMISE aims at delivering a unified infrastructure and environment for data, knowledge, tools, methodologies and the user community in order to advance the experimental evaluation of complex multimedia and multilingual information systems.

The contribution of this deliverable to the overall goal is to set the specification for the PROMISE evaluation infrastructure in order to:

- Manage and provide access to the scientific data produced during evaluation activities
- Support the organization and running of evaluation campaigns
- Increase automation in the evaluation process
- Provide component-based evaluation
- Foster the usage of the managed scientific data.

The specification adopts a user-centric approach and it is based on the requirements gathered during the first 12 months of PROMISE activities. In particular, it is based on the experience gained in using an early prototype of it during the CLEF 2011 campaign (see D3.1 “Initial prototype of the evaluation infrastructure”), on the requirements defined for the initial evaluation tasks (see D2.1 “Initial specification of the evaluation tasks”), for the collaborative user interface (see D5.1 “Collaborative user interface requirements”), and for the visual analytics environment (see D5.2 “User interface and Visual analytics environment requirements”).

This specification will serve as a basis for subsequent deliverables, namely D3.3 “Prototype of the evaluation infrastructure” due at month 18 and D5.3 “Collaborative user interface prototype with annotation functionalities” due at month 24.

The first key contribution of this deliverable is a conceptual schema which describes the entities involved in the experimental evaluation and the relationships between them. Some of the main new features of this schema are:

- the capability to support not only evaluation campaigns but also general evaluation activities, such as experimentation internal to research groups or industries, and even educational activities to teach students how to carry out experimental evaluation;
- the introduction of new types of experiments, beyond traditional TREC ones, to move a step forward towards living laboratories;
- the introduction of support for visualizations to allow users to produce their own visualizations, save their state, and interact with them later on.

The conceptual schema provides the means for designing the underlying database needed to manage, keep over the time, and curate the scientific data produced during experimental evaluation. Moreover, it serves also to define the resources to be made accessible via RESTful Web services and derive their XML and JSON representations.

The second key contribution is the architecture of the evaluation infrastructure, which is modular, scalable and completely decouples the representation and access to the managed scientific data offered via RESTful Web services from the applications built over them. Moreover, this architecture strives to simplify the development and sharing of code.

The third key contribution is a set of Web Application Program Interfaces (API) to interact with all the resources managed by the system via a RESTful Web service.

Finally, the conceptual schema and the Web API are checked and exemplified in a concrete use case, which is the access to performance measures for retrieving the data needed to create advanced visualizations.

1 Introduction

Experimental evaluation is a key activity for driving and supporting the development of multilingual and multimedia information access systems. It is an essential part of the scientific process because using shared data sets and evaluation scenarios systems can be compared, performances can be better understood, and progress can be pursued and demonstrated.

Large-scale evaluation initiatives, such as Text Retrieval Conference (TREC)[1] in the United States, the Cross-Language Evaluation Forum (CLEF)[2] in Europe, and the NII/NACSIS Test Collection for IR Systems (NTCIR)[3] in Asia, contribute significantly to advancements in research and industrial innovation in the information retrieval sector, and to the building of strong research communities. A study conducted by the National Institute for Standards and Technology (NIST) reports that "for every $1 that NIST and its partners invested in TREC, at least $3.35 to $5.07 in benefits accrued to IR researchers. The internal rate of return (IRR) was estimated to be over 250% for extrapolated benefits and over 130% for unextrapolated benefits" [Rowe et al., 2010].

During their life-span, large-scale evaluation campaigns have produced a huge amount of scientific data which are extremely valuable. These experimental and scientific data provide the foundations for all the subsequent scientific production and system development and constitute an essential reference for all the literature produced in the field.

Moreover, these data are valuable also from an economic point of view, due the great amount of effort devoted to their production: [Rowe et al., 2010] estimates the overall investment in TREC at about 30 million dollars.

Nevertheless, much less attention has been paid over the years to the modelling, management, curation, and access to the scientific data produced, even though the importance of scientific data in general has been highlighted by many different institutional organizations, such as the European Commission [EU 2007], the US National Scientific Board [NSB, 2005], and the Australian Working Group on Data for Science [PMSEIC, 2006].

Our goal is to deliver a unified infrastructure and environment for data, knowledge, tools, methodologies and the user community in order to advance the experimental evaluation of complex multimedia and multilingual information systems.

The evaluation infrastructure will:

- manage and provide access to the scientific data produced during evaluation activities;
- support the organization of evaluation campaigns;
- increase the automation of the evaluation process;

[1] http://trec.nist.gov/

[2] http://www.clef-campaign.org/

[3] http://research.nii.ac.jp/ntcir/

- allow for continuous evaluation;
- foster the usage and understanding of the scientific data;

A user-centred design approach has been adopted involving the different stakeholders (e.g. scientists, evaluation campaign organizers, system developers, students) in the development of the infrastructure.

The first outcome of this effort is the Distributed Information Retrieval Evaluation Campaign Tool (DIRECT)[4] system [Agosti and Ferro, 2009; Ferro, 2011], which has been described in D3.1 "Initial prototype of the evaluation infrastructure" [PROMISE D3.1, 2011].

This deliverable takes a step forward with respect to D3.1 and operates a complete redesign of the DIRECT system by taking into consideration the experience of use during the CLEF 2011 campaign and the input coming from other PROMISE activities [PROMISE D2.1, 2011; PROMISE D5.1, 2011; PROMISE D5.2, 2011].

In particular, the contributions of this deliverable are:

- a brand new conceptual schema for describing the entities involved in the experimental evaluation and the relationships between them. Some of the main new features of this new schema are:
  - the capability to support not only evaluation campaigns but also general evaluation activities, such as experimentation internal to research groups or industries, and even educational activities to teach students how to carry out experimental evaluation;
  - the introduction of new types of experiments, beyond traditional TREC ones, to move a step forward towards living laboratories;
  - the introduction of support for visualizations to allow users to produce their own visualizations, save their state, and interact with them later on;
- a fully re-designed architecture, which is more modular and completely decouples the representation and access to the managed scientific data offered via RESTful Web services from the applications built over them;
- a set of Web Application Program Interfaces (API) to interact will all the resources managed by the system.

The deliverable is organized as follows: Section 2 provides an overview to the approach taken for the conceptual modelling of the information space entailed by the experimental evaluation; Section 3 provides a detailed description of the designed conceptual schema; Section 4 introduces the overall architecture of the evaluation infrastructure; Section 5 provides the API to interact with evaluation infrastructure; and, lastly, Section 6 provides some examples of how the evaluation infrastructure can be exploited for the interaction and visualization of the performance measures related to an evaluation activity.

[4] http://direct.dei.unipd.it/

2 Conceptual Modelling

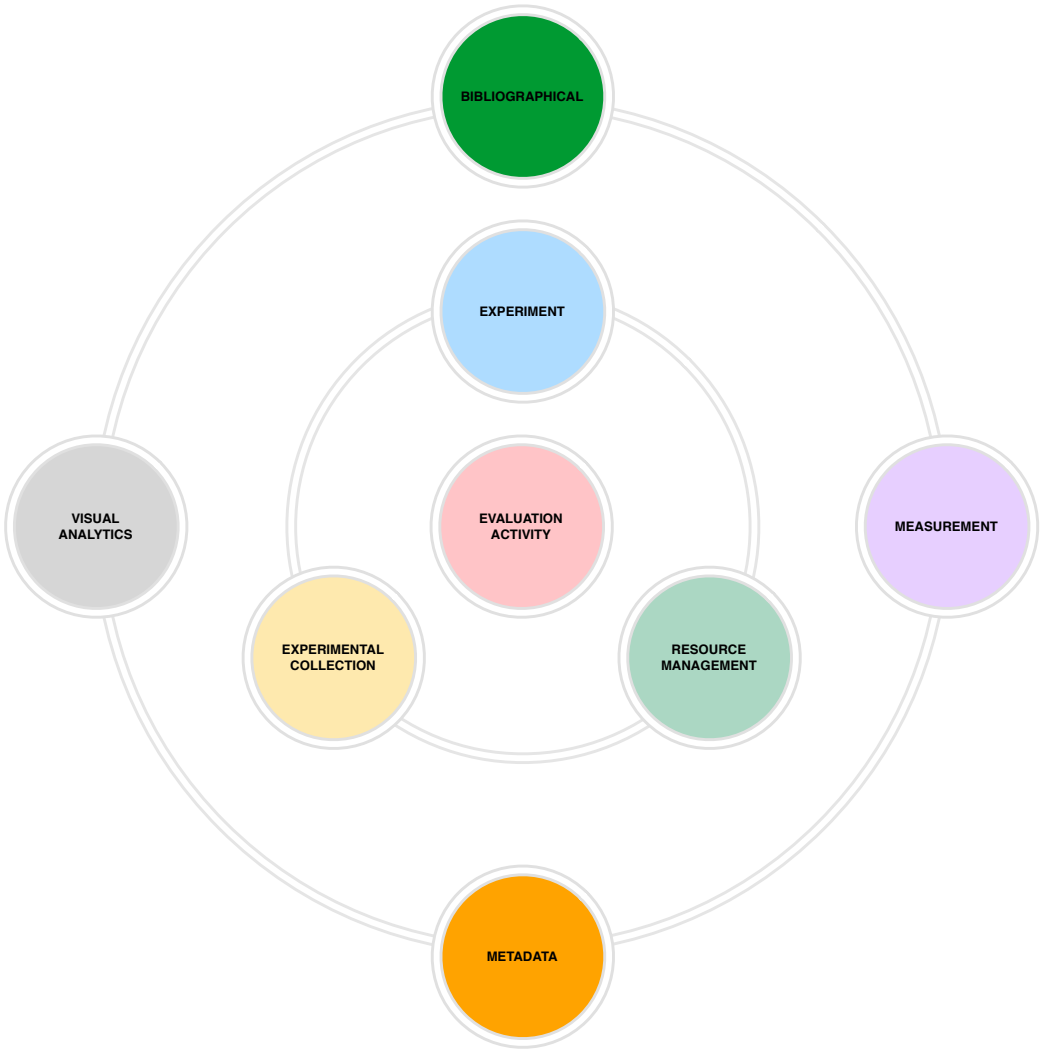


Figure 1 - The conceptual areas of the PROMISE Evaluation Infrastructure

The conceptual design of the PROMISE Evaluation Infrastructure has been divided into eight functional areas, as shown in Figure 1, which provides an intuitive representation of the conceptual areas; each distinct area has its own colour, which helps the reader to distinguish among them.

The description of the conceptual schema of the evaluation infrastructure is going to be dealt and presented in Section 3 where the presentation is going to be divided into the eight identified macro-areas, which are:

Resource Management area: The main logical core of the PROMISE Evaluation Infrastructure is strictly bounded to the *Resource Management* area (Section 3.1), which

supports the interaction between users/groups and the resources handled by the infrastructure. Resources can be actual data adopted in or produced by evaluation activities, e.g. experimental collections or experiment results, as well as the evaluation activities and tasks carried out within them.

Metadata area: The *Metadata* area (Section 3.2), which supports the description and the enrichment through metadata of the resources handled by the infrastructure.

Evaluation Activity area: The *Evaluation Activity* area (Section 3.3) identifies the core of the infrastructure. In particular, it refers to whichever type of activity that aims at the evaluation of applications, systems, and methodologies for multimodal and multimedia information access and retrieval.

Experimental Collection area: The Evaluation Activity area is connected to the *Experimental Collection* area (Section 3.4), which allows us to set up a traditional IR evaluation environment and, more specifically, it regards the different collections made available by an evaluation forum.

Experiment area: The scientific data produced during an evaluation activity are covered by the *Experiment* area (Section 3.5), which concerns the system or the application under evaluation and its components.

Measurement area: The *Measurement* area (Section 3.6), which concerns the measures adopted for evaluation activities.

Visual Analytics area: The *Visual Analytics* area (Section 3.7), which manages the information used by the infrastructure to store and recover whichever visualization of the data that the users do.

Bibliographical area: The *Bibliographical* area (Section 3.8), which is responsible for making explicit and retaining the relationship between the data that result from the evaluation activities and the scientific production based on these data.

The minimum set of attributes has been identified to describe the characteristics of each entity of the conceptual schema of the evaluation infrastructure.

3 Conceptual Schema

The same colours used in Figure 1 are used in the conceptual schemas to highlight the entities belonging to different conceptual areas.

3.1 Resource Management Area

In the PROMISE infrastructure the term *resource* refers to a generic entity that concerns evaluation activities and with which a user or a group of users can interact. Resources can be actual data adopted in or produced by these activities (e.g. experimental collections or experiment results), as well as the evaluation activities and tasks carried out within them. The remainder of Section 3 will provide a brief description of these concepts (evaluation activities, experiments ...) and discuss how they are modelled within the infrastructure.

This section concerns the management of the resources retained and made accessible by the infrastructure, specifically the relationship between the properties common to many of the resources discussed in Section 3.2-3.8 and the relationship between resources and users that interact with them.

3.1.1 Resource Management Entities

The **`Resource`** entity is introduced in this section to describe the relationships that involve many resources of a generic evaluation activity. The relationships with the specific resources will be detailed in the section concerning the area to which the resource belongs. Every **`Resource`** entity has an attribute called **`scope`** that defines the extent of the resource taken into account; it is a controlled vocabulary: {PUBLIC, PRIVATE, SHARED}.

3.1.1.1 User

The entity **`User`** refers to a generic user of the infrastructure. Each user is characterized by the following attributes:

- **`id`**: The unique identifier of the user;
- **`pwd`**: The password of the user;
- **`last_name`**: The last/family name of the user;
- **`first_name`**: The first name of the user;
- **`affiliation`**: The affiliation of the user;
- **`email`**: The email of the user;
- **`country`**: The country of the user;
- **`lang`**: The language of the user;
- **`birth_date`**: The birth date of the user;

- **gender**: The gender of the user;
- **address**: The address, i.e. the street address, of the user;
- **city**: The city of the user;
- **state**: The state/province/region of the user;
- **zip**: The postcode of the user;
- **phone**; The telephone number of the user;
- **facsimile**: The facsimile number of the user;
- **mobile**: The mobile telephone number of the user;
- **voip_caller_id**: The VoIP caller identifier of the user;
- **homepage**: The URL of the homepage of the user;
- **picture_media_type**: The MIME media type of the picture of the user;
- **created**: The creation time stamp of the user;
- **last_modified**: The last modification time stamp of the user.

3.1.1.2 Role

The **Role** entity indicates the role a user can assume in the evaluation infrastructure. The attributes associated to the **Role** entity are:

- **id**: The identifier of the role;
- **description**: The description of the role.

3.1.1.3 Group

The **Group** entity indicates a set of users that are classed together. The attributes of the **Group** entity are:

- **id**: The identifier of the group;
- **description**: The description of the group;
- **created**: The creation time stamp of the group;
- **last_modified**: The last modification time stamp of the group.

3.1.1.4 Namespace

The **Namespace** entity refers to a logical grouping of identifiers and allows the disambiguation of homonym identifiers belonging to different namespaces. The **Namespace** entity is characterized by the following attributes:

- **id**: The unique identifier of the namespace;
- **prefix**: The unique prefix for the namespace;
- **description**: The description of the namespace;
- **created**: The creation time stamp of the namespace;
- **last_modified**: The last modification time stamp of the namespace.

3.1.2 Resource Management Relationships

Figure 2 depicts the relationship existing between the **User**, **Role** and **Group** entity. A **User** can **play** none, one or more **Role**(s): for instance, a user can be both an organizer of an evaluation activity and a researcher that carries out the activity, i.e. a participant to the evaluation activity. A **Role** can be played by none, one or more users; for instance, an evaluation activity can have one or more participants, e.g. the researchers that are carrying out the experiments for writing a paper. A **User** belongs to none, one or more **Groups**; a **Group** needs not necessarily contain a **User**.

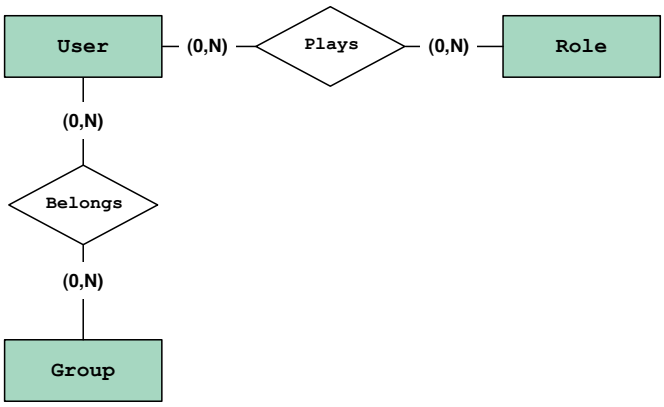


Figure 2 Relationship between the User, the Role and the Group entities

As depicted in Figure 3, a user can **create** a resource while a **Resource** must be created by a user. A **Resource** can be shared with a **Group** of **Users**. It is important to underline that the **Shares** relationship has an attribute called **privilege**, which takes values in a controlled vocabulary – i.e. {READ, READ_WRITE, DENIED}. This attribute is used only if the **scope** of the considered **Resource** is SHARED, otherwise the **share** relationship between **Resource** and **Group** does not hold.

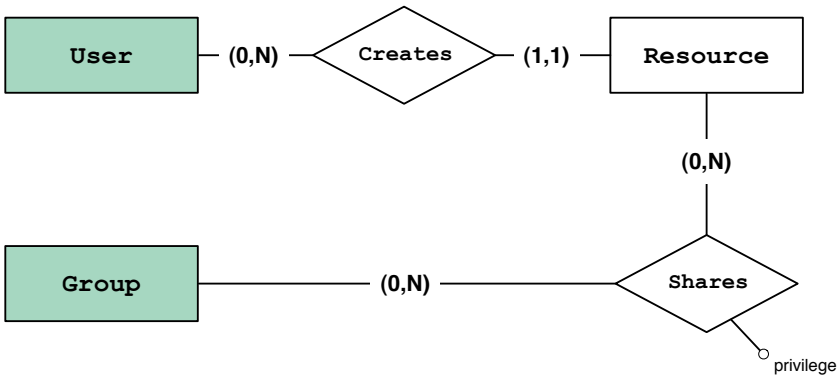


Figure 3 - Relationships between the Resource, User and Group entities

Lastly, a Resource **`Is Associated To`** one and only one **`Namespace`**; the same namespace can be associated to none, one or many resources. When a **`Resource`** is associated to a namespace it has a *mixed identifier* [Batini Et 1992] that involves the **`id`** of the resource and the **`id`** of the namespace to which the resource is associated as we can see in Figure 4 – the mixed identifier notation is the one adopted in [Batini Et 1992].

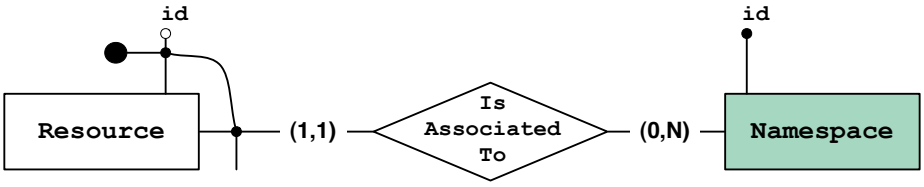


Figure 4 - Relationship between Resource and Namespace using the [Batini Et 1992] notation.

3.2 Metadata Area

3.2.1 Metadata Entities

3.2.1.1 Metadata

A metadata is usually defined as “data over data” and it is used to describe the resources of the evaluation infrastructure. A metadata is itself a resource of the infrastructure and thus it can be recursively described by another metadata; this fact is modelled by means of the recursive relationship **`Relates`** on the **`Metadata`** entity, as shown in Figure 5.

The **`Metadata`** entity is characterized by the following attributes:

- **`id`**: The unique identifier of the metadata;
- **`fields`**: The blob XML of the metadata fields;

- `created`: The creation time stamp of the metadata;
- `last_modified`: The last modification time stamp of the metadata.

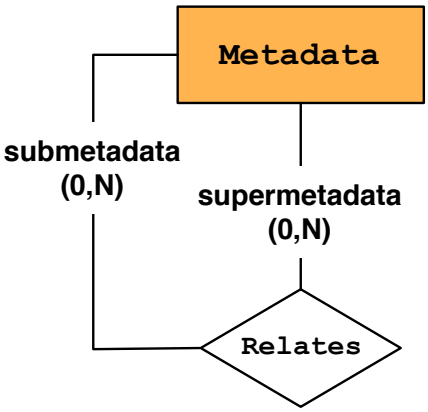


Figure 5 Relationship that involves the Metadata entity

Furthermore, in the evaluation infrastructure we can set proper access permissions on the metadata, as shown in Figure 6.

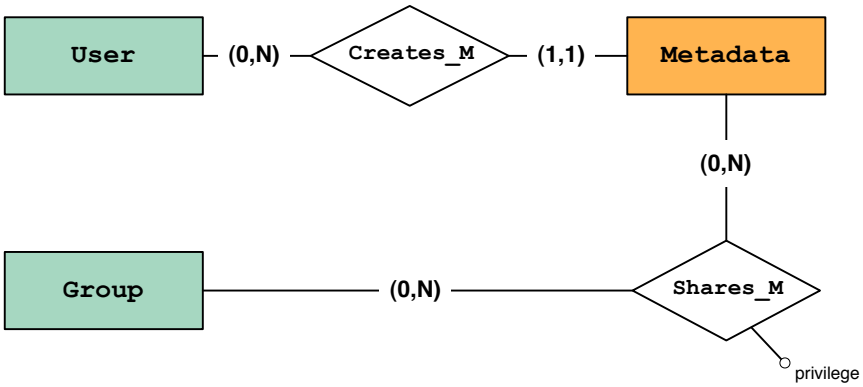


Figure 6 - Relationship between Metadata, User and Group entitie6s

Every metadata has to be associated to a namespace as we can see in Figure 7.

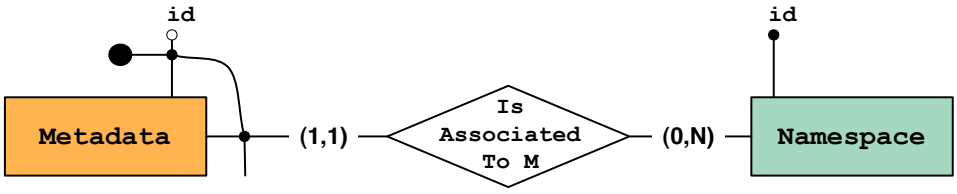


Figure 7 - Relationship between Metadata and Namespace

3.2.1.2 Metadata Set

The **Metadata Set** entity represents a logical grouping of **Metadata**. We can create hierarchies of metadata sets and this fact is modelled by means of a recursive relationship – i.e. **Includes** – on the **Metadata Set** entity as we can see in Figure 8.

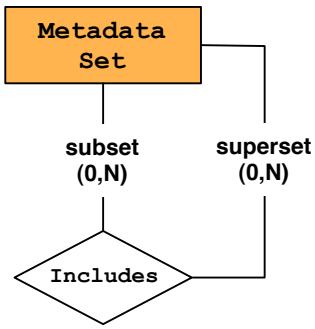


Figure 8 - Recursive relationship on the Metadata Set entity

The **Metadata Set** entity is characterized by the following attributes:

- **id**: The unique identifier of the metadata set;
- **name**: The unique prefix for the metadata set;
- **description**: The description of the metadata set;
- **created**: The creation time stamp of the metadata set;
- **last_modified**: The last modification time stamp of the metadata set.

Every metadata set has to be associated to a namespace as we can see in **Figure 9**.

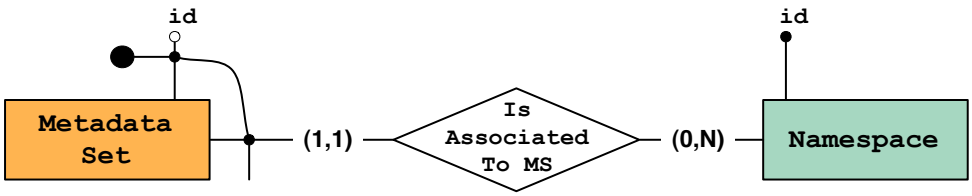


Figure 9 - Relationship between the Metadata Set entity and the Namespace entity

Furthermore, in the evaluation infrastructure we can set proper access permissions on the metadata, as shown in Figure 10.

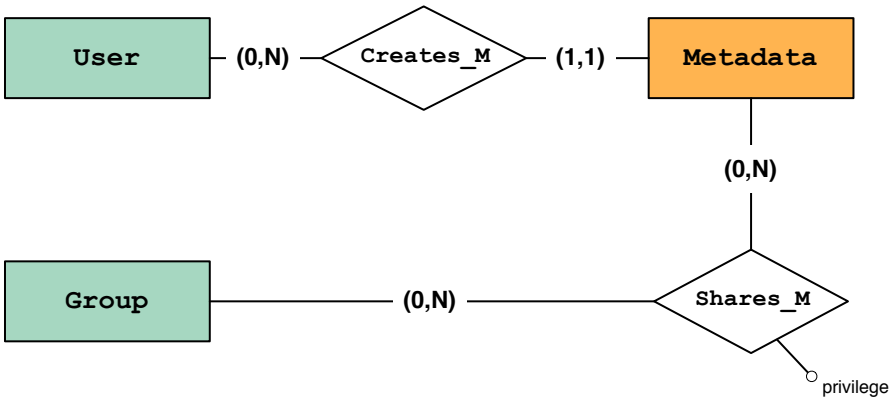


Figure 10 Relationship between the Metadata Set, the User and the Group entities.

3.2.2 Metadata Relationships

Figure 11 depicts the **Owns** relationship between **Metadata** and **Metadata Set**: a metadata can belong to no, one or more sets. As we can see from Figure 11, empty metadata sets can also be created.

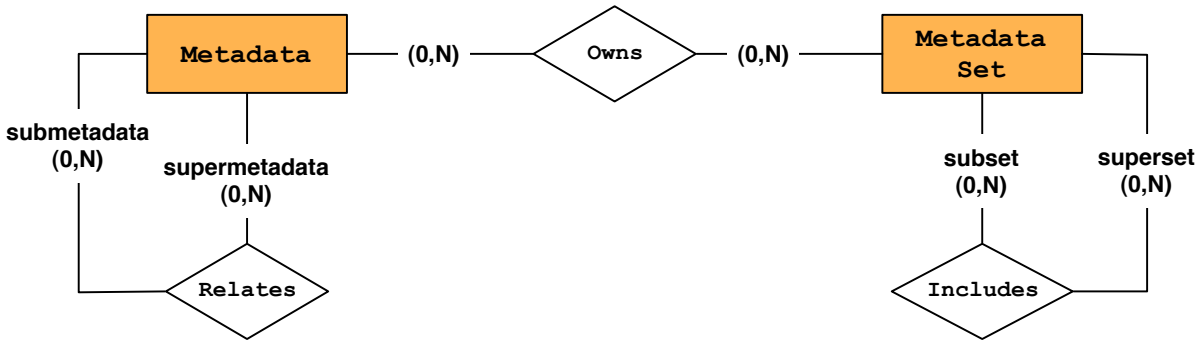


Figure 11 - Relationships that involve the Metadata and the Metadata Set entity

3.3 Evaluation Activity Area

3.3.1 Evaluation Activity Entities

3.3.1.1 Evaluation Activity

The **Evaluation Activity** entity is the core of the conceptual schema of the evaluation infrastructure. An **Evaluation Activity** is any type of activity that aims at the evaluation of applications, systems, and methodologies for multimodal and multimedia information access and retrieval. As we can see in Figure 12 the set of subclasses {**Campaign, Trial, Education**} is a specialization of the superclass **Evaluation**

Activity that distinguishes between the different typology of activities handled by the infrastructure. The **Campaign** entity deals with the different aspects of an evaluation forum, such as the different campaigns and the different editions of each campaign, the tracks along which the campaign is organized and the tasks in which each track is divided. A peculiar characteristic of a **Campaign** is to be a public and shared activity that may be undertaken by, say, academic, commercial and governmental groups that are interested in the activity organized and structured by a third-party body. Participating groups share the data on which evaluation is based and the evaluation metrics, thus allowing comparison across the techniques adopted by the diverse groups. The concept of **Campaign** derives from the traditional IR view of an evaluation activity on which basis the major international

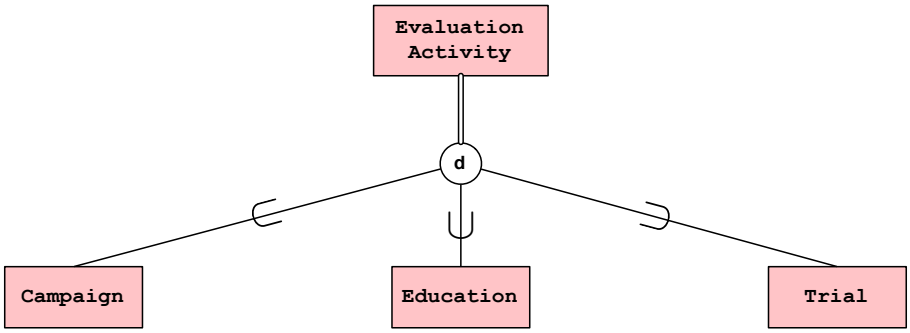


Figure 12 - Evaluation Activity entities and its subclasses

evaluation initiatives – e.g. TREC, CLEF and NTCIR – rely.

Conversely, the **Trial** entity identifies an evaluation activity that may be actively run by, say, a research group, a person or a corporate body for their own interest. This evaluation activity may be or may not be shared with the community of interest; for instance, a trial activity may be the experiments performed to write a research paper or the activities conducted to evaluate a Web application. The **Trial** activity does not have a standard organization like the **Campaign** activity and the body that undertakes the activity defines its organization. In the evaluation infrastructure we assume that a **Trial** activity has to be organized in tasks (at least one task). In a **Trial** activity there is room for defining heterogeneous organizations as well as new types of evaluation activities that may arise in the course of time.

The **Education** entity allows us to envision evaluation activities carried out for educational purposes. For instance, we can consider the case of homework assigned by a teacher as an exercise in an information retrieval academic course; in this case the course may be considered an educational evaluation activity and the homework may be seen as a task[5] within this activity.

We can point out the following attributes for the **Evaluation Activity** entity:

[5] Task will be defined in Section 3.3.1.2.

- **id**: The identifier of the evaluation activity;
- **name**: The name or acronym of the evaluation activity;
- **description**: A short description of the evaluation activity;
- **created**: The creation time stamp of the evaluation activity;
- **last_modified**: The last modification time stamp of the evaluation activity.

The **name** of the evaluation activity allows us to associate the activity with an easily recognizable name such as TREC7 or CLEF2010 for a campaign or IR_Paper_LM_2011 for a trial evaluation activity; this attribute could also be an acronym identifying the activity.

An **Evaluation Activity** has a **Namespace** as depicted by the relationship in Figure 13.

Figure 13 - Relationship between the Evaluation Activity entity and the Namespace entity

Furthermore, in the evaluation infrastructure we can set proper access permissions on the evaluation activities, as shown in Figure 14. It is worthwhile to point out that a **Campaign** is always PUBLIC, whereas a **Trial** and an **Education** activity may be PUBLIC, PRIVATE or SHARED.

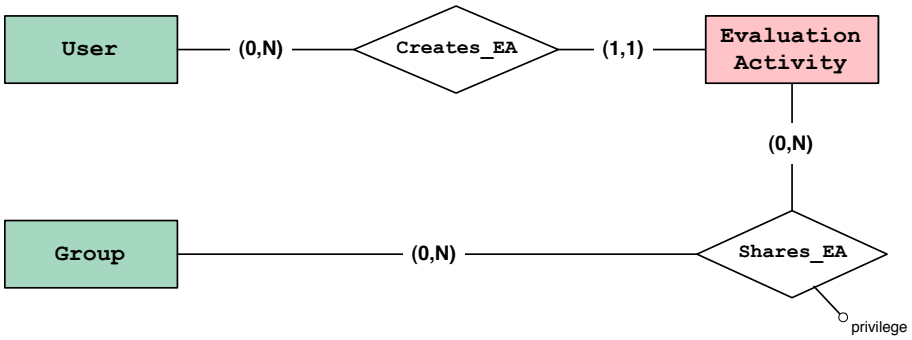


Figure 14 - Relationship between the Evaluation Activity entity, the User and the Group entities

Lastly, we can describe an evaluation activity by means of metadata, as shown in the figure below.

Figure 15 - Relationship between the Metadata entity and the Evaluation Activity entity

3.3.1.2 Task

A **Task** refers to a specific piece of work that is undertaken within the evaluation activity and aims at testing a specific (research) hypothesis. An example is the ad-hoc task in an evaluation campaign, e.g. TREC or CLEF; the aim of the ad-hoc task is to test the ability of retrieval systems to retrieve accurate and complete ranked list of documents in response to a set of information need statements [Voorhees and Harman 2005]. The research hypothesis does not necessarily refer to the effectiveness of a retrieval technique, but it may concern the effect of a pooling strategy or a user-centric analysis of an application. e.g. a web portal, where the retrieval system may be only one of the constituting blocks. The definition of the **Task** entity is therefore more general than the one adopted in traditional campaigns since the infrastructure aims at retaining and sharing information on a generic evaluation activity.

A **Task** entity is characterized by the following atomic attributes:

- **id**: The identifier of the task;
- **description:** A short description of the task;
- **created**: The creation time stamp of the task;
- **last_modified**: The last modification time stamp of the task.

Every Task has to be associated to a Namespace as we can see in Figure 16.

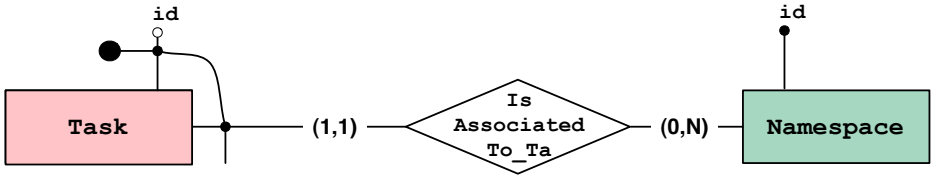


Figure 16 - Relationship between the Task entity and the Namespace entity

Furthermore, in the evaluation infrastructure we can set proper access permissions on the task, as shown in Figure 17.

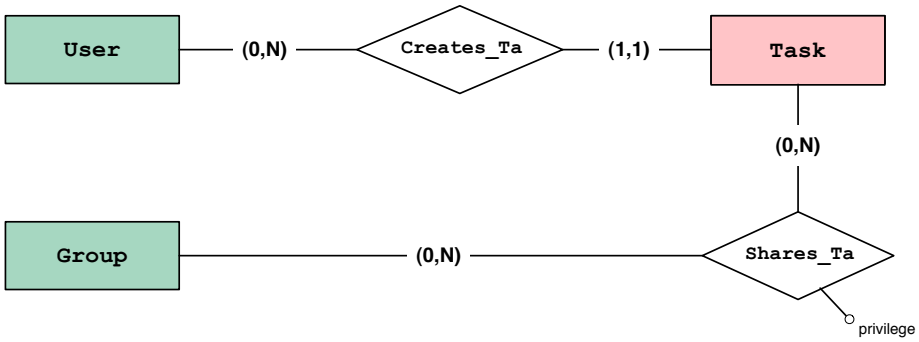


Figure 17 - Relationship between the Task entity, the User and the Group entities

Lastly, we can describe a task by means of metadata, as shown in the figure below.

Figure 18 - Relationship between the Metadata entity and the Task entity

3.3.1.3 Track

Tasks carried out within a campaign are grouped into **Track**s. Tasks within a track share a common aspect or research area within a **Campaign**. Please note that the concepts of track and task may be assume different meanings on the reality of interest basis; for instance, we may need to model a task composed by many sub-tasks, in this case the main task is represented by a **Track** and the sub-tasks as **Task**s. An example is the Web Track in TREC 2009 where two tasks are carried out, i.e. the *ad-doc* and the *diversity* task. They share the same research area, i.e. web retrieval, and the same goal: searching a static set of documents using previously-unseen information need statements and ranking the document in the collection in decreasing probability of relevance. But they differ in the judging process and in the adopted effectiveness measures; indeed, the objective of the diversity task is to investigate the retrieval technique capability in returning a ranked list of results that provides complete coverage of an information need statement, thus minimizing the redundancy in the result list.

The following attributes can be identified for the **Track** entity:

- **id**: The identifier of the track;
- **description**: A short description of the track;
- **submission_deadline**: The date when the participants must submit the experiments;
- **topic_status**: The status of the topic creation procedure for topics adopted in the track;
- **submission_status**: The status of the experiment submission for the track;
- **pool_status**: The status of the pool;
- **metric_status**: The status of the computation of the values for the metrics adopted in the track;
- **created**: The creation time stamp of the track;
- **last_modified**: The last modification time stamp of the track.

The four status attributes **topic_status**, **submission_status**, **pool_status** and **metric_status** can have one of the following values: {NOT_STARTED_YET, ONGOING, ENDED, AVAILABLE}.

Every track has to be associated to a namespace as we can see in Figure 19.

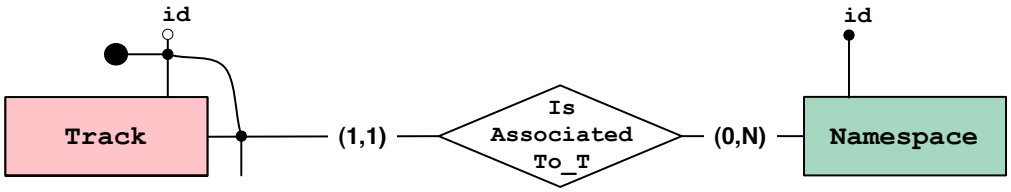


Figure 19 - Relationship between Track and Namespace

Furthermore, in the evaluation infrastructure we can set proper access permissions on the track, as shown in Figure 20.

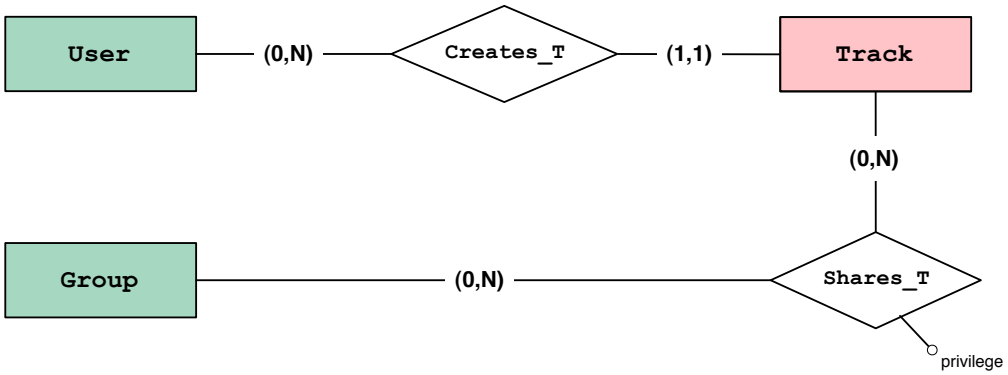


Figure 20 - Relationship between Track, User and Group

Lastly, we can describe a track by means of metadata, as shown in the figure below.

Figure 21 - Relationship between Metadata and Track

3.3.1.4 Series

The **Series** entity represents a family of campaigns or tracks; e.g. the campaigns *CLEF2008*, *CLEF2009* and *CLEF2010* belong to the same series, which may be called "*Cross-Evaluation Evaluation Forum*". This entity can be seen as a taxonomy of campaigns and tracks handled by the infrastructure.

The following attributes can be identified for the **Series** entity:

- **id**: The identifier of the series;
- **name**: A representative name of the series;
- **description**: A short description of the series.

The **Series** entity is not associated to a namespace and it has no access permissions. We can describe a track by means of metadata, as shown in the figure below.

Figure 22 - Relationship between Metadata and Series

3.3.2 Evaluation Activity Relationships

Figure 23 shows the relationships between entities of the Evaluation Activity area.

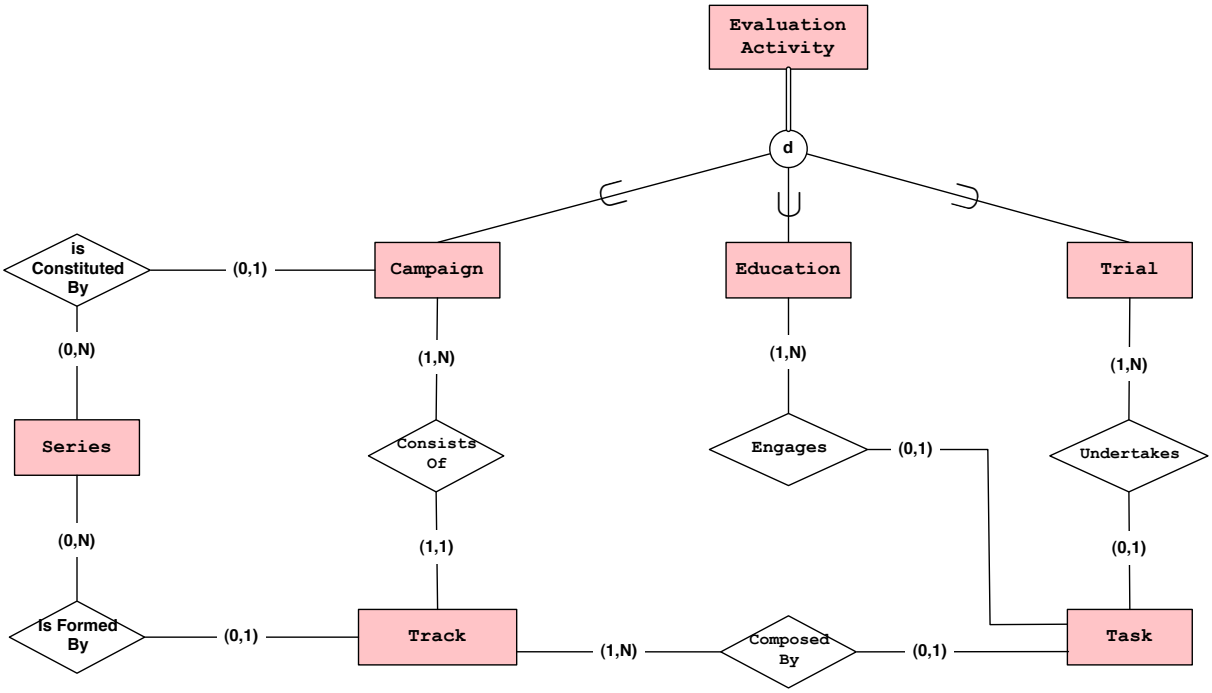


Figure 23 - Relationship between entities of the Evaluation Activity area

A **Campaign** must **consist of** one or more **Tracks**, which in turn must be **composed by** one or more **Tasks**. The organization in tracks is a peculiarity of the **Campaigns**. It is important to highlight that we may need to store information about the whole history of a track in a series of campaigns organized by the same body; e.g. we may need to know that the *Ad hoc track* in TREC has been organized for 8 years from 1992 to 1999. To this end **Track** has a relationship with the **Series** entity as well as the **Campaign** entity.

On the other hand, a **Trial** and an **Education** activity must undertake one or more **Tasks** which are not necessarily organized into tracks.

3.4 Experimental Collection Area

3.4.1 Experimental Collection Entities

3.4.1.1 Experimental Collection

The **`experimental collection`** is a logical entity that allows us to set up a traditional IR evaluation environment. A classical IR experimental collection is a triple composed by a corpus of documents, a group of topics and a set of assessments on the documents with regard to the considered topics. The attributes of this entity are:

- **`id`**: The identifier of the experimental collection;
- **`description:`** A short description of the experimental collection (e.g. "TREC 2001 Web Track Ad Hoc Test Collection" or "TREC 7-8 Ad Hoc Test Collection with Graded Judgements [Järvelin and Kekäläinen 2002]");
- **`created`**: Creation time stamp of the experimental collection;
- **`last_modified`**: The last modification time stamp of the experimental collection.

Every experimental collection has to be associated to a namespace as we can see in Figure 24.

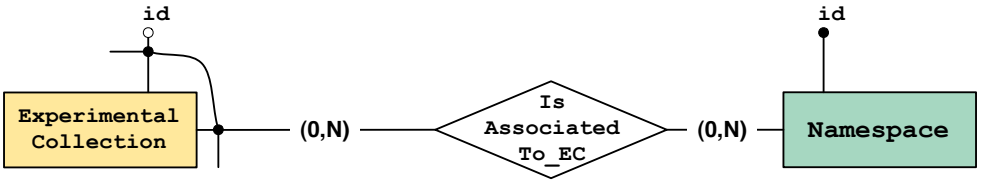


Figure 24 - Relationship between Experimental Collection and Namespace

Furthermore, in the evaluation infrastructure we can set proper access permissions on the experimental collection, as shown in the figure below.

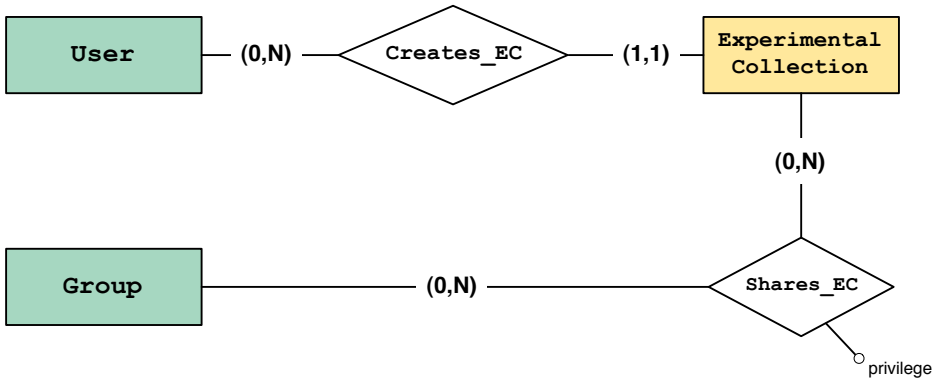


Figure 25 - Relationship between Experimental Collection, User and Group

Lastly, we can describe an experimental collection by means of metadata, as shown in the figure below.

Figure 26 - Relationship between Metadata and Experimental Collection

3.4.1.2 Relation

The **Relation** entity represents the taxonomy of relationship types that we can define between the informative objects handled by the infrastructure. The **Relation** entity has the following attributes:

- **id**: The identifier of the relation;
- **description**: A short description of the relation.

3.4.1.3 Document

In general we call **Document** the object on which the evaluated system acts, e.g. the object which is retrieved by the system under evaluation.

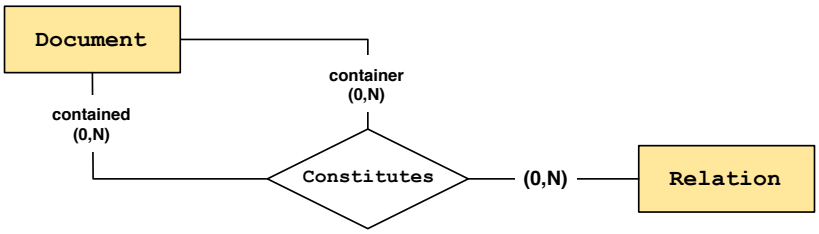


Figure 27 - Relationship between Document and Relation

In Figure 27, the recursive relationship **Constitutes** indicates that a document may be constituted by one or more other documents. This allows us to consider documents with a variable granularity, i.e. the object of the access/retrieval evaluation can be a whole document (e.g. a full-text document) or just an identifiable part of it (e.g. a section of a chapter of a book). **Constitutes** is a ternary relationship which also involves the **Relation** entity which allows us to specify the type of the relationships between a document and its parts, if any.

The attributes of the **Document** entity are:

- **id**: The identifier of the document;
- **content:** The actual content of the document;
- **language:** The language of the document;
- **media_type:** The media type of the document;

- **created**: The creation time stamp of the document;
- **last_modified**: The last modification time stamp of the document.

In the evaluation infrastructure we can set proper access permissions on the document, as shown in Figure 28.

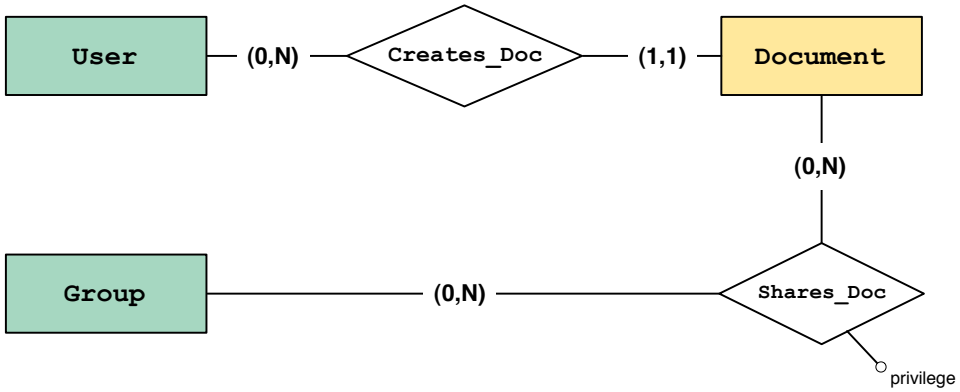


Figure 28 - Relationship between Document, User and Group

We can describe a document by means of metadata, as shown in the figure below.

Figure 29 - Relationship between Metadata and Document

3.4.1.4 Corpus

The **Corpus** is a set of informative resources, which allows us to perform a series of investigations in a research area; thus, a corpus is composed by **Document**(s). The attributes of the corpus are:

- **id**: The identifier of the corpus;
- **description**: A short description of the corpus (e.g. “Wall Street Journal 1996”);
- **language (0,N)**: The language of the corpus, which can have no, one, or more languages;
- **media_type (0,N)**: The media type of the corpus. A corpus can have no, one, or more media types;

- **created**: The creation time stamp of the corpus;
- **last_modified**: The last modification time stamp of the corpus.

The language attribute is useful to select the corpora on the language basis; for instance, we may ask for all the corpora in Italian and English. The media type has the same purpose. This information can be useful for building a new experimental collection from existing corpora.

Every corpus has to be associated to a namespace as we can see in Figure 30.

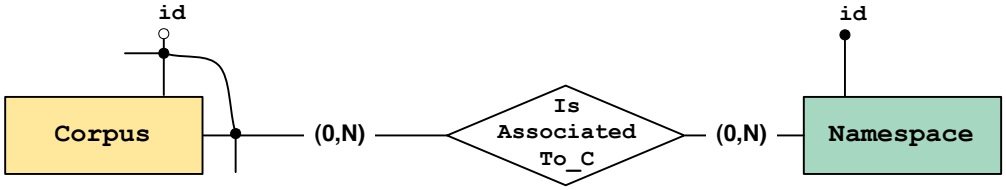


Figure 30 - Relationship between Corpus and Namespace

Furthermore, in the evaluation infrastructure we can set proper access permissions on the experimental collection, as shown in Figure 31.

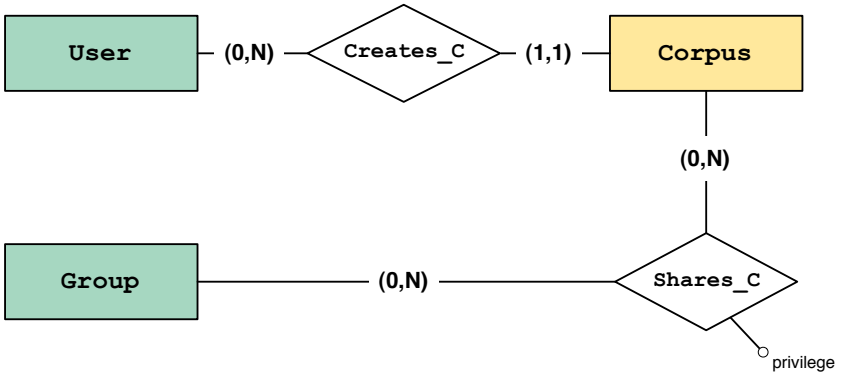


Figure 31 - Relationship between Corpus, User and Group

Lastly, we can describe an experimental collection by means of metadata, as shown in the figure below.

Figure 32 - Relationship between Metadata and Corpus

3.4.1.5 Topic Type

The **Topic Type** entity represents the taxonomy of types in which we can classify the topics handled by the infrastructure. For instance, “faceted” and “ambiguous” are two topic types adopted in the TREC Web Track. This entity has the following attributes:

- **id**: The identifier of the topic type;

- **description**: A short description of the topic type.

3.4.1.6 Topic

A **Topic** entity represents the materialization of an information need. The attributes of this entity are:

- **id**: The identifier of the topic;
- **created**: The creation time stamp of the topic;
- **last_modified**: The last modification time stamp of the topic.

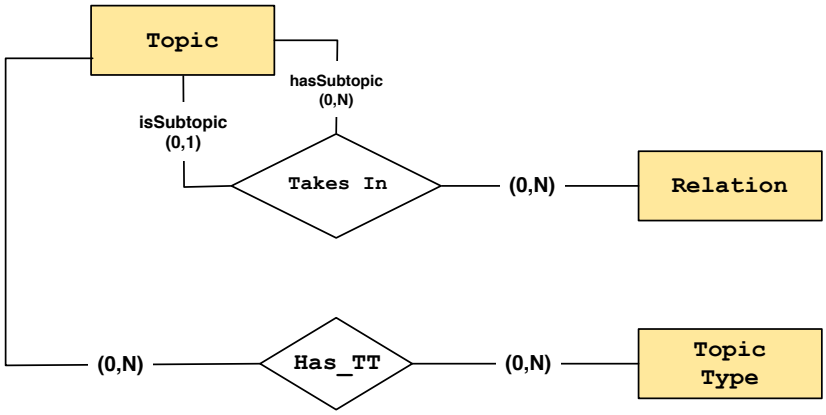


Figure 33 - Relationship between Topic, Topic Type and Relation

Figure 33 shows the recursive relationship **Takes In** on the **Topic** entity; indeed, a topic may be made up of one or more subtopics (in this case we call it a *compound topic*). A relevant example of this scenario is represented by the *Diversity Task* of the *TREC Web Track* [Clark et al. 2009]. In this task each topic is structured as a representative set of subtopics, each related to a different user need; documents are judged with respect to the subtopics, as well as with respect to the topic as a whole. By means of this recursive relationship we can express the fact that a topic can have no, one or more subtopics and that a topic can be a subtopic of at most one other topic. The type of the recursive relationship on **Topic** is modelled by means of the **Relation** entity which is the third entity involved in the ternary relationship **Takes In**.

A topic may have a proper topic type that is expressed by means the relationship **Has_TT** depicted in Figure 33.

We can describe topic by means of metadata, as shown in the figure below.

Figure 34 - Relationship between Metadata and Topic

3.4.1.7 Topic Field

A topic is made up of one or more components, which are traditionally called fields – e.g. "title", "description", and "narrative". The **`Topic Field`** entity has the following attributes:

- **`id`**: The identifier of the topic field; the identifier can be the name of the topic field itself;
- **`description`**: A short description of the topic field.

3.4.1.8 Topic Content

The **`Topic Content`** entity represents the actual content of a topic field and it has the following attributes:

- **`id`**: The identifier of the topic content;
- **`content`**: The actual content of the topic field;
- **`language`**: The language, if any, of the content;
- **`media_type`**: The media type of the topic content.

3.4.1.9 Topic Group

The **`Topic Group`** entity represents a set of topics, which are grouped together because they are used to address a research task carried out in an evaluation activity. The attributes of this entity are:

- **`id`**: The identifier of the topic group;
- **`description:`** A short description of the topic group;
- **`created`**: The creation time stamp of the topic group;
- **`last_modified`**: The last modification time stamp of the topic group.

Every topic group has to be associated to a namespace as we can see in Figure 35.

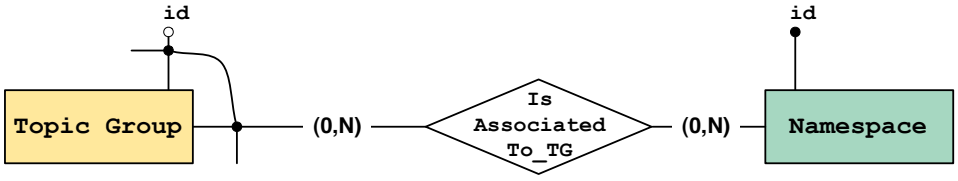


Figure 35 - Relationship between Topic Group and Namespace

Furthermore, in the evaluation infrastructure we can set proper access permissions on the topic group, as shown in Figure 36.

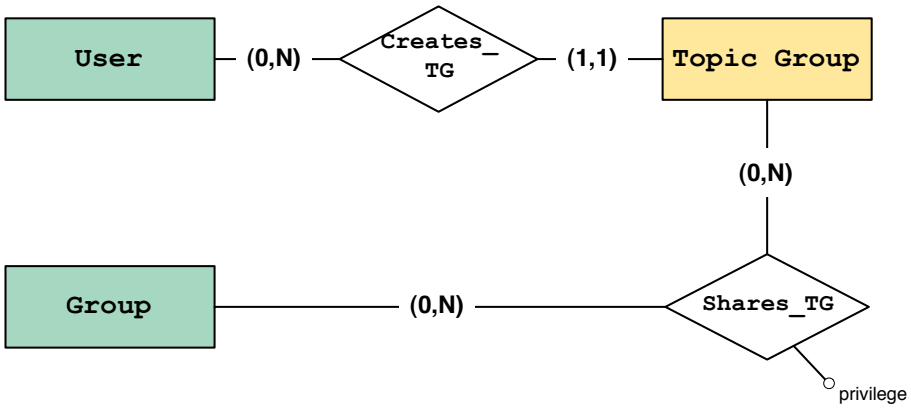


Figure 36 - Relationship between Topic Group, User and Group

Lastly, we can describe a topic group by means of a metadata, as shown in Figure 37.

Figure 37 - Relationship between Metadata and Topic Group

3.4.1.10 Pool

For each topic, a list of relevant documents needs to be compiled which should be as complete as possible. When collections were very small, most of the documents in a collection could be evaluated for relevance, but for very big collection this is clearly impossible [Croft et al. 2009]. For this reason a well-known technique called *pooling* is adopted; this consists in making relevance judgements on "the sample of documents selected by the various participating systems" [Voorhees and Harman 2005]. For instance, one of the most popular techniques used in creating relevance judgments is *depth pooling* adopted by TREC. In the case of depth **k** pooling, only the top **k** documents retrieved by the systems are judged and the rest of the documents in the collection are assumed to be non-relevant [Yilmaz and Aslam (2008)].

The entity **Pool** refers to a container of assessments obtained through the pooling technique.

The following attributes characterize the **Pool** entity:

- **id**: The identifier of the pool;
- **description:** A short description of the pool;
- **assessment_status:** The status of the pool;
- **created**: The creation time stamp of the pool;
- **last_modified**: The last modification time stamp of the pool.

The **assessment_status** attribute refers to the status of the pool under consideration; this attribute can take the following values: { NOT_STARTED_YET, ONGOING, ENDED, AVAILABLE }.

Every pool has to be associated to a namespace as we can see in Figure 38.

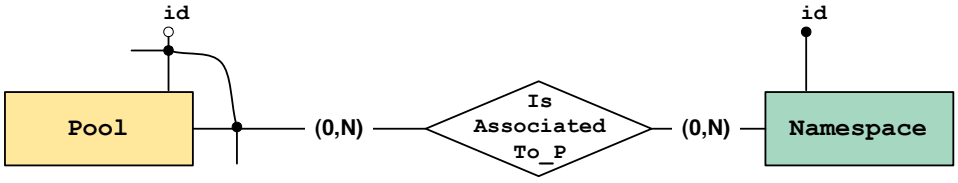


Figure 38 - Relationship between Pool and Namespace

Furthermore, in the evaluation infrastructure we can set proper access permissions on the topic group, as shown in Figure 39.

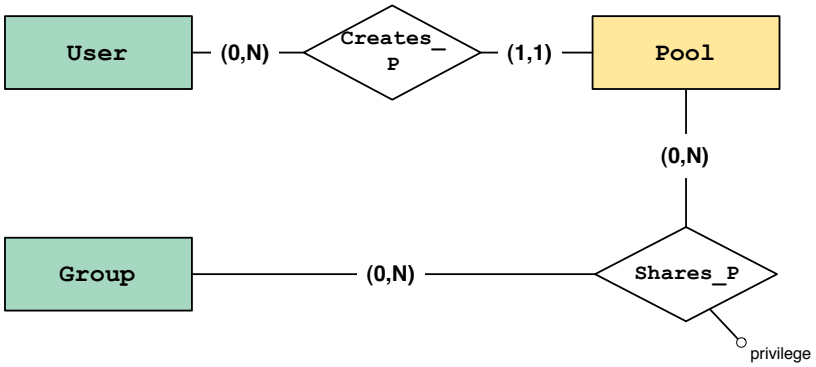


Figure 39 - Relationship between Pool, User and Group

Lastly, we can describe a pool by means of metadata, as shown in Figure 40. The **description** attribute consists of a short description of the pool; additional information on the pool is maintained in the associated metadata. The metadata associated to a **Pool** can contain information on the pooling strategy adopted. A **Pool** can be partly constituted by other pools or part of them; in these cases, the metadata also contain information on these pools or their adopted parts. An example is the pool described in [Järvelin and Kekäläinen 2002] where a set of users were asked to reassess all the documents judged as relevant and a number of documents judged as non-relevant by TREC assessors, for a number of topics among those in the TREC7 and TREC8 ad-hoc track test collections. In this case the new pool is obtained by the reassessed documents and the remaining documents judged as non-relevant in the TREC7 and TREC8 ad-hoc track pools, for the considered topics. In other words, items of a pool that are inherited by other pools are duplicated and considered as new items; but history on their original pool is maintained in the metadata.

Figure 40 - Relationship between Metadata and Pool

3.4.1.11 Relevance

The **Relevance** entity represents the possible "relevance values" which can be assigned to a document in the context of a topic.

The attributes of the **Relevance** entity are:

- **id**: The relevance identifier – e.g. "relevant", "marginally relevant", "not relevant";
- **value:** The numerical value associated to the relevance.

We can describe relevance by means of metadata, as shown in the figure below.

Figure 41 - Relationship between Metadata and Relevance

3.4.1.12 Pool Item

The **Pool Item** entity refers to relevance judgements, which are provided on a **Document** in the **Pool** for a given **Topic**. Figure 42 shows the **Assesses** relationship between: **Pool Item**, **Document**, **Topic**, and **Pool**.

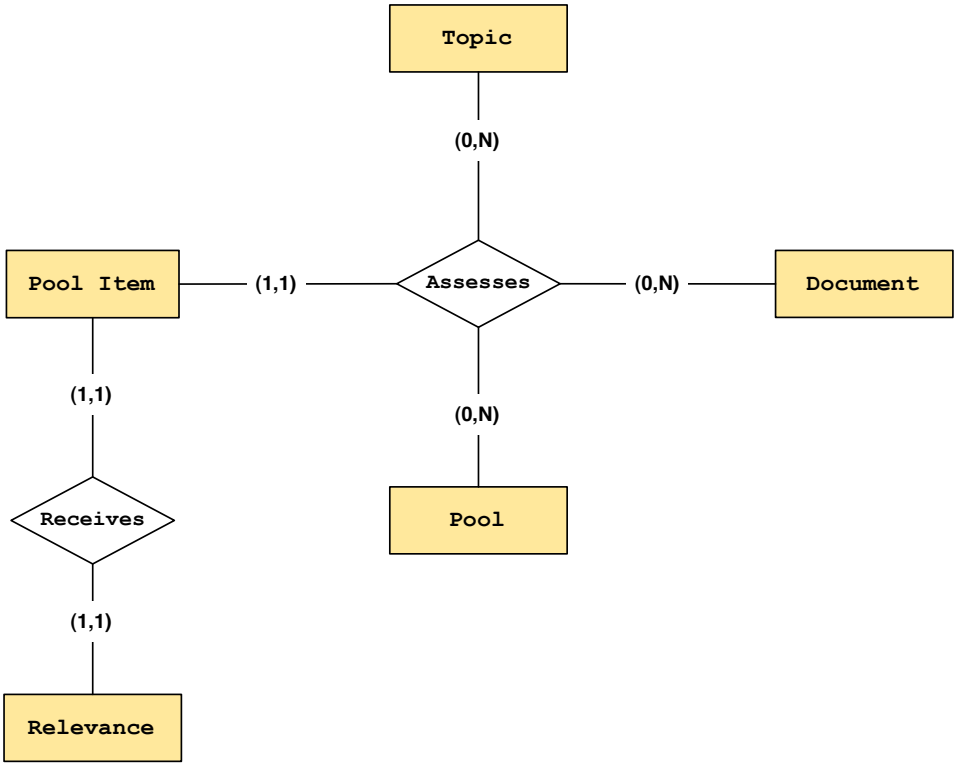


Figure 42 - Assesses relationship between: Pool Item, Document, Topic and Pool

The attributes of the **Pool Item** entity are:

- **id**: The identifier of the pool item;
- **created**: The creation time stamp of the pool item;
- **last_modified**: The last modification time stamp of the pool item.

We can describe a pool item by means of metadata, as shown in the figure below.

Figure 43 - Relationship between Metadata and Pool Item

3.4.2 Experimental Collection Relationships

Figure 44 shows the relationships of the Experimental Collection with the main entities in the Experimental Collection area.

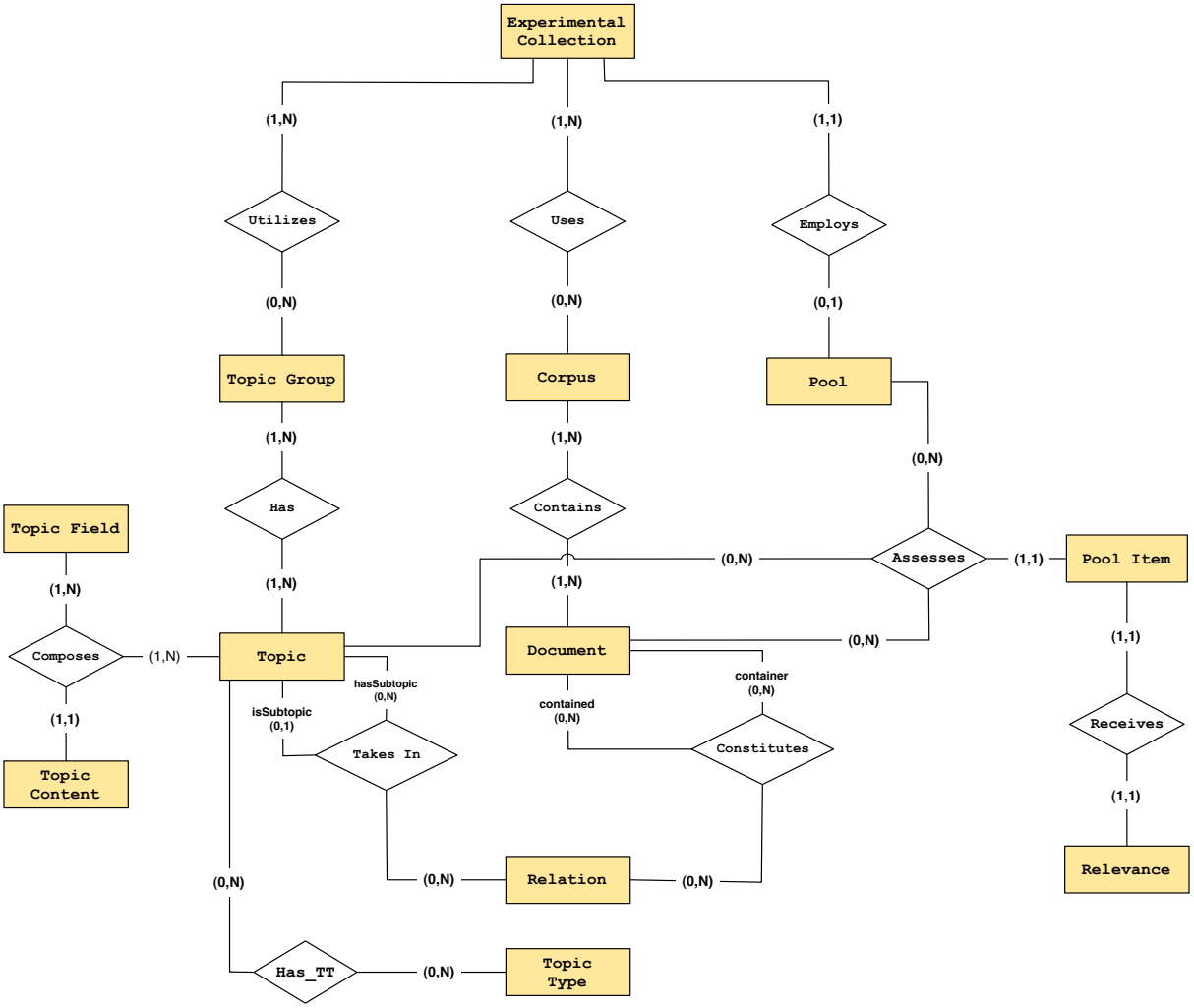


Figure 44 - Relationships of the Experimental Collection with the main entities in the Experimental Collection area

As we can see one or more corpora, one or more topic groups, and one and only one pool must compose an experimental collection. This means that two experimental collections composed by the same corpus and topic group but by different pools are considered distinct collections. In the left-hand side of the figure we can see that a topic can be composed by one or more topic fields and contents; it is important to highlight that a specific topic content can belong to only one topic and topic field couple.

As mentioned above, the recursive relationship on the **`Topic`** entity allows us to express information about the subtopics related to a *compound* topic. This means that it is possible to assess a document with regard to both a topic and a subtopic; e.g. this is necessary to

maintain the pooling data in a consistent way for the diversity task of the TREC Web track. When we deal with compound topics it can happen that an assessment is given on the basis of the subtopics and not on the overall topic, which contains them; this is for instance the case of the diversity task of the TREC Web Track [Clark et al. (2009)]. On the other hand, an assessment can be given only to the overall topic and not to its subtopics like in the Ad Hoc task of the TREC Web Track [Clark et al. (2009)]. In general, we could maintain independent assessments both on the topic as a whole and on its subtopics in the same pool.

3.5 Experiment Area

3.5.1 Experiment Area Entities

3.5.1.1 Experiment

An **`Experiment`** is part of the data produced by a system under evaluation. Figure 45 shows the **`Experiment`** entity which is specialized in three subclasses {**`Run, Guerrilla, Living`**} which refer to the different typologies of experimental activities handled by the infrastructure.

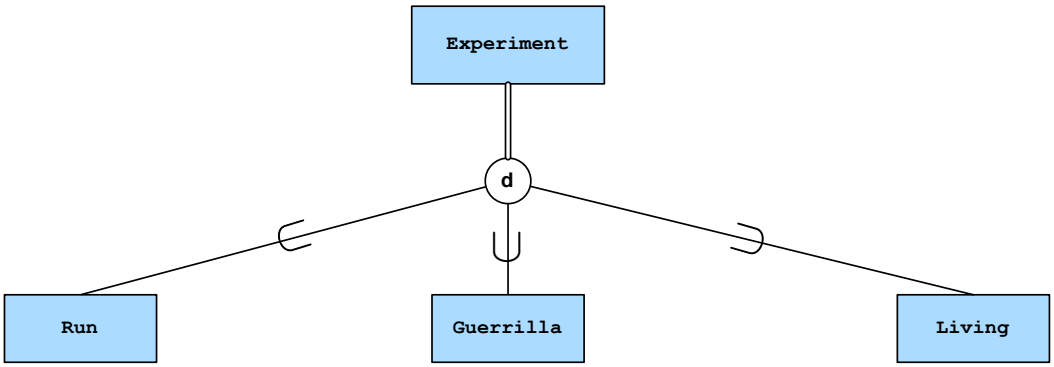


Figure 45 - The Experiment entity and its specializations

A **`Run`** is defined as a ranked list of documents for each topic in the experimental collection. "The documents in a list are sorted such that the document the system believes is most likely to match the topic is retrieved in the first rank" [Voorhees and Harman (2005)].

A **`Guerrilla`** experiment identifies an evaluation activity performed on corporate IR systems (e.g. a custom search engine integrated in a corporate Web site). In this case, the evaluation process is defined by a set of experimental activities aimed at assessing different aspects of the application such as the completeness of the index of an ad-hoc search engine or the effectiveness of the multilingual support. For this reason the evaluation metrics can differ from those used during a **`Run`** experiment, such as precision.

The **`Living`** entity deals with the specific experimental data resulting from the Living Retrieval Laboratories defined in Task 4.4, which will examine the use of operational systems as experimental platform on which to conduct user-based experiments to scale.

The attributes of this entity are:

- **id**: The identifier of the experiment;
- **description**: A short description of the experiment;
- **created**: The creation time stamp of the experiment;
- **last_modified**: The last modification time stamp of the experiment.

The **Run** entity as the following attribute:

- **query_costruction**: The way the query is built from the topic fields (e.g. "automatic", "manual").

Every experiment has to be associated to a namespace as we can see in Figure 46.

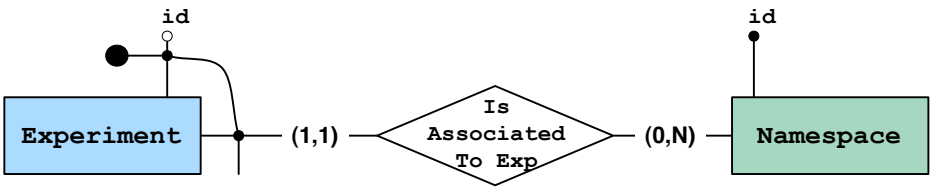


Figure 46 - Relationship between Namespace and Experiment

Furthermore, in the evaluation infrastructure we can set proper access permissions on the experiment, as shown in Figure 47.

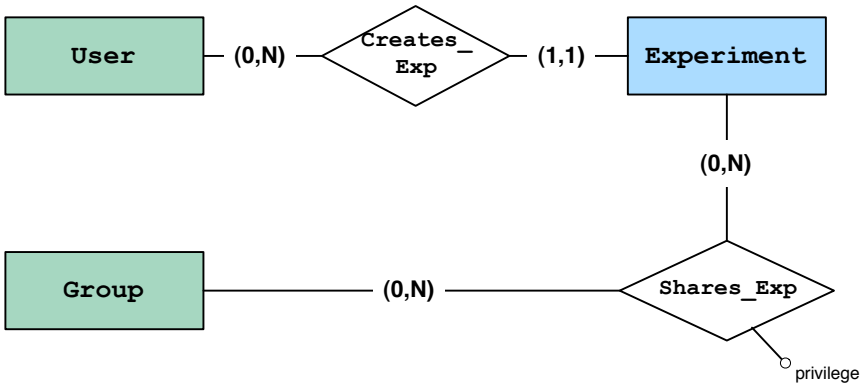


Figure 47 - Relationship between Experiment, User and Group

Lastly, we can describe an experiment by means of metadata, as shown in the figure below.

Figure 48 - Relationship between Metadata and Experiment

3.5.1.2 Run Item

A **Run Item** is a component of a **Run** and relates a **Run** with a **Document** retrieved for a given **Topic**; it represents an entry of the result file usually provided by a participant of a campaign. This entity has the following attributes:

- **id**: The identifier of the run item;
- **rank**: The rank of the run item, i.e. the rank of the document associated to the run item in the ranked list;
- **score**: The score of the run item, i.e. the score provided by the system under evaluation to the document corresponding to the run item.

3.5.1.3 Experiment Type

The **Experiment Type** entity represents a taxonomy through which we can classify an experiment; e.g. “official”, “extra”, “private”, “helper”.
The attributes of this entity are:

- **id**: The identifier of the experiment type;
- **description:** A short description of the experiment type.

3.5.1.4 Component

A **Component** is a building block of a running system. This entity has the following attributes:

- **id**: The identifier of the component;
- **name:** The name or acronym of the component;
- **description:** A short description of the component;
- **created**: The creation time stamp of the component;
- **last_modified**: The last modification time stamp of the component.

Every component has to be associated to a namespace as we can see in Figure 49.

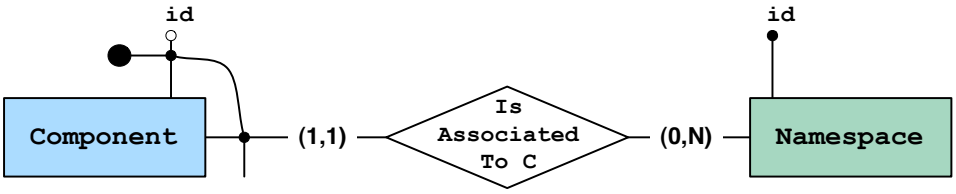


Figure 49 - Relationship between Namespace and Component

Furthermore, in the evaluation infrastructure we can set proper access permissions on the component, as shown in Figure 50.

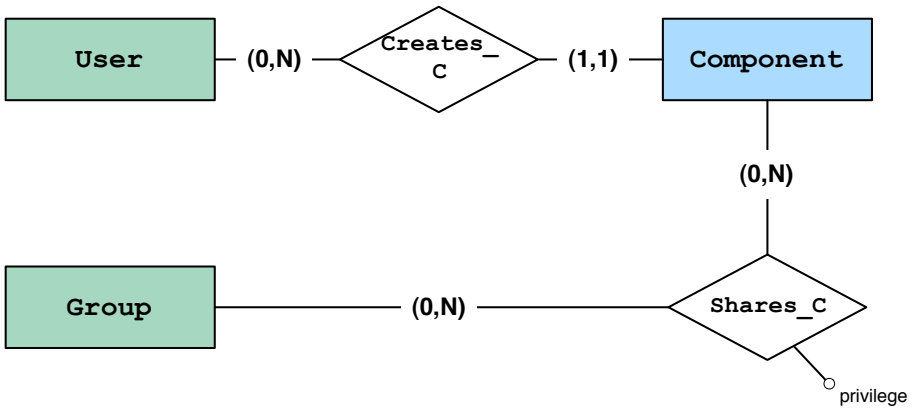


Figure 50 - Relationship between Component, User and Group

Lastly, we can describe a component by means of metadata, as shown in the figure below.

Figure 51 - Relationship between Metadata and Component

3.5.1.5 Component Type

The **`Component Type`** entity classifies the component according to a taxonomy; e.g. the *tokenizer* separates an input document into a stream of tokens, the *stemmer* maps a token to its root or stem and the *scorer* assigns scores to documents with respect to queries.

The attributes of the **`Component Type`** entity are:

- **`id`**: The identifier of the component type;
- **`description:`** A short description of the component type.

We can describe a component type by means of metadata, as shown in the figure below.

Figure 52 - Relationship between Metadata and Component Type

3.5.1.6 System

The **System** entity represents a running software engine, which is under evaluation. It is characterized by the following attributes:

- **id**: The identifier of the system;
- **name:** The name or acronym of the system;
- **description:** A short description of the system;
- **created**: The creation time stamp of the system;
- **last_modified**: The last modification time stamp of the system.

Every system has to be associated to a namespace as we can see in Figure 53.

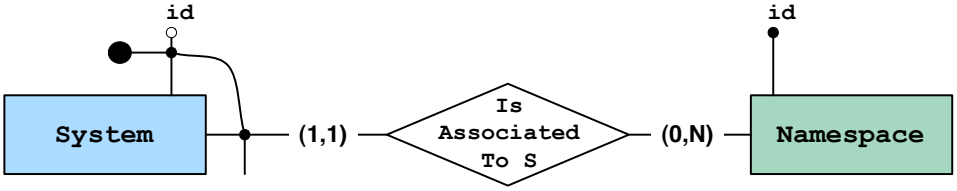


Figure 53 - Relationship between Namespace and System

Furthermore, in the evaluation infrastructure we can set proper access permissions on the system, as shown in Figure 54.

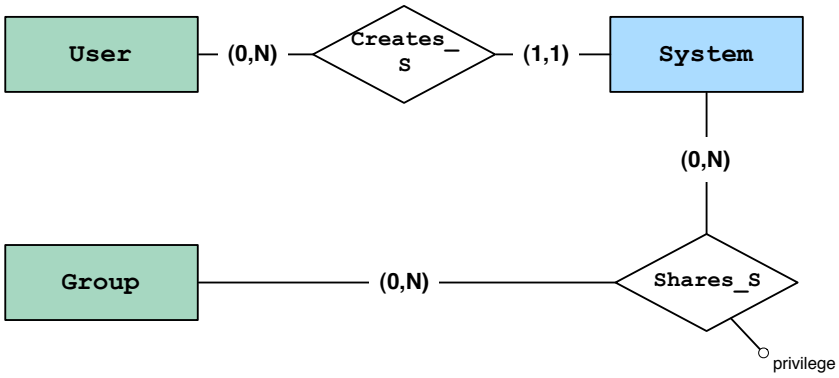


Figure 54 - Relationship between System, User and Group

Lastly, we can describe a system by means of metadata, as shown in the figure below.

Figure 55 - Relationship between Metadata and System

3.5.1.7 Application

The **Application** entity identifies a running software application evaluated during a Guerrilla experiment. . It is characterized by the following attributes:

- **id**: The identifier of the application;
- **name:** The name or acronym of the application;
- **description:** A short description of the application;
- **created**: The creation time stamp of the application;
- **last_modified**: The last modification time stamp of the application.

Every application has to be associated to a namespace as we can see in Figure 56.

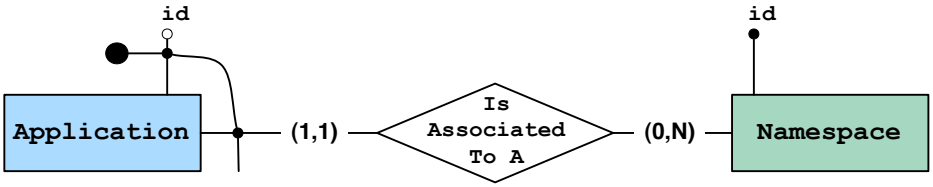


Figure 56 - Relationship between Namespace and Application

Furthermore, in the evaluation infrastructure we can set proper access permissions on the application, as shown in Figure 57.

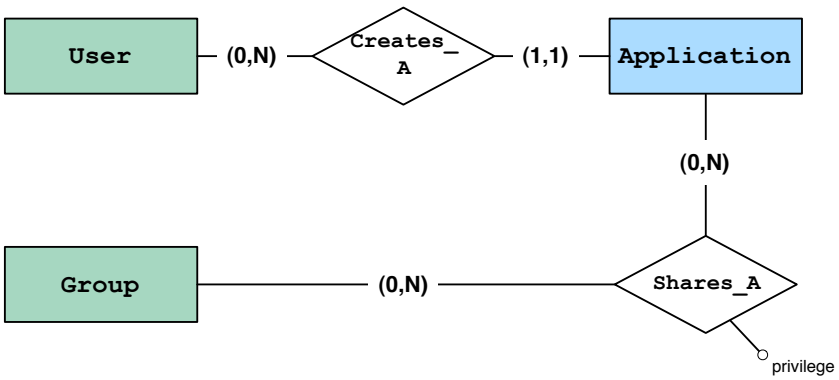


Figure 57 - Relationship between Application, User and Group

Lastly, we can describe a system by means of metadata, as shown in the figure below.

Figure 58 - Relationship between Metadata and Application

3.5.1.8 Configuration

The **Configuration** entity identifies the configuration of a **component**, a **system** or an **application** under evaluation. It is characterized by the following attributes:

- **id**: The identifier of the configuration;
- **description:** A short description of the configuration;
- **parameters:** The blob containing the state of the stored configuration.

In the evaluation infrastructure we can set proper access permissions on the configuration, as shown in the figure below.

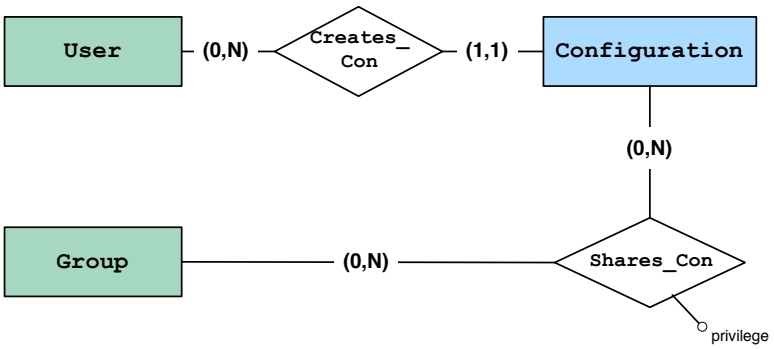


Figure 59 Relationships between Configuration, User and Group.

Lastly, we can describe a configuration by means of metadata, as shown in the figure below.

Figure 60 Relationship between Metadata and Configuration.

3.5.1.9 Guerrilla Item

A **Guerrilla Item** is a component of a **Guerrilla** experiment and relates a **Guerrilla** with a **Configuration** (of a System, an Application or a Component) and a **Metric**. This entity has the following attributes:

- **id**: The identifier of the guerrilla item;
- **weight**: The weight of the guerrilla item;
- **value**: The value of the guerrilla item.

3.5.2 Experiment Area Relationships

Figure 61 shows relationships of the **Experiment** subclasses with three entities of the evaluation infrastructure: **Component**, **System** and **Application**. A **Run** refers to the evaluation activity on one and only one **System**. A **Guerrilla** experiment **tests** one and only one **Application** at a time as well as a **Living** experiment. A **Component** can be **part of** a system and, similarly, it can be **integrated in** an **Application**. A **System** can **compose** an **Application**. As we can see the **Experiment Type** and the **Component Type** taxonomies are associated respectively to **Experiment** and **Component** through the relationships **Is Classified by_E** and **Is Classified by_C**.

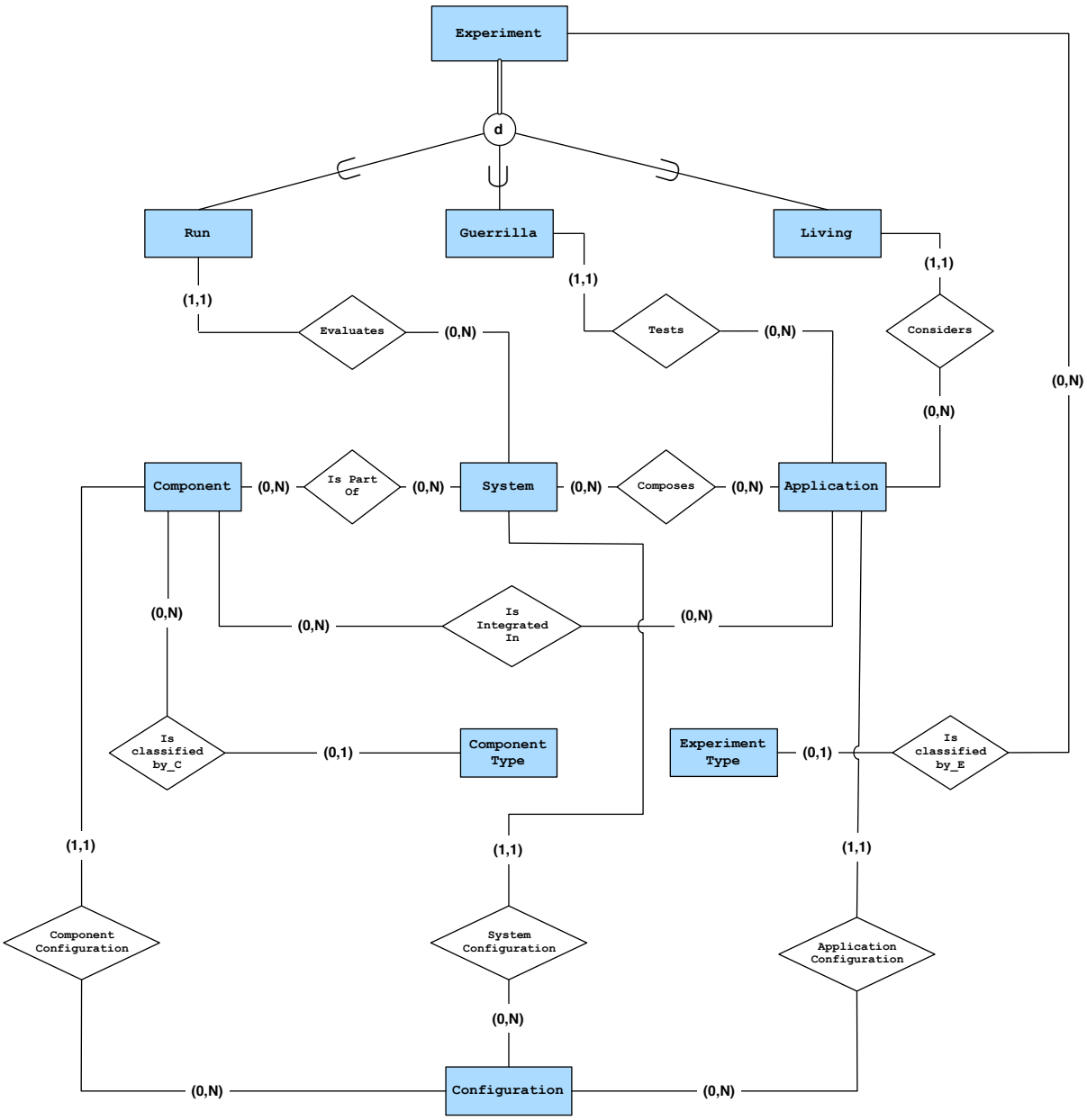


Figure 61 - Relationship of the Experiment subclasses with Component, System and Application

3.6 Measurement Area

3.6.1 Measurement Area Entities

3.6.1.1 Metric

In general a metric is a standard of measurement allowing us to quantify the effectiveness and the efficiency of a system under evaluation and also to optimize systems themselves. For instance, two classical metrics used in Information Retrieval are precision and recall. The **Metric** entity has the following attributes:

- **id**: The identifier of the metric;
- **name:** The name or acronym of the metric;
- **unit:** The unit of measurement of the metric (e.g. milliseconds for the execution time);
- **description:** A short description of the metric.

A **Metric** can be connected to another **Metric** (e.g. in order to create a taxonomies of metrics) by means of the recursive relationships **Is Connected To** and it can be described by means of **Metadata**, as shown in the figure below.

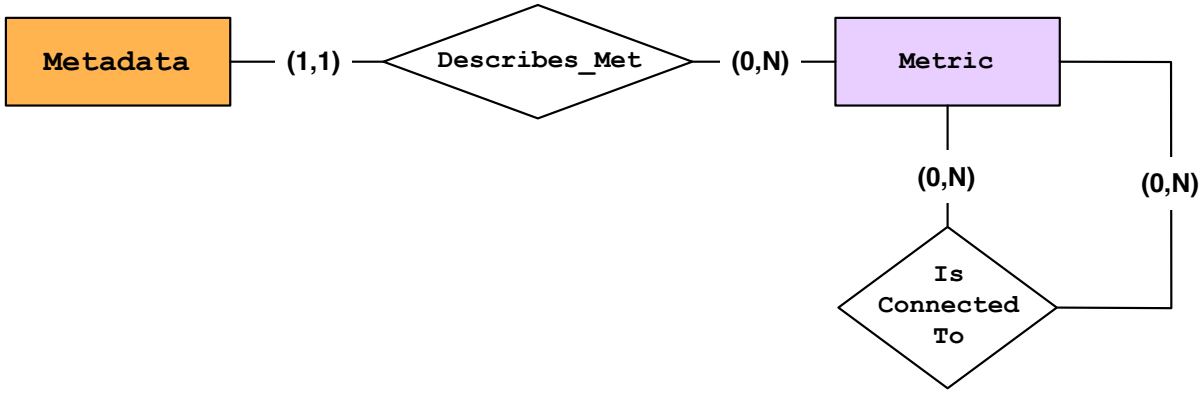


Figure 62 - Relationship between Metric and Metadata

3.6.1.2 Statistical Analysis

This entity represents a list of the statistical analyses supported by the infrastructure. The attributes of the **Statistical Analysis** entity are:

- **id**: The identifier of the statistical analysis;
- **description:** A short description of the statistical analysis.

A statistical analysis can be described by means of metadata, as shown in the figure below.

Figure 63 - Relationship between Statistical Analysis and Metadata

3.6.1.3 Descriptive Statistic

Descriptive statistics are used to describe the basic features of the data in a study; e.g. “quantile”, “frequency distributions”, “variability”, and “normal distribution” are descriptive statistics. A **`Descriptive Statistic`** entity is characterized by the following attributes:

- **`id`**: The identifier of the descriptive statistic;
- **`name:`** The name or acronym of the descriptive statistic;
- **`description:`** A short description of the descriptive statistic.

3.6.1.4 Statistical Test

A statistical test provides a mechanism for making quantitative decisions about a process or processes. A **`Statistical Test`** in the evaluation infrastructure represents an example of statistical analysis which can be carried out on the available data. A **`Statistical Test`** is characterized by the following attributes:

- **`id`**: The identifier of the statistical test;
- **`scope:`** The scope of the statistical test;
- **`parameters:`** The parameters for reconstructing the statistical test;
- **`created`**: The creation time stamp of the statistical test;
- **`last_modified`**: The last modification time stamp of the statistical test.

Every statistical test has to be associated to a namespace as we can see in Figure 64.

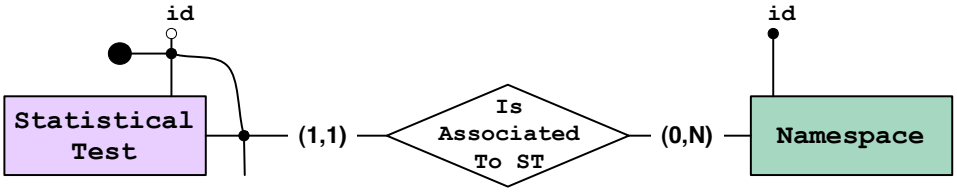


Figure 64 - Relationship between Namespace and Statistical Test

Furthermore, in the evaluation infrastructure we can set proper access permissions on the Statistical Test, as shown in Figure 65.

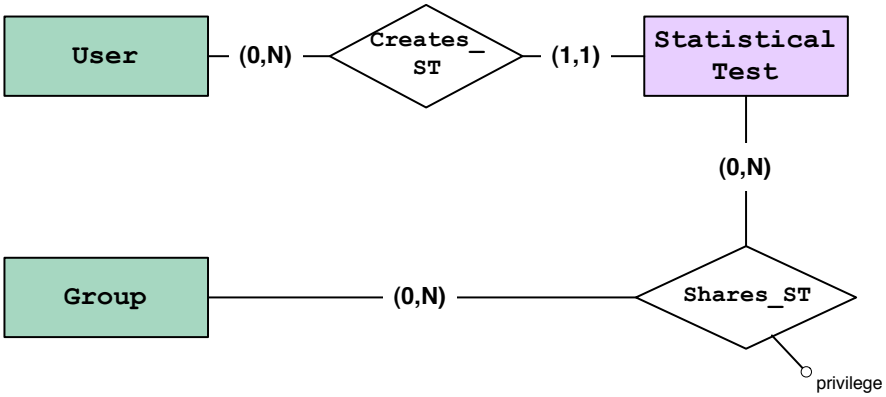


Figure 65 - Relationship between Statistical Test, User and Group

Lastly, we can describe a statistical test by means of metadata, as shown in Figure 66. The metadata contain information on the statistical test, e.g. the value adopted for the test parameters, if any.

Figure 66 - Relationship between the Statistical Test and Metadata entity

3.6.1.5 Measure

The **Measure** entity represents the value of a **Metric** calculated on some **Experiments** handled by the infrastructure. The **Measure** entity has the following attributes:

- **id**: The identifier of the measure;
- **value**: The numerical value of the measure;
- **created**: The creation time stamp of the measure.

We can describe a measure by means of metadata, as shown in the figure below.

Figure 67 - Relationship between the Measure and Metadata entity

3.6.1.6 Estimate

The **Estimate** entity represents the estimated numerical value of a **Descriptive Statistic** calculated by the infrastructure. The **Estimate** entity has the following attributes:

- **id**: The identifier of the estimated value;
- **value**: The estimated numerical value;
- **created**: The creation time stamp of the estimate.

We can describe an estimated value by means of metadata, as shown in the figure below.

Figure 68 - Relationship between the Estimate and the Metadata entity

3.6.2 Measurement Area Relationships

A statistical test can be performed on the results obtained for a **`Metric`**; in Figure 69 we can see the relationships between the **`Statistical Test`** entity and the **`Metric`** entity. Furthermore, we can see that a **`Statistical Test`** is related to a **`Statistical Analysis`** by the **`Produces`** relationship; this means that the statistical test is the specific type of analysis carried out on the data under examination. More details about the measurement area will be given in the Inter-area relationship section.

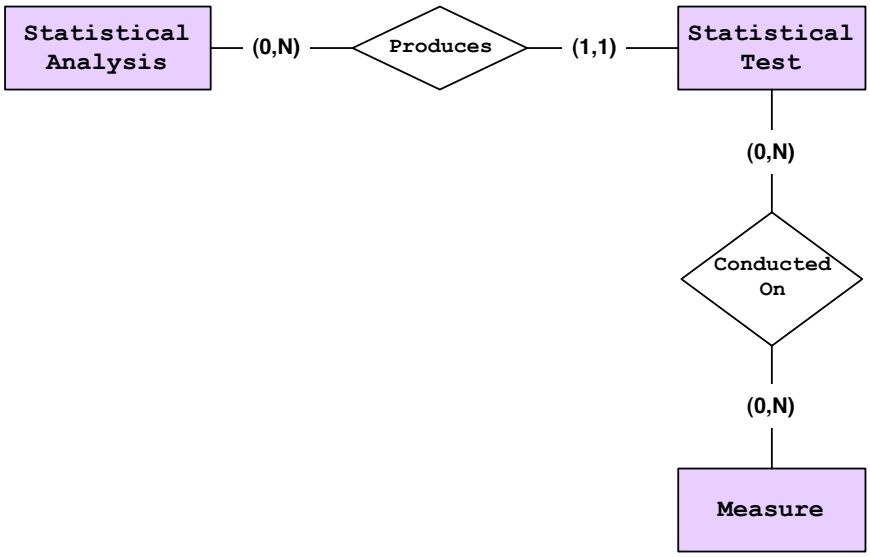


Figure 69 - Relationship between entities in the Measurement Area

3.7 Visual Analytics Area

3.7.1 Visual Analytics Area Entities

3.7.1.1 Visualization

The **`Visualization`** entity refers to the information used by the infrastructure to store and recover whichever visualization of the data that the users do. The following attributes characterize this entity:

- **`id`**: The identifier of the visualization;

- **`parameters:`** The blob containing the state of the stored visualization (e.g. the URL to load the data and the type of visualization such as bar chart or scattered plot);
- **`created`**: The creation time stamp of the visualization;
- **`last_modified`**: The last modification time stamp of the visualization.

Every visualization has to be associated to a namespace as we can see in Figure 70.

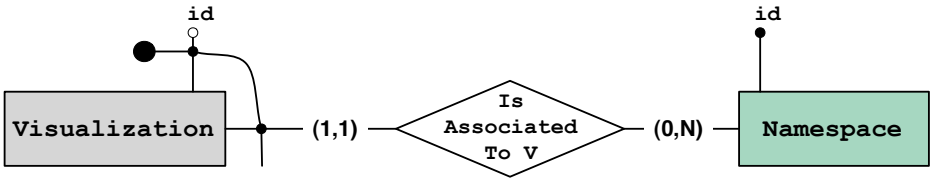


Figure 70 - Relationship between Visualization and Namespace Entity

Furthermore, in the evaluation infrastructure we can set proper access permissions on the visualization, as shown in Figure 71.

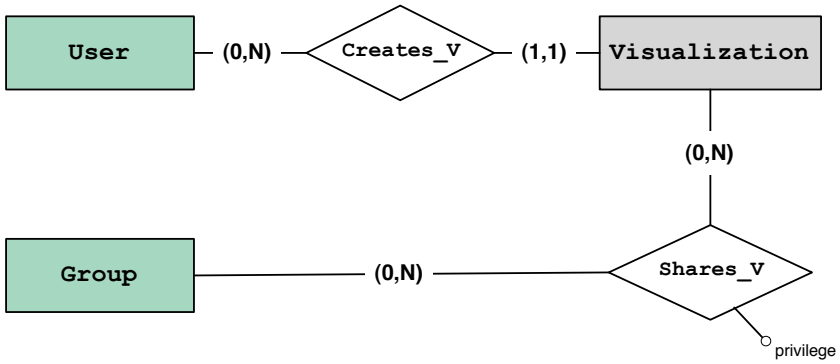


Figure 71 - Relationships between User, Visualization and Group entities

Lastly, we can describe a visualization by means of metadata, as shown in the figure below.

Figure 72 - Relationship between the Visualization and Metadata entity

3.7.1.2 Visualization Type

The **`Visualization Type`** entity represents a controlled vocabulary which identifies the types of visualizations supported by the infrastructure (e.g. scattered plot, histogram, bar chart). The **`Visualization Type`** entity has the following attributes:

- **`id`**: The identifier of the visualization type;
- **`description:`** A short description of the visualization type.

3.7.1.3 Snapshot

The **`Snapshot`** entity stores the snapshots of a visualization and it is characterized by the following attributes:

- **`id`**: The identifier of the snapshot;
- **`media_type:`** The media type of the snapshot (e.g. “pdf”, “jpeg”);
- **`content:`** The actual snapshot of the visualization;
- **`created`**: The creation time stamp of the snapshot;
- **`last_modified`**: The last modification time stamp of the snapshot.

Every snapshot has to be associated to a namespace as we can see in Figure 73.

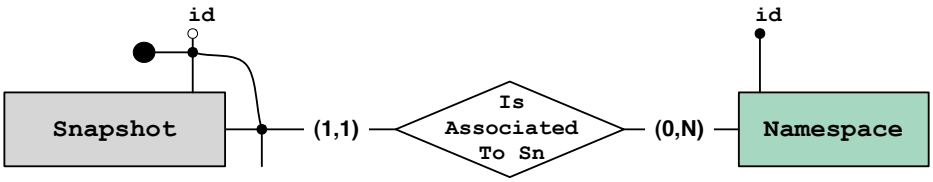


Figure 73 - Relationship between the Snapshot and the Namespace entity

Furthermore, in the evaluation infrastructure we can set proper access permissions on the snapshot, as shown in Figure 74.

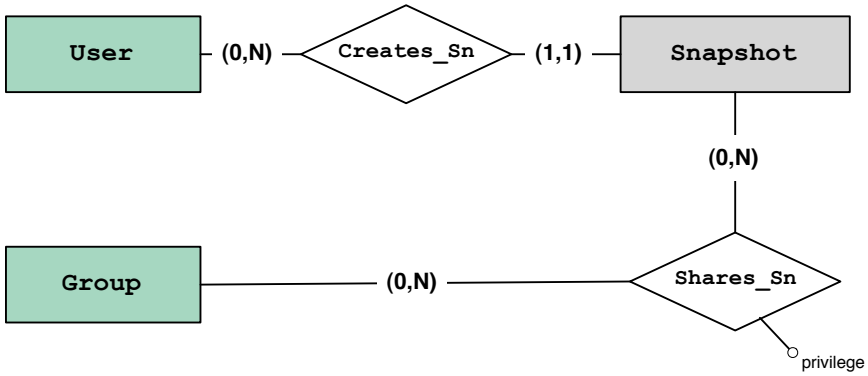


Figure 74 - Relationships between the Snapshot, the User and the Group entity

Lastly, we can describe a snapshot by means of metadata, as shown in the figure below.

Figure 75 - Relationship between Snapshot and Metadata entity

3.7.2 Visual Analytics Area Relationships

Figure 76 depicts entities in the visual analytics area; as we can see a visualization can have no, one or more snapshots, but a snapshot has to be related to one and only one visualization. Visualization has to be connected to one and only one visualization type. It is worthwhile to notice that a visualization type can be recursively associated to one or more subtypes; this is useful to customize different types of visualizations made from some other types (e.g. the recall-precision graph is a subtype of scatterplot).

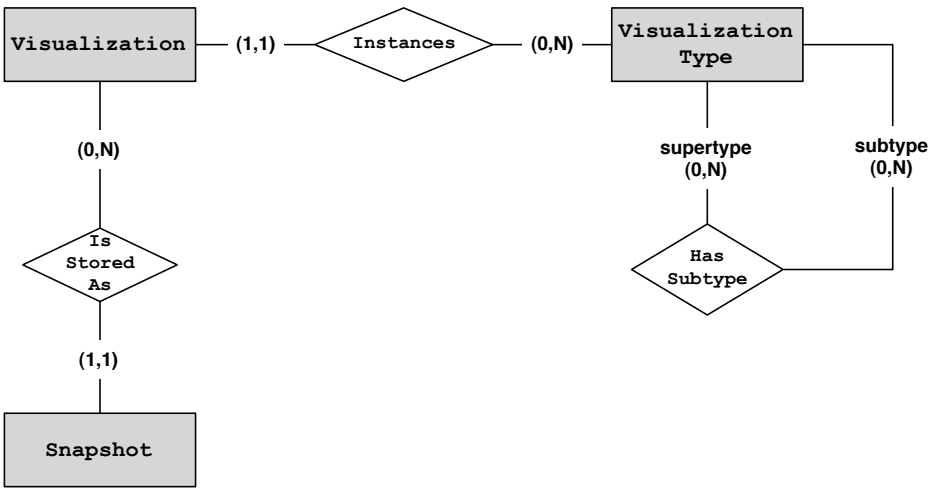


Figure 76 - Relationship between Visualization Type, Visualization and Snapshot entity

3.8 Bibliographical Area

One of the objectives of the PROMISE infrastructure is to make explicit and retain the relationship between the data that result from the evaluation activities and the scientific production based on these data. The bibliographical area concerns the entities that are involved in the scientific production and the relationship among these entities.

3.8.1 Bibliographical Area Entities

3.8.1.1 Contribution

The **Contribution** entity refers to a **piece of writing submitted for** a publication. A conference or a workshop paper, a journal article, a book, a technical report, a thesis or a manual are examples of contributions. The **Contribution** entity is characterized by the following attributes:

- **id**: The identifier of the contribution;
- **title**: The title of the contribution;
- **year**: The year when the contribution was published or made available;
- **month**: The month when the contribution was published or made available;
- **created**: The creation time stamp of the contribution.

Figure 79 shows two recursive relationships in which the **Contribution** entity is involved: **Cited By** and **Is In**.

The recursive relationship **Cited By** is used to retain information of the citations to a contribution; this information is useful for measuring the impact of the knowledge produced by the evaluation activities supported and retained in the infrastructure. A contribution has no, one or more citations and a contribution can cite none, one, or more other contributions.

The recursive relationship **`Is In`** allows us to maintain information on part of contributions, e.g. chapter in books.

A contribution can be described by means of metadata, as shown in the figure below.

Figure 77 - Relationship between Contribution and Metadata entity

3.8.1.2 Venue

This entity refers to possible venues where a contribution can be published or made available. Examples of venues are conferences or workshop proceedings, journals or paper collections. A venue can be described by means of metadata, as shown in the figure below.

Figure 78 - Relationship between Venue and Metadata

3.8.2 Bibliographical Area Relationships

Figure 79 depicts the relationships that involve the bibliographical area entities. Each contribution is published in one and only one venue; when an already published contribution is published in a new venue, it is considered as a distinct contribution (e.g. a paper or a journal article published in a new collection). A venue can have none, one or more contributions; therefore we can maintain information on venue independently from its contributions. Figure 78 also depicts the relationship between the **`Contribution`** and the **`User`** entity: a user can author none, one, or more contributions, but a contribution has to be authored by at least one author.

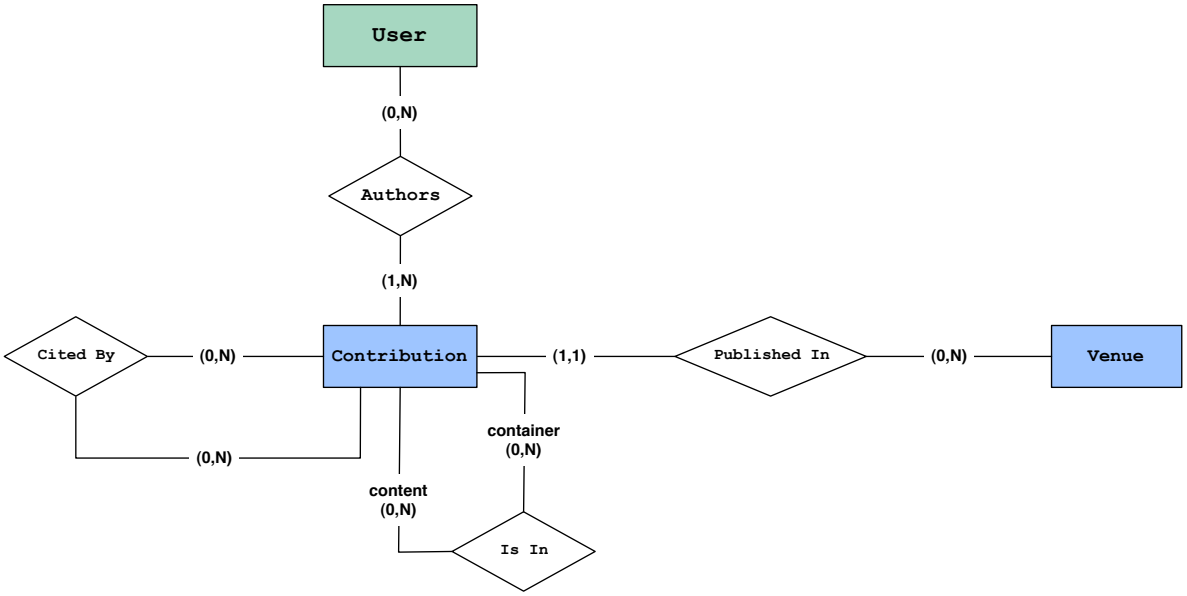


Figure 79 - Relationship between entities in the Bibliographical area and User

3.9 Inter-area Relationships

Figure 80 depicts the relationships between entities in the Experimental Collection area, the Evaluation Activity area, and the Resource Management area.

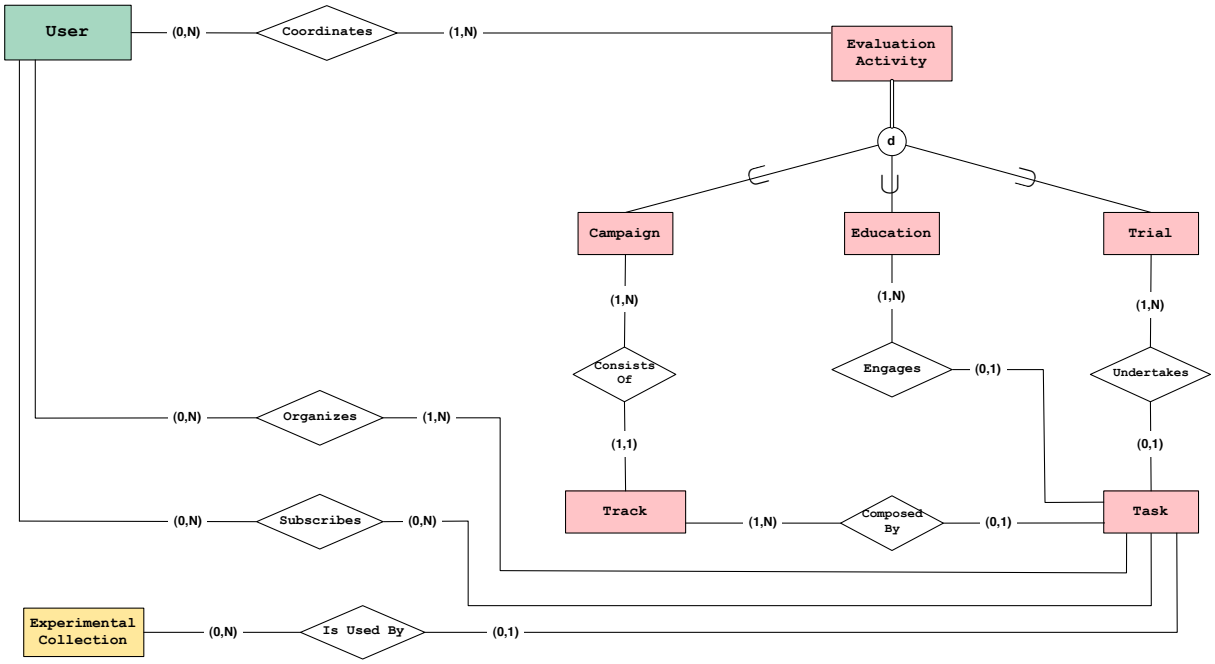


Figure 80 - Relationships between entities in the Resource Management, Evaluation Activity and Experimental Collection area

According to the **`Is Used By`** relationship between **`Task`** and **`Experimental Collection`** a task may or may not use an experimental collection; this allows us to consider tasks where the activity is not based on an experimental test collection (e.g. a task of a *trial* evaluation activity that is connected to a *guerrilla* experiment type). Moreover, a task performed within an evaluation activity can exploit more than one experimental collection; for instance, this is the case of a *trial* evaluation activity where the same weighting scheme or the same methodology is tested across different experimental collections, e.g. TREC 7 and TREC 8 Ad-hoc Test Collection, and TREC2001 Web Track Ad-hoc Test Collection. For tasks that involve a training phase and a test phase – e.g. the CLEF-IP Patent Classification task – the two phases are considered as distinct tasks.

An evaluation activity must have at least a coordinator and a task must have at least an organizer: this is modelled respectively by the relationships **`Coordinates`** and **`Organizes`**. User subscriptions are retained at a task level: a user can **`subscribe`** no, one, or most tasks.

Figure 81 depicts relationships that involve entities in the Evaluation Activity, the Experimental Collection, the Experiment and the Resource Management area. A user can submit no, one or more experiments for a given task; each experiment refers to one and only one user-task pair. A task can use no, one or more topic fields, where some of the adopted fields can be mandatory: this is modelled by the attribute **`mandatory`** of relationship **`Uses Topic Field`** that involves the **`Task`** and the **`Topic Field`** entity. Of the three subclasses of the **`Experiment`** entity, Figure 81 is mainly focused on the **`Run`** subclass. A run **`Comprises`** at least one **`Run Item`**, where each **`Run Item`** refers to a specific experiment-topic-document triple; a document as well as a topic can be related to no, one or more run items through the **`Comprises`** relationship. Some of the runs retained in the infrastructure are adopted to constitute the pool: a run is pooled in no, one or more pools, while a pool is constituted by run items in at least one run. We can also see that a **`Guerrilla is made of`** one or more **`Guerrilla Item`**, **`Configuration`** and **`Metric`**.

Lastly, the **`Is Assessor`** relationship states that a user can be an assessor for no, one or more pools, and that a pool must have at least one assessor.

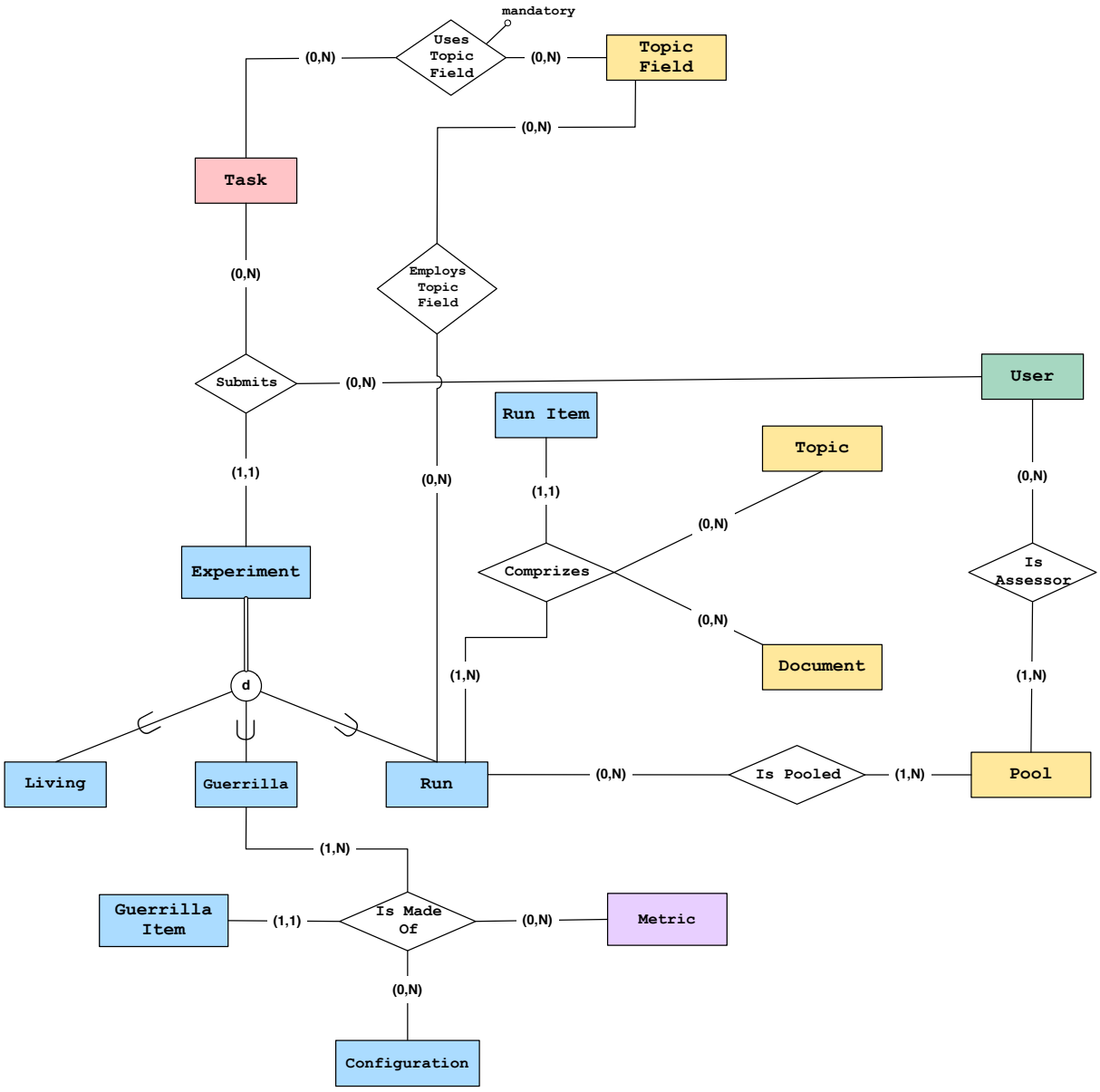


Figure 81 - Relationship between entities in the Evaluation Activity, Experimental Collection, Experiment and Resource Management area

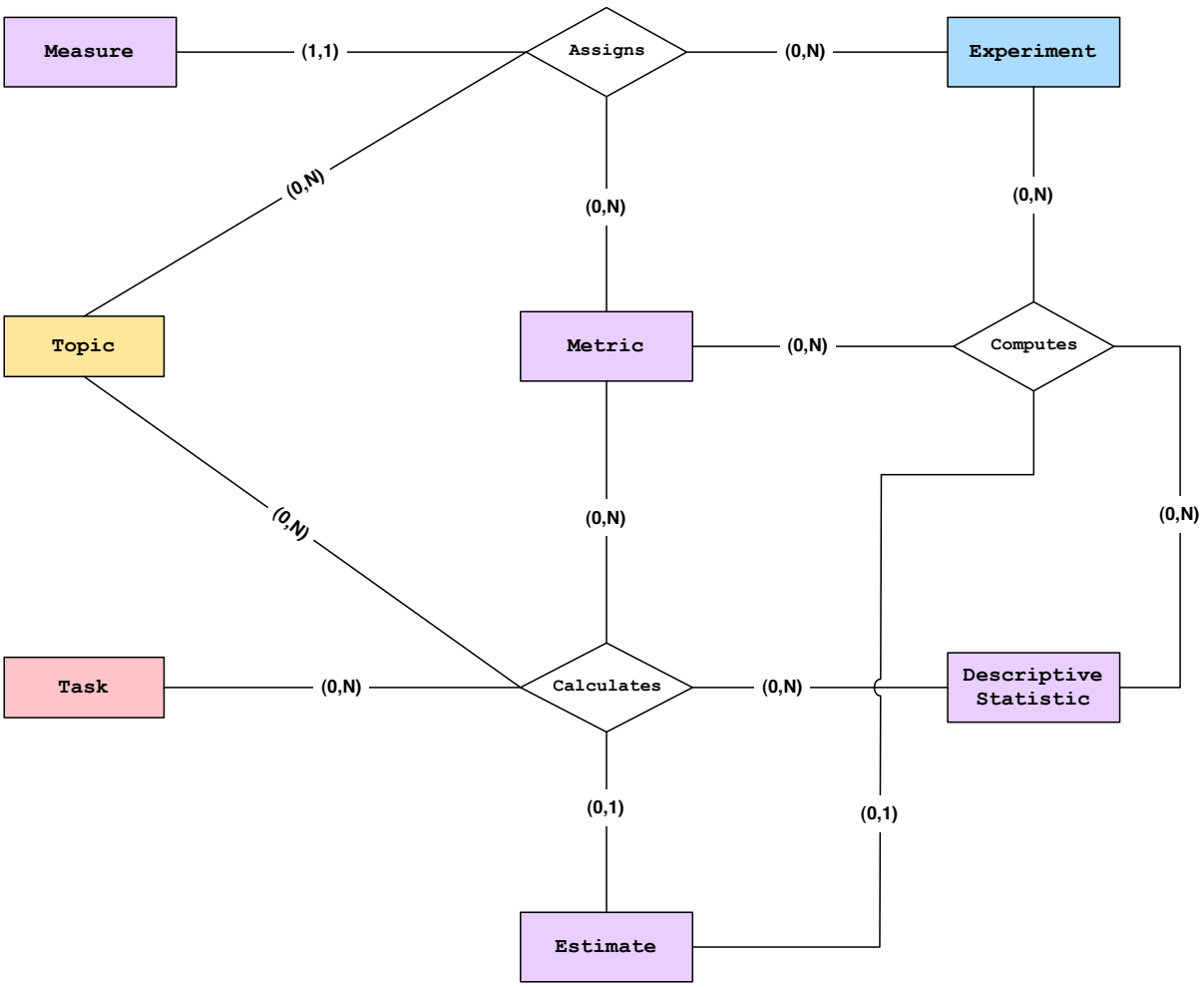


Figure 82 – Relationship between entities in the Evaluation Activity, Experimental Collection, Experiment and Measurement area

Figure 82 concerns entities in the measurement area and its relationship with entities of other areas, i.e. Evaluation Activity area, Experimental Collection area and Experiment area. For a topic-experiment pair a specific value of a metric, namely a measure, is assigned – i.e. a **`Measure`** refers to one and only one **`Experiment-Topic-Metric`** triple through the relationship **`Assigns`**; an example is the value computed for the metric average precision on the data of an experiment for a specific topic. When considering the results on a per-experiment basis **`Descriptive Statistic`**s can be computed for a given **`Metric`**, e.g. the mean *Average Precision* over all the topics adopted for the **`Experiment`** under consideration; this is modelled through the **`Computes`** relationship in Figure 82. **`Descriptive Statistic`**s can be computed also on a per task basis, e.g. the *variance* for a given **`Topic`** over all the **`Experiment`**s submitted for a specific **`Task`**; this is modelled by the relationship **`Calculates`** that involves the **`Task`**, the **`Metric`**, the **`Descriptive Statistic`** and the **`Estimate`** entities.

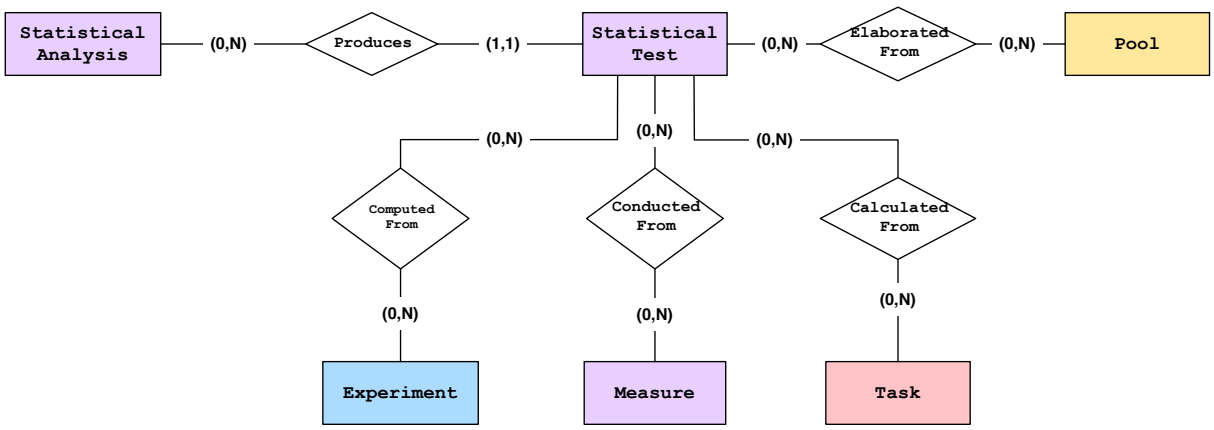


Figure 83 - Relationship between entities of the Evaluation Activity, the Experimental Collection and the Measurement Area

As depicted in Figure 83 a **Statistical Analysis** can produce a value for a specific statistical test; the **Statistical Test** value can be **Elaborated From** data in no, one or more **Pool**s, or **Calculated From** data from no, one or more **Task**s, or **Computed From** an **Experiment**. Lastly, a Statistical Test value can be obtained by the test **Conducted on** no, one or more **Measure**s.

Figure 84 depicts the relationship between the Visualization entity and entities in the Evaluation Activity, the Experimental Collection, the Experiment and the Measurement area. Every visualization can be related to no, one or more **Task**s – see relationship **ViTa**, to no, one or more **Pool**s – see relationship **ViPo**, to no, one or more **Experiment**s – see relationship **ViEx**, to no, one or more **Statistical Test**s – and see relationship **ViSt**.

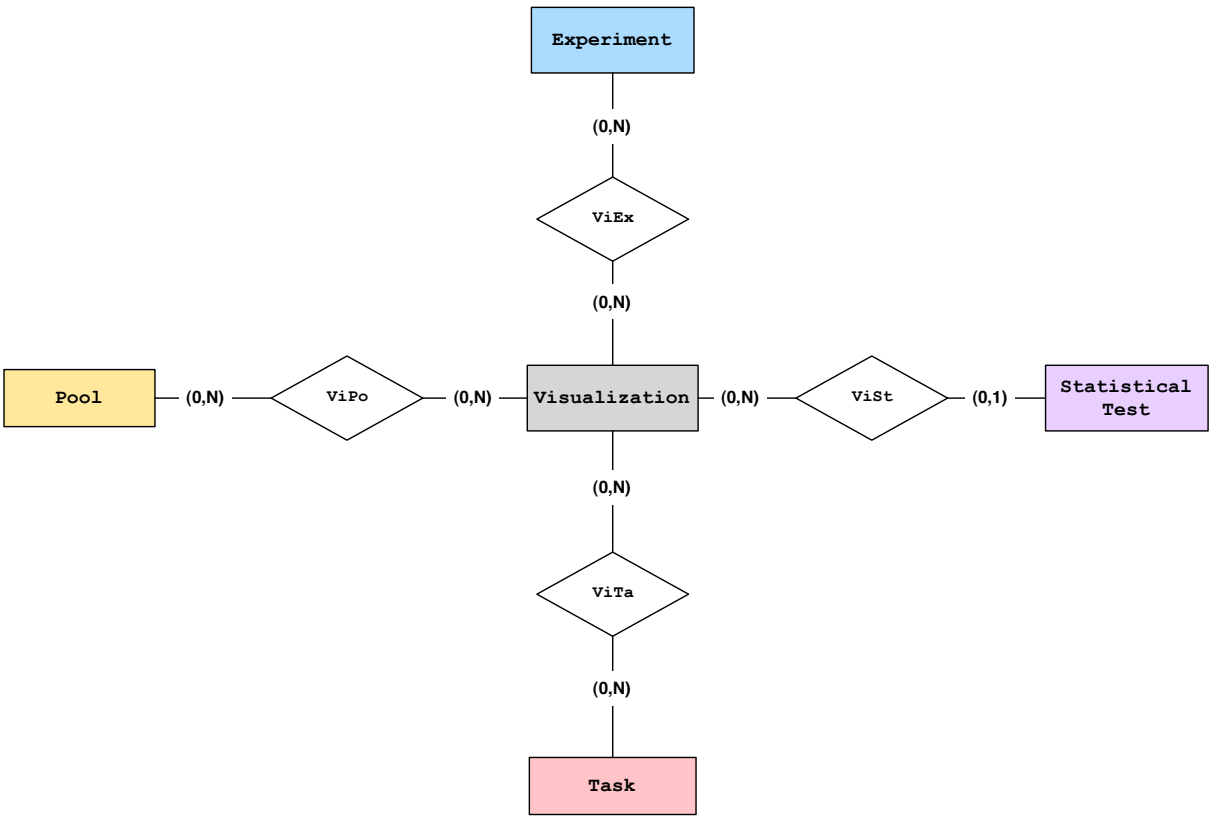


Figure 84 – Relationship between the Visualization entity and entities in the Evaluation Activity, the Experimental Collection, the Experiment and the Measurement area.

Figure 85 depicts the relationship between the **`Contribution`** entity and the entities in the Evaluation Activity, the Experimental Collection, and the Experiment area. The basic rationale behind the introduction of these relationships is that a contribution can refer to data stored in the infrastructure: besides experimental collections and its constituting components – i.e. corpus, pool and topic group – a contribution can refer to no, one or more experiments, evaluation activities, tracks and tasks. That allows us to measure the impact of the PROMISE project both in terms of citations to papers on PROMISE related evaluation activities and citations on data that has resulted from such activities, e.g. experiments and experimental collections. Moreover, that can help identify previous works that exploit the same experimental collection or their constituting component, or concern similar tasks – i.e. experimental hypotheses to be tested.

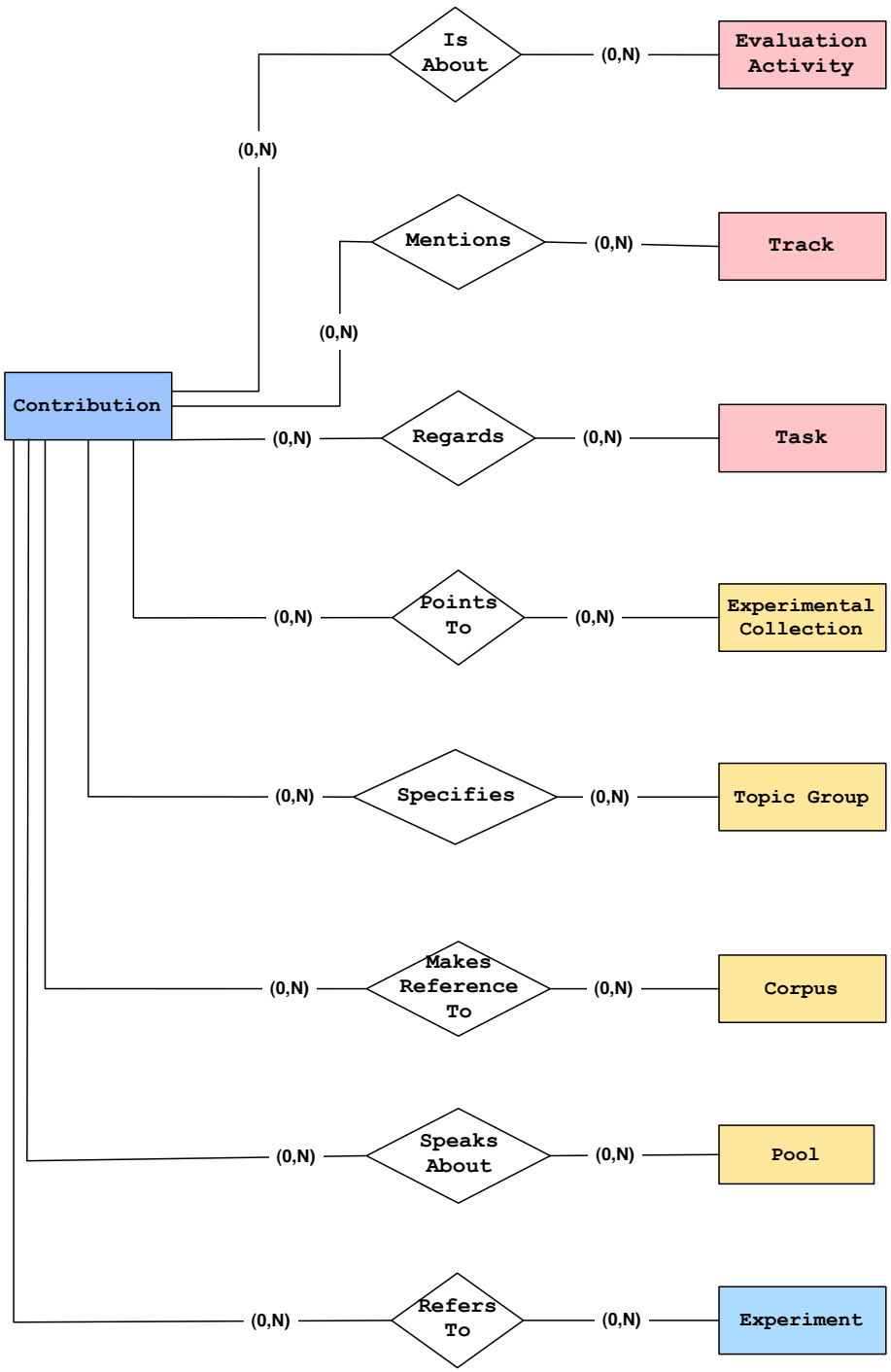


Figure 85 - Relationship between the Contribution entity of the bibliographical area and the entities in the Evaluation Activity, the Experimental Collection, and the Experiment area

4 Architecture

This section describes the enhancements on the architecture of the *Distributed Information Retrieval Evaluation Campaign Tool* (DIRECT) evaluation system presented on [PROMISE D3.1, 2011] to make it accessible as a Representational State Transfer (REST) Web Service.

The architecture and the implementation of the system have been developed by exploiting open source technologies, software and frameworks, in order to guarantee a platform which is cooperative, modular, scalable, sustainable over time and allowing interoperability among different systems.

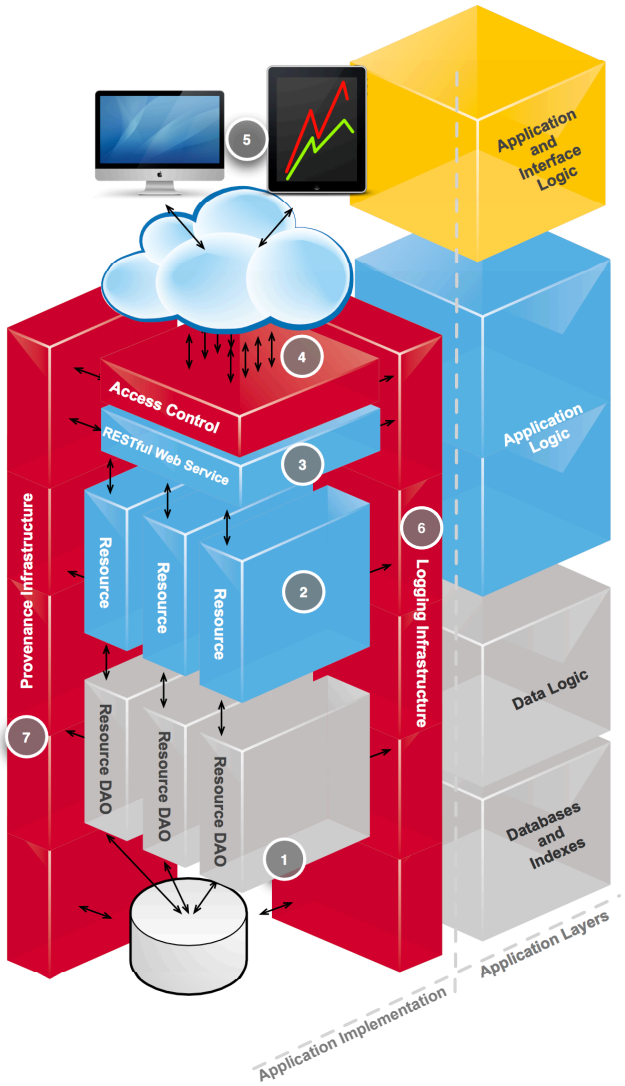


Figure 86 - The Architecture of the DIRECT System as a REST Web Service

Figure 86 shows the architecture of the DIRECT system. The right stack summarizes the layers modelling the application, as presented in [Deliverable 3.1], while the left stack shows the building blocks of the implementation of the system.

4.1 Database and Indexes Infrastructure

At the lowest levels of the stack – see point (1) on Figure 5 – data stored into database and indexes are mapped to resources and vice versa. The communication with the upper levels is granted through the mechanism of the *Data Access Object* (DAO)[6] pattern.

4.2 Access to Resources Infrastructure

The application logic layer is in charge of the high-level tasks made by the system, such as the enrichment of raw data, the calculation of metrics and the carrying out of statistical analyses on experiments. These resources, shown at point (2), are therefore accessible by remote devices via HTTP through a RESTful Web service, described in details in Section 5 and represented by points (3) and (5).

The *Access Control Infrastructure*, point (4), takes care of monitoring the various resources and functionalities offered by the system. It performs *authentication*, asking for user credentials to log it into the system, and *authorization*, verifying if the logged in user requesting an operation holds sufficient rights to perform it.

The access control policies can be dynamically configured and changed over time by defining *roles*, i.e. groups of users, entitled to perform given operations. Examples of such roles could be ADMINISTRATORS, PARTICIPANTS, ASSESSORS, VISITORS, SCIENTIFIC_BOARD_MEMBERS.

4.3 Logging Infrastructure

The logging infrastructure, which lays behind all the components of the DIRECT system, captures information such as the user name, the Internet Protocol (IP) address of the connecting host, the action invoked by the user, the messages exchanged among the components of the system, and any error condition, if necessary.

It also captures the HyperText Transfer Protocol [Fielding et al., 1999] logs and represents them according to the W3C Extended Log File Format[7].

Log files can be accessed and searched interactively to mine and fully exploit them.

4.4 Provenance Infrastructure

The *Provenance Infrastructure* – point (7) in Figure 85 – is in charge of keeping track of the full lineage of each resource managed by the system since its first creation, allowing granted users to reconstruct its full history and modifications over time.

Provenance events are presented as statements about a resource of the form

<when><who><predicate><what><why>

[6] http://java.sun.com/blueprints/corej2eepatterns/

[7] http://www.w3.org/TR/WD-logfile.html

where:

- **when** is the timestamp at which the event occurred;
- **who** is the user who caused the event;
- **predicate** is the action carried out in the event, i.e. CREATED, READ, or DELETED;
- **what** is the resource originated by the event;
- **why** is the motivation that originated the event, i.e. the operation performed by the system that led to a modification of the resource.

5 DIRECT RESTful Web Service

As discussed in the previous sections, the DIRECT evaluation system is accessible to remote client applications by means of a RESTful Web Service [Fielding and Taylor 2002; Richardson and Ruby, 2007].

The DIRECT RESTful Web Service offers several APIs built around the following main resources:

- **namespace** manages the operations related to namespaces and their provenance;
- **group** manages the operations related to groups of users and their provenance;
- **user** manages the operations related to users and their provenance;
- **metadata** manages the operations related to metadata and their provenance;
- **metadata set** manages the operations related to metadata sets and their provenance;
- **log event** manages the operations related to log events;
- **search** manages the search of resources, to provide a ranked list of results as response, specified by identifier, namespace, score and rank;
- **list** manages the search and listing of resources, to provide a list of complete objects as response;
- **evaluation activity** manages the operations related to evaluation activities and their provenance;
- **track** manages the operations related to tracks and their provenance;
- **task** manages the operations related to tasks and their provenance;
- **topic** manages the operations related to topics and their provenance;
- **experiment** manages the operations related to experiments and their provenance;
- **pool** manages the operations related to pools and their provenance.

The API for accessing the various resources are described in detail in the following. Each section presents: the Uniform Resource Identifier (URI) to be used to refer to the desired resource; the method to be used to access the resource (GET, POST, PUT, DELETE, HEAD); the request parameters; the response HTTP status code [Fielding et al., 1999] and body for the different possible cases.

As discussed in Section 4.2 about the *Access Control Infrastructure*, some resources are publicly available, some others require authentication before being accessed. The DIRECT RESTful Web Service makes use of the basic HTTP authentication scheme [Fielding et al. 1999; Franks et al. 1999].

When the system needs to access a resource that requires authentication, the user will receive, as response, an authentication challenge with HTTP status code `401 – Unauthorized` asking for user name and password.

Since DIRECT uniquely identifies users by means of their unique identifier and namespace, such information must be provided in the user name field of the HTTP Basic Authentication Scheme. To separate between the unique user identifier and the namespace, the `;`

(semicolon) symbol should be used. Therefore, the user name must be provided with the following syntax:

```
user-identifier;namespace
```

Moreover, since the namespace is usually identified by means of a URI which may contain characters that needs to be escaped, the proper URI encoding has to be performed according to [Berners-Lee et al., 2005]. For example, for the user **myuser** in the namespace **http://name.space.com/**, the following user name field for the HTTP Basic Authentication Scheme should be used:

```
myuser;http%3A%2F%2Fname%2Espace%2Ecom%2F
```

Lastly, note that all the URIs presented in the following sections are relative to a base URI which depends on the installation of the DIRECT digital library system. Therefore, these URIs needs to be appended to the base URI.

For example, if the DIRECT system is available at the following URL **http://ims-ws.dei.unipd.it:8080/direct**, the following complete URL **http://ims-ws.dei.unipd.it:8080/direct/namespace** needs to be used to access the namespace resource, and so on.

All the resources support two input and output formats: eXtensible Markup Language (XML) [W3C, 2008] and JavaScript Object Notation (JSON)[8] [Crockford, 2006]. This can be set by using the standard HTTP headers: **Content-Type** for specifying the input format and **Accept** for the desired output format followed by either **application/xml** or **application/json** MIME media types [Freed and Borenstein, 1996].

5.1 Error Messages

The following table summarizes the error conditions reported by the system. These error conditions are common across all the resources managed by the system.

For each error condition, the table contains:

- the HTTP status code
- the Error Code
- a short description

For each error condition, the response body contains detailed diagnostic messages further explaining it.

[8] http://www.json.org/

HTTP Status Code	Error Code	Description
400 – Bad Request	C2002 – INVALID_PARAMETER	An invalid parameter (null, empty, missing, ...) has been provided
400 – Bad Request	C2003 – MALFORMED_REPRESENTATION	A malformed representation of a resource (not well-formed, not valid, ...) has been provided
401 – Unauthorized	C3000 – AUTHENTICATION_REQUIRED	An attempt to access a resource without the required authentication has been performed
403 – Forbidden	C3001 – INSUFFICIENT_ACCESS_RIGHTS	An attempt to access a resource with insufficient access rights has been performed
404 – Not Found	C4003 – NOT_FOUND_RESOURCE	An attempt to refer to an inexistent resource has been performed
405 – Method Not Allowed	C1001 – UNSUPPORTED_OPERATION	An unsupported operation has been requested
406 – Not Acceptable	C2000 – UNSUPPORTED_OUTPUT_FORMAT	An unsupported output format has been requested
409 – Conflict	C4002 – DUPLICATED_RESOURCE	An attempt to create an already existing resource has been performed
409 – Conflict	C4004 – NOT_MODIFIABLE_RESOURCE	An attempt to update or delete a resource that cannot be modified has been performed
409 – Conflict	C4005 – CONCURRENT_RESOURCE_MODIFICATION	An attempt to update a resource that has been concurrently updated has been performed
415 – Unsupported Media Type	C2001 – UNSUPPORTED_INPUT_FORMAT	An unsupported input format has been provided
500 – Internal Server Error	C1000 – INTERNAL_ERROR	An error internal to the system has occurred

5.2 Namespace Resource

Action	HTTP Method	URL
CREATE_NAMESPACE	POST	`/namespace`
READ_NAMESPACE	GET	`/namespace/{id}`
UPDATE_NAMESPACE	PUT	`/namespace/{id}`
DELETE_NAMESPACE	DELETE	`/namespace/{id}`
LIST_NAMESPACES	GET	`/namespace`
LIST_NAMESPACE_PROVENANCE_EVENTS	GET	`/namespace/{id}/provenance`

5.3 User Resource

Action	HTTP Method	URL
CREATE_USER	POST	/user
READ_USER	GET	/user/{id};{ns}
UPDATE_USER	PUT	/user/{id};{ns}
DELETE_USER	DELETE	/user/{id};{ns}
CHANGE_USER_PASSWORD	PUT	/user/{id};{ns}/changePassword
AUTHENTICATE_USER	GET, PUT, POST, DELETE, OPTIONS, HEAD	/user/authenticate
ADD_USER_TO_GROUP	GET, PUT, POST	/user/{id};{ns}/member/group/{resource-id};{resource-ns}
REMOVE_USER_FROM_GROUP	DELETE	/user/{id};{ns}/member/group/{resource-id};{resource-ns}
ADD_USER_TO_ROLE	GET, PUT, POST	/user/{id};{ns}/member/role/{resource-id};{resource-ns}
REMOVE_USER_FROM_ROLE	DELETE	/user/{id};{ns}/member/role/{resource-id};{resource-ns}
ADD_USER_AS_PARTICIPANT_TO_TASK	GET, PUT, POST	/user/{id};{ns}/participant/task/{resource-id};{resource-ns}
ADD_USER_AS_ORGANIZER_TO_TASK	GET, PUT, POST	/user/{id};{ns}/organizer/task/{resource-id};{resource-ns}
ADD_USER_AS_COORDINATOR_TO_CAMPAIGN	GET, PUT, POST	/user/{id};{ns}/coordinator/campaign/{resource-id};{resource-ns}
ADD_USER_AS_COORDINATOR_TO_EDUCATION	GET, PUT, POST	/user/{id};{ns}/coordinator/education/{resource-id};{resource-ns}
ADD_USER_AS_COORDINATOR_TO_TRIAL	GET, PUT, POST	/user/{id};{ns}/coordinator/trial/{resource-id};{resource-ns}
ADD_USER_AS_ASSESSOR_TO_POOL	GET, PUT, POST	/user/{id};{ns}/assessor/pool/{resource-id};{resource-ns}
LIST_USERS	GET	/user
LIST_USER _PROVENANCE_EVENTS	GET	/user/{id};{ns}/provenance

5.4 Group Resource

Action	HTTP Method	URL
CREATE_GROUP	POST	/group
READ_GROUP	GET	/group/{id};{ns}

Action	HTTP Method	URL
UPDATE_GROUP	PUT	`/group/{id};{ns}`
DELETE_GROUP	DELETE	`/group/{id};{ns}`
LIST_GROUPS	GET	`/group`
LIST_GROUP_PROVENANCE_EVENTS	GET	`/group/{id};{ns}/provenance`

5.5 Role Resource

Action	HTTP Method	URL
CREATE_ROLE	POST	`/role`
READ_ROLE	GET	`/role/{id};{ns}`
UPDATE_ROLE	PUT	`/role/{id};{ns}`
DELETE_ROLE	DELETE	`/role/{id};{ns}`
LIST_ROLES	GET	`/role`
LIST_ROLE_PROVENANCE_EVENTS	GET	`/role/{id};{ns}/provenance`

5.6 Metadata Resource

Action	HTTP Method	URL
CREATE_METADATA	POST	`/metadata`
READ_METADATA	GET	`/metadata/{id};{ns}`
UPDATE_METADATA	PUT	`/metadata/{id};{ns}`
DELETE_METADATA	DELETE	`/metadata/{id};{ns}`
ADD_METADATA_TO_METADATA_SET	GET, PUT, POST	`/metadata/{id};{ns}/member/metadataset/{resource-id};{resource-ns}`
REMOVE_METADATA_FROM_METADATA_SET	DELETE	`/metadata/{id};{ns}/member/metadataset/{resource-id};{resource-ns}`
LIST_METADATA_BELONGING_TO_METADATA_SET	GET	`/metadata/member/metadataset/{resource-id};{resource-ns}`
LIST_METADATA	GET	`/metadata`
LIST_METADATA _PROVENANCE_EVENTS	GET	`/metadata/{id};{ns}/provenance`

5.7 Metadata Set Resource

Action	HTTP Method	URL
CREATE_METADATA_SET	POST	/metadataset
READ_METADATA_SET	GET	/metadataset/{id};{ns}
UPDATE_METADATA_SET	PUT	/metadataset/{id};{ns}
DELETE_METADATA_SET	DELETE	/metadataset/{id};{ns}
INCLUDE_SUBSET_INTO_SUPERSET	GET, PUT, POST	/metadataset/{id};{ns}/member/{resource-id};{resource-ns}
EXCLUDE_SUBSET_FROM_SUPERSET	DELETE	/metadataset/{id};{ns}/member/{resource-id};{resource-ns}
LIST_METADATA_SETS	GET	/metadataset
LIST_METADATA_SET_PROVENANCE_EVENTS	GET	/metadataset/{id};{ns}/provenance

5.8 Log Event Resource

Action	HTTP Method	URL
READ_LOG_EVENT	GET	/logevent/{id}
LIST_LOG_EVENTS	GET	/logevent

5.9 Search Resource

Action	HTTP Method	URL
SEARCH_RESOURCES	POST	/search
SEARCH_RESOURCES	GET	/search?query={query}
SEARCH_RESOURCES	GET	/search/{query}

5.10 List Resource

Action	HTTP Method	URL
LIST_RESOURCES	POST	/list
LIST_RESOURCES	GET	/list?query={query}
LIST_RESOURCES	GET	/list/{query}

5.11 Campaign Resource

Action	HTTP Method	URL
CREATE_CAMPAIGN	POST	/campaign
READ_CAMPAIGN	GET	/campaign/{id};{ns}
UPDATE_CAMPAIGN	PUT	/campaign/{id};{ns}
DELETE_CAMPAIGN	DELETE	/campaign/{id};{ns}
ADD_CAMPAIGN_TO_SERIES	GET, PUT, POST	/campaign/{id};{ns}/series/{name}
REMOVE_CAMPAIGN_FROM_SERIES	DELETE	/campaign/{id};{ns}/series/{name}
LIST_CAMPAIGNS	GET	/campaign
LIST_CAMPAIGN_PROVENANCE_EVENTS	GET	/campaign/{id};{ns}/provenance

5.12 Trial Resource

Action	HTTP Method	URL
CREATE_TRIAL	POST	/trial
READ_TRIAL	GET	/trial/{id};{ns}
UPDATE_TRIAL	PUT	/trial/{id};{ns}
DELETE_TRIAL	DELETE	/trial/{id};{ns}
LIST_TRIALS	GET	/trial
LIST_TRIAL_PROVENANCE_EVENTS	GET	/trial/{id};{ns}/provenance

5.13 Education Resource

Action	HTTP Method	URL
CREATE_EDUCATION	POST	/education
READ_EDUCATION	GET	/education/{id};{ns}
UPDATE_EDUCATION	PUT	/education/{id};{ns}
DELETE_EDUCATION	DELETE	/education/{id};{ns}
LIST_EDUCATION	GET	/education
LIST_EDUCATION_PROVENANCE_EVENTS	GET	/education/{id};{ns}/provenance

5.14 Track Resource

Action	HTTP Method	URL
CREATE_TRACK	POST	/track
READ_TRACK	GET	/track/{id};{ns}

Action	HTTP Method	URL
UPDATE_TRACK	PUT	/track/{id};{ns}
DELETE_TRACK	DELETE	/track/{id};{ns}
LIST_PARTICIPANTS_TO_TRACK	GET	/track/{id};{ns}/participant
ADD_TRACK_TO_SERIES	GET, PUT, POST	/track/{id};{ns}/series/{name}
REMOVE_TRACK_FROM_SERIES	DELETE	/track/{id};{ns}/series/{name}
LIST_TRACKS	GET	/track
LIST_TRACK_PROVENANCE_EVENTS	GET	/track/{id};{ns}/provenance

5.15 Series Resource

Action	HTTP Method	URL
CREATE_SERIES	POST	/series
READ_SERIES	GET	/series/{name}
UPDATE_SERIES	PUT	/series/{name}
DELETE_SERIES	DELETE	/series/{name}
LIST_SERIES	GET	/series
LIST_SERIES	GET	/series/{name}/provenance

5.16 Task Resource

Action	HTTP Method	URL
CREATE_TASK	POST	/task
READ_TASK	GET	/task/{id};{ns}
UPDATE_TASK	PUT	/task/{id};{ns}
DELETE_TASK	DELETE	/task/{id};{ns}
LIST_TOPICS_OF_TASK	GET	/task/{id};{ns}/topic
READ_TOPIC_OF_TASK	GET	/task/{id};{ns}/topic/{resource-id};{resource-ns}
LIST_METRICS	GET	/task/{id};{ns}/topic/{resource-id};{resource-ns}/metric
LIST_METRICS	GET	/task/{id};{ns}/topic/{resource-id};{resource-ns}/metric/{name}/experiment
LIST_METRICS	GET	/task/{id};{ns}/topic/{resource-id};{resource-ns}/experiment/{other-resource-id};{other-resource-ns}/metric
LIST_METRICS	GET	/task/{id};{ns}/topic/{resource-id};{resource-ns}/experiment/{other-resource-

Action	HTTP Method	URL
		id};{other-resource-ns}/metric/{name}
LIST_EXPERIMENTS_OF_TASK	GET	/task/{id};{ns}/experiment
READ_EXPERIMENT_OF_TASK	GET	/task/{id};{ns}/experiment/{resource-id};{resource-ns}
LIST_METRICS	GET	/task/{id};{ns}/experiment/{resource-id};{resource-ns}/metric
LIST_METRICS	GET	/task/{id};{ns}/experiment/{resource-id};{resource-ns}/metric/{name}/topic
LIST_METRICS	GET	/task/{id};{ns}/experiment/{resource-id};{resource-ns}/topic/{other-resource-id};{other-resource-ns}/metric
LIST_METRICS	GET	/task/{id};{ns}/experiment/{resource-id};{resource-ns}/topic/{other-resource-id};{other-resource-ns}/metric/{name}
LIST_METRICS	GET	/task/{id};{ns}/experiment/{resource-id};{resource-ns}/statistic
LIST_METRICS	GET	/task/{id};{ns}/experiment/{resource-id};{resource-ns}/statistic/{name}
LIST_METRICS	GET	/task/{id};{ns}/metric/
READ_METRIC_OF_TASK	GET	/task/{id};{ns}/metric/{name}
LIST_METRICS	GET	/task/{id};{ns}/metric/{name}/topic/{resource-id};{resource-ns}/experiment
LIST_METRICS	GET	/task/{id};{ns}/metric/{name}/experiment/{resource-id};{resource-ns}/topic
LIST_METRICS	GET	/task/{id};{ns}/metric/{name}/experiment/{resource-id};{resource-ns}/topic/{other-resource-id};{other-resource-ns}
LIST_STATISTICS_OF_TASK	GET	/task/{id};{ns}/statistic
READ_STATISTIC_OF_TASK	GET	/task/{id};{ns}/statistic/{name}
LIST_METRICS	GET	/task/{id};{ns}/statistic/{name}/metric/{other-name}
LIST_METRICS	GET	/task/{id};{ns}/statistic/{name}/topic/{resource-id};{resource-ns}
LIST_METRICS	GET	/task/{id};{ns}/statistic/{name}/topic/{resource-id};{resource-ns}/metric/{other-name}
LIST_POOLS_OF_TASK	GET	/task/{id};{ns}/pool
LIST_CORPORA_OF_TASK	GET	/task/{id};{ns}/corpus
LIST_EXPERIMENTAL_COLLECTIONS_OF_TASK	GET	/task/{id};{ns}/collection

Action	HTTP Method	URL
LIST_CONTRIBUTIONS_OF_TASK	GET	/task/{id};{ns}/contribution
LIST_VISUALIZATIONS_OF_TASK	GET	/task/{id};{ns}/visualization
LIST_PARTICIPANTS_OF_TASK	GET	/task/{id};{ns}/participant
LIST_ORGANIZERS_OF_TASK	GET	/task/{id};{ns}/organizer
LIST_TASKS	GET	/task
LIST_TASK_PROVENANCE_EVENTS	GET	/task/{id};{ns}/provenance

5.17 Topic Resource

Action	HTTP Method	URL
CREATE_TOPIC	POST	/topic
READ_TOPIC	GET	/topic/{id};{ns}
UPDATE_TOPIC	PUT	/topic/{id};{ns}
DELETE_TOPIC	DELETE	/topic/{id};{ns}
ADD_TOPIC_TO_TOPIC_GROUP	POST	/topic/{id};{ns}/member/topic-group/{resource-id};{resource-ns}
REMOVE_TOPIC_FROM_TOPIC_GROUP	DELETE	/topic/{id};{ns}/member/topic/{resource-id};{resource-ns}
ADD_TOPIC_TO_RELATION	POST	/topic/{id};{ns}/relation/{name}
REMOVE_TOPIC_FROM_RELATION	DELETE	/topic/{id};{ns}/relation/{name}
LIST_TASKS_OF_TOPIC	GET	/topic/{id};{ns}/task
LIST_TOPICS	GET	/topic
LIST_TOPIC_PROVENANCE_EVENTS	GET	/topic/{id};{ns}/provenance

5.18 Relation Resource

Action	HTTP Method	URL
CREATE_RELATION	POST	/relation
READ_RELATION	GET	/relation/{name}
UPDATE_RELATION	PUT	/relation/{name}
DELETE_RELATION	DELETE	/relation/{name}
LIST_RELATION	GET	/relation
LIST_RELATION	GET	/relation/{name}/provenance

5.19 Topic Group Resource

Action	HTTP Method	URL

Action	HTTP Method	URL
CREATE_TOPIC_GROUP	POST	/topic-group
READ_TOPIC_GROUP	GET	/topic-group/{id};{ns}
UPDATE_TOPIC_GROUP	PUT	/topic-group/{id};{ns}
DELETE_TOPIC_GROUP	DELETE	/topic-group/{id};{ns}
LIST_TOPIC_GROUPS	GET	/topic-group
LIST_TOPIC_GROUP_PROVENANCE_EVENTS	GET	/topic-group/{id};{ns}/provenance

5.20 Experiment Resource

Action	HTTP Method	URL
CREATE_EXPERIMENT	POST	/experiment
READ_EXPERIMENT	GET	/experiment/{id};{ns}
UPDATE_EXPERIMENT	PUT	/experiment/{id};{ns}
DELETE_EXPERIMENT	DELETE	/experiment/{id};{ns}
LIST_EXPERIMENT_ITEMS	GET	/experiment/{id};{ns}/item
ADD_ITEMS_TO_EXPERIMENT	POST	/experiment/{id};{ns}/item
LIST_ITEMS_OF_TOPIC_OF_EXPERIMENT	GET	/experiment/{id};{ns}/topic/{resource-id};{resource-ns}/item
READ_STATISTICS_OF_EXPERIMENT	GET	/experiment/{id};{ns}/statistic
LIST_EXPERIMENTS	GET	/experiment
LIST_EXPERIMENT_PROVENANCE_EVENTS	GET	/experiment/{id};{ns}/provenance

5.21 Metric Resource

Action	HTTP Method	URL
CREATE_METRIC	POST	/metric
READ_METRIC	GET	/metric/{id}
UPDATE_METRIC	PUT	/metric/{id}
DELETE_METRIC	DELETE	/metric/{id}
LIST_METRICS	GET	/metric
LIST_METRIC_PROVENANCE_EVENTS	GET	/metric/{id}/provenance

5.22 Pool Resource

Action	HTTP Method	URL
CREATE_POOL	POST	/pool

Action	HTTP Method	URL
READ_POOL	GET	/pool/{id};{ns}
UPDATE_POOL	PUT	/pool/{id};{ns}
DELETE_POOL	DELETE	/pool/{id};{ns}
LIST_POOLS	GET	/pool
LIST_ITEMS_OF_POOL	GET	/pool/{id};{ns}/item
ADD_ITEMS_TO_POOL	POST	/pool/{id};{ns}/item
UPDATE_ITEM_OF_POOL	PUT	/pool/{id};{ns}/item
REMOVE_ITEMS_FROM_POOL	DELETE	/pool/{id};{ns}/item
READ_ITEM_OF_POOL	GET	/pool/{id};{ns}/item/{id}
UPDATE_ITEM_OF_POOL	PUT	/pool/{id};{ns}/item/{id}
LIST_POOL_PROVENANCE_EVENTS	GET	/pool/{id};{ns}/provenance
LIST_ITEM_OF_POOL_PROVENANCE_EVENTS	GET	/pool/{id};{ns}/item/{id}/provenance

5.23 Visualization Resource

Action	HTTP Method	URL
CREATE_VISUALIZATION	POST	/visualization
READ_VISUALIZATION	GET	/visualization/{id};{ns}
UPDATE_VISUALIZATION	PUT	/visualization/{id};{ns}
DELETE_VISUALIZATION	DELETE	/visualization/{id};{ns}
LIST_VISUALIZATION	GET	/visualization
ADD_SNAPSHOT_TO_VISUALIZATION	POST	/visualization/{id};{ns}/snapshot/{name}
UPDATE_SNAPSHOT_OF_VISUALIZATION	PUT	/visualization/{id};{ns}/snapshot/{name}
REMOVE_SNAPSHOT_FROM_VISUALIZATION	DELETE	/visualization/{id};{ns}/snapshot/{name}

5.24 Contribution Resource

Action	HTTP Method	URL
CREATE_CONTRIBUTION	POST	/contribution
READ_CONTRIBUTION	GET	/contribution/{id};{ns}
UPDATE_CONTRIBUTION	PUT	/contribution/{id};{ns}
DELETE_CONTRIBUTION	DELETE	/contribution/{id};{ns}
LIST_CONTRIBUTIONS	GET	/contribution
LIST_CITATIONS_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/cite
ADD_CONTRIBUTION_TO_CONTRIBUTION	POST	/contribution/{id};{ns}/member/cont

Action	HTTP Method	URL
		ribution/{resource-id};{resource-ns}
REMOVE_CONTRIBUTION_FROM_CONTRIBUTION	DELETE	/contribution/{id};{ns}/member/contribution/{resource-id};{resource-ns}
LIST_CONTRIBUTIONS_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/contribution
ADD_CONTRIBUTION_TO_CAMPAIGN	POST	/contribution/{id};{ns}/member/campaign/{resource-id};{resource-ns}
REMOVE_CONTRIBUTION_FROM_CAMPAIGN	DELETE	/contribution/{id};{ns}/member/campaign/{resource-id};{resource-ns}
LIST_CAMPAIGNS_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/campaign
ADD_CONTRIBUTION_TO_TRACK	POST	/contribution/{id};{ns}/member/track/{resource-id};{resource-ns}
REMOVE_CONTRIBUTION_FROM_TRACK	DELETE	/contribution/{id};{ns}/member/track/{resource-id};{resource-ns}
LIST_TRACKS_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/track
ADD_CONTRIBUTION_TO_TASK	POST	/contribution/{id};{ns}/member/task/{resource-id};{resource-ns}
REMOVE_CONTRIBUTION_FROM_TASK	DELETE	/contribution/{id};{ns}/member/task/{resource-id};{resource-ns}
LIST_TASKS_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/task
ADD_CONTRIBUTION_TO_CORPUS	POST	/contribution/{id};{ns}/member/corpus/{resource-id};{resource-ns}
REMOVE_CONTRIBUTION_FROM_CORPUS	DELETE	/contribution/{id};{ns}/member/corpus/{resource-id};{resource-ns}
LIST_CORPORA_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/corpus
ADD_CONTRIBUTION_TO_EXPERIMENTAL_COLLECTION	POST	/contribution/{id};{ns}/member/experimental-collection/{resource-id};{resource-ns}
REMOVE_CONTRIBUTION_FROM_EXPERIMENTAL_COLLECTION	DELETE	/contribution/{id};{ns}/member/experimental-collection/{resource-id};{resource-ns}
LIST_EXPERIMENTAL_COLLECTIONS_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/experimental-collection
ADD_CONTRIBUTION_TO_EXPERIMENT	POST	/contribution/{id};{ns}/member/experiment/{resource-id};{resource-ns}
REMOVE_CONTRIBUTION_FROM_EXPERIMENT	DELETE	/contribution/{id};{ns}/member/experiment/{resource-id};{resource-ns}
LIST_EXPERIMENTS_OF_CONTRIBUTION	GET	/contribution/{id};{ns}/experiment
LIST_CONTRIBUTION_PROVENANCE_EVENTS	GET	/contribution/{id};{ns}/provenance

6 Use Case Scenarios

6.1 Visualization of Topics, Experiments and Metrics

This use case scenario, made in collaboration with ROMA1, describes how the users of the DIRECT system can get and handle resources to create visualizations with data about topics, experiments, and metrics of a topic.

Assuming that the DIRECT system is available at the URL

`http://direct.dei.unipd.it/`

then the HTTP GET request for a task with identifier **`id_tsk`** and namespace **`ns_tsk`**

`http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/metric`

will provide, according to Section 5.16, data about topics, experiments, and metrics as response, which can be envisioned as the three-dimensional matrix, or Online Analytical Processing (OLAP) data cube [Elmasri and Navathe, 2003], sketched in Figure 87.

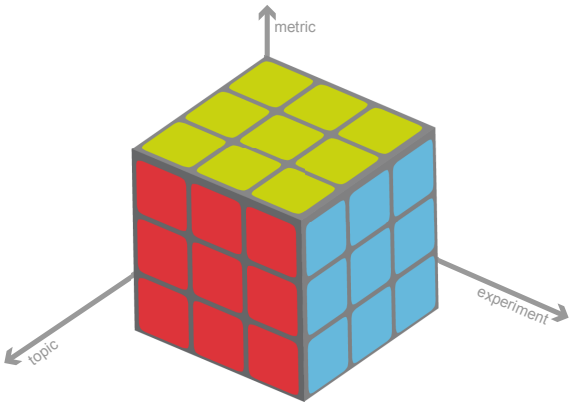
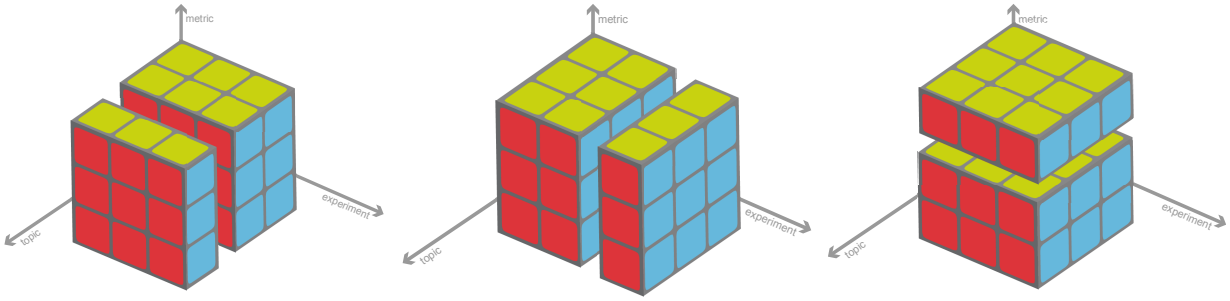


Figure 87 - Topics, Experiments, and Metrics Data as a Three Dimensional Matrix or OLAP Data Cube

The data cube can be rotated (*pivot operation*) to show topics, experiments and metrics as rows or columns, providing alternative visualizations of data that the user can save and export as snapshots. It is also possible to select and reorder rows or columns, and *slice* portions of cube, as represented by the following figures.



(a) (b) (c)

Figure 88 - Topics, Experiments, and Metrics Data Matrix Sliced on a Fixed Topic (a), Experiment (b), or Metric (c)

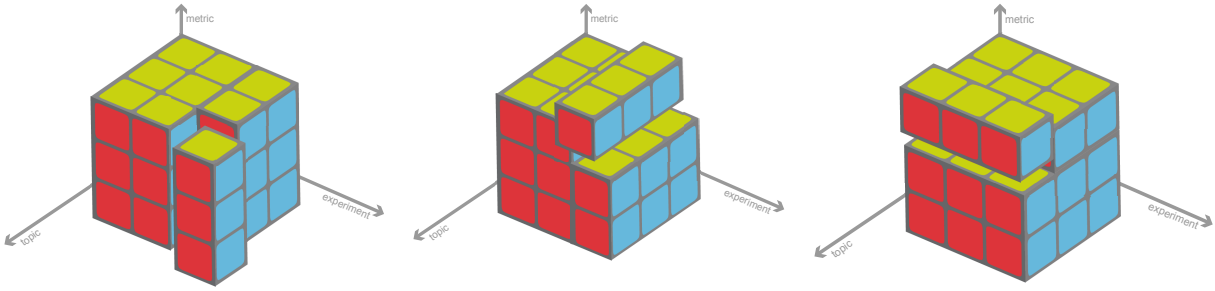
Figure 88 shows how to slice the data cube to provide information about a fixed topic (a), a fixed experiment (b), or a fixed metric (c).

The HTTP GET request provided to the DIRECT system to reach this goal will be:

a) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/topic/{id_tpc};{ns_tpc}/metric`
b) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/experiment/{id_exp};{ns_exp}/metric`
c) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/metric/{id_mmt};{ns_mtc}`

For each slice it is possible to refine the request specifying two parameters instead of one, as shown in Figure 89 where the corresponding requests are:

a) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/topic/{id_tpt};{ns_tpc}/experiment/{id_exp};{ns_exp}/metric` or `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/experiment/{id_exp};{ns_exp}/topic/{id_tpt};{ns_tpc}/metric`
b) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/experiment/{id_exp};{ns_exp}/metric/{id_mmt};{ns_mtc}/topic` or `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/metric/{id_mmt};{ns_mtc}/experiment/{id_exp};{ns_exp}/topic`
c) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/metric/{id_mmt};{ns_mtc}/topic/{id_tpt};{ns_tpc}/experiment` or `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/topic/{id_tpt};{ns_tpc}/metric/{id_mmt};{ns_mtc}/experiment`



(a) (b) (c)

Figure 89 Topics, Experiments, and Metrics Data Matrix Sliced on a Fixed Topic and Experiment (a), Experiment and Metric (b), or Metric and Topic (c)

Lastly, when specifying three parameters we have the selection of the cube of Figure 90 where the corresponding requests are:

a) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/topic/{id_tpt};{ns_tpc}/experiment/{id_exp};{ns_exp}/metric/{id_mmt};{ns_mtc}` or
`http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/experiment/{id_exp};{ns_exp}/topic/{id_tpt};{ns_tpc}/metric/{id_mmt};{ns_mmt}` or
`http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/metric/{id_mmt};{ns_mtc}/experiment/{id_exp};{ns_exp}/topic/{id_tpt};{ns_tpc}`

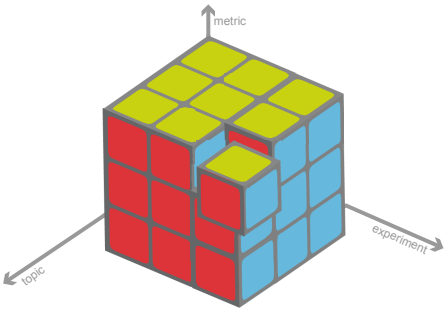


Figure 90 - Topics, Experiments, and Metrics Data Matrix Sliced on a Fixed Topic, Experiment, and Metric

6.2 Visualization of Experiments, Statistics, and Metrics

This use case scenario describes how the user of the DIRECT system can access information about experiments, experiment statistics, and metrics, using the OLAP data cube approach presented in 6.1.

The cube (a) in Figure 91 can be obtained, according to the parameters presented in 6.1, using the HTTP GET request

```
http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/experiment-statistic
```

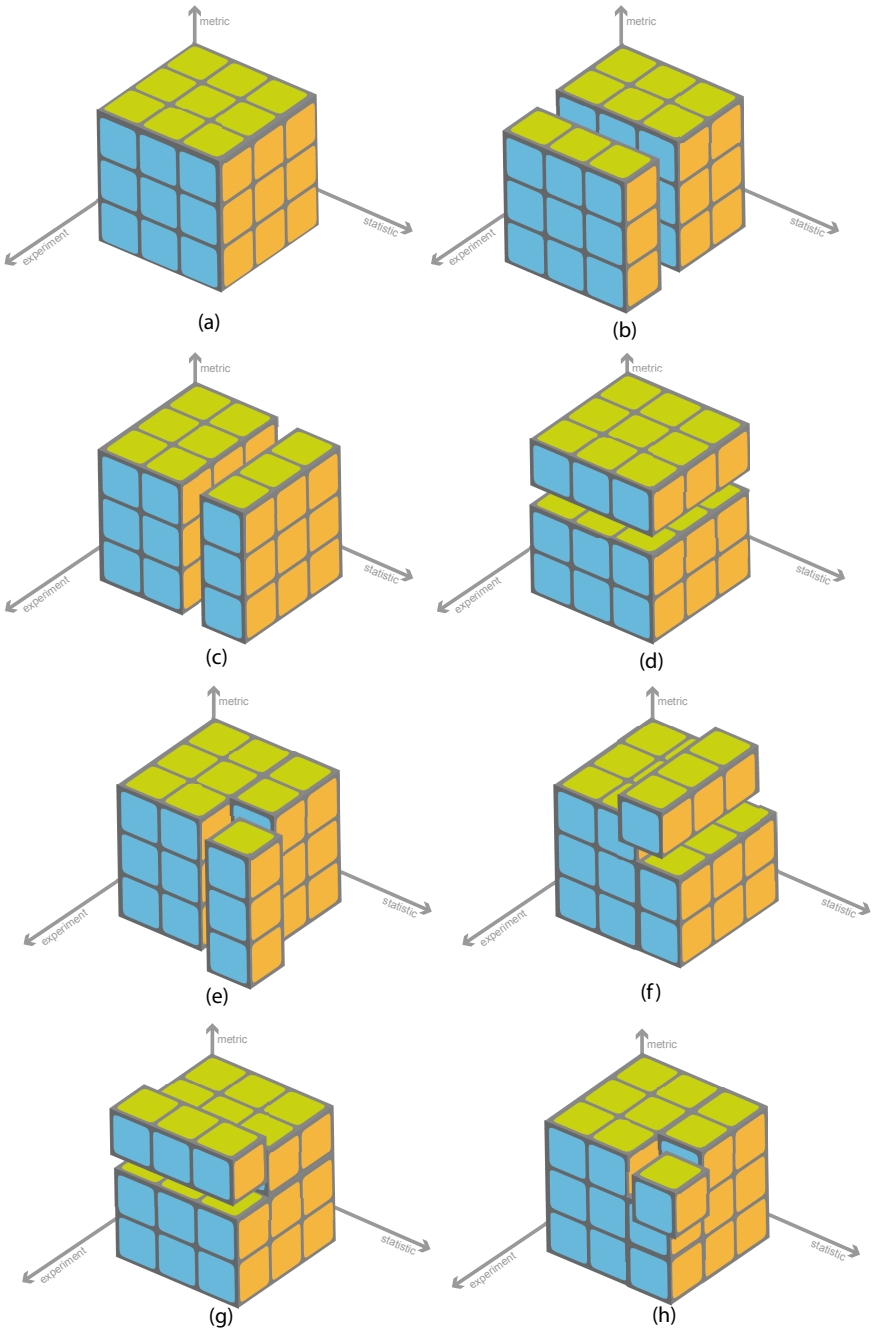


Figure 91 - Experiments, Experiment Statistics, and Metrics Data as a Three Dimensional Matrix or OLAP Data Cube

As an example, the slices (b), (c), and (d) can be obtained using:

a) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/experiment/id_exp};{ns_exp}/experiment-statistic`
b) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/experiment-statistic/{id_est};{ns_est}`
c) `http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/metric/{id_mmt};{ns_mtc}`

6.3 Visualization of Topics, Statistics, and Metrics

This last use case scenario describes how the user of the DIRECT system can access information about topics, statistics, and metrics, using the OLAP data cube approach presented in 6.1.

The cube (a) in Figure 92 can be obtained, according to the parameters presented in 6.1, using the HTTP GET request

`http://direct.dei.unipd.it/task/{id_tsk};{ns_tsk}/statistic`

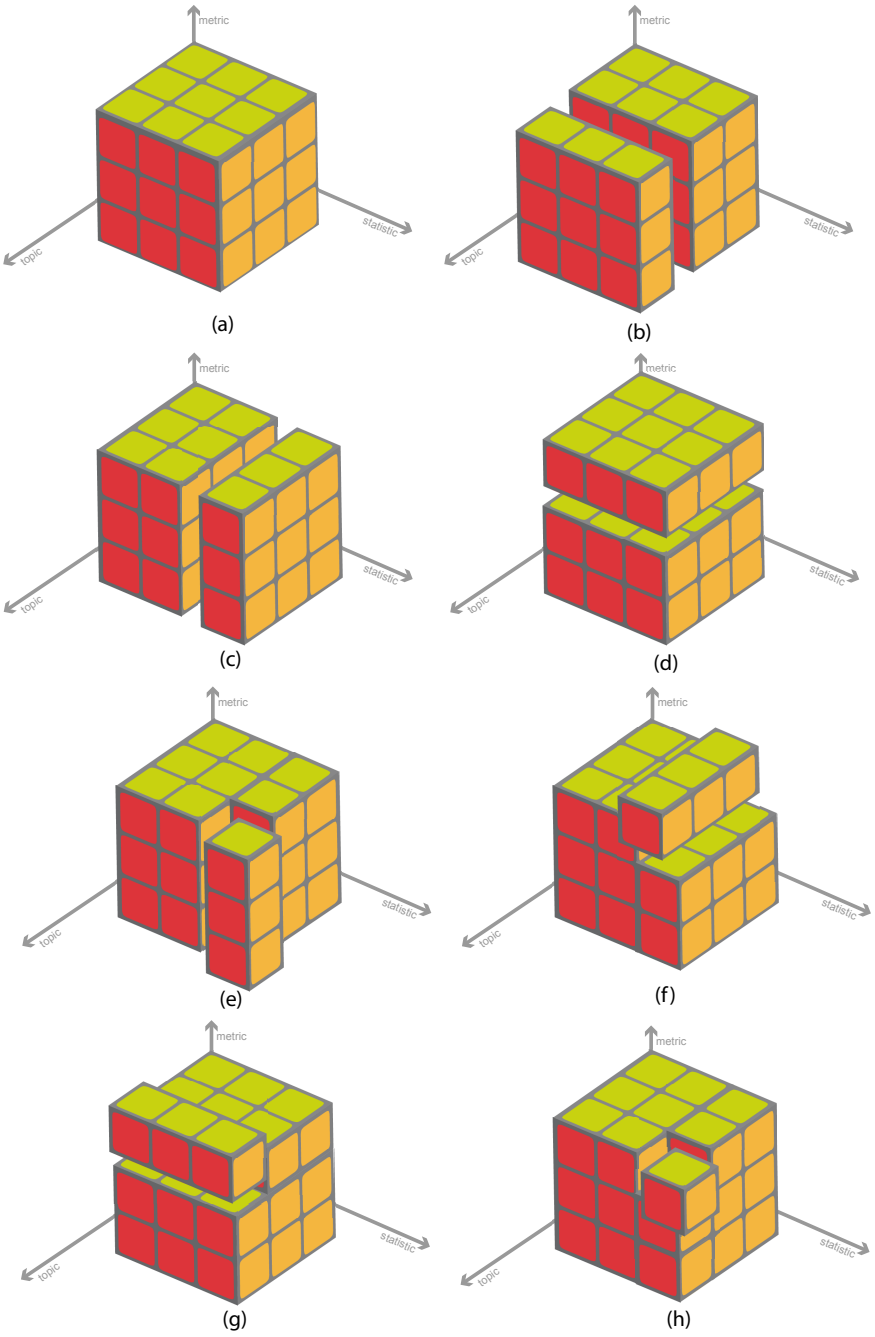


Figure 92 - Experiments, Statistics, and Metrics Data as a Three Dimensional Matrix or OLAP Data Cube

References

[Agosti and Ferro, 2009] M. Agosti, N. Ferro. Towards an Evaluation Infrastructure for DL Performance Evaluation. In G. Tsakonas, and C. Papatheodorou, editors, *Evaluation of Digital Libraries: an insight into useful applications and methods*, pages 93-120, Chandos Publishing, Oxford, UK, 2009.

[Batini et al., 1992] C. Batini, S. Ceri, S. B. Navathe. (1992) *Conceptual Database Design* (1st ed.). The Benjamin/Cummings Publishing Company, Inc. Redwood City, California, USA.

[Berners-Lee et al., 2005] Berners-Lee, T., Fielding, R., and Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax. RFC 3986.

[Croft et al., 2009] Croft, B., Metzler, D., and Strohman, T. (2009). *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley Publishing Company, USA.

[Clark et al., 2009] Clarke C. L. A., Craswell N., and Soboroff, I. (2009). Overview of the TREC 2009 Web Track. In *Proc. of TREC-2009*.

[Crockford, 2006] Crockford, D. (2006). The application/json Media Type for JavaScript Object Notation (JSON). RFC 4627.

[Elmasri and Navathe, 2003] R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*, 4th Edition, Addison-Wesley, 2003.

[EU, 2007] Commission of the European Communities (2007). *Communication from the Commission to the European Parliament, the Council and the European Economic and Social Committee on scientific information in the digital age: access, dissemination and preservation*. COMM(2008) 56 Final.

[Ferro, 2001] Ferro, N. (2011). DIRECT: the First Prototype of the PROMISE Evaluation Infrastructure for Information Retrieval Experimental Evaluation. *ERCIM News*, 86:54–55.

[Fielding and Taylor, 2002] R. T. Fielding and R. N. Taylor. Principled design of the modern web architecture. ACM TOIT, 2:115-150, 2002.

[Fielding et al., 1999] Fielding, R., Gettys, Y., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (1999). Hypertext Transfer Protocol – HTTP/1.1. RFC 2616.

[Franks et al., 1999] Franks, J., Hallam-Baker, P. M., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and Stewart, L. (1999). HTTP Authentication: Basic and Digest Access Authentication HTTP Authentication: Basic and Digest Access Authentication. RFC 2617.

[Freed and Borenstein, 1996] Freed, N. and Borenstein, N. (1996). Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045.

[Järvelin and Kekäläinen, 2002] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

[NSB, 2005] National Science Board (2005). *Long-Lived Digital Data Collections: Enabling Research and Education in the 21st Century (NSB-05-40)*. National Science Foundation (NSF). http://www.nsf.gov/pubs/2005/nsb0540/.

[PMSEIC, 2006] Working Group on Data for Science (2006). *FROM DATA TO WISDOM: Pathways to Successful Data Management for Australian Science. Report to Prime Minister's Science, Engineering and Innovation Council (PM-SEIC)*, http://www.innovation.gov.au/Science/PMSEIC/Documents/FromDatatoWisdom.pdf.

[PROMISE D2.1, 2011] Karlgren, J., Eriksson, G., Frieseke, M., Gäde, M., Hansen, P., Järvelin, A., Lupu, M., Müller, H., Petras, V., and Stiller, J. (2011). Deliverable D2.1 – Initial specification of the evaluation tasks. PROMISE Network of Excellence, EU 7FP, Contract N. 258191. http://www.promise-noe.eu/documents/10156/5fce1a18-d6c2-4063-958c-bb3e73a27456.

[PROMISE D3.1, 2011] Agosti, M., Di Nunzio, G. M., and Ferro, N. (2011). *Deliverable D3.1 – Initial prototype of the evaluation infrastructure*. PROMISE Network of Excellence, EU 7FP, Contract N. 258191. http://www.promise-noe.eu/documents/10156/e0df8a3c-388f-40e8-bfbd-04434a393004.

[PROMISE D5.1, 2011] Croce, M., Di Reto, E., Granato, G. L., Hansen, P., Sabetta, A., Santucci, G., and Veltri, F. (2011). Deliverable *D5.1 – Collaborative user interface requirements*. PROMISE Network of Excellence, EU 7FP, Contract N. 258191. http://www.promise-noe.eu/documents/10156/50834686-2118-48f8-a57b-8553ec3d7981.

[PROMISE D5.2, 2011] Granato, G. L., Santucci, G., and Tino, G. (2011). *Deliverable D5.2 – User interface and Visual analytics environment requirements*. PROMISE Network of Excellence, EU 7FP, Contract N. 258191.

[Richardson and Ruby, 2007] Richardson, L. and Ruby, S. (2007). *RESTful Web Services*. O'Reilly Media, Inc., Sebastopol (CA), USA.

[Rowe et al., 2010] Rowe, B. R., Wood, D. W., Link, A. L., and Simoni, D. A. (2010). *Economic Impact Assessment of NIST's Text REtrieval Conference (TREC) Program*. RTI Project Number 0211875, RTI International, USA. http://trec.nist.gov/pubs/2010.economic.impact.pdf

[Voorhees and Harman. 2005] E. M. Voorhees, D. Harman, editor. *TREC: Experiment and Evaluation in Information Retrieval.* The MIT Press, Cambridge, MA, 2005.

[Yilmaz and Aslam, 2008] Yilmaz, E. and Aslam, J. A. (2008). *Estimating average precision when judgments are incomplete*. Knowl. Inf. Syst. 16, 2 (July 2008), 173-211.

[W3C, 2008] W3C (2008). Extensible Markup Language (XML) 1.0 (Fifth Edition) – W3C Recommendation 26 November 2008. http://www.w3.org/TR/ xml/.