Chapter 2

Dataspace management

The objective of this chapter is to present a general perspective over the existing works oriented to define the notion of dataspace and to specify the main functions and components of a dataspace management system. Although this project is focused on solving the problem related to the dataspace's indexation, we consider it is important to introduce the definition of a dataspace. As this notion is recent, it is necessary to introduce this concept and emphasize the importance of having an indexing structure within this context. Particularly, this chapter analyzes existing works contributing to the construction of a level with respect to different abstraction levels in order to satisfy the requirements over the dataspace's organization.

The rest of this chapter is organized as follows. Section 2.1 provides an overview over the works introducing notion of dataspaces as a new abstraction for data management and the first approximations to specify the architecture and components of a DSSP. Section 2.2 discusses existing works contributing in a significant way to the problem related to the resource's indexation in different levels. Section 2.2.4 describes a recent data integration paradigm adapted to the construction of indexes in the context of dataspaces. Finally, Section 2.3 concludes this chapter by discussing the importance of proposing an integral multi-level index satisfying the characteristics of a dataspace.

2.1 Dataspaces

The notion of dataspaces is an abstraction arisen from the requirements presented over recent years after the development of multiple environments such as the Web, business, government, and in general any environment in which great amounts of data are produced and consumed. The rest of this section describes the firsts approximations leading to the definition of a dataspace and the specification of a system providing a transparent layer between users and the resources (data or applications) published into the dataspace, like a DBMS.

2.1.1 The origin of dataspaces

The study of computer science over recent years has promoted the production of applications aiming to solve particular requirements. [10] analyze the requirements presented over technologies in the world, which promotes the construction of applications integrated at different data and system abstraction levels (e.g., user interface, middleware, schemes, semantics, etc.).

As applications have been continuously developed and the requirements to integrate different applications has remained, the notion of databases has been broadly spread mainly because of the diverse media used to produce data (e.g., text document, blogs, multimedia files, etc.) This data has structured, semi-structured and non-structured data models, therefore they cannot be managed using an existent system in an homogeneous way (e.g., RDBMS, OODBMS). As a consequence of these deficiencies, the notion of *dataspaces* [18, 13] arises as a new data management abstraction that considers a great amount of data sources implicitly related among each other under different granularity levels. Unlike traditional structured databases where meta-data (e.g., data model) is presented, resources in a dataspace may lack of meta-data describing their intension, structure, and/or semantics.

The notion of dataspaces analyzes the vocabulary used in databases in order to retake problems previously addressed by the databases' community. A subset of these problems are the following:

- Data mediation and integration [15, 8]
- Information integration [4]
- Semantic mappings [22, 23]
- Data model management [6]
- Continuous data [17, 16]

- etc.

In order to provide efficient solutions to these problems, different authors have aimed to define the first requirements to build systems providing support capabilities to the dataspaces [13]:

- Dataspace catalogs containing an up-to-date register of subscribed participants and their associated meta-data (if there exists).
- Data lineage in order to determine relationships among data with respect to diverse granularities.
- Indexing definition to provide an homogeneous view over the dataspace and the resources location.

These requirements arise new challenges and motivate researchers to revisit the achievements and solutions proposed within the databases area in order to partially fulfill them. Particularly, we consider important to reanalyze the proposals oriented to the development of current DBMS.

2.1.2 Dataspace support platform

Database management systems (DBMS) have been evolving since the 60's with the emergence of first approximations to manage data and particularly in the 70's with the specification of the relational model [7].. The result of the construction of such systems has culminated with the development of the DBMS capable to provide data access through the definition of structured queries (e.g., SQL) as well as to guarantee data to be persistent, available, coherent, etc. This is achieved thanks to the definition of the ACID properties, provided by the transactional execution model. Nowadays, with the emergence of dataspaces, it would be desirable to develop systems capable to provide base functionality over their management and querying as similar as possible to a DBMS.

Franklin et al. [13] define a dataspace support platform (DSSP) as a platform providing a set of collaborative services with general functions in order to satisfy the requirements presented over the dataspaces.

Table 2.1: DBMS vs DSSP

	Data model	Data control	Answers	Semantic integration
DBMS	Relational/OO	Complete	Precise	Explicit
DSSP	All	Partial	Approximate	Implicit/Incremental

To situate the DSSP within the data management domain, we present a general comparison between the main properties of a DBMS and a DSSP (see Table 2.1).

A DBMS is oriented to the management of structured data models (e.g., relational, object-oriented), offering a transactional execution model guaranteeing the ACID properties. Schema or data structure in such systems allows to adopt a querying strategy oriented to the retrieval of precise results. On the contrary, a DSSP is oriented to support any data model. For this reason, the system delegates certain functions to autonomous services (e.g., persistence services). Due to the lack of schemas in such environments, querying strategies are oriented to retrieve approximate results.

[13] define a set of interrelated services to build a DSSP that allows developers to focus on the specific challenges of their applications rather than dealing with the heterogeneity and distribution of data within the dataspace. The minimal services provided by a DSSP are the following:

- **Catalog**: Represents an inventory of the participants subscribed into the dataspace (producers and consumers in our vocabulary) and the minimal required information in order to model the infrastructure of the dataspace and to provide functionalities to the rest of the DSSP's services.

- **Search and querying**: Search is a characteristic presented over current search engines that allows users to find data associated to unfamiliar domains. Structured query capabilities is a characteristic provided by current DBMS to find data in environments in which vocabulary and structure are well known. This service must provide users with both capabilities, e.g. to define a structured query (SQL style) over a set of unfamiliar data obtaining approximate results. These results may be refined as more information is known about the data sources.

- **Indexing**: Maintains available the meta-data associated to the participants in order to provide continuous access to data. An indexing service must be able to model relationships among data that may allow to query the dataspace, identify data and producers, and locate them within the dataspace. Additional desirable functions include to maintain a cache increasing the availability of resources and providing recovery capabilities when certain resources become unavailable or fail.

- **Discovery**: Participants may connect explicitly to the dataspace (e.g., subscription) or through a discovery process. This way, this service must be able to detect data sources, consumers and producers that may form part of the dataspace by registering them into the catalog and tagging them. For example, a participant may be tagged with respect to the queries he executes and the feedback he maintains with the DSSP, while a resource may be tagged using a term collecting process over the data it contains.

- **Participant enhancement**: Producers, consumers, and external services must implement a set of minimal functions that will allow them to interact with the DSSP.

Elsayed et al. [20] define an specialization of a DSSP into an infrastructure in order to build a dataspace management system (DSMS). They extend the components of the DSSP by adding services such as a query processor and a query translator. Also, they retake certain characteristics of Grid technologies as they provide pertinent contributions to the services of the DSMS. An example of a DSMS is iMeMex [1] [9], a first implementation of a DSMS for personal data management.

Moreover, Abiteboul and Polyzotis [1] propose Data Ring, a system for data sharing based in a P2P architecture. Such architecture delegates query expression and processing to each *peer* connected to the system. Each peer represents an autonomous and heterogeneous software entity providing data and computing capabilities, similarly as the components of a dataspace. Unlike other data sharing systems (e.g., GNutella, FreeNet, etc.), DataRing supports heterogeneous documents and allows to retrieve particular data located in the documents. Additionally, the system does not use network flooding over the peers in order to process a query. The system locates the pertinent peer capable to solve the query [33].

[1] imemex.ethz.ch

Both the DSSP and DSMS emphasize the existence of indexing structures. In this project, we consider that in order to enable the correct functionality of the DSSP to process and rewrite queries, integrate data, etc., it is necessary to have an integrated global view over the dataspace. The rest of this chapter describes our analysis over existing works related to the indexation of dataspaces.

2.2 Indexing dataspaces

From databases perspective, an index is a defined as a data structure in which partial replicas of data are stored in memory in order to optimize the execution of operations over data. From the search engine perspective, an index is a collection of organized data, that enables information retrieval. In our context of dataspaces, both perspectives are useful and may be merged in order to produce an integrated and homogeneous view (GaV) over the dataspace according to terms belonging to a particular vocabulary. Through this view, consumers will be able to identify the data sources published in the dataspace satisfying their requirements, and locate physically this data in order to retrieve them.

The rest of this section focuses on describing existing works aiming to provide a solution to the construction of an indexing service over the dataspace. First, we present recent works related to the characterization of the dataspace in order to build a first layer interacting with the consumers. Then, we analyze works aiming to characterize the resource's content (e.g., sources, documents, producers) through annotation-based schemas. Finally, we describe classical solutions proposed to build an indexing structure to organize the physical location of data (e.g., hard-disks or memories).

2.2.1 Dataspace semantization

Weikum [37] analyzes and defines a state-of-art of semantic expressiveness applied to search engines. Particularly, we are interested in the proposal of Suchanek et al. [34] who developed an ontology denominated YAGO [2] composed by a set of entities and binary relations of the

[2] www.mpi-inf.mpg.de/yago-naga/yago/

form

$$< \mathtt{entity_i}, \mathtt{relation}, \mathtt{entity_j} >$$

. Binary relations are organized into two main categories:

- Taxonomic: ($< \mathtt{Automovil}, \mathtt{isA}, \mathtt{Transport} >$) y
- Non-taxonomic: ($< \mathtt{Automovil}, \mathtt{Color}, \mathtt{Red} >$).

YAGO was built through an integration process between knowledge contained in Wikipedia [3] web encyclopedia and the concept's taxonomy defined in WordNet [4] [12]. The main contribution in this project is the definition of the integrated and organized view from the knowledge stored in the web encyclopedia Wikipedia (as a set of semi-structured documents).

Dong and Halevy [11] propose the definition of an integrated view over heterogeneous sources by modeling exhaustively the sources schemas (if there exist). They are supported in the definition of an indexing structure to provide users with capabilities to execute *keyword* and structured queries [5] even when the schemas (if there exist) are unknown by the consumers. In order to achieve this, they propose a representation of data items as tuples attribute-value to express the following hierarchical queries:

- `'value'`
- `attribute='value'`
- `relation/attribute='value'`
- `schema/relation/attribute='value'`

In order to process any of these queries it is necessary to have annotations associated to the resources describing their content. Such annotations may be manually or automatically defined using services for collecting terms. The rest of this section describes existing works oriented to define the resource's annotations.

[3] en.wikipedia.org

[4] wordnet.princeton.edu

[5] Structured queries refer to predicate queries of the form attribute-value (e.g., auto.color='red'.

2.2.2 Dataspace description

Annotations describe the content of a particular resource and facilitate the identification of terms contained in them, and building resource groups (domains) defined by their common annotations. Annotations may be defined: (i) manually through a tagging system, or (ii) automatically using a term collecting service (stemization and lematisation).

In digital libraries and collections, Sánchez et al. [30] propose the notion of *induced tagging* to define a collaborative tagging of resources using an annotation schema. Collaborative tagging is an alternative to build a knowledge base based on annotations and has presented a great impact in social networks such as Flickr, [6] , Facebook [7] , as well as in feed and evaluation sites such as REC [8] , Delicious [9] y Google Bookmarks [10] .

[11] propose to describe resources by adapting the annotation descriptions into a more structured model. This is achieved by defining a schema in an exhaustively and approximate way e.g., retrieving meta-data embedded in documents [24, 31].

Howe et al. [19] apply the profile notion used in databases [28] into dataspaces. They retrieve the implicit structure of data and expose it within a browser in order to identify resources manually.

To achieve the identification of annotated resources automatically given a certain query (keyword-based or structured-based), it must be considered that each resource is annotated using a set of terms giving them the multidimensionality property. For these reasons, it will not be always possible to find an exact result but an approximate one. This is done by defining the notion of *neighborhood* representing the distance existing among entities with common characteristics (in our context, terms). An entity is described by a set of terms and in the resource location context, q is an entity describing a query and R is the set of entities (resources) partially ordered. Order is determined by the proximity of each R_i with respect to q. This way, for each R_i, there exist a neighborhood relation v so that $q \underset{\longrightarrow}{v} R_i$. In order to identify the "nearest neighbor", there are diverse works [25, 38, 21, 3] that propose the construction of different indexing structures and algorithms to identify data according to their

[6] www.flickr.com

[7] www.facebook.com

[8] ict.udlap.mx:9090/reduc/

[9] delicious.com

[10] www.google.com/bookmarks

recall proportion with respect to a particular query.

2.2.3 Storage space

Storage space within dataspaces is a set of persistence services sharing and managing a storage subspace. Several works have been developed to optimize the response time related to data retrieval from an storage device given the problems of rotational latency, workload distribution, and network latency, since services are disposed over a network (as well as latency time [32]).

A classical solution is the proposal of Bayer and McCreight [2] who developed the B-Tree structure which is one of the most used structures for indexing within the databases and file organization areas. In principle, this structure was defined to organize and optimize the access to files located in storage devices in order to reduce the latency immersed in these devices. Nowadays, latency remains present in these devices, and despite the fact that there have been developed several strategies to reduce it, now we must have to consider the network latency as well as the one from services built on top of the network like a dataspace.

On the other hand, there exist several works who have addressed not only the data location and optimization of data retrieval, but also the problem related to the multidimensionality (Section 2.2.2). An example of such works is the proposal of Bertchold et al. [5] who organize storage space to enable a pertinent retrieval of data from their physical location.

2.2.4 Index construction

Previously, we have mentioned that classic proposals have aimed to integrate data within heterogeneous environments. However in the context of dataspaces, these solutions are not sufficient as they require to have a priori meta-data describing the structure and intension of the resources in order to achieve data integration. As we mentioned before, certain sources in a dataspace may be described using annotations or may export a data model or schema, however it is assumed the absence of meta-data within the resources.

Data integration in heterogeneous environments has always implied the definition of mappings and semantic assertions [22, 23] (*Schema First Aproach*). However, in dataspaces this approach is not possible at least when we are dealing with first attempts to integrate the data of the dataspace. This way, several questions arise within this context such as: how could we determine which data are related among each other?, how can we integrate them despite of not knowing their structure and data models?, how can we determine if results are pertinent?, etc.

Madhavan et al. [26] propose the *pay-as-you-go* strategy as a new integration paradigm which assumes the absence of schemas (*No Schema Aproach*). In our approach, *pay-as-you-go* provides the fundamentals to achieve the construction of the indexing structures along time through:

- having user feedback,
- measuring and qualifying results
- building clusters based on annotations,

and mechanisms that allow to retrieve information from the systematic monitoring of the dataspace. This monitoring is performed by the DSSP when queries are answered along time.

Pay-as-you-go is a paradigm we consider for the index construction as the index is the main element of the DSSP used to solve data management problems such as the query rewriting, information retrieval, and data integration.

Vaz Salles et al. [36] developed iTrails, one of the first systems implementing the *pay-as-you-go* paradigm. This system shows how data integration in dataspaces can be achieved by defining and refining automatically semantic mappings.

2.3 Conclusions

Our state-of-art enumerates some existing works defining the notion of dataspaces [18, 13], and some architectures of support and management systems over the dataspace. Particularly, [13, 20, 9, 1] locate the indexing service as a fundamental component to provide a solution to problems previously addressed in traditional databases (e.g., data integration, query rewriting, information retrieval, etc.), which remain of interest within the perspective of absence of schemas (*No Schema Aproach*).

With respect to indexation, we have retaken works contributing considerably to the construction of an integral index considering three main aspects: (i) an integrated view over the content of the dataspace [37, 34, 11] as a transparent layer for the consumers, (ii) the pertinent and/or approximate identification of resources and data sources based on annotation schemas [30, 11, 19, 25], and finally (iii) the pertinent location of data into hard-disks[2, 5] within the storage space. Additionally, results are refined with respect to their precision and recall, thus the paradigm *pay-as-you-go* [36] provides the foundations to build and feed the indexing structure, and to refine results (e.g., if A is a set of results ordered with respect to time where A_0 was presented before A_1, then A_{i+1} must be a more precise and pertinent result thant A_i).

Considering the three main aspects enunciated before, it is how arises our proposal to built an index from the consumer point of view (external layer), passing by the description and precise identification of data sources and documents given a specific query (logical layer) to the location and pertinent extraction of data from the entity managing the data and the hard-disk under it is located (physical layer).