Chapter 2
Data Governance

Data and information are the fundamental resources managed by public administrations to provide services to users. So, the reader should not be surprised that at the beginning of the book we focus on the part of the eG4M methodology that deals with data. To give an example, in the Italian Central Public Administration more than 500 large-sized databases are managed, while in each one of the 21 regions more than 1000 large-sized databases are used. Each one of the above databases has been designed independently from the other by different teams and is updated by different sources and data flows.

As a consequence, administrative processes making use of such heterogeneous variety of data result in costly and low-quality services. To mention another typical phenomenon of eGovernment initiatives, data are often considered ancillary to software applications, so in order to solve a problem, the most important issue is considered to produce a new software application, without worrying about the data involved. We arrive at the conclusion that data deserve a special attention in eGovernment initiatives, from different perspectives, that are discussed in this chapter. Section 2.1 introduces the concept of data governance and of its different facets. Section 2.2 discusses the most important of such facets, namely *data quality*, providing an introduction to data quality dimensions and methodologies for data quality assessment and improvement. Section 2.3 introduces the reader the representation of data by means of graphical models that allow us to represent data classes by means of schemas, so as to understand more clearly their meaning and relationships among them. In Sects. 2.4 and 2.5 we deal with the problem of integrating data schemas, to achieve a comprehensive and reconciled description of the information content managed in an administration or a set of public administrations; in Sect. 2.4 we discuss schema integration in the small, say, when performed on 5–10 schemas, while Sect. 2.5 discusses schema integration in large, when abstraction mechanisms are needed to govern the complexity of schema representation. An encompassing comment is needed for the terms data and information. When we use the term *data* we refer to values represented in a database by means of n-ples of attributes whose domains are usually numeric or alphanumeric, such data are

This chapter is authored by Carlo Batini.

G. Viscusi et al., *Information Systems for eGovernment*,
DOI 10.1007/978-3-642-13571-2_2,

also called in the literature *structured data*. The term *information* is referred to as any type of representation such as maps, images, videos, semi-structured text, and unstructured text.

2.1 Data Governance Issues

Data governance can be defined as the formal orchestration of people, processes, and technology to enable an organization to leverage data as an enterprise asset. Several different issues related to data governance exist:

1. *Data quality*: The set of issues that allow to assess different dimensions of data quality (e.g., accuracy, timeliness, consistency) and to improve such dimensions by means of activities that may operate directly on data or else on processes that interchange or elaborate data.
2. *Data modeling*: The representation of classes of data in terms of a conceptual model, namely a model whose linguistic categories highlight the aspects related to the meaning of data, instead of their representation in a computer.
3. *Data integration*: The technologies that allow to query and access different independent databases as they were virtually a single, integrated database.
4. *Schema integration*: The process of harmonizing conceptual descriptions of data across heterogeneous databases.
5. *Data architecture governance*: The process that considers the overall architecture of data, namely the representation of data in different databases of the organization, and conceives a new architecture that, making use of data integration solutions, maximally increases the amount of queries that can be expressed on it.
6. *Data governance management*: The asset of responsibilities and activities and their collocation in the organization that allow to manage, monitor, govern, improve the quality and level of integration of data.

Although all of the above issues are relevant in data governance, in this chapter we focus on data quality, data modeling, and schema integration. Data quality is relevant due to the negative impact of loose quality data in administrative processes and in service provision. Data modeling provides all the users of data a common model and, consequently, a common understanding of the data resource. Schema integration is relevant since it gives all the users of data to proceed to their reconciliation in terms of a common integrated description, notwithstanding the heterogeneous representation of data in databases.

2.2 Data Quality

In this section we deal with data quality issues at an introductory level, since the whole matter has been thoroughly discussed by one of the authors in the book [23]. Data quality has serious consequences, of far-reaching significance, for the

efficiency and effectiveness of organizations and businesses. The report on data quality of the Data Warehousing Institute (see [70]) estimates that data quality problems cost US businesses more than 600 billion dollars a year. The findings of the report were based on interviews with industry experts, leading edge customers, and survey data from 647 respondents.

A frequent problem of data quality in organizations concerns the so-called customer matching problem. Information systems of public and private organizations can be seen as the result of a set of scarcely controlled and independent activities producing several databases which are very often characterized by overlapping information. In private organizations, such as marketing firms or banks, it is not surprising to have several (sometimes dozens!) customer registries, updated with different organizational procedures, resulting in inconsistent, duplicate information. The customer matching problem is indicative of the growing need to integrate information across completely different data sources, an activity in which poor quality hampers integration efforts.

Awareness of the importance of improving the quality of data is increasing in many public domains. In the public sector a number of initiatives address data quality issues at international, European, and national levels. Two of the main initiatives concern the Data Quality Act in the US and European directives on reuse of public data.

In 2001 the president of the USA signed into law important new data quality legislations, concerning "Guidelines for Ensuring and Maximizing the Quality, Objectivity, Utility, and Integrity of Information Disseminated by Federal Agencies," in short the Data Quality Act. The Office of Management and Budget (OMB) issued referred guidelines for policies and procedures on data quality issues (see [158]). Obligations mentioned in the guidelines concern agencies, which are to report periodically to the OMB regarding the number and nature of data quality complaints received and how such complaints were handled. OMB must also include a mechanism through which the public can petition agencies to correct information that does not meet the OMB standard. In the OMB guidelines data quality is defined as an encompassing term comprising utility, objectivity, and integrity. Objectivity is a measure to determine whether the disseminated information is accurate, reliable, and unbiased, and whether that information is presented in an accurate, clear, complete, and unbiased manner. Utility refers to the usefulness of the information for its anticipated purpose, by its intended audience. Integrity refers to the security of information, namely protection of the information from unauthorized, unanticipated, or unintentional modification, to prevent from being compromised by corruption or falsification. Specific risk-based, cost-effective policies are defined for assuring integrity.

The European directive 2003/98/CE on the reuse of public data (see [76]) highlights the importance of reusing the vast data assets owned by public agencies. The public sector collects, produces, and disseminates a wide range of information in many areas of activity, such as social, economic, geographical, meteorological, business, and educational information. Making public all generally available documents held by the public sector, concerning not only the political process but also

the legal and administrative processes, is considered a fundamental instrument for extending the right to information, which is a basic principle of democracy. Aspects of data quality addressed by such a directive are the accessibility of public data and availability in a format which is not dependent on the use of specific software. At the same time, a related and necessary step for public data reuse is to guarantee its quality in terms of accuracy and currency, through data cleaning campaigns. This makes it attractive to new potential users and customers.

2.2.1 Data Quality Dimensions

Data are normally considered to be of poor quality if typos are present or wrong values are associated with a concept instance, such as an erroneous birth date or age associated with a person. However, data quality is more than simply data accuracy. Other significant dimensions such as completeness, consistency, and currency are necessary in order to fully characterize the quality of data. In Fig. 7.4 we provide some examples of these dimensions. The relation in Fig. 2.1 describes movies, with title, director, year of production, number of remakes, and year of the last remake.

In the figure, the cells with data quality problems are shaded. At first, only the cell corresponding to the title of movie 3 seems to be affected by a data quality problem. In fact, there is a misspelling in the title, where `Rman` stands for `Roman`, thus causing an accuracy problem. Nevertheless, another accuracy problem is related to the exchange of the director between movies 1 and 2; `Weir` is actually the director of movie 2 and `Curtiz` the director of movie 1. Other data quality problems are a missing value for the director of movie 4, causing a completeness problem, and a 0 value for the number of remakes of movie 4, causing a currency problem because a remake of the movie has actually been proposed. Finally, there are two consistency problems: first, for movie 1, the value of `LastRemakeYear` cannot be lower than the value of `Year`; second, for movie 4 the value of `LastRemakeYear` cannot be different from null, because the value of `#Remakes` is 0.

Over 50 quality dimensions have been proposed in the literature, referring both to qualities of data schemas and to quality of data values. The most frequently mentioned concerns are as follows:

Id	Title	Director	Year	#Remakes	LastRemakeYear
1	Casablanca	Weir	1942	3	1940
2	Dead poets society	Curtiz	1989	0	NULL
3	Rman Holiday	Wylder	1953	0	NULL
4	Sabrina	null	1964	0	1985

Fig. 2.1 A relation `Movies` with data quality problems

1. *Accuracy* is defined as the closeness between a value v and a value v', considered as the correct representation of the real-life phenomenon that v aims to represent. For example, "Jon" is an inaccurate representation of the name "John."
2. *Completeness* is defined as the extent to which data are of sufficient breadth, depth, and scope for the task at hand. A null value in a data set is an example of incomplete data.
3. *Currency* concerns how promptly data are updated. A change of address of a business that is updated after 1 month in a business registry is an example of out of date data.
4. *Consistency* is the absence of any violation of a business rule in a database. In the relational model of data, any violation of referential integrity is an example of inconsistency.

A detailed list of DQ dimensions can be found in [23]; we suggest that the reader read this book for a thorough description of dimensions and proposed classifications.

2.2.2 A Methodology for Data Quality Assessment and Improvement

We define a data quality methodology as a set of guidelines and techniques that, starting from the input information concerning a set of databases, defines a rational process for using the information to measure and improve the quality of data of an organization through given phases and decision points. In [23] the methodology shown in Fig. 2.2 is proposed.

Phase 1: State reconstruction

1. Reconstruct the state and meaning of most relevant databases and data flows exchanged between organization structures and build the *database + dataflow/organization matrices.*
2. Reconstruct most relevant business processes performed by organizations and build the *processes/organizations matrix.*
3. For each process or group of processes, reconstruct the norms and organizational rules that discipline the processes and the services provided .

Phase 2: Assessment

4. Check with the internal and final users the major problems related with the services provided. Fix these drawbacks in terms of process and service qualities and identify the causes of the drawbacks due to low data quality.
5. Identify relevant DQ dimensions and metrics, measure data quality of databases and data flows, and identify their critical areas.

Phase 3: Choice of the optimal improvement process

6. For each database and data flow, fix the new DQ levels that improve process quality and reduce costs under a required threshold.
7. Conceive process re-engineering activities and choose DQ activities that may lead to DQ improvement target sets in step 6, relating them in the data/*activity matrix* to clusters of databases and data flows involved in DQ improvement targets.
8. Choose optimal techniques for the DQ activities.
9. Connect crossings in the *data/activity matrix* in reasonable candidate improvement processes
10. For each improvement process defined in the previous step, compute approximate costs and benefits and choose the optimal one, checking that the overall cost–benefit balance meets the targets of step 6.

Fig. 2.2 Phases and steps of a methodology for data quality assessment and improvement

The overall strategy of the methodology sees the measurement and improvement activities as being deeply related to the business processes, services provided by processes, and the costs of the organization due to low data quality. In phase 1 all the most important relationships between organization units, processes, services, and data if not known are reconstructed. Phase 2 sets new target quality dimensions that are needed to improve process qualities and evaluates cost savings and new benefits. Phase 3 identifies the optimal improvement process, i.e., the sequence of activities that has the optimal cost-effectiveness ratio.

We refer the reader again to [23] for a detailed description of the methodology and the discussion of a case study.

2.3 Data Modeling

In our life we frequently need to conceptualize the physical objects we perceive in everyday life. With the term "conceptualize" we mean the process of extracting and representing common features from physical objects pertaining to the same class, e.g., human beings are very different among populations, but are characterized by common physical and intellectual features, such as arms, eyes, and the ability to reflect and think.

The public administration makes intensive use of data to provide services. Databases are the most common technology we use to represent, manage, query data. Database management systems (DBMSs) are software applications that are built in functionalities to represent, manage, and query data. Every DBMS represents data in terms of a *logical model* that provides linguistic primitives to represent classes of data (e.g., the classes `Person`, `Business`, `City`, here and in the following we give singular names to classes of data) and relationships between them (e.g., the relationship `Born` that relates the class `Person` to the class `Municipality`). In the early years of data management programs were written using programming languages such as Cobol that uses a representation such as the one shown in Fig. 2.3 to represent classes of data. At a first glance it is quite difficult without knowing the technicalities of Cobol to understand that the Cobol representation describes two different classes of `Person`, namely `Man` and `Woman`, their `birth` relationship with the birthplace, represented as the administrative class `Municipality`, and the administrative/territorial relationship of municipalities with regions that are represented by the class `Region`.

As another example, relational DBMSs, the most frequently used ones, adopt the relational model which represents data in terms of tables, tuples, and columns. Also the relational model, though simple to understand, reveals unfit to represent classes of data and relationships among them. When we conceive classes of data which are used to provide services in eGovernment information systems, we need a model that is at the same time (i) simple to understand and (ii) rich of linguistic features that allow an intuitive description of classes of data and hide physical aspects of the computer implementation. Since the 80s of the past century, the model adopted

```
DATA DIVISION.
    WORKING-STORAGE SECTION.
    01      PERSON.
    05          MAN.
    10              MAN.-CODE     PIC X(5).
    10              MAN-DESCR     PIC X(80).
    05          WOMAN REDEFINES   UOMO.
    10              RECORD-TYPE   PIC X.
    10              WOMAN-CODE    PIC 9(5).
    10              WOMAN-DESCR   PIC X(80).
    01      AGENCY.
    05          REGION.
    10              REGION-CODE   PIC X(3).
    10              DUMMY         PIC X(6).
    10              REGION-DESCR  PIC X(80).
    005         MUNICIPALITY REDEFINES REGION.
    10              REGION-CODE   PIC X(3).
    10              MUNIC-CODE    PIC X(3).
    10              MUNIC-DESCR   PIC X(80).
    01      PERSON-ADMINISTRATION
    05          LINK.
    10              .CODE   PIC X(5).
    10              MUNIC-CODE    PIC X(3).
```

Fig. 2.3 A piece of a Cobol program

for this goal is the *entity – relationship model* (ER model in the following) that uses intuitive linguistic primitives for classes, properties of classes, relationships among classes. Furthermore, the ER model adopts a diagrammatic representation that provides even more intuitive flavor. Here we provide an example of schema represented in the ER model, while the next section is dedicated to a more detailed introduction to the model. In Fig. 2.4 we see the same piece of reality described in Fig. 2.3 represented using the ER model. We see the class `Person`, with a `Code`, a `Description`, and classified in terms of the classes `Man` and `Woman`; then we see `Municipality`, with a `Code` and a `Description`, and the relationship `Born` between `Person` and `Municipality`. Finally we see the class `Region`, again with a `Code` and a `Description` and its administrative/territorial relationship, called `Located in`, with `Municipality`.

2.3.1 The Entity – Relationship Model

We provide here a simplified description of the entity – relationship model, for a more comprehensive discussion see [21]. The ER model makes use of the following concepts to represent in an easy-to-understand way a reality of interest.

1. An *entity* is a class of things or events of the reality of interest having common properties, e.g., a `Person` is an entity in a registry of living persons having residence address in a `Municipality`. Things or events are also called *instances* of the entity. An *attribute* is an elementary property of an entity, e.g., a `Social Security Number`, a `Name`, a `BirthDate`. An *identifier* is an attribute or a set of attributes of an entity whose values uniquely identify a single instance of the entity, e.g., `Social Security Number` is an identifier of the entity `Person`, since every person has a specific `Social Security Number` that uniquely identifies the person.
2. A *relationship* is a set of facts relating instances of two entities, e.g., `Owns` defined among entities `Person` and `Car` describes for each person the cars he/she owns. Also the relationships may have attributes associated, e.g., the relationship `Owns` may have an attribute `StartDate` that, for each person and for each car owned, provides the start date of the ownership.
3. An *IS-A relation* is defined among two entities, Entity1 and Entity2 and expresses the property that every instance of `Entity2` is also an instance of `Entity1`, e.g., the IS-A relation among the entities `Person` and `Woman` expresses the property that all women are persons.
4. A *generalization* is defined among an entity `Entity0`, called ancestor, and a set of entities `Entity1`, `Entity2`,..., `Entityn`, called children, and expresses the property that each child entity is in the IS-A relation with the `Entity0`.

We call *ER schema* a set of interconnected entities, relationships, attributes, IS-A relations, and generalizations representing a reality of interest. The ER model is popular due to its intuitive and simple diagrammatic notation. In Fig. 2.5 we provide a notation that simplifies the one adopted in Fig. 2.4. We will use this notation in the following.

In Fig. 2.6 we reproduce the ER schema of Fig. 2.4 using the notation of Fig. 2.5. We have added some more attributes to the entities.

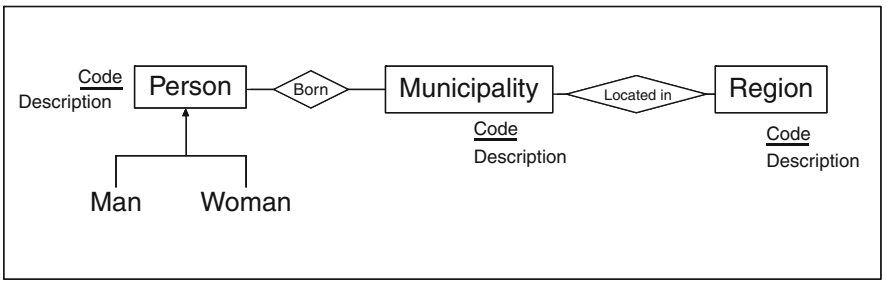

Fig. 2.4 An example of entity – relationship schema

Symbol	Concept represented
Person	Name of entity/attribute/relationship
Owns	Binary relationship
Person (SSN, Name, Last Name)	Attributes of an entity, identifier
	IS-A Relationship
	Generalization among entities

Fig. 2.5 A diagrammatic notation for the entity – relationship model

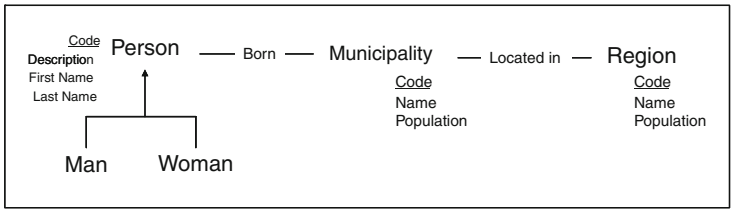

Fig. 2.6 The previous ER schema represented with the notation of Fig. 2.5

2.4 Schema Integration in the Small

The structure of public administration (PA) consists in many countries of central and local administrations that together interact with and offer services to citizens and businesses. For example, in Italy central PAs are of two types: ministries such as internal affairs, finance and other central agencies such as social security, accident insurance, and the chambers of commerce. Main types of local PAs correspond to regions (21 in Italy), provinces (about 100), and municipalities (about 8000). Each one of these administrations usually manages its own databases and registries. A crucial aspect in changing the relationship between PAs and citizens consists in the design of a new technological architecture that, contrary to the past, offers the services to citizens by means of a common front-office layer, on the basis of the one-stop shop paradigm; furthermore, a cooperative back-office layer has to be developed that allows administrations to share information and application services, in order to re-engineer the administrative procedures and reduce the burden to users. Concerning the data architecture, redundancies should be discovered and controlled, data have to be interchanged in an interoperable format, all the administrations have to assign the same meaning to the same data, achieving integration in the long term. To be able to

1. discover redundancies and heterogeneities among databases of different administrations;
2. reconcile the different meanings of data;
3. reuse entities in the design of new databases achieving semantic interoperability

a unified conceptual and reconciled description is needed of the different databases. In the following, we call such a description an *integrated schema*. Integration is the mechanism by which a set of local schemas is merged into a unique global schema, after solving all heterogeneities present in the input schemas.

In order to be able to reconcile the (usually) heterogeneous representations of data managed in databases of different administrations, we have to perform an activity of *schema integration*, whose goal is to homogenize two or more ER schemas and produce a reconciled representation of all the entities and relationships into a new schema called the *integrated schema*. Schema integration has different approaches when the number of schemas is small, say, less than 10, and in the case when the number of schemas is large. In the two cases we will use the terms *schema integration in the small* and *schema integration in the large*. Schema integration in the small is discussed in this section, while schema integration in the large is described in Sect. 2.5.

The activity of schema integration can be divided into three main steps: (i) conflict analysis, (ii) schema merging, and (iii) schema enrichment and rearrangement. We will describe the methodology for schema integration, considering as an example the land department inside a hypothetical ministry of finance. This department is in charge of the evaluation of real property in order to determine direct and indirect tax assessment and to issue real estate certifications. Moreover, this department administers and records all state properties in regard to their financial affairs. Its responsibilities include the acquisition of new state properties; the disposal of properties when authorized; the care and supervision of state properties; and the maintenance of an inclusive inventory. The above activities are in charge of two offices, the general land office and the state property office.

Seven databases are located within the information system of the land department, namely

1. General land office: urban database
2. General land office: land database
3. Mortgage registry database
4. State property office: real estate database
5. State property office: property grant database
6. State property office: confiscated private database
7. State property office: private estate renting database

Among them, in the following we consider the first three databases, whose schemas are described in Figs. 2.7, 2.8, and 2.9. Notice that we represent only entities, relationships, IS-A relations, and generalizations; we do not represent identifiers, attributes, and names of relationships. Only in the case of urban schema we

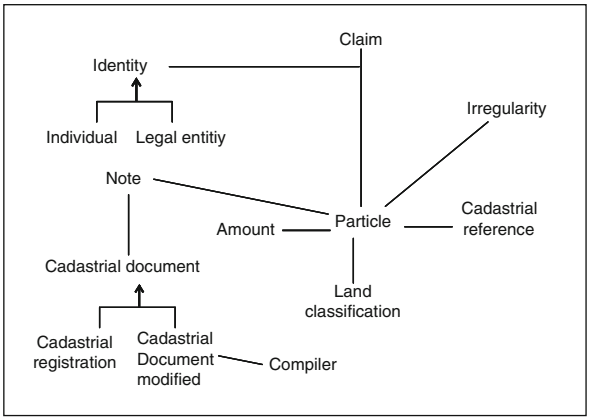

Fig. 2.7 The land schema

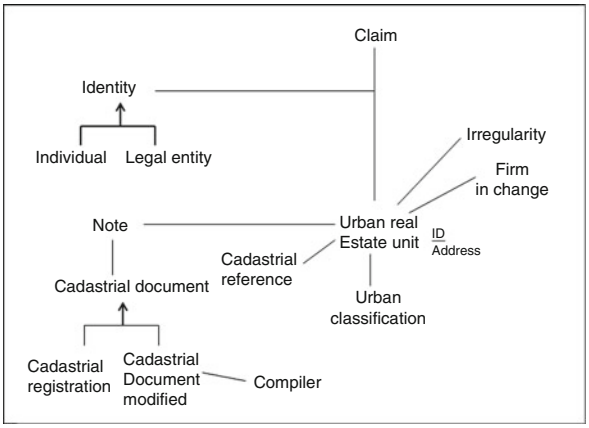

Fig. 2.8 The urban schema

represent for the `Urban real estate unit` entity two attributes, the identifier `Id` and the attribute `Address`.

2.4.1 Conflict Analysis and Schema Merging

The aim of the conflict analysis and schema merging step is to discover and solve every type of conflict among data representations in different schemas. Two main activities may be distinguished:

1. *Name conflict analysis*, to establish naming correspondences for concepts. There are basically two sources of name conflicts: synonyms and homonyms. *Synonyms* occur when schema objects with different names represent the same concept

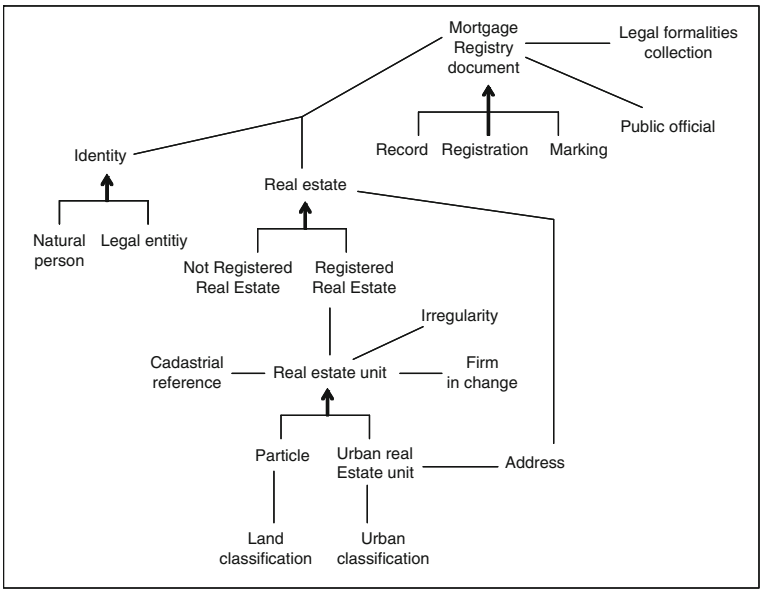

Fig. 2.9 The mortgage registry schema

while *homonyms* occur when the names are the same, but different concepts are represented. Therefore, whenever synonyms or homonyms are detected, a concept renaming is required to solve the conflict.

2. *Structural conflict analysis*, to discover conflicts between different representations of the same concept. The use of an entity and an attribute to represent the same concept in two different schemas is a typical example of structural conflict. Each difference in representing the same reality can be solved by applying an *equivalence transformation* (a transformation which does not change the schema information content) to the schemas involved. At the end of this stage, we obtain a set of amended schemas that can be syntactically integrated, all the name and structural conflicts having been solved.

In our case study we have the following conflicts.

1. A synonym among `Individual` in the `Urban` and `Land` schemas and `Natural person` in the `Mortgage` schema; we choose the term `Individual` and consequently amend the `Mortgage` schema.
2. A structural conflict between the attribute `Address` in the `Mortgage` schema and the entity `Address` in the `Urban` schema; we choose for `Address` the entity type and consequently transform the attribute `Address` in the `Mortgage` schema into an entity.

The activity required at this point, called *schema merging*, is a simple superimposition of common concepts belonging to the amended schemas, thus building the integrated schema.

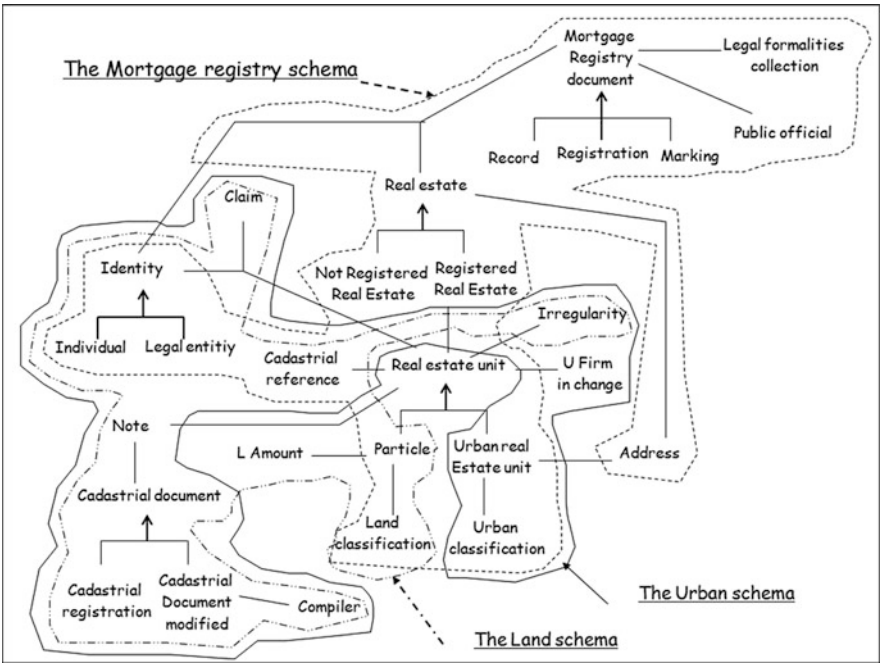

Fig. 2.10 The integrated schema together with the three input schemas

2.4.2 Enrichment and Rearrangement

This phase aims to detect *interschema properties*, corresponding to redundancies and cycles within the global schema in order to build the final *integrated schema*; with the term "interschema property" we mean mutual constraints between concepts appearing in different schemas. Due to their cross-schema nature, in fact, these relationships have not yet been represented in the global schema and therefore require a specific analysis at this point. In the case study, analyzing the `Urban` schema and the `Land` schema it is easy to discover that the entity `Particle` in the `Land` schema and `Urban real estate unit` in the `Urban` schema have many related concepts in common, so they are concepts which are in an IS-A relation with a common generalized concept, whose name can be `Real estate Unit`. At the end of the integration activity we obtain the integrated schema, represented in Fig. 2.10 together with the three amended schemas, identified by differently shaped closed lines.

2.5 Schema Integration in the Large: The Repository of Schemas

When the number of schemas to be integrated is high, building the integrated schema becomes unfeasible, as Fig. 2.11 expresses in graphical/metaphorical terms. Reconsidering the integrated schema of Fig. 2.10 we perceive that when a schema has

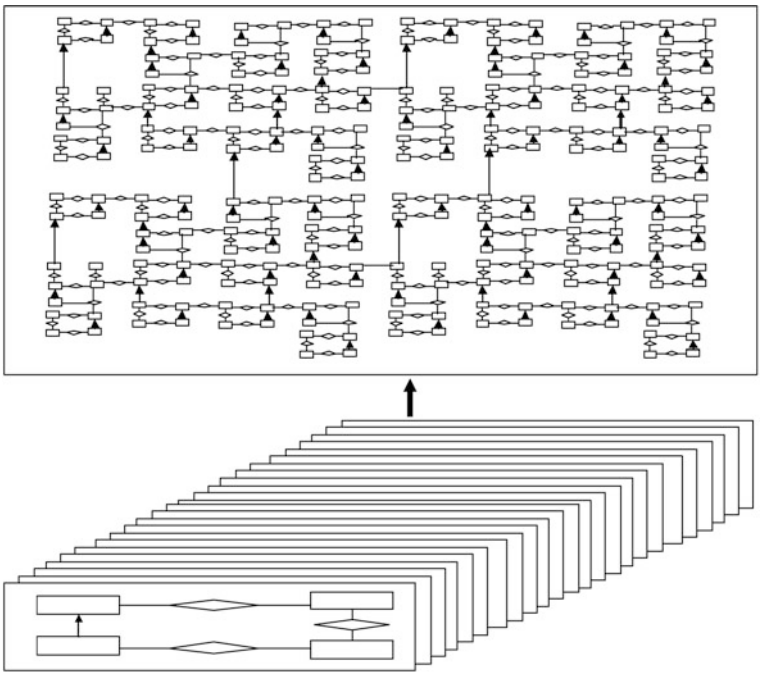

Fig. 2.11 Unfeasibility of the integration of a large number of schemas

more than 20/30 entities, it becomes difficult to perceive at a glance the *semantics* of the reality represented, namely the meaning of concepts and of relationships among them.

When the reality becomes complex, besides integration we have to adopt another paradigm that we call *abstraction* (see also [20]). Given a schema S, we define abstraction of S a new schema obtained from S, clustering and collapsing groups of concepts into a unique entity. The new schema, in a sense, describes the same reality of S with a more concise representation. Given a schema S, we may iterate the use of abstractions producing schemas that describe the same reality at different levels, from detailed to abstract ones. We will call *refinement* the inverse primitive that allows to proceed from abstract representations to more detailed ones.

We call *repository of schemas*:

1. A set of ER schemas, representing the set of databases of an organization, one schema for each database. We will call these schemas *basic schemas*.
2. A set of ER schemas obtained from basic schemas by iterative joint usage of the two integration and abstraction primitives.

In Fig. 2.12 we show an example of repository, where in the bottom row in the second, third, and last columns the `Production`, `Sales`, `Department` basic schemas of a production and sales organization are represented. The `Company`

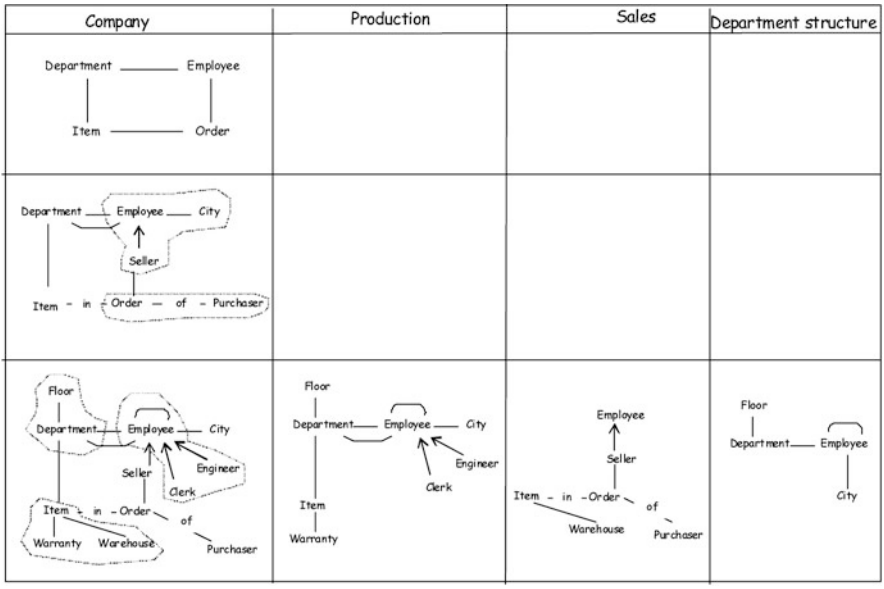

Fig. 2.12 An example of repository

schema in the bottom row, in the first column, is the result of their integration. For the `Company` schema we show in the second and first rows two abstractions. In the first column groups of concepts abstracted at the upper level into single concepts are highlighted in closed lines (see next section).

We now have to provide more detail on the abstraction step (Sect. 2.5.1); furthermore, putting together all issues introduced so far, in Sect. 2.5.2 we discuss a methodology for building a repository of schemas.

2.5.1 Schema Abstraction

In Fig. 2.13 we show a methodology for producing an abstract schema.

We apply the methodology to the schema represented in the bottom of Fig. 2.14. Typical candidates for groups of concepts to be abstracted are generalization hierarchies, sometimes alone (as in the case of `Identity` (ancestor), `Individual`, `Legal entity`), other times together with less relevant concepts related to the generalization (as in the case of the group `Cadastral document` (ancestor), `Cadastral registration` and `Cadastral document modified`), to which the `Compiler` entity is added due to its unique relationship with the generalization.

Another group concerns two relationships, among respectively: `Urban Real Estate Unit` and `Urban Classification`, and `Urban Real Estate Unit` and `Address`. In this case, the most relevant concept is clearly `Urban Real Estate Unit` that is chosen as the abstract concept in the upper level abstract

1. Group concepts in the schema S according to the following rules:
   - Each group is made up of a small number of entities, relationships, IS-A relations, and generalizations
   - Concepts in each group have a "strongly related" meaning, while they have a looser relationship w.r.t. entities in other groups.
   - The set of groups covers the whole schema.
2. Associate to each group a unique abstract entity or abstract relationship, whose name represents the whole information content of the group in an abstract way.
3. Link abstract entities or relationships resulting from step 2 with the relationships that link in S the corresponding groups.

Fig. 2.13 A methodology for producing an abstract schema

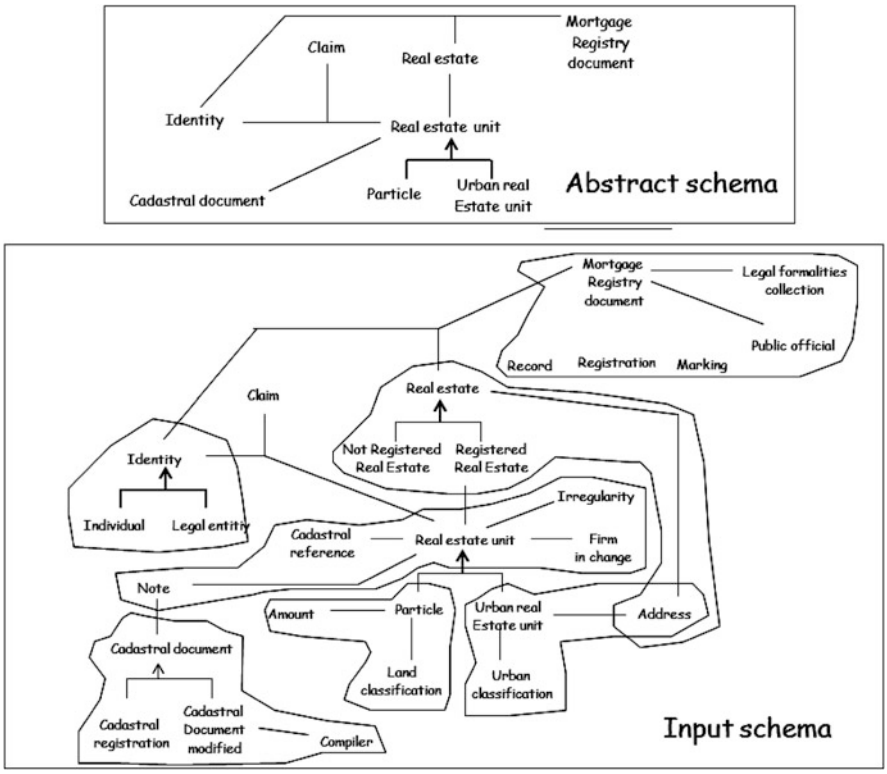

Fig. 2.14 A schema and one possible abstraction

schema. In previous cases the ancestors in generalizations were chosen as abstract concepts.

Notice that the abstraction process is subjective, and it is not easy to provide quality criteria that distinguish between a good abstraction and a bad abstraction. In our example it is not evident why we leave the entity `Claim` untouched in the process. Groups made of a single entity have to be avoided, since their presence tends to create an asymmetry in the balancing of concepts in groups. In our case, we could adjoin `Claim` to one of the adjacent groups.

2.5.2 A Methodology for the Construction of a Repository of Schemas

A methodology for the construction of a repository of conceptual schemas is described in Fig. 2.15 (see also [20] for a more comprehensive discussion). In the case of a repository made of dozens or hundreds of schemas, schemas have first to be clustered; the clustering activity can be performed putting together schemas that pertain to the same topic, e.g., finance, internal affairs, justice. As a second step, schemas in the same cluster are integrated producing a unique integrated schema. At this point integrated schemas are abstracted and the process of clustering/integration/abstraction proceeds until a unique schema is produced. Notice that when the group of schemas to be integrated is made of a huge number of concepts, resulting in a complex integrated schema, the two integration and abstraction steps can be applied in the inverse order.

We apply the methodology to the land office case study. Initially we populate the bottom level of the repository with the basic `Land`, `Urban`, and `Mortgage registry` schemas. Then we generate the integrated schema.

At this point we may abstract the integrated schema at two different levels, leading in the end to a six-entity schema, compact enough to conclude the abstraction procedure (see Fig. 2.16).

1. Produce basic schemas.
2. Cluster schemas in groups, using areas of interest for choosing clusters
3. For each cluster of schemas, produce an integrated/abstract schema through
   - 3.1 Integration
     - Perform integration activities on the schemas in the cluster
   - 3.2. Abstraction
     - Perform abstraction activities on the integrated schemas

   Until a unique abstract schema is obtained

Fig. 2.15 A methodology for the construction of a repository of conceptual schemas

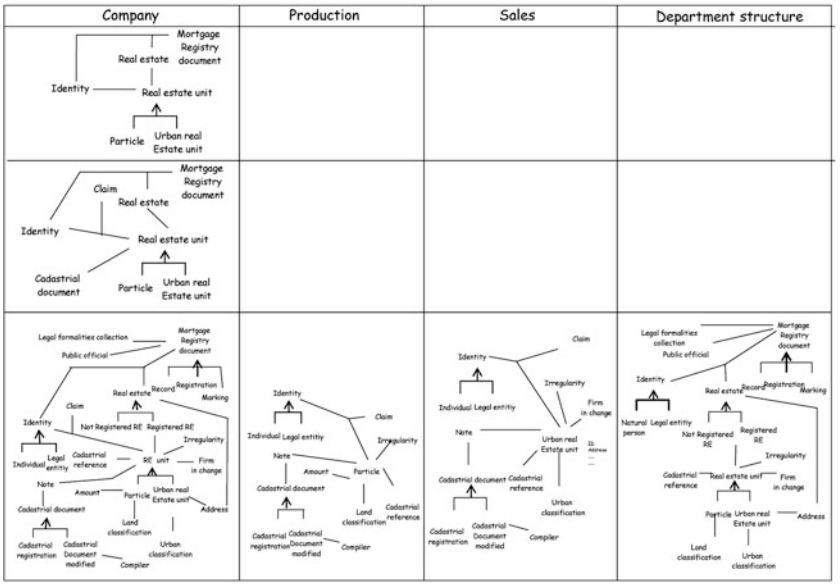

Fig. 2.16 The repository of the land office

2.5.3 Usages of the Repository of Schemas in eG4M Planning Activities

In this section we discuss several analyses that can be made on the repository of schemas that provide useful insight for planning activities and for defining joint eGovernment projects in groups of administrations.

The analyses we show have been performed on a repository made of about 300 databases managed in the Italian Central PA (*central PA repository* in the following); the repository has been produced using the methodologies for schema integration, schema abstraction, and the methodology for repository structuring described above. In order to build the repository, about 200 person-months were needed to first produce the 300 basic ER schemas, while about 24 person-months were needed to produce the 59 abstract schemas of the upper part of the repository (approximately 14 person-days per schema, both for the basic and for the abstract schemas).

The analyses performed on the central PA repository led to significant planning decisions and to the conception of innovative projects that significantly improved the relationships among PAs and citizens and businesses.

2.5.3.1 Choosing Priorities and Planning New Initiatives

The repository provides useful knowledge on the information resource to define priorities in eGovernment projects, e.g., at a very high level of analysis we may

Area	Subarea	Min of foreign affairs	Foreign trades	De-fence	Reve-nues	Justice	Internal affairs	Cultural heritage	Com-merce and trades	Wel-fare	Edu-cation	Agri-culture	Health	Trea-sury	Transpor-tations	Re-search	Total	%
Resources	Financial	21	19	8	345		82	51		5		40		145			716	20
	Real estate				68		59							86			213	6
	Support				25		28							3			56	2
	Human	67	16		102	6	136	12		11				127	14		491	14
Total resources		88	35	8	540	6	305	63		16		40		361	14		1476	41
Services	Direct	78		24		203	149							93			547	15
	Economic		156				24		55	107		70		0	84		496	14
	General				66		143		27					0	84		320	9
	Social	14					40	153			120	93	204	0		116	740	21
Total services		92	156	24	66	203	356	153	82	107	120	163	204	93	168	116	2103	59
Grand total		180	191	32	606	209	661	216	82	123	120	203	204	454	182	116	3579	100
%		5	5	1	17	6	18	6	2	3	3	6	6	13	5	3	100	

Fig. 2.17 Macroareas and corresponding number of entities of schemas owned by a set of central administrations

evaluate the distribution of entities among the different areas of interest and among the different administrations which are owners of data. In Fig. 2.17 we show such a distribution for the two service and resource macroareas and the related areas of interest.

We see that the 50% of the entities are concentrated in three agencies, namely finance, treasury, and internal affairs, while referring to areas, over 40% of entities are managed for resource-related support processes, while less than 60% of entities are used for processes that produce services to citizens; this means that for every five employees, two of the five work for managing themselves and the other three for the employees! Such distribution increases the costs of public administration and reduces the effectiveness of its mission in providing services to users. Similar figures can be produced for the distribution of instances. On the basis of such figures, several projects can be conceived and set up to balance this unequal distribution of information.

A second analysis concerns the redundancy related to managing the same entity in different schemas and administrations; in such analysis we may initially focus on macro-entities of interest in public administration that are `Person` and `Business`. Focusing on `Business`, in Fig. 2.18 we represent attributes associated with the `Business` entity in common with the three agencies that own national registries on businesses, namely chambers of commerce, social insurance, and social security. Due to such overlapping, common attributes regarding any particular business are likely to be duplicated, with no guaranteed consistency among the copies. A project that can be launched concerns the coordination of updates, choosing, e.g., to update only one reference database and subsequently to coordinate the updates to be performed in other databases with an asynchronous communication between the reference and the other databases.

Furthermore, high costs for agencies and for businesses are related to the multiple updates. In [23] a project setup to tackle the above issues is described, showing that in the new coordinated ICT architecture overall costs for agencies and businesses can be reduced yearly by approximately 200 million euros.

Attributes	Chambers of commerce	Social security	Social insurance
Fiscal code	X	X	X
Vat number	X	X	X
Name	X	X	X
Company deed of partnership	X		X
Activity	X		X
Legal status	X		X
Registered office	X	X	X
Code in the national registry	X		X
Registration date	X		X
Company structure	X	X	X
Administrative office	X	X	X
Address	X	X	X
Start date of activity	X	X	X
Suspension date of activity	X	X	X
End date of activity	X	X	X
Number of workers	X	X	X

Fig. 2.18 Common attributes of the entity company among different administrations

2.5.3.2 Coverage Analysis

Public administration in its relationship with citizens exercises a different degree of attention as to different types of individuals, such as workers, retired persons, emigrants, immigrants. In Fig. 2.19 we show an analysis of the instances of entities referring to individuals, state employees, retired persons, and students, compared with the corresponding size of the universe in Italy, as results from the national bureau of census statistical tables.

The comparison puts in evidence uneven coverage between the four categories, e.g., students are neglected, despite a much greater availability of information (in terms of instances) for public employees. These analyses can also be used for choosing priority areas to focus eGovernment projects on.

Concept observed	# of instances	Size of the universe
Individual	250,000,000	58,000,000
State employee	2,500,000	1,700,000
Retired person	4,900,000	10,500,000
Student	240,000	3,500,000

Fig. 2.19 Instances represented in databases and sizes of the real universe for several relevant entities in the repository of schemas

Type of identifier	Number	Percentage
Fiscal code	124	25
Other non-standard identifiers	370	75

Fig. 2.20 Common standard identifiers and other identifiers of individuals

2.5.3.3 Reconciliation of Identifiers and Knowledge Potential

The knowledge represented in the information systems of public administrations is huge, but is fragmented in databases managed by different administrations. The possibility of integrating the different databases and retrieving and joining related data are enabled by having common identifiers defined in the different databases. An analysis on the repository has shown (see Fig. 2.20 referring to individuals) that among all the identifiers of individuals and companies only 25% standard identifiers such as fiscal code are for individuals and VAT code for companies.

2.6 Summary

Languages, models, tools, and methodologies introduced in this chapter can be used in eGovernment projects for several purposes, namely to improve the quality of data and consequently, of administrative processes, to understand more clearly the types of data managed in an information system, to analyze the redundancies and overlapping existing between different databases, to reconcile heterogeneous databases, to launch projects for shared usage of databases managed by different administrations, in a word, to govern data, the most important resource managed in public administration. Further usages will be shown in the following chapters.