REPRINTED FROM:

journal of information science principles & practice

Volume 15, No. 6, 1989

Towards data modelling in information retrieval

Maristella Agosti, Fabio Crestani and Girolamo Gradenigo
Dipartimento di Elettronica e Informatica, Università di Padova, Via Gradenigo 6/A, 35131 Padova, Italy

pp. 307–319

published for the institute of information scientists by
ELSEVIER

Towards data modelling in information retrieval

Maristella Agosti, Fabio Crestani
and Girolamo Gradenigo
Dipartimento di Elettronica e Informatica, Università di Padova, Via Gradenigo 6/A, 35131 Padova, Italy

Received 31 October 1988
Revised 21 April 1989

The diffusion of automated information retrieval (IR) applications and of applications which share many of the peculiarities of IR applications is increasing. The demand for these types of applications draws attention to the lack of methodologies available for IR data design. The data modelling of an IR application, that is, the data modelling of the IR part of a complete information system, has not yet been approached and studied as a complete process and in a structured way.

The first part of this paper introduces the motivations and the scope of the DIRD (Design of Information Retrieval Data) project, which is devoted to the development of an environment for the design of advanced applications of IR.

The second part of the paper addresses the design of the IR part (or IR application) of an information system. The conceptual paradigm necessary for the design of advanced IR applications is investigated. The Entity Relationship (ER) approach is compared with that conceptual modelling paradigm and is examined as a candidate data model for the conceptual design of IR data. A new ER approach is then introduced: this new approach extends the constructs of the ER model to manage the complexity of IR data. Two design examples are given to present the use of the new ER approach in designing IR data.

1 Introduction

The information retrieval (IR) part (or IR application) of an information system is the part concerned with the management of a databank of unformatted data; an IR application is related to the storage, management and associative retrieval of weakly structured and unstructured data.

In IR applications the objects handled are usually termed "documents," where by document we mean a unit of information such as a book, report or letter (all textual media), or an image or drawing (both visual media).

The software tool which automatically manages the descriptions of the information content of the documents is usually identified as an information retrieval system (IRS). The description of the content of a document does not derive from a deterministic process, because the content can be seen and described in different ways by different people and must reflect different users' information requirements. The descriptions are created by an indexing procedure and are managed by an IRS as subsidiary data to the complete information on the documents; they are variously known in the literature as document representation, auxiliary data and paradata [1]; in this paper they will be called IR data.

The IRS is one of many different information processing systems which manage data; a diagrammatic representation of the spectrum of information processing systems is given in Fig. 1. In this diagram, the systems are clustered by nature of the data they manage. The systems managing a prevalent proportion of structured data are towards the left of the spectrum, while systems managing a prevalent proportion of document descriptions are towards the right. The systems on the left of the scale can be characterized by restricted input selection criteria, fixed-format records and by the importance of fast access to the data to get quick answers. In contrast, the systems on the right are generally less response-time critical, but are able to retrieve all documents which might be pertinent to a user's query. They also need to be flexible, as they essentially manage variable format records of descriptive information.

For the design of the data to be managed by the systems on the left of the spectrum, different data models have been studied and developed which can be effectively used during the design process. The lack of a similar design approach for the data managed by the systems on the right of the spectrum was the reason for starting this work. The authors asked themselves whether data modelling, widely used for the design of the databases managed by a database management system (DBMS), can be effectively applied to the design of IR applications; the paper addresses this crucial problem. Section 2 explains the motivation for introducing a design methodology for IR applications and introduces the scope of the DIRD (De-

Journal of Information Science 15 (1989) 307–319
Elsevier

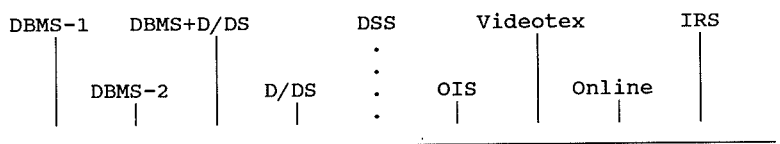

Fig. 1. Spectrum of information processing systems (source: [1]).

Key:

DBMS-1	indicates database management systems (DBMS) with two levels of data representation; external and internal levels;
DBMS-2	implies DBMS with three levels of data representation; external, conceptual and internal;
DBMS+D/DS	are the DBMS which permit the development of the database by means of a data dictionary system (D/DS); and the D/DS can be consulted by the users;
D/DS	indicates the data dictionary/directory systems that are used in the enterprise independently from a DBMS, for documentation purposes;
DSS	stands for decision support systems;
OIS	is used for automated office information systems;
Videotex	indicates all the different viewdata systems;
Online	stands for the online bibliographic databases, namely the commercial systems which manage bibliographic references;
IRS	stands for automatic information retrieval systems.

sign of Information Retrieval Data) project devoted to the development of an environment for IR data design. Section 3 addresses the design of the IR part of an information system. The conceptual paradigm necessary for the design of IR applications is investigated. In section 4, the Entity Relationship (ER) approach is compared with that conceptual paradigm and is examined as a candidate modelling paradigm for the conceptual design of IR applications. Two examples are given in section 5.

2 Motivations

The design of IR data has not yet been approached and studied as a complete process and in a structured way, as it has in the database area. Furthermore, suitable design tools for the design of IR data have not yet been developed. The result of this is that the designer of an IR application cannot call upon any complete and proven design methodology or development environment with specification languages and prototyping tools.

The authors' experience in the design of IR applications has shown that it is necessary to use suitable methodological tools if all the user's requirements are to be met. It is common in actual applications to have more than one type of user—e.g. an expert user and an end-user. For different types of user it is necessary to adopt different views of the application, representing different levels of abstraction. These views must be represented by different design tools, called a "data model" in the database management area. The use of a data model in the representation of the application gives a schema which clearly represents the design decisions to the expert user and the end-user. Also, for the IR application it is necessary to have a schema representing the IR data that are going to be managed by an IRS. The design tools can correspond to different underlying methodological approaches (e.g. the ER approach, relational algebra, logic, etc.). Different methodological approaches can be implemented with different data model tools.

These motivations led to the starting in 1987 of the DIRD project at the Department of Electronics and Informatics of the University of Padua. The aim of this project is the development of an environment for the design of information retrieval applications. Models and methodologies are studied for the design of IR data.

The project is being developed in different phases. In [3] an initial proposal for the requirements design has been made and the problem of the conceptual design broached. The project also addresses the dynamic aspects of the design of IR applications; some initial findings, related to work in the office information systems area [9,10], have been presented in [4].

3 The design of IR applications

To introduce the design methodology of IR applications it is useful to recall the experience and knowledge gained in the database design process (see for example [2] for an introductory bibliography of the literature on the subject).

The database design process is used to transform and organize unstructured information and processing requirements concerning the application, through different intermediate representations, to a complex representation which defines schemas and functional specifications. The design process is usually divided into components which produce intermediate representations.

The process is divided into four components by many experts in database design:

(1) information requirements analysis and design;
(2) conceptual database design;
(3) logical (or implementation) database design; and
(4) physical database design.

The divisions are useful for managing the process and permitting the exchange of information and intermediate results between the designers and users involved in the design. Errors made during the design process can affect the entire life of the application. The process is therefore of great importance to the enterprise and it is necessary to pay close attention to it. In this paper it is shown that the design of IR data can be approached in a similar way to design in the database management area, but important modifications have to be made, to take into account the peculiarities of IR data (see Fig. 2).

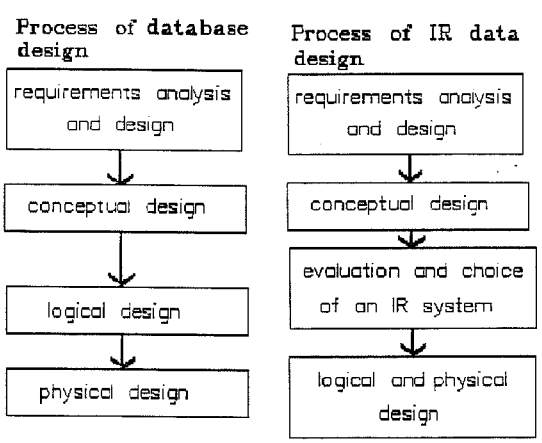

Fig 2. Database and IR data design process.

Components 1 and 2 of the database design process are independent of the software system which is going to be used for the implementation of the application. These two components are concerned with the representations of the user's general requirements. These two steps are usually neither well developed nor well designed in IR practice. Generally, the design of IR applications starts from component 3, when an IR system has been chosen for its processing characteristics and not for its ability to meet the user's requirements.

Some of the reasons for introducing the information requirements and conceptual application design steps in IR application design practice are:

- the application is studied independently of any software tool; the data and the relationships among data are studied independently of any logical or physical representation and implementation;
- the user's requirements play the central role in the application design;
- the user's requirements can be transformed into a conceptual representation (a conceptual schema), which provides a unique central description of the various information contents that may be managed by the application—in fact the conceptual schema is obtained through the consolidation of the user's information requirements specifications;
- the availability of a conceptual schema of the application allows the software tool adopted to be changed, if necessary, without the need to redesign the conceptual level of the application.

The implementation and physical design steps of an IR application depend on the type of IRS available. The main difference between the IRS and the DBMS which are currently available is that the IRS lack an explicitly defined data model which can be used for the description and manipulation of the data that will be managed by the system.

The IRS available on the market which can be effectively used for the implementation of real applications fall into two categories: the traditional IRS based on an inverted file architecture [16]; and DBMS which integrate IR functions (often referred to as a database management information retrieval system—DBMIRS). The mapping from the conceptual component to the logical component is beyond the scope of this paper. It

seems important to note here that the mapping from a conceptual component to a DBMIRS can be developed with the extension of available design tools because the present DBMIRS is based on extensions of the relational model which allow the relational model to manage unstructured data and complex objects [8,14,17]. In any case, the extensions of the relational model cannot solve the problems that are addressed in this work.

The introduction of components 1 and 2 in the design of IR applications requires the identification and the development of appropriate tools.

3.1 *Information requirements analysis and design*

The information requirements analysis and design is the first component of the application design. It is the interface between the analysis process and the design process, and represents the process of mapping analysis into design. It leads to the specification of the design requirements for the relevant part of the information system to meet the information requirements of the users.

The information requirements design can be conducted with the support of the relevant part of an information system design methodology. Practical experience has been gained in the DIRD project with the information requirements analysis and design part of the ISAC methodology [15]. The shortcoming that the designer has to face in using one of these methodologies for the design of IR data is that they have been developed for the representation of "objective" and structured data. That is, when the relevant data of the application are identified they have to be represented with a deterministic process. The IR data are identified through a non-deterministic process. A simple example can clarify this concept: let us assume that the documents which are going to be managed with an IR application are an engineering company's projects. The description of the information content of one of these projects is different if it has to meet the requirements of a professional engineer or those of a draughtsman. In this case, more than one description of the same project needs to be managed to meet users' requirements. The description of the information content of a project can also be different for two different users in the same category. For example, a power station project can be described in a different way by a civil engineer and by an electrical engineer.

The requirements design of IR applications has to be more detailed to consider and incorporate information on IR peculiarities. In fact, the requirements design of structured data can be conducted as it is usually done for common database applications. The design of unstructured data requires:

- the identification of the characteristics of the application objects which have to be considered as documents, that is, the objects that the user can retrieve through descriptions of their information contents; and
- different descriptions able to meet different users' category profiles.

This supplementary information can be represented in any form which is suitable for integration with the adopted design method.

3.2 *The conceptual modelling paradigm*

The conceptual design of IR data is more problematic than the analysis and design of the information requirements. In fact to conduct the conceptual design it is necessary to have available a conceptual modelling paradigm with the abstraction mechanisms necessary for the classification of the IR objects and for the design of the relationships among the IR objects. The authors' experience in the design of IR applications [3,4] shows the need for a conceptual design paradigm which includes certain kinds of abstraction. It seems important to discuss here all the necessary conceptual mechanisms of such a conceptual paradigm.

The abstraction mechanisms found necessary are:

(1) the classification mechanism;
(2) the generalization/specialization mechanism [19];
(3) the aggregation mechanism [18]; and
(4) the part hierarchy mechanism.

These four mechanisms are illustrated using the following examples, which are pertinent to an IR application.

(1) The *classification mechanism* is a fundamental and intuitive mechanism. The identification of the document types and of the other application object types as entities is done by applying the classification mechanism. With this mechanism, the application designer identifies the class of objects which are to be taken into consideration in the automation of the application. An

entity type and its instances are related with an "instance of" relation. An example of the classification mechanism is the construction of a document class from single document instances.

(2) The *generalization / specialization mechanism* is the mechanism which relates a set to its subsets or a class to its subclasses. The representation of the generalization/specialization mechanism is usually given by a "subset of" or "subclass of" relation.

An example can be given by considering the relations between documents of a collection in the computing science area and the subclasses of documents that can be created by classifying the documents of the collection into classes of different computing science topics. A classification scheme of this sort can be seen as a mechanism which builds a hierarchical tree structure of subclasses of documents, which inherit all the characteristics of the document class of the previous level of the tree but which also have some characteristics which could not have been inherited; each subclass has one or more distinctive, added characteristics which make its documents distinguishable from the document of the class.

The simplest relation among a class and its subclasses is hierarchical, but a conceptual paradigm for the design of IR data must have an abstraction mechanism powerful enough for the representation of network relations among a class and its subclasses. In fact, it is possible to extend the previous example to show the necessity of network relations among a class and subclasses. It is quite common to classify specialized collections of documents using two or more classification schemes: for a computing science collection it is common to see the documents classified by a specialized classification scheme such as the Computing Reviews Classification System, and by a general classification scheme such as the Dewey Classification scheme. When all documents are classified by the two schemes, a network structure of classes and subclasses is created.

(3) The *aggregation mechanism* [18] relates objects of the same or different types, transforming a relationship between several objects into a higher level single object. This new object can have specific individual characteristics. The aggregation abstraction mechanism is used to model user-perceived relations between objects.

(4) The *part hierarchy mechanism* is a relevant mechanism for the conceptual design of IR data. This mechanism relates different types of objects with an assembly or composition relationship to produce a new assembled object. The new object is a whole and not the simple sum of its parts, so the characteristics of the whole which are produced by the assemblage cannot be derived from the characteristics of the single parts. An example in the IR situation is an application where each IR document is the complete documentation of an engineering project. The documentation is comprised of a set of drawings, a set of photographs and a technical report. The drawing type, the photograph type and the report type are all parts of the project type; it is important to note that the project type is not the simple aggregation or sum of its parts, but has some individual characteristics that are not present in its three parts, such as, for example, a textual description of the project history. As this example shows, the abstraction mechanism models different situations from those modelled by the aggregation mechanism, but it is not usually supported by the conceptual data models.

The above presentation of data modelling requirements is a framework which permits the evaluation of existing conceptual data models as tools for the conceptual design process of IR data.

3.3 *The conceptual design*

The conceptual data model which has been taken into consideration for the design of IR applications is the ER model [11]. The constructs of the ER model are semantically powerful and it has been developed as a technique for producing ER diagrams, to enhance the communication between the application designer and the user. The constructs of the ER model and the associated diagrammatic technique constitute a complete design approach called the ER approach [11,12]. This approach has been shown to be sound for the conceptual design of database applications, mainly because of its basic simplicity [7].

The fundamental constructs of the ER model are the entity and the relationship. An entity is a class of (real or conceptual) objects or individuals which share common characteristics; an entity is identified through the classification mechanism. Different entities can be related in the reality which has to be designed, so it is necessary to have

a construct to express the relationship between different entities: this construct is called a relationship in the ER model. The relationship is an association between two or more entities. An entity can be connected with itself, with another entity or with more entities.

In an ER diagram, an entity type is represented by a rectangular box and a relationship type is represented by a diamond box. A name is associated with each rectangle and with each diamond to identify respectively the corresponding entity and relationship. Since the entity and the relationship are the two main constructs of the model, the reason for the name of the model is evident.

The reality can be represented with the ER model as a collection of entities which are connected together through a network of relationships.

Entities and relationships may have properties which need to be taken into consideration during the design process and which have to be represented in the conceptual schema. These properties are expressed in terms of attribute/value-set pairs. In an ER diagram the value sets are expressed by arcs. When an attribute has the same name as the value set, the name of the attribute is omitted to simplify the diagram.

Many extensions have been proposed to the original ER approach to make it more powerful in capturing the semantics of the information managed by the applications. The extensions which have introduced the category concept to represent generalization hierarchies and subclasses [13,20] are fundamental to the use of the approach in the design of IR applications, as is shown in the examples in section 5. Nevertheless, the ER approach which incorporates these extensions is not yet adequate for the design of IR applications, although experience has been gained in its use as an abstract tool for the design of applications of information retrieval [3,4]. The use of the ER approach has shown that it lacks the semantic power to represent the unstructured data of IR applications, mainly for two reasons:

(1) the model does not support a part hierarchy; and
(2) the value sets are pre-defined and the model does not support the definition of new data types; that is, the model does not have type mechanisms for the construction of new data types.

A new use of the ER approach is considered here, in an effort to enhance its ability to represent data in the IR application domain. The findings obtained through the development of the ER approach to make it suitable as a conceptual design tool of IR applications are presented in the rest of the paper.

4 The ER approach

In the ER approach, an entity is a (real or conceptual) object or event which can be distinctly identified through the values of its attributes [11,12]. For IR applications, the entity which carries the IR peculiarities is the entity "document," and a document has the characteristics previously defined in section 1 of this paper. But an IR document cannot be identified through key attributes; the identification of a document has to be possible independently of the values of some specific attributes.

An attribute is a key attribute of an entity when the entity which carries the attribute can be identified unequivocally through the value of the key attribute. The constraint of the key attribute is too limiting for the IR objects; it seems necessary to remove this constraint to be able to model the IR objects.

An IR document is a real object which can be represented as an ER entity and which can be connected to other entities through relationships. In an application it is necessary to represent the characteristics of each type of object; in fact an application has to support the access to each object type through its particular characteristics.

In the ER approach the characteristics of the entity DOCUMENT can only be expressed in terms of attribute/value-set pairs. If the DOCUMENT entity represents technical reports, some value sets can be: AUTHOR-NAME, TITLE, NUMBER-OF-PAGES, PUBLICATION-YEAR. But the crucial property of the entity DOCUMENT is its information content. As has been shown in section 3.1, the requirements of the different users can require the management of different descriptions of the information content of a DOCUMENT. The ER approach has only a pre-defined (or built-in) set of value sets; in this pre-defined set there is not a value set that can be used for the representation of the information

content of an IR document. To represent the information content of a DOCUMENT it is necessary to consider the information content as an entity itself and not as an attribute, because for an IR system the managed document is not the document itself but a token object which is the description of the information content of the document.

The concept of an entity in the ER approach needs to be extended to make the entity concept more powerful, so as to be able to identify the information content of a document as an entity. In fact, the information content cannot be identified through a deterministic process, but has to be distinctly identified for different categories of users. The identification of the information content as an entity permits distinct properties to be associated with it.

The INFORMATION-CONTENT entity needs to be related to the DOCUMENT entity through a relationship named CONTENT. The INFORMATION-CONTENT entity is a weak entity because it cannot be uniquely identified by the values of its attributes; in fact it is necessary to use the relationship CONTENT to identify it. INFORMATION-CONTENT can be uniquely identified by the values of the attributes of the DOCUMENT entity. The INFORMATION-CONTENT entity is identifier-dependent (ID) on a DOCUMENT entity. Since the INFORMATION-CONTENT entity cannot exist if the DOCUMENT entity does not exist, the INFORMATION-DOCUMENT entity is also existence-dependent (E) on a DOCUMENT entity. In an ER diagram this is represented by E&ID.

Some value sets for the INFORMATION-CONTENT entity are not of the ER pre-defined set; the new value sets must be defined by the designer; these new value-set types are user-defined types and they are application-dependent; for each application or each application type, it is necessary to define the data structures and the operations that are admissible on the data. Two intuitive examples of value sets to be used for the representation of the INFORMATION-CONTENT entity can be TEXT and KEYWORDS; the data structures and the operations on TEXT and KEYWORDS depend on the specific application. For example, the value set TEXT can be defined similarly to the full-text management of many operational IR systems. The value set KEYWORDS can be defined as the set of words of a pre-defined list of terms which are pertinent to the domain of the application.

In the ER approach the information about an entity or a relationship is obtained by observation or measurement, and is expressed by a set of attribute-value pairs; values are classified into different value sets [11]. Each class defined as a value set has occurrences which are values of a pre-defined set: that is, the set of occurrences of a value set is a set of pre-defined values; any class that is a value set is a pre-defined class; and the extensional level of one of these classes is completely characterized [6]. In the conceptual design of IR data, it happens that an attribute maps from an entity into a value set, but the entity has unstructured characteristics. The corresponding value set, for example TEXT, cannot be a pre-defined class, but has to be a "variable" class. A variable class is a class which groups occurrence sets that are different for different database instances.

To use the ER approach in the context of IR applications, it has been necessary to extend the approach to incorporate value sets which have to be defined by the application designer. For this reason it has been necessary to extend the constructs and the notation to allow the possibility of incorporating attributes which map into value sets that are not pre-defined. An extension of the ER approach has been developed to enhance the semantic power of the ER model for the design of IR data.

The fundamental concepts of this new ER approach are:

- a new type of attribute, the *unstructured descriptor*; this new type of attribute maps from an entity which represents information content into a value set which is not pre-defined; and
- the *part hierarchy abstraction mechanism*.

These new concepts are depicted in Fig. 3, which extends the ER concept representation given in [20]. This new representation incorporates the unstructured descriptor; for the graphical representation of the unstructured descriptor a rectangular box in a circle is proposed. The circle is always used for the representation of attributes; a rectangular box is inserted in the circle to denote that this new type of descriptor is similar to an entity. The new ER approach includes also the part hierarchy abstraction mechanism. Figure 3 also includes the part hierarchy abstraction mech-

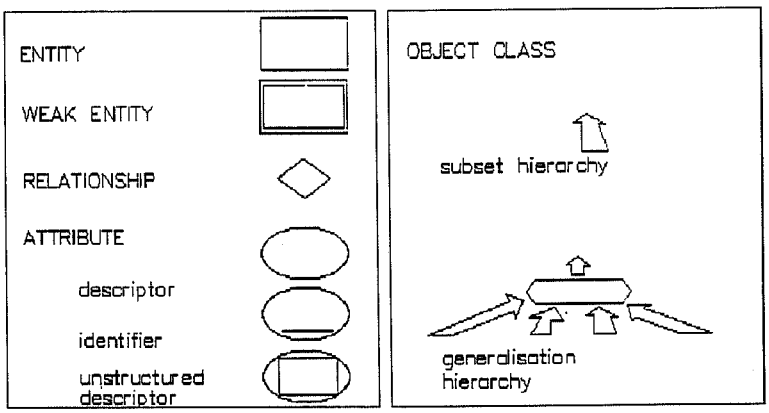

Fig. 3. Concept representation of the extended ER approach for the conceptual design of IR data.

anism in a graphical representation. The next section shows in two detailed design examples the use of the proposed extensions.

5 Use of the proposed extensions

The proposed extensions will now be treated in two real-life design examples: the design of a file of projects for an engineering company where the documents are highly structured; and the design of a file of bibliographic references and supporting abstracts where the documents are simpler and well known.

5.1 The design of a file of engineering projects

The company examined in this example is an engineering company which develops and implements public utility projects such as roads, bridges, canals and gasworks.

The company has been active for more than 20 years. Around 1,700 projects have been developed by the company, a critical number according to management. The documentation on each project is a thick folder divided into three parts. The first is a technical report on the project; the second a set of drawings that represent the works of the project; the third contains photographs of the area around the works.

All the folders are classified by year and kind of work and are stored in different places within the company; a copy of the drawings is classified by year, size and type of work and is stored in the drawing office; copies of the photographs are stored in the documentation centre of the company and each photograph is classified by type of work. All the information retrieval work is carried out manually and it is only possible to retrieve the document entity by its deterministic attributes; for example, a drawing can only be retrieved by year, size and kind of work.

The company management has judged the number of projects to be critical in the sense that it is no longer possible to re-use parts of previously developed projects because the staff are not able to recall to mind the relevant contents of the projects. For this reason, management has decided to consider the project collection as a subsystem of the information system of the company and to automate it as an IR application.

There are three categories of users in the application: the documentalist, the draughtsman and the manager. Each of them needs to retrieve, for different purposes, different parts of the projects by information content.

Three external views of the same application need to be developed at the beginning of the design. When the different views are consolidated, they have to be integrated in a global conceptual view of the application.

For each of the above views it is necessary to develop an ER diagram to represent it. It is assumed that the requirements design has been completed with the ISAC methodology [15]; the mapping from the ISAC to the ER representations is done following the method proposed in [5].

The documentalist stores the technical reports of the projects and photographs of them. He has to be able to retrieve a project through the TECH-

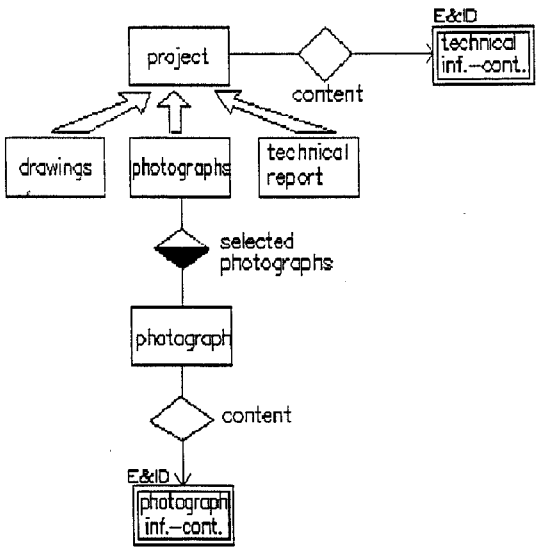

Fig. 4. External view of the documentalist.

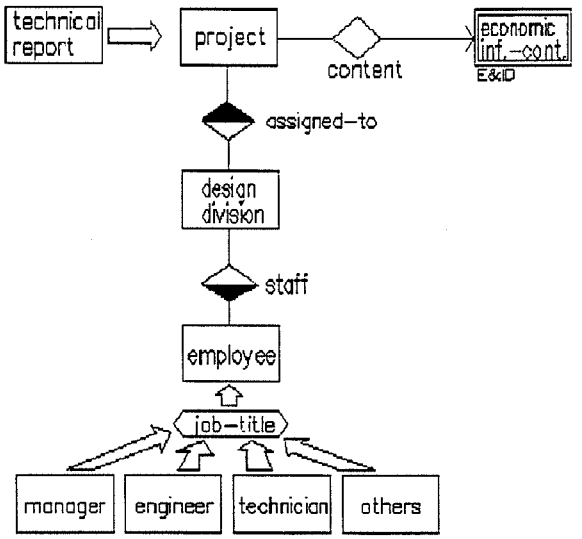

Fig. 6. External view of the manager.

NICAL INFORMATION-CONTENT, the photograph attributes or the PHOTOGRAPH INFORMATION-CONTENT. Figure 4 shows the external view of the documentalist.

The draughtsman stores the drawings of each project. He needs to be able to retrieve the drawings through structured and unstructured representations of the work depicted by a design. Figure 5 shows the external view of the draughtsman.

The manager needs to be able to see the economic aspects of the projects; Fig. 6 shows his external view.

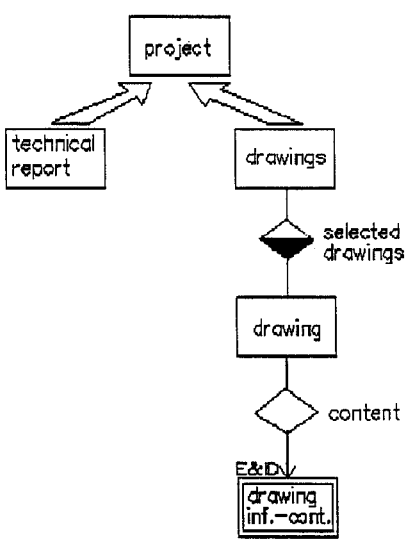

Fig. 5. External view of the draughtsman.

Figure 7 shows the global conceptual schema which is derived from the previous external views after consistency checks and resolution of conflicting views.

Figure 8 expands the part of Fig. 7 that represents the PROJECT entity and its relationship with the CONTENT entity. In Fig. 8 all the value sets and attributes of the entities, together with the representation of the new type of attribute, are shown. The user has access to the PROJECT by means of the attributes of the INFORMATION-CONTENT, the TECHNICAL INFORMATION-CONTENT or the ECONOMIC INFORMATION-CONTENT entities. For example, the user can retrieve data about the ECONOMIC INFORMATION-CONTENT entity through the ECONOMIC DATA, KEYWORDS, DATA-BUDGET and COST-CLASS attributes; the ECONOMIC INFORMATION-CONTENT entity is the starting point for accessing the PROJECT entity through the CONTENT relationship.

5.2 *The design of a file of bibliographic references*

It has been thought necessary to supplement the previous example with a simpler one; a common example of a file of bibliographic references and supporting abstracts was chosen.

The file of bibliographic references is supposed to be used by students and research staff (re-

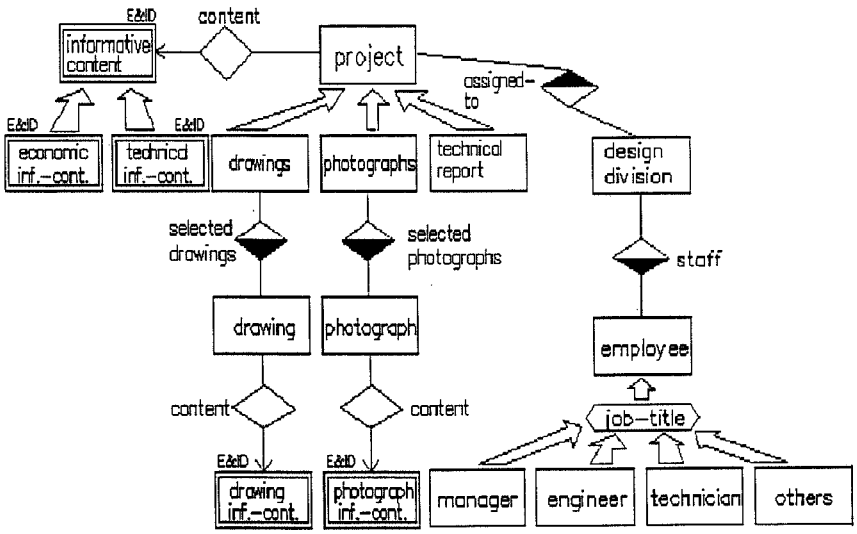

Fig. 7. Conceptual schema for the management of the file of engineering projects.

searchers, professors, etc.) who are involved in a research project in a university department.

Each bibliographic reference has some deterministic attributes (for example, year of publication and title), and some attributes which have to represent the information content of the reference (for example, index terms and abstract).

The number of bibliographic references has grown so large that it has been decided that it is necessary to manage automatically the collection of bibliographic references as an IR application.

There are two categories of users of the application: the student and the research staff of the department. Both types of user need to be able to retrieve the bibliographic references by information content, but the users' profiles are different and suggest the need to produce different views of the application. During the requirements design phase, the needs of the two different categories are evaluated, and at the end it is decided to provide different views. Two external views need to be developed during the design, and these views must be integrated in the conceptual schema of the application. For each external view, it is necessary to develop an ER diagram to represent it. It is supposed that the requirements design has been completed with the ISAC methodology and the mapping has been done from the ISAC to the ER representation. The two external views are represented in Figs. 9 and 10.

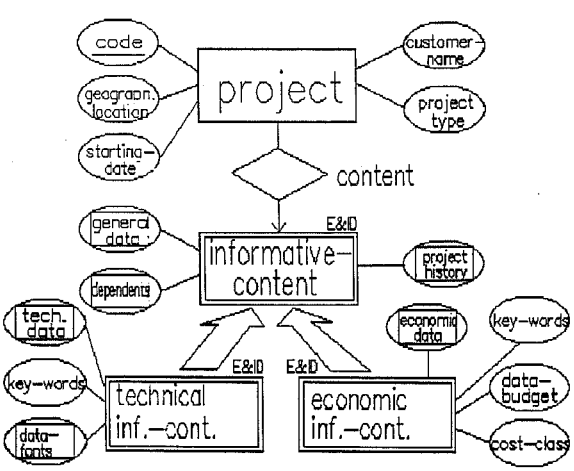

Fig. 8. Usual and new types of attributes.

Different information requirements will be expressed by the use of different types of abstracts: one for the student and one for the research staff. The research staff, in addition, have index terms which represent the information content of the bibliographic reference in detail and with a controlled vocabulary; the connections between index terms of the controlled vocabulary are made visible to the research staff as an aid for more precise retrieval. A bibliographic reference needs to be retrieved through the bibliographic reference attributes (e.g. year of publication) or by the information content, which can be ABSTRACT and INDEX TERMS, for the research staff, or AB-

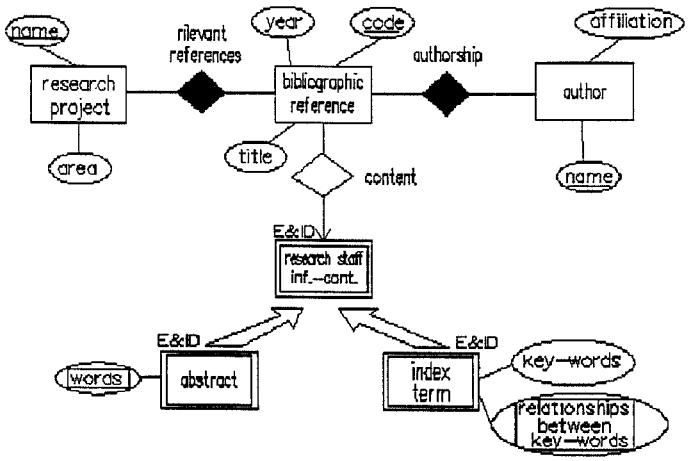

Fig. 9. External view of the research staff.

STRACT, for the student. The user can retrieve data about the ABSTRACT entity through the WORDS attribute; the WORDS attribute is an unstructured descriptor and therefore its value set is defined by the application designer. In this case the designer will specify the algorithms for extraction of the significant words from the abstract in the input phase and for the use of words in the retrieval technique. The user can retrieve data about the INDEX-TERMS entity through the KEYWORDS and the RELATIONSHIP-BETWEEN-KEYWORDS attributes. Since the RELATIONSHIP-BETWEEN-KEYWORDS attribute is an unstructured descriptor, the designer will need to specify the specific use to which this attribute will be put, as has been necessary for the WORDS attribute. Figure 11 shows schematically the conceptual schema of the entire application.

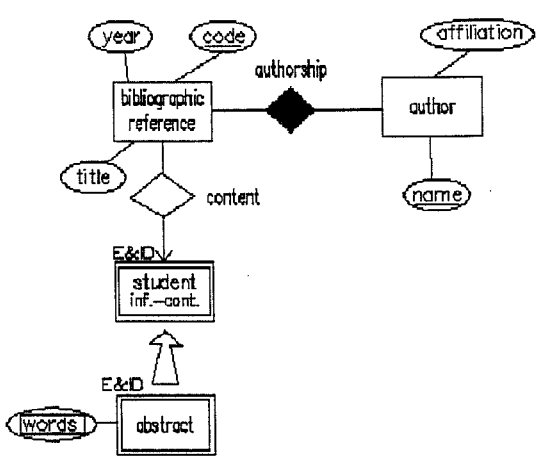

Fig. 10. External view of the student.

6 Conclusions

The reasons for proposing a methodology for the design of IR data were discussed at the beginning of this paper; the second part addressed the conceptual design of IR data. The conceptual paradigm necessary for the design of that type of application was presented. The Entity Relationship approach was compared to the conceptual paradigm introduced and was examined as a possible modelling paradigm for the conceptual design of IR applications.

To use the ER approach in the context of IR applications, it was necessary to extend the approach in order to incorporate value sets, which have to be defined by the application designer, and to include the part hierarchy abstraction mechanism. The new ER approach described extends the constructs of the model to manage the complexity of IR data. This new approach has a new type of attribute, the unstructured descriptor, which maps from an entity which represents information content into a value set which is not pre-defined and which has to be defined by the application designer during the design process. The new ER approach also includes the part hierarchy abstraction mechanism. Two design examples have been given to present the use of the new ER approach in designing IR data.

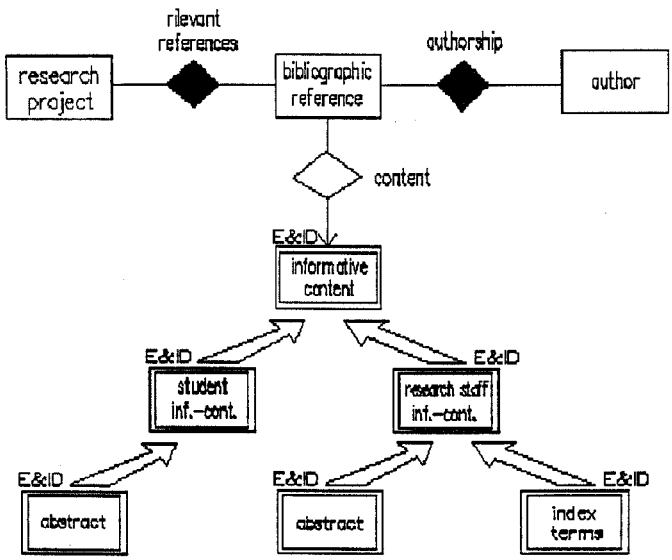

Fig. 11. Conceptual schema for the management of the file of bibliographic references.

The work presented here is part of work in progress on the formulation of a complete design methodology for IR applications.

Acknowledgements

This work was partly supported by the Italian National Research Council (CNR) under the project "Progetto Strategico—Banche Dati."

The constructive comments of the anonymous referees were of great value in the revision of an earlier version of this paper.

References

[1] M. Agosti, Special purpose hardware and effective information processing, *Information Technology: Research and Development* 3 (1984) 3–14.

[2] M. Agosti, *Database Design: a Classified and Annotated Bibliography*. BCS Monographs in Informatics (Cambridge University Press, Cambridge, 1986).

[3] M. Agosti and F. Crestani, La progettazione di applicazioni di Information Retrieval, *Atti del Congresso AICA*, Vol. 1, Trento, 1987, 353–380.

[4] M. Agosti and F. Crestani, Un modello delle applicazioni di Information Retrieval, *Atti del Convegno Text Processing III*, Milan, 2–3 December 1987.

[5] M. Agosti, F. Dalla Libera, F. Lestuzzi and R. Locatelli, Dall'analisi del sistema informativo alla progettazione della base di dati, *Atti del congresso AICA 80*, Bologna, 1980, 1364–1373.

[6] C. Batini, G. De Petra, M. Lenzerini and G. Santucci, *La progettazione concettuale dei dati* (Franco Angeli, Milan, 1986).

[7] C. Batini, S. Ceri and S.B. Navathe, *Database Design Using the Entity-Relationship Approach* (Benjamin/Cummings, Menlo Park, CA, in press).

[8] D.C. Blair, An extended relational document retrieval model, *Information Processing & Management* 24 (1988) 349–371.

[9] G. Bracchi and B. Pernici, Specification of control aspects in office information systems. In: A. Sernadas, J. Bubenko and A. Olivé (Editors), *Information Systems: Theoretical and Formal Aspects* (North-Holland, Amsterdam, 1985) 211–224.

[10] G. Bracchi and B. Pernici, SOS: un modello semantico del lavoro d'ufficio. In: G. Degli Antoni and G. Occhini (Editors), *Office Automation: metodi e tecnologie* (Masson, Milan, 1986) 15–29.

[11] P.P. Chen, The entity-relationship model—toward a unified view of data, *ACM Transactions on Database Systems* 1 (1976) 9–36.

[12] P.P. Chen, Applications of the entity-relationship model. In: S.B. Yao, S.B. Navathe, J.L. Weldon and T.L. Kunii (Editors), *Data Base Design Techniques I: Requirements and Logical Structures* (Springer-Verlag, Berlin, 1982) 87–113.

[13] R. Elmasri, J. Weeldreyer and A. Hevner, The category concept: an extension to the entity-relationship model, *Data & Knowledge Engineering* 1 (1985) 75–116.

[14] R.L. Haskin and R.A. Lorie, On extending the functions of a relational database system. In: M. Schkolnick (Editor), *Proceedings of the ACM-SIGMOD Conference*, 1982, 207–212.

[15] M. Lundeberg, G. Goldkuhl and A. Nilsson, *Information System Development — a Systematic Approach* (Royal Institute of Technology and University of Stockholm, Sweden, 1978).

[16] G. Salton and M. McGill, *Introduction to Modern Information Retrieval* (McGraw-Hill, New York, 1983).

[17] H.J. Schek and P. Pistor, Data structures for an integrated data base management and information retrieval system, *Proceedings of the 8th International Conference on VLDB*, 1982, 197–207.

[18] J.M. Smith and D.C.P. Smith, Database abstractions: aggregation, *Communications of the ACM* 20 (1977) 405–413.

[19] J.M. Smith and D.C.P. Smith, Database abstractions: aggregation and generalisation, *ACM Transactions on Database Systems* 2 (1977) 105–133.

[20] T.J. Teorey, D. Yang and J.P. Fry, A logical design methodology for relational databases using the extended entity-relationship model, *ACM Computing Surveys* 18 (1986) 197–222.