A General Methodological Framework for the Development of Web-Based Information Systems*

Silvana Castano[1], Luigi Palopoli[2], and Riccardo Torlone[3]

[1] DSI, Università di Milano, Italy. castano@dsi.unimi.it
[2] DEIS, Università della Calabria, Italy. palopoli@si.deis.unical.it
[3] DIA, Università di Roma Tre, Italy. torlone@dia.uniroma3.it

Abstract. In this paper, we present a general methodological framework, called WISDOM (Web Based Information System Development with a cOmprehensive Methodology), for the development of Web-based information systems (WIS). WISDOM is generally applicable and supports the design and the development of a wide spectrum of WIS applications. This is achieved by defining the WISDOM framework as a family of inter-related activities. The most suitable combination of activities for the development of a target WIS application is obtained by dropping and/or specializing one or more of the component activities of WISDOM.

1 Introduction

In the last few years, an important growth of Web-based approaches to the development of information systems has been witnessed within very diverse application scenarios. Accordingly, there has been a significant proliferation of technologies supporting developers of Web-based information system (WIS). Moreover, the development of new tools and techniques for WIS management represents nowadays one of the most relevant issues for theoriticians and practitioners in the area. As an example, several tools equipped with high-level interfaces have been defined and implemented for retrieving and processing information extracted from the Web and/or to build Web sites over data repositories organized in the most proper way for this purpose [8,12].

Unfortunately, methodological issues pertaining Web-based information system design and development have not received much attention, even though the development of WIS applications is usually a complex task, often involving the solution of quite diverse problems. Actually, WIS applications may vary significantly from one another, both in terms of application objectives and in terms of architecture and structure and also of supplied functionality. This makes the development of a methodological framework spanning a WIS life-cycle, complex to define. In particular, with WIS applications, aspects related to distribution,

* This research has been partially supported by MURST, within the *InterData* project, and by CNR.

S.W. Liddle, H.C. Mayr, B. Thalheim (Eds.): ER 2000 Workshop, LNCS 1921, pp. 128–139, 2000.

heterogeneity, and availability of information sources as well as to workflow processes are important and have to be considered. In the development of WIS applications, both *information export activities* (devoted to the organization and publication of information over the Internet/Intranet), and *information import activities* (devoted to the collection, extraction, and integration of information available in existing Web information sources for transaction and analytical processing purposes) are to be considered into account and properly inter-related.

The goal of this paper is the definition of a methodology general enough to effectively support the development of complex and diverse WIS applications in which the contents of information sources are analyzed, described, extracted, integrated, processed and possibly reorganized in order to create new information sources. To this aim we propose a *structured methodological framework*, called WISDOM (Web Based Information System Development with a cOmprehensive Methodology) that can be properly customized to a specific WIS application development. WISDOM is made of a large number of activities, independent of each other but strongly coordinated. The whole scheme is structured in that complex activities are iteratively refined in a number of sub-activities. Once the application requirements are known, a specific methodology for the development of the target application can be identified in WISDOM by selecting the most suitable sequence of activities. In this sense, WISDOM provides a framework of reference for the development of *generic* WIS applications: a "view" or "instance" of WISDOM can be obtained by dropping and/or specializing one or more of its component activities, and constitutes a specific methodology for a given application.

The work reported in this paper has been developed within a national Italian project, called InterData, whose aim is to study and develop methodologies and techniques for managing data in Web environments. Some of the issues presented in this paper have been therefore largely influenced by the results of this project. In more detail, Web site design borrows ideas from [1,11], workflow design from [2,9], data integration from [6,7,13,14] and, finally, warehousing from [5].

Providing an integrated framework where both information export and import activities are put together and coordinated is an important step for WIS application development. The effort we made in this respect constitutes the novel contribution of the work. To the best of our knowledge, this is a first paper addressing the development of generic WIS applications with such a comprehensive view of both information import and export activities.

The paper is organized as follows. In Section 2, we settle down the general WISDOM framework. In Section 3, we analyze the activity of information export, whereas in Section 4, we focus on information import. Then, in Section 5, we describe the customization of WISDOM to target WIS applications.

2 The WISDOM Framework

The general structure of WISDOM is illustrated in Figure 1. In the figure, (*i*) a box represents an activity, possibly further decomposed into sub-activities; (*ii*) a link between two activities represents both an order relationship between their

execution and a flow of information; (*iii*) free texts represent input/output data expected/produced by an activity, as indicated by an arrow.

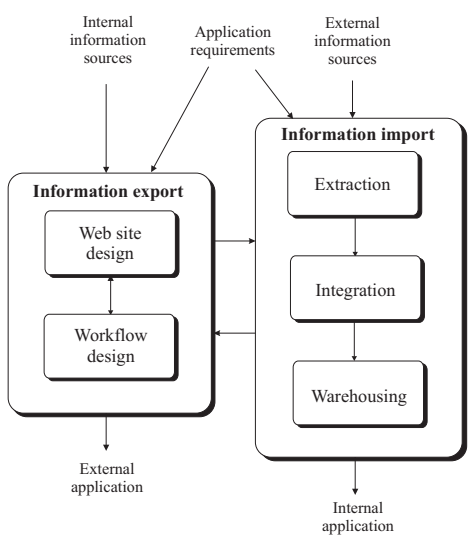

Fig. 1. The WISDOM methodological framework

According to what we have said in the introduction, WISDOM considers two basic activities in the context of Web applications: information *import* and information *export*.

Information export. This activity is devoted to the organization and publication of information over the Internet/Intranet and Web, and possibly to the design of workflow processes over the Web. Published information can originate from (possibly pre-existent) internal data sources or can originate from the integration of proprietary information with data extracted from external data sources. The result of this activity is an *external application* in the sense that it can be accessed through the Web. Information export is decomposed into the following macro-activities (see Fig. 1): *Web site design*, where data to be published are properly organized and managed, possibly using a database management system, and presentation and navigation features of the Web-interface are defined; *Workflow design*, where processes and services provided by the WIS under development are identified, and basic tasks of these processes are defined and coordinated using workflows. If both activities are required, they often need to be executed in parallel since they influence each other.

Information import. Information import consists in the collection, extraction and integration of relevant information available in existing Web sites or data

sources for transaction and analytical processing purposes. As shown in the picture, the result of this can be an *internal application*, that is, an application available in a local network/intranet within the organization. Information import is decomposed into the following main activities (see Fig. 1): *Extraction* of the information of interest from external information sources through wrappers; *Integration* of extracted data, to provide a unified representation of heterogeneous information; *Warehousing* which consists of a further processing of integrated data, mainly based on multidimensional analysis and usually oriented to decision support. Differently from the case of Web site and workflow design, these activities proceed in sequence since each of them needs the outcomes of the preceding activity. As often happens however, a feedback among phases is required to improve and correct results of the previous phases.

Note that the development of a target WIS application often requires both import and export activities. For instance, an application could require the integration of internal data with data extracted from the Web. Integrated data can can be then processed within an internal application or an external one.

In the following sections, we will describe in a top-down way, the various activities reported in Figure 1, resulting in the sub-activities depicted in Figure 2.

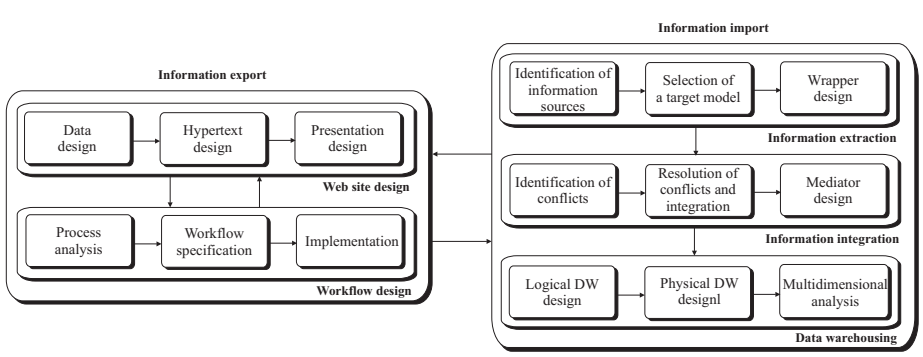

Fig. 2. The WISDOM methodological framework - Refined view

3 Information Export in WISDOM

The Web is rapidly becoming a standard interface to information systems and, more in general, a uniform platform for sharing data, and traditional methodologies for the design of information systems must be reconsidered in this context. The first goal of WISDOM is to provide a support to the development of modern information systems that need to publish and exchange data using the Web. Two major activities can be identified in this context: Web site design and workflow design. In the former, the Web site and the underlying database used as back-

end are defined. In the latter, services and applications are developed to support business process execution over the Web.

3.1 Web Site Design

In the Web scenario, information systems usually maintain large amounts of well structured data that are stored in database management systems and are accessed through a Web site. In WISDOM, the main objectives of Web site design are (a) associating a Web site with a high-level description of its content, that can be used for querying, evolution, and maintenance; and (b) separating the information content, stored in the database from navigation and presentation features, which can be independently defined. Thus, the methodology distinguishes three main sub-activities: the data design, the hypertext design, and the presentation design (see Figure 2).

Data design. This phase is required when the Web based information system is built from scratch. For data design, we follow a traditional approach to database design based on a separation between a conceptual and a logical phase [4]. The output of the first phase is a conceptual scheme (E-R or O-O), describing the organization of the underlying domain in an implementation-independent manner. From the conceptual scheme, the logical (and possibly physical) scheme of the database can be derived using standard techniques. It may happen that the database containing the information to publish is already existent. In this case, data design can be omitted, provided that a conceptual scheme is available. If this is not the case, a reverse-engineering step may be necessary in order to obtain a conceptual scheme starting from the existing database.

Hypertext design. In order to better isolate the structural properties of the resulting hypertext, it is useful to consider two different description levels: the hypertext conceptual level and the hypertext logical level. At the conceptual level, the hypertext is simply described in terms of nodes and paths to navigate between them. This can be done by using a specific data model inspired by known hypertext data model. At the logical level the hypertext organization is detailed in terms of Web pages and links. This can be done using a logical model for Web hypertexts [1]. The hypertext conceptual design aims at describing how application domain concepts can be organized in hypertextual form. This activity is independent of the physical implementation, and concentrates on two essential aspects: (i) deciding which concepts (or combination thereof) of the input scheme will correspond to hypertextual nodes; (ii) choosing the paths to navigate between concepts. The hypertext logical design describes the actual organization in terms of pages and links of the Web site. This phase is specific of the Web framework, and aims at detailing the structure of the hypertext as shown by the browser. At this level we concentrate on abstracting the relevant pieces of information in the page (e.g., text or images) and their organization (e.g., at page, list or nested list). To do this, the hypertext conceptual scheme is translated into a logical scheme, based on a suitable logical data model for hypertext applications providing a notion of *page type* and where each page is

seen as an object with an URL plus a set of attributes. Page types are connected using links, used to describe navigation in the site.

Presentation Design. In the Presentation Design, the final graphical layout of Web pages is defined. The input of this phase is the logical hypertext description of the Web site and the output are a collection of Web pages in the form of HTML documents or XML files plus XSL style sheets. We associate a page style with a page type of the hypertext logical schema. Such a page style specifies all format directives for each piece of information in the page, plus the graphical features to be associated with the page itself, like, for example, page background and banners. This can be done using an abstract description or a standard language like XSL, whereby Web pages can be automatically derived [11].

3.2 Workflow Design

Workflow design in WISDOM pertains to all the activities to be carried out in order to build applications supporting business process execution over the Web. As an example of business process execution over the Web, consider an e-commerce process for sales. Its description consists of information about the involved activities and information objects (e.g., selection of items to purchase, filling of a form, checking fund availability for a credit card, etc.), the involved agents (e.g., seller, buyer), and the business goals (e.g., high-level perceived security). This process can be naturally modeled in the form of a workflow. The activities composing the workflow design macro-activity in WISDOM are described in the following (see Figure 2).

Process analysis. Process analysis starts from a description of the business process to be implemented covering the following perspectives [2]: (i) functional, concerning the process activities and the involved information objects; (ii) organizational, concerning the agents and roles involved in process execution; (iii) business, concerning the goals capturing business rules and objectives of the process. A workflow is suggested for each group of activities that are loosely coupled with the outside and have a high number of connection points with activities within group, and which are performed within different organizational units. For each candidate workflow, roles, pre-conditions (i.e., the event(s) starting the workflow), post-conditions (i.e., how the workflow ends), and associated business goals, are specified.

Workflow specification. The goal of this activity is the definition of a schema for each candidate workflow according to a conceptual workflow model [2,9]. According to such a model, a workflow schema is composed of sub-processes and tasks. Tasks are organized into a directed graph, which defines their execution order. Arcs in the graph can be labeled with transition predicates defined over process data, with the meaning that the tasks connected through outgoing arcs are executed only if the corresponding transition predicate evaluates to true. A top-down development is assumed in the workflow design phase. A candidate workflow resulting from the process analysis phase is decomposed into

sub-processes and then into tasks, which are properly inter-related into a flow structure, to reflect the correct sequencing of activities in the sub-process. In a WIS application, sub-processes can describe self-contained activity fragments to be executed, for example, at a given site. Interactions with external information systems/applications are also modeled during this phase, by specifying at a conceptual level the expected interaction modalities between the elements of the workflow schema and external information systems/applications.

Workflow implementation. The final activity of workflow design in WISDOM consists in implementing the designed workflow, either by developing ad hoc software solutions or by using a commercial workflow management system.

4 Information Import in WISDOM

In order to run desired analysis and manipulation procedures over information coming from one or more existing sources, information must be first located and extracted from world-spread information source sites. Information coming from different sources can be heterogeneous and possible conflicting situations have to be identified and resolved to come up with an integrated, unified information representation. Moreover, integrated information may not have the right format to undergo needed decision-oriented analysis, and a warehousing activity can be required. Consequently, the goal of WISDOM for the information import activity is to provide a methodological support to guide the user in the activities of extraction, integration and warehousing.

4.1 Extraction

The extraction macro-activity in WISDOM pertains all the activities to be carried out in order to import the information at the site where analysis procedures take place. Obtaining information means either to materialize data at the application site (materialized approach) or maintaining suitable query templates to be executed to extract the information on demand (virtual approach) [10]. The extraction activity requires the identification of external information sources where relevant information to be extracted can be found. A target data model is selected to represent extracted information uniformly in view of subsequent integration activity, and wrapper tools are developed to perform data restructuring according to the selected target model. Hence, the Extraction macro-activity is refined as described in the following (see also Figure 2).

Identification of information sources. With this activity, application requirements must be analyzed to single out required information content as target elements. A *target element* refers to a single concept or to a restricted number of related concepts of interest for the application. Moreover, it provides a textual description of the information to be retrieved for it (e.g., *Employee* target element, with information regarding employee name, address, salary, qualification). The designer must identify all concepts relevant to the application, and define

appropriate target elements for representing them. Potential information sources where data sets for the target elements required by the application can be located are established. Since we deal with WIS applications, information sources where to find data sets could be not known a priori, due to the lack of sufficient information on contents and location of the sources worldwide. In such a case, the entire Web is considered as the default information source, and the selection of most specific information sources is demanded to the subsequent phases.

Selection of a target data model. When retrieved from multiple, existing information sources, data sets can be formatted in different ways and may be not ready to be used for the purposes of developing the target WIS application. Therefore, the designer has to define a target data model, according to which extracted data sets will be restructured for the integration activity. In particular, the target data model can be a database model, or a semi-structured data model. After defining the target data model, the designer gives indication also of the storing data structures and presentation formats for schema elements.

Wrapper design. Data sets can be extracted from a WIS using various techniques. This include specialized query languages, search robots and custom procedures [12]. A WIS application development will typically require to choose more than one of such tools in order to locate and retrieve all the needed data sets for each element. It is important to stress that performing retrieval over the Web may also serve the purpose of data location. Moreover, note that a correct choice of retrieval tools is a key issue if efficiency is to be achieved, specially if "on-the-fly" retrieval of data is involved. For each target element, one or more query templates are defined to import the corresponding data sets. A query template specifies the location where to retrieve data (in the default case, the Web) and the extraction strategy. For materialized elements, query templates are executed and their results (i.e., data sets) are gathered according to the data format they have in the original source. As for virtual elements to be retrieved "on-the-fly", query templates are maintained in form of stored procedures at the local source. A wrapper is designed for each different data set to generate its corresponding representation according to the target data model, and vice versa. As the result of the wrapping, we obtain the so called *restructured data sets* are obtained to be used in the subsequent integration step.

4.2 Integration

This activity serves the purpose of resolving conflicts between restructured data sets obtained in the previous phase for a given target element. The goal is to obtain an integrated, unified representation of various target elements, and make them suitable for subsequent elaboration. Sub-activities composing the integration macro-activity are described in the following (see Figure 2).

Identification of conflicts. This activity is concerned with pointing out conflicts among restructured data sets. Conflicts are due to the fact that different external information sources may use different terminologies and design structures to describe the same concept. Using the target data model and wrapper tools

facilitates conflict identification and resolution, in that all data representations are reduced to the same model. Following [3,10], we distinguish the following main categories of conflicts: (*i*) *lexical conflicts*: due to a different terminology adopted to denote a certain concept; (*ii*) *structural conflicts*: due to the use of different data structures used for representing a certain concept; (*iii*) *semantic conflicts*: due to the differences between utilized domain values (e.g., format, currency, unit). In addition, restructured data sets should be analyzed to discover also possible application-dependent conflicts. Semi-automatic techniques can be employed to assist the designer in the conflict identification activity [7,14].

Resolution of conflicts and integration. Given the list of conflicts identified in the previous step, this activity deals with conflict resolution, to obtain an integrated definition of the target application schema. Conflict resolution is a process heavily custom-interactive, since (specially when semi-structured data are involved), it is largely application-specific. However, semi-automatic techniques have been developed to support the designer in deriving the integrated schema out of restructured data sets plus conflicts [6,13]. As the result of the conflict resolution activity, the final target application schema is obtained.

Mediator design. The goal of this activity is to define mappings and conversion functions to materialize each target element by mediating the different structures of underlying restructured data sets. Given a target element, mappings defined for it specify the correspondences between its structure and the structure of restructured data sets from which it has been derived. Conversion functions are defined for attributes of the target element to implement transformations to convert mismatching domain values of its corresponding data sets, if necessary. In case of materialized target elements, defined mappings and conversion functions are executed on data sets to populate the target application schema. In case of virtual target elements, mapping and conversion functions are maintained together with the target application schema, and will be used for populating target elements "on-the-fly", in combination with stored query templates. Mappings and conversion functions together with the target application schema constitute the so called mediator module. Mediator functionality coupled with wrapping translation functionality previously described enforce the integration of heterogeneous data sets.

4.3 Warehousing

This macro-activity is concerned with the construction of an integrated collection of operational data, called *data warehouse*, followed by the processing of its content, usually oriented to decision making. The warehousing macro-activity of WISDOM consists of the following sub-activities (see Figure 2).

Logical data warehouse design. In this phase, the structure of the data warehouse is defined starting from an integrated schema. The integrated schema may have a format that is often not suitable for the analysis purposes. Therefore, a first step of the data warehousing process consists in the transformation

of the input scheme in order to provide a better support for analysis operations. The output consists in a schema of the data warehouse according to a logical data model, which describes the multidimensional aspects of data analysis and is independent of the implementation in a specific data storage system. We make use of a logical data model for multidimensional databases [5]. The logical data warehousing design follows a structured approach consisting of a number of activities. The first activity consists in a careful analysis of the given E-R scheme whose aim is the selection of the facts, the measures, and the dimensions of interest for our business processing. The second activity consists in a reorganization of the original E-R scheme in order to describe facts and dimensions in a better, more explicit way. The goal of this step is the production of a new E-R scheme that can be easily translated into the logical multidimensional model. A dimensional graph is used to represent, in a succinct way, facts and dimensions of the restructured E-R scheme. The final activity consists in translating the dimensional graph into the logical multidimensional model.

Physical data warehouse design. In this phase the input schema is translated into the data model adopted by the storage system chosen. Since the multidimensional database is defined according to a logical data model, it can be in fact implemented in several ways (e.g., using either ROLAP or MOLAP systems). In the first case a multidimensional database can be implemented in the form of a "star scheme" (or variant thereof, e.g., snowflake scheme). In the second case, an f-table is represented by a n-dimensional matrix, storing each measure corresponding to a certain symbolic entry in the cell having the corresponding physical coordinates. A dimension can be then represented by means of a special data structure, with a hierarchical organization according to the hierarchy defined on it. We can then use this structure as an index to access the multidimensional array.

Multidimensional analysis. In this last step the data warehouse is finally used to perform the analytical processes for which it was originally designed. This can done by using specific OLAP tools having querying and reporting capabilities. Aggregate views of the warehouse can be materialized to support analysis needs. In this phase, the propagation to the warehouse of updates on source data is periodically required. The outcomes of the analysis can be published through the Web site: this clearly requires some of the activities described in Section 3.

5 Application of WISDOM

In order to pinpoint the effective applicability of our methodological framework, we will describe possible instantiations/customizations with respect to significant practical examples of applications.

Development of a data analysis application with a Web site interface. This is an example of *mixed* application, where WIS design involves aspects related to both information import to design an integrated warehouse supporting expected types of analysis by selecting and integrating pre-existing data sources and information

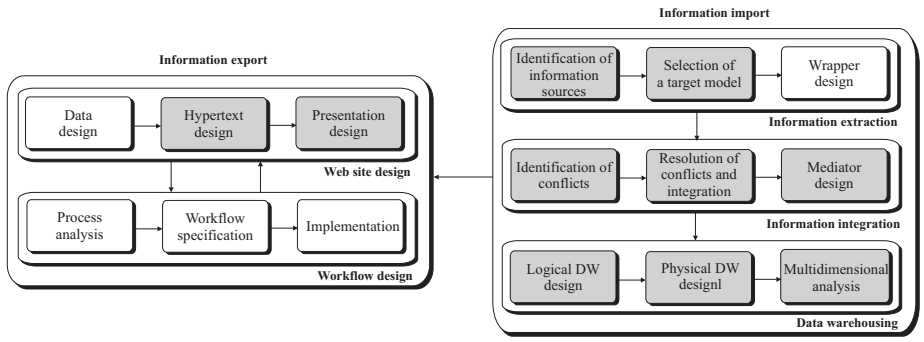

Fig. 3. Example of customized view of WISDOM

export, to design a Web site on top of the warehouse, to provide a common interface for querying the warehouse and publish analysis results. In this case, WISDOM can be customized by selecting the activities related to extraction, integration and warehousing. Moreover, for each macro-activity, only its sub-activities necessary in the specific application scenario are kept. For example, a possible customization of the methodological framework is shown in Fig. 3.

Design of an integrated schema of heterogeneous databases. This is an example of information import, where the goal is to define an integrated mediator schema of pre-existing heterogeneous databases to support uniform queries at the global level. In this case, WISDOM is customized by selecting the activities related to extraction and integration. Also in this case, some sub-activities can be skipped if not necessary.

Design of a Web-based document management workflow. This is an example of information export, where WIS design involves aspects related to coordination of activities across distributed organization units together with correct document exchanges among the involved units. This kind of application is typical, for example, of the Public Administration domain, where document management workflows are required to automate and make more efficient administration processes involving many organization units, often distributed over the territory. In this case, WISDOM can be customized by keeping only the activities related to workflow design. The emphasis is on designing activity and document flows correctly, by identifying all involved organization units, their responsibilities and interconnection modalities, and their expected interactions.

6 Concluding Remarks

In this paper, we have presented the WISDOM methodological framework for the development of WIS applications, conceived as a family of inter-related activities to support both information export and import activities. Future research work

will be devoted to set out a complete instantiation scenario for a real case study of WIS application development. In this paper, we discussed the applicability of WISDOM by referring to the general typologies of WIS applications of Section 5. Specific case studies related to web site design, workflow design, integration, and warehousing activities have been separately studied in the framework of the Interdata project (see, for example, [6,13,5]), and are under consideration to develop a comprehensive case study. Another goal of future research work will be the enrichment of the framework with a set of pre-defined "methodological patterns", suggesting to the WIS designer a reference methodology for a specific application to be developed, that can be customized if necessary, based on the specific requirements of the application at hand.

References

1. P. Atzeni, G. Mecca, and P. Merialdo. Design and maintenance of data-intensive Web sites. In *Sixth Int. Conf. on Extending Database Technology (EDBT'98)*, 1998.
2. L. Baresi, F. Casati, S. Castano, M.G. Fugini, I. Mirbel, B. Pernici. WIDE Workflow Development Methodology. In *Int. Joint Conf. on Work Activities Coordination and Collaboration (WACC'99)*, 1999.
3. C. Batini, M. Lenzerini, S.B. Navathe. A Comprehensive Analysis of Methodologies for Database Schema Integration. *ACM Computing Surveys,* Vol.18, No.4, 1986.
4. C. Batini, S. Ceri, and S. Navathe. *Conceptual Database Design: an Entity-Relationship Approach.* Benjamin & Cummings, 1992.
5. L. Cabibbo, R. Torlone. A logical approach to multidimensional databases. In *Sixth Int. Conf. on Extending Database Technology (EDBT'98)*, 1998.
6. S. Castano, V. De Antonellis, S. De Capitani Di Vimercati. Global Viewing of Heterogeneous Data Sources. *IEEE Trans. on Knowledge and Data Engineering*, to appear.
7. S. Castano, V. De Antonellis. A Discovery-Based Approach to Database Ontology Design. *Distributed and Parallel Databases*, Vol.7, N.1, 1999.
8. A. Deutsch et al. XML-QL: A Query Language for XML. World Wide Web Consortium, Working paper, *(http://www.w3.org/TR/NOTE-xml-ql)*, 1998.
9. P. Grefen, B.Pernici, G. Sanchez, (eds.), *Database Support for Workflow Management, The WIDE Project.* Kluwer Academic Publishers, 1999.
10. R. Hull. Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. Tutorial presented at *PODS'97*, 1997.
11. G. Mecca, P. Merialdo, P. Atzeni, V. Crescenzi. The Araneus Guide to Web Site Development. In *ACM SIGMOD Workshop on the Web and Databases (WebDB'99)*, 1999.
12. A. Mendelzon, G. Mihaila, T. Milo. Querying the World Wide Web. In *First Int. Conf. on Parallel and Distributed Information Systems (PDIS'96)*, 1996.
13. L. Palopoli, L. Pontieri, G. Terracina, D. Ursino. Intensional and Extensional Integration and Abstraction of Heterogeneous Databases. *Data and Knowledge Engineering*, to appear.
14. L. Palopoli, D. Saccà and D. Ursino. Semi-automatic, semantic discovery of properties from database schemes. In *IDEAS'98*, 1998.