Integration and Provenance of Cereals Genotypic and Phenotypic Data[⋆,⋆⋆]

Domenico Beneventano, Sonia Bergamaschi, and Abdul Rahman Dannaoui

Department of Computer Science
University of Modena and Reggio Emilia
via Vignolese 905, 41125 Modena, Italy
firstname.lastname@unimore.it

Abstract. This paper presents the ongoing research on the design and development of a Provenance Management component, PM_{MOMIS}, for the MOMIS Data Integration System. PM_{MOMIS} aims to provide the provenance management techniques supported by two of the most relevant data provenance systems, the *Perm* and *Trio* systems, and extends them by including the data fusion and conflict resolution techniques provided by MOMIS.
PM_{MOMIS} functionalities have been studied and partially developed in the domain of genotypic and phenotypic cereal-data management within the CEREALAB project. The CEREALAB Data Integration Application integrates data coming from different databases with MOMIS, with the aim of creating a powerful tool for plant breeders and geneticists. Users of CEREALAB played a major role in the emergence of real needs of provenance management in their domain.

1 Introduction

The CEREALAB Data Integration Application has been developed to create a powerful tool for plant breeders and geneticists [21]. It stores genotypic and phenotypic cereal-data collected within the CEREALAB project and integrates them with already existing well known public data sources. The integrated database can help breeders and geneticists in: unravelling the genetics of economically important phenotypic traits, identifying and choosing molecular markers associated to key traits, and in choosing the desired parentals for breeding programs. The CEREALAB Data Integration Application development was one of the objectives of the CEREALAB and SITEIA projects and of the BIOGEST-SITEIA laboratory (www.biogest-siteia.unimore.it), funded by Emilia-Romagna (Italy) regional government, and aims to increase the competitiveness of Regional seed companies through the use of modern selection technologies, i.e. the Marker-Assisted Selection (MAS).

Data integration is obtained by using the MOMIS system (Mediator envirOnment for Multiple Information Sources), a framework to perform integration of structured

[⋆] Extended abstract of the paper "D. Beneventano, S. Bergamaschi, A.R. Dannaoui, N. Pecchioni: Integration and Provenance of Cereals Genotypic and Phenotypic Data, to appear as a Poster at Data Integration in the Life Sciences (DILS 2012)".

[⋆⋆] This work is partially supported by the BIOGEST-SITEIA laboratory (www.biogest-siteia.unimore.it), funded by Emilia-Romagna (Italy) regional government.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

and semi-structured data sources [3,7]. MOMIS is characterized by a classical wrapper/mediator architecture: the local data sources contain the real data, while a Global Schema (GS) provides a *reconciled, integrated, read-only view* of the underlying sources. The GS and the mappings between the GS and the local sources are semi-automatically defined at design time by the Integration Designer component of the system [3]. After GS creation end-users can pose queries over this GS in a transparent way w.r.t. the local sources. MOMIS has been developed by the DBGROUP of the University of Modena and Reggio Emilia[1]. An open source version of the MOMIS system is delivered and maintained by the academic spin-off DataRiver[2].

As discussed in [21] the global classes of the *CEREALAB GS* plays the role of performing *data fusion*, i.e. the process of fusing multiple records representing the same real-world object into a consistent representation. Data fusion may involve the resolution of possible conflicts between data coming from different sources; several high level strategies to handle inconsistent data have been described and classified in [9]. MOMIS supports: *conflict avoiding* strategies (such as the *trust your friends* strategy which takes the value of a preferred source), *conflict ignoring* strategies (such as the *pass it on* strategy, which presents all values deferring conflict resolution to the user) and *resolution* strategies (such as the *meet in the middle* strategy which takes an average value). These strategies are implemented by means of *Resolution functions* in the *full outerjoin-merge operator* proposed in [23] and adapted to the MOMIS System in [7].

A requirement emerging from CEREALAB users, the breeders, was that in many cases they would prefer to give a look at the data coming from the local sources, i.e. they need *provenance*. A need to support detailed data provenance is one of the main requirements of biological data management identified in [19][3].

Lineage, or *provenance*, in its most general definition, describes where data came from, how it was derived and how it was modified over time. Lineage provides valuable information that can be exploited for many purposes, ranging from simple statistical resumes presented to the end-user to more complex applications, such as, managing data uncertainty or identifying and correcting data errors. Lineage has been studied extensively in data warehouse systems [11], but it is still an open research problem in Data Integration systems [15,14].

In [4] we introduced the notion of provenance into MOMIS, by defining the provenance for the *full outerjoin-merge operator*; this definition is based on the concept of *PI-CS*-provenance (Perm Influence Contribution Semantics) proposed in *Perm* (Provenance Extension of the Relational Model) [12] to produce more precise provenance information for outerjoins. Another important reason behind the choice of using the *PI-CS*-provenance, was that it is fully implemented in an open-source provenance management system that is capable of computing, storing and querying provenance for relational databases. At present, we are using the *Perm* system as the SQL engine of MOMIS, so that to obtain the provenance in our CEREALAB Application.

[1] `http://www.dbgroup.unimore.it`

[2] `http://www.datariver.it`

[3] This article is an extract from the Report of the NSF Workshop on Data Management for Molecular and Cell Biology, edited by H. V. Jagadish and Frank Olken he workshop was held at the National Library of Medicine, Bethesda, MD, Feb. 2-3,2003)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

2 The MOMIS Data Integration System

A MOMIS Data Integration System [3,7,8] is constituted by: a set of *local schemas* $\{LS_1, \ldots, LS_k\}$, a *global schema* GS and *Global-As-View* (*GAV*) mapping assertions [20] between GS and $\{LS_1, \ldots, LS_k\}$. For each global class G of GS, a *Mapping Table* (*MT*) is defined, whose columns represent the set of local classes $\{L_1, \ldots, L_n\}$ of G; an element $MT[GA][L]$ represents the local attribute of L which is mapped onto the global attribute GA, or $MT[GA][L]$ is empty (there is no local attribute of L mapped onto the global attribute GA)[4]. *GAV* mapping assertions are expressed by specifying for each G a *mapping query* over its local classes, which defines the instance of G.

2.1 The CEREALAB Data Integration Application

In [21] we described the MOMIS semi-automatic approach to build the *GS* of the CEREALAB Data Integration Application. In this paper, we focus on the *GermPlasm* global class (see Figure 1) obtained by integrating two local classes `A` and `B` which store data about germplasms coming from two different data sources: `A` stores data obtained by experimental results within the CEREALAB project, `B` stores data coming from other public data sources (see [21] for a fully description of these sources).

The attributes of these tables represent the following information: `GPN`: the name of a variety; `FHB`: the resistance of the germplasm to the *FHB* disease (Fusarium Head Blight) (as values are S=susceptible, MR= Moderately Resistant and R= Resistant); *Type*: the variety's type; `yield` : the *grain yield* expressed in tons/hectare.

Figure 1 shows the Mapping Table of the global class `GERMPLASM`, with schema `GERMPLASM(GPN,Yield,FHB,Type)`, obtained by integrating `A` and `B`. As discussed in [21] `GERMPLASM` plays the role of performing *data fusion*. To identify multiple local tuples coming from local classes and representing the same real-world object, we assume that error-free and shared object identifiers exist among different sources: two local tuples with the same object identifier ID indicate the same object in different sources; thus we can use L^{id} to denote the tuple t of a local class L with ID equal to id, i.e. $t[ID] = id$. In our example, we assume `GPN` as an object identifier; then the first tuple of the local class `A`, i.e. the tuple with $\texttt{GPN} = Eureka$, will be denoted with $\texttt{A}^{Eureka}$. For conflicting attributes the following strategies are used:

- `FHB` (*trust your friends* strategy): `FHB`=COALESCE(`A.FHB`,`B.FHB`).
  Conflicts are solved preferring the data coming from the `A` Italian source to offer the Italian breeders information from Italian studies and so nearest to their needs;
- `Type` (*pass it on* strategy): `Type`= ALLVALUES(`A.type`, `B.type`)
  A germplasm type is unique, so when we find two different types for the same

[4] For the sake of simplicity, we consider a simplified version of the MOMIS framework proposed in [3,7], where $MT[A][L]$ is a set of local attributes and *Data Transformation Functions* specify how local attribute values have to be transformed into corresponding global attribute values. Moreover we assume $S(G) = \cup_i S(L_i)$, i.e. global and local attribute names are the same. Finally, both the global and the local schemas are expressed in the ODL_{I^3} language [8]. However we consider both GS and LS_i as relational schemas, but we will refer to their elements respectively as global and local classes to comply with the MOMIS terminology.

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors

Local classes

A (GermPlasmA)

GPN	yield	FHB	type
Eureka	18	MR	
Fortuna	7	MR	
Mentana		S	line
Kenora	20	MR	landrace
Oasis	21	MR	cultivar

B (GermPlasmB)

GPN	yield	FHB	type
Eureka	6	S	cultivar
Fortuna	15	S	landrace
Mentana	20	MR	line
Kenora			cultivar

Mapping Table of the global class GERMPLASM

GERMPLASM	A	B
GPN	GPN	GPN
Yield	yield	yield
FHB	FHB	FHB
Type	type	type

Fig. 1. Example: two local classes with two conflicting attributes

Mapping Query of GERMPLASM

```
SELECT GPN = GPN,
       Yield=AVG(A.yield, B.yield)
       FHB = COALESCE(A.FHB,B.FHB),
       Type = ALLVALUES(A.type, B.type)
FROM   A FULL OUTER JOIN B
       USING (GPN)
```

Instance of GERMPLASM

GPN	Yield	FHB	Type
Eureka	12	MR	cultivar
Fortuna	11	MR	landrace
Mentana	20	S	line
Kenora	20	MR	landrace,cultivar
Oasis	21	MR	cultivar

Fig. 2. Mapping Query and Instance of the global class GERMPLASM

germplasm we know that uncertain data are present. In the case of *Kenora*: we have two types but we do not know which value is correct. To avoid choosing the wrong value all values are maintained and conflict resolution is deferred to the user;

- Yield (*meet in the middle* strategy): Yield=AVG(A.yield, B.yield) represents the average of grain yield in the local classes.

Finally, the mapping query of the global class GERMPLASM (and then the instance of GERMPLASM) is defined by means of the *full outerjoin-merge operator*. Intuitively, as shown in Figure 1, this corresponds to the following two operations: (1) Computation of the *full outerjoin*, on the basis of the shared object identifier, of the *local classes* of GERMPLASM; (2) Application of the *Resolution Functions* [5].

2.2 Provenance at present implemented in the MOMIS system

In [4] the concept of *PI-CS*-Provenance was applied and extended to the full outerjoin-merge operator. As an example, let us consider a query on GERMPLASM (the user is searching for the types of the varieties that are resistant to the fusarium head blight):

```
TYPE_MR = SELECT  DISTINCT Type
              FROM  GERMPLASM
              WHERE FHB = 'MR'
```

[5] COALESCE is the (standard SQL) function which returns its first non-null value; AVG is a (non standard SQL) function to compute the average value; ALLVALUES is a (non standard SQL) function which returns all non-null values.

`Type`	*PI-CS* Provenance as a set of witness lists
landrace	$\{ \langle \texttt{A}^{Fortuna}, \texttt{B}^{Fortuna} \rangle \}$
cultivar	$\{ \langle \texttt{A}^{Eureka}, \texttt{B}^{Eureka} \rangle, \langle \texttt{A}^{Oasis}, \perp \rangle \}$
landrace,cultivar	$\{ \langle \texttt{A}^{Kenora}, \texttt{B}^{Kenora} \rangle \}$

Relational Representation of the *PI-CS* Provenance

`Type`	`A.GPN`	`A.yield`	`A.FHB`	`A.type`	`B.GPN`	`A.yield`	`B.FHB`	`B.type`
landrace	Fortuna	7	MR		Fortuna	15	`S`	landrace
cultivar	Eureka	18	MR		Eureka	6	`S`	cultivar
cultivar	Oasis	21	MR	cultivar				
landrace,cultivar	Kenora	20	MR	landrace	Kenora			cultivar

Fig. 3. Example: *PI-CS* Provenance for the query `TYPE_MR`

The *PI-CS*-Provenance of an output tuple is a set of *witness lists*, where each witness list represents one combination of local tuples that were used together to derive the output tuple; a witness list contains a local tuple from each local class or the special value $\perp$, indicating that no tuple from a local class was used to derive the output tuple (useful in modeling outerjoins). For example, the *PI-CS*-Provenance of the output tuple cultivar (see Figure 3) is a set of two *witness lists*, where the second one $\langle \texttt{A}^{Oasis}, \perp \rangle$ indicates that $\texttt{A}^{Oasis}$ paired with no tuples of `B` is a possible derivation of the output tuple. Witness lists are represented in a relational form, as shown in Figure 3: each witness list of an output tuple is represented by a single tuple.

The main drawback of this solution is that often conflicting values represent *alternative*. For example if we want to select other germplasms of the same type of *Kenora*, the two values landrace and cultivar must be considered as *alternative values*; if not, we might obtain Fortuna and Eureka as germplasms of the same type. Our proposal to overcome this drawback is to consider the output of the full outer join merge operator as an *uncertain relation* and then manage it with a system that supports uncertain data and data lineage, the *Trio* system [1,6].

3 Provenance based Conflict Handling Strategies

The *Trio* system is based on the *ULDB* data model [5], which extends the relational model with:

- *Alternatives*, representing uncertainty about the contents of a tuple. ULDB uncertain relations have a set of *certain* attributes and a set of *uncertain* attributes; each tuple in a ULDB relation has one value for each certain attribute, and a set of possible values for the uncertain attributes.
- *maybe* ('?') annotations, representing uncertainty about the presence of a tuple.
- *Lineage*, connecting tuple-alternatives to other tuple-alternatives from which they were derived: *Trio*-Provenance is a boolean formula λ over tuple-alternatives.
- *Confidences*: numerical confidence values optionally attached to alternatives.

(A) Global class `GERMPLASM`

`GPN`	`Yield`	`FHB`	`Type`	
Eureka	12	MR	cultivar	?
Fortuna	11	MR	landrace	?
Mentana	20	S	line	?
Kenora	20	MR	landrace || cultivar	?
Oasis	21	MR	cultivar	

(B) query `TYPE_MR`

`Type`	
landrace	?
cultivar	
landrace || cultivar	?

Fig. 4. Global class `GERMPLASM` and query `TYPE_MR` as *uncertain relations*

Intuitively, these concepts might be applied to our data fusion context as follows. As shown in Figure 4.A, the global class `GERMPLASM` is considered as an *uncertain relation* where *non-conflicting* attributes and *solved-conflicting* attributes (like `FHB` and `Yield`) are modelled as *certain* attributes and *not-solved-conflicting* attributes (like `Type`) are modelled as *uncertain* attributes. The global tuple identified by `GPN` $=$ $Kenora$, denoted by $Kenora$, has two *alternatives*, $(Kenora, 1)$ and $(Kenora, 2)$. A global tuple coming from local tuples with conflicting values (solved or not solved) is annotated with '?' ; in the example, all global tuples are *maybe* tuples '?', with the exception for $Oasis$. The *Trio*-Provenance of a global tuple-alternative is a boolean formula over the local tuples from which the alternatives were derived; for example:
$\lambda(Kenora, 1) = \mathtt{A}^{Kenora}$: this alternative derives from a local tuple in `A`;
$\lambda(Kenora, 2) = \mathtt{A}^{Kenora} \wedge \mathtt{B}^{Kenora}$: this alternative derives from the conjunction of two local tuples. At present *Confidences* are not considered in our framework.

How can we implement the three above TRIO concepts in PM$_{\mathrm{MOMIS}}$? Can we obtain the `GERMPLASM` *uncertain relation* shown in the example by means of the *Trio* system?

GERMPLASM computation is based on outerjoins and uses resolution functions, thus such computation is not simple as in *Trio* outerjoins are not allowed[6] and resolution functions have to be implemented in *Trio* (as we already did with a strong effort in *Perm*). For this reason, our choice is to use PERM for computing global classes and Trio for querying global classes. The computation of the full outer join merge operator is extended to obtain global classes including conflicts as *uncertain relations* with their related *TRIO*-Provenance; this computation is discussed in [2] and its result is intuitively shown in Figure 4.A. Thus, the *Trio* system[7] may be used to execute queries on global classes (i.e. exploiting the uncertain database theory); intuitively, the uncertain relation resulting from query `TYPE_MR_S` is shown in Figure 4.B. In this way, the user knows that only cultivar (the second tuple) is coming from non-conflicting local tuples, since it is not a *maybe* tuples; then he may obtain the provenance of this tuple either[8] at

[6] In a *TRIO* database U, *base tables* are uncertain relations and the result of a relational query Q on U is an uncertain relation: in [16] the problem of incorporating outerjoins into uncertain databases is considered but the authors argued that standard possible-worlds semantics may be inappropriate for outerjoins.

[7] The *Trio* source code is freely available and the system is based also on PostgreSQL.

[8] *TRIO*-Provenance is a multilevel (transitive) relationships: formulas specify direct derivations, but when the alternatives in a formula are themselves derived from other alternatives, it is possible to recursively expand a formula until it specifies local tuples only.

the level of global classes, $(Eureka, 1) \vee (Oasis, 1)$, and at the level of local classes, $(\mathtt{A}^{Eureka} \wedge \mathtt{B}^{Eureka}) \vee \mathtt{A}^{Oasis}$.

A relevant application and extension of the *Trio* system to query conflicting data, is to use the possible instances of a query to provide the user with different *search strategies* for querying the Global Schema. As an example (the user is searching for the types of the varieties with a yield value greater than 11):

```
HIGH_PROD_TYPE =
    SELECT DISTINCT Type
    FROM GERMPLASM
    WHERE Yield > 11
```

HIGH_PROD_TYPE as an *uncertain* relation

Type	
line	?
cultivar	
landrace || cultivar	?

The user, by interacting with the framework, might obtain: At first, only *consistent global tuples* coming from non-conflicting local tuples are viewed : { cultivar }. Then, after the application of conflict handling strategies (`yield` solved with `AVG`, `Type` considered as an uncertain attribute) the uncertain relation `HIGH_PROD_TYPE`, provides two different answers: (1) tuples in *every* possible instance: { cultivar,line }; (2) tuples in *some* possible instance: { cultivar,line,landrace}.

4 Conclusion and Future Work

In this paper we presented the ongoing research on the design and development of a Provenance Management component, $\mathrm{PM}_{\mathrm{MOMIS}}$, for the MOMIS System. $\mathrm{PM}_{\mathrm{MOMIS}}$ functionalities have been studied and partially implemented in the domain of genotypic and phenotypic cereal-data management within the CEREALAB project.

Several notions of provenance for database queries have been proposed and studied in the past years, see [10] for a survey. Among these approaches, one of the most expressive ones is the *Provenance Semiring* [13]; an extension of the *Provenance Semiring* to schema mappings is used in ORCHESTRA data sharing system [18]; as in our framework, provenance in ORCHESTRA is also used to perform reconciliation based on user preferences on the sources the data come from; moreover, the ORCHESTRA system is being prototyped in applications with biological collaborators [17]. *Provenance Semirings* and ORCHESTRA are then important references for our future work.

Another important references for our future work is the Open Provenance Model (OPM) [22], which aims to define a generic and comprehensive representation of data provenance and which is becoming a standard to share provenance information. OPM represents provenance through a graph; a graphical representation of provenance graph may be used to formulate queries from a visual and intuitive interface, and then enabling end-users (plant scientists) to be able to directly query the provenance information.

An interesting idea of provenance application in the biology community is sketched in [19]: mechanisms similar to the bibliographic citation index for articles and authors are needed to acknowledge ”publication” of datasets in shared databases, so as to encourage rapid, effective sharing of data; data management support for tracking data provenance can provide the analog of citations. Following this idea we are ”ranking” the local sources integrated in the CEREALAB application on the basis of users’ queries.

References

1. Agrawal, P., Benjelloun, O., Sarma, A.D., Hayworth, C., Nabar, S., Sugihara, T., Widom, J.: Trio: a system for data, uncertainty, and lineage. In: VLDB '06. pp. 1151–1154 (2006)
2. Beneventano, D.: Provenance based conflict handling strategies. In: Data Quality in Data Integration Systems, DASFAA-Workshop, 18 April, South Korea (2012), to appear
3. Beneventano, D., Bergamaschi, S., Guerra, F., Vincini, M.: Synthesizing an integrated ontology. IEEE Internet Computing 7(5), 42–51 (2003)
4. Beneventano, D., Dannoui, A.R., Sala, A.: Data lineage in the momis data fusion system. In: ICDE-Workshops, April 11-16, 2011, Hannover, Germany. pp. 53–58 (2011)
5. Benjelloun, O., Sarma, A.D., Halevy, A., Widom, J.: Uldbs: databases with uncertainty and lineage. In: VLDB '06. pp. 953–964. VLDB Endowment (2006)
6. Benjelloun, O., Sarma, A.D., Hayworth, C., Widom, J.: An introduction to uldbs and the trio system. IEEE Data Eng. Bull. 29(1), 5–16 (2006)
7. Bergamaschi, S., Beneventano, D., Guerra, F., Orsini, M.: Data integration. In: Handbook of Conceptual Modeling: Theory, Practice and Research Challenges. Springer Verlag (2011)
8. Bergamaschi, S., Castano, S., Vincini, M., Beneventano, D.: Semantic integration of heterogeneous information sources. Data Knowl. Eng. 36(3), 215–249 (2001)
9. Bleiholder, J., Naumann, F.: Data fusion. ACM Comput. Surv. 41(1), 1–41 (2008)
10. Cheney, J., Chiticariu, L., Tan, W.C.: Provenance in databases: Why, how, and where. Foundations and Trends in Databases 1(4), 379–474 (2009)
11. Cui, Y., Widom, J., Wiener, J.L.: Tracing the lineage of view data in a warehousing environment. ACM Trans. Database Syst. 25(2), 179–227 (2000)
12. Glavic, B., Alonso, G.: Perm: Processing provenance and data on the same data model through query rewriting. In: ICDE '09. pp. 174–185 (2009)
13. Green, T.J., Karvounarakis, G., Tannen, V.: Provenance semirings. In: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 31–40. PODS '07, ACM, New York, NY, USA (2007)
14. Halevy, A., Li, C.: Information integration research: Summary of nsf idm workshop breakout session. NSF IDM Workshop (2003)
15. Halevy, A., Rajaraman, A., Ordille, J.: Data integration: the teenage years. In: VLDB '06. pp. 9–16. VLDB Endowment (2006)
16. Ikeda, R., Widom, J.: Outerjoins in uncertain databases. In: Management of Uncertain Data (MUD). Stanford InfoLab (2009), `http://ilpubs.stanford.edu:8090/925/`
17. Ives, Z.G.: Data integration and exchange for scientific collaboration. In: Proceedings of the 6th International Workshop on Data Integration in the Life Sciences. pp. 1–4. DILS '09, Springer-Verlag, Berlin, Heidelberg (2009), `http://dx.doi.org/10.1007/978-3-642-02879-3_1`
18. Ives, Z.G., Green, T.J., Karvounarakis, G., Taylor, N.E., Tannen, V., Talukdar, P.P., Jacob, M., Pereira, F.: The orchestra collaborative data sharing system. SIGMOD Rec. 37(3), 26–32 (Sep 2008), `http://doi.acm.org/10.1145/1462571.1462577`
19. Jagadish, H.V., Olken, F.: Database management for life sciences research. SIGMOD Rec. 33, 15–20 (June 2004), `http://doi.acm.org/10.1145/1024694.1024697`
20. Lenzerini, M.: Data integration: A theoretical perspective. In: PODS. pp. 233–246 (2002)
21. Milc, J., Sala, A., Bergamaschi, S., Pecchioni, N.: A genotypic and phenotypic information source for marker-assisted selection of cereals: the cerealab database. Database (2011)
22. Moreau, L., Freire, J., Futrelle, J., Mcgrath, R.E., Myers, J., Paulson, P.: The open provenance model: An overview. In: Provenance and Annotation of Data and Processes, pp. 323–326. Springer-Verlag (2008)
23. Naumann, F., Freytag, J.C., Leser, U.: Completeness of integrated information sources. Inf. Syst. 29(7), 583–615 (2004)

Nicola Ferro and Letizia Tanca (Eds.): SEBD 2012, Edizioni Libreria Progetto, Padova, Italia
ISBN: 978-88-96477-23-6, Copyright (c) 2012 - Edizioni Libreria Progetto and the authors