An Approach for Synergically Carrying out Intensional and Extensional Integration of Data Sources Having Different Formats

Luigi Pontieri[1], Domenico Ursino[2], and Ester Zumpano[3]

[1] ISI, Consiglio Nazionale delle Ricerche
Via Pietro Bucci, 87036 Rende (CS), Italy
pontieri@si.deis.unical.it
[2] DIMET, Università di Reggio Calabria
Via Graziella, 89060 Reggio Calabria, Italy
ursino@ing.unirc.it
[3] DEIS, Università della Calabria
Via Pietro Bucci, 87036 Rende (CS), Italy
zumpano@si.deis.unical.it

Abstract. In this paper we propose a data source integration approach capable of uniformly handling different source formats, ranging from databases to XML documents and other semi-structured data. The proposed approach consists of two components, performing intensional and extensional integration, respectively; these are strictly coupled, since they use related models for representing intensional and extensional information and are synergic in their behaviour.

1 Introduction

The activity of integrating different data sources involves two different levels, namely the intensional and the extensional ones. As a matter of fact, these two forms of integration consider two different, yet complementary, aspects of the problem. In more detail, *Intensional Integration* [1,4,5] concerns with the activity of combining *schemes* of involved data sources for obtaining a global scheme representing all of them. *Extensional Integration* [2,3] is the activity of producing (either virtual or materialized) data representing all *instances* present in the involved data sources. In [5] an *intensional integration* technique, based on a model, called *SDR-Network* [6], capable of uniformly handling data sources with different representation formats, has been presented. In this paper we propose a technique for carrying out *extensional integration* of data sources with different formats. In order to uniformly represent and handle instances of involved data sources, our technique exploits a logic model, called *E-SDR-Network*, which is the counterpart, at the extensional level, of the SDR-Network conceptual model. The definition of an extensional integration technique, whose behaviour and reference model are strictly related to those ones relative to the intensional integration technique proposed in [5], allows us to obtain, in the whole, an approach

A. Banks Pidduck et al. (Eds.): CAISE 2002, LNCS 2348, pp. 752–756, 2002.

consisting of two components, synergically carrying out the intensional and the extensional integration, respectively. Interestingly enough, our extensional integration technique is also capable of handling null or unknown values. In addition, it is able to reconstruct, at the extensional level, the information sources which an integrated E-SDR-Network has been derived from; this important feature is directly inherited from the specific properties of the E-SDR-Network model. Last, but not the least, our technique is capable of producing consistent query answers from inconsistent data.

2 The E-SDR-Network Logic Model

The E-SDR-Network logic model extends, at the extensional level, the SDR-Network conceptual model [6], conceived to operate at the intensional one. An E-SDR-Network $E\_Net(DS)$ representing, at the extensional level, a data source DS, is a rooted directed labeled graph:

$$E\_Net(DS) = \langle E\_NS(DS), E\_AS(DS)\rangle = \langle E\_NS_A(DS) \cup E\_NS_C(DS), E\_AS(DS)\rangle$$

Here, $E\_NS(DS)$ is a set of nodes, each representing an instance of a concept of DS. Nodes in $E\_NS(DS)$ are subdivided in two subsets, namely, the set of *atomic nodes* $E\_NS_A(DS)$ and the set of *complex nodes* $E\_NS_C(DS)$. A node is atomic if it has no outgoing arcs, complex otherwise. Each atomic node $N_A \in E\_NS_A(DS)$ is an instance of an atomic node of the corresponding SDR-Network and represents a value; it is identified by a pair $[N, V]$, where N is the name of the SDR-Network node which N_A is an instance of, and V is the value which N_A represents. Each complex node $N_C \in E\_NS_C(DS)$ is an instance of a complex node in the corresponding SDR-Network; it is identified by a pair $[N, I]$, where N is the name of the SDR-Network node which N_C is an instance of, and I is an identifier allowing to uniquely single out N_C among all nodes of $E\_NS_C(DS)$.

$E\_AS(DS)$ denotes a set of arcs; an E-SDR-Network arc represents a relationship between two E-SDR-Network nodes; in addition, it is an instance of an SDR-Network arc. An E-SDR-Network arc A from S to T, labeled L_{ST} and denoted by $\langle S, T, L_{ST}\rangle$, indicates that the instance represented by S is related to the instance denoted by T. S is called the *source node* of A whereas T represents its *target node*. At most one arc may exist from S to T. The label L_{ST} of A has been conceived for supporting the integration task. In particular, L_{ST} is a set; if $E\_Net(DS)$ does not derive from an integration task then L_{ST} is equal to $\{E\_Net(DS)\}$. Vice versa, if $E\_Net(DS)$ has been obtained from the integration of some E-SDR-Networks, then A derives from the merge of some E-SDR-Network arcs, each belonging to one of the E-SDR-Networks that have been integrated[1]; in this case L_{ST} is equal to the set of E-SDR-Networks containing the arcs which have been merged to obtain A. L_{ST} plays a relevant role in the integration task since it stores information about the original sources of

[1] Observe that, for some of these E-SDR-Networks, it could not exist an arc taking part to the construction of A.

integrated data. It provides the extensional integration technique with the capability of reconstructing the original contents of data sources involved in the integration task; as we pointed out in the Introduction, this is one of the most interesting features of our technique.

Observe that any data source can be generally represented, at the extensional level, as a set of instances and a set of relationships among instances. Since E-SDR-Network nodes and arcs are well suited to represent both instances and their relationships, the E-SDR-Network can be used to uniformly model, at the extensional level, most existing data sources.

3 Extensional Integration Algorithm

Our Extensional Integration Algorithm takes in input: *(i)* a global SDR-Network SDR_G, obtained by applying the Intensional Integration Algorithm, described in [5], on two SDR-Networks SDR_1 and SDR_2;*(ii)* two E-SDR-Networks $ESDR_1$ and $ESDR_2$ such that $ESDR_1$ (resp., $ESDR_2$) is an E-SDR-Network corresponding to SDR_1 (resp, SDR_2). It returns a global E-SDR-Network $ESDR_G$ which corresponds, at the extensional level, to the global SDR-Network SDR_G. The algorithm first examines all SDR_G nodes; let N_G be one of them. If N_G derives from the merge of two nodes N_1 of SDR_1 and N_2 of SDR_2, the algorithm determines the set of nodes $ENSet_1$ of $ESDR_1$ and $ENSet_2$ of $ESDR_2$ corresponding, at the extensional level, to N_1 and N_2; then it transforms all nodes of $ENSet_1$ and $ENSet_2$ so that they become instances of N_G in $ESDR_G$. If there exist two nodes $EN_1 \in ENSet_1$ and $EN_2 \in ENSet_2$ representing the same real world instance, they are merged for obtaining a unique $ESDR_G$ node EN_G being an instance of the SDR_G node N_G; arcs incoming to and outgoing from EN_1 and EN_2 are transferred to EN_G. If N_G derives from only one node N of either SDR_1 or SDR_2, the algorithm determines the set of nodes $ENSet$ corresponding, at the extensional level, to N and transforms each node $EN_i \in ENSet$ so that it becomes an instance EN_G of N_G in $ESDR_G$. Arcs incoming to and outgoing from EN_i are transferred to EN_G. By applying these operations to all SDR_G nodes, the algorithm constructs all nodes and arcs of $ESDR_G$. After these operations, it could be possible that two arcs exist in $ESDR_G$ from a node EN_S to a node EN_T; if this happens, they must be merged. The Extensional Integration Algorithm can be encoded as follows:

Algorithm for the Extensional Integration of two data sources

```
Input: a global SDR-Network $SDR_G$, obtained from the integration of two
          SDR-Networks $SDR_1$ and $SDR_2$; two E-SDR-Networks $ESDR_1$ and
          $ESDR_2$ corresponding to $SDR_1$ and $SDR_2$, resp.;
Output: a global E-SDR-Network $ESDR_G$;
var
    $NSet$: a set of SDR-Network nodes; $N_G$: an SDR-Network node;
    $ENSet$: a set of E-SDR-Network nodes;
    $EN_S, EN_T$: an E-SDR-Network node;
    $EASet$: a set of E-SDR-Network arcs; $A_1, A_2$: an SDR-Network arc;
```

```
begin
    $NSet := Get\_Nodes(SDR_G)$;
    for each $N_G \in NSet$ do
        $Construct\_Node\_Instances(N_G, ESDR_1, ESDR_2, ESDR_G)$;
    $ENSet := E\_Get\_Nodes(ESDR_G)$;
    for each $EN_S \in ENSet$ do
        for each $EN_T \in ENSet$ do begin
            $EASet := E\_Get\_Arcs(EN_S, EN_T)$;
            if $(EASet = \{A_1, A_2\})$ then $E\_Merge\_Arcs(A_1, A_2, ESDR_G)$
        end
end
```

The presence of arc labels in the E-SDR-Network logic model provides our algorithm with the capability to reconstruct, at the extensional level, the information sources which an integrated E-SDR-Network has been derived from. This is an important feature "per se", but it becomes even more relevant in that it allows to produce consistent query answers from inconsistent data. Indeed, the user can be requested to associate a reliability degree with each data source; whenever she/he poses a query involving inconsistent data, derived from different sources, those coming from the source which she/he has considered more reliable are taken. Finally, our algorithm is capable to handle null or unknown values. Indeed, whenever two complex instances, say EN_1 and EN_2, relative to two E-SDR-Networks $ESDR_1$ and $ESDR_2$, and representing the same real world instance, are merged for obtaining a global instance EN_G of $ESDR_G$, if EN_1 (resp., EN_2) has a property, say P, having a null or unknown value, and EN_2 (resp., EN_1) has a specific value, say V, for P, then V will appear as the value assumed by P into EN_G. Interestingly enough, our technique allows to substitute the null or unknown value, which P had in EN_1 (resp., EN_2), with the value V, which P assumed in EN_2 (resp., EN_1). In this way it is able to restore the missing information relative to a given source taking part to the integration task.

References

1. S. Castano, V. De Antonellis, and S. De Capitani di Vimercati. Global viewing of heterogeneous data sources. *Transactions on Data and Knowledge Engineering*, 13(2), 2001. 752
2. Y. Kanza, W. Nutt, and Y. Sagiv. Queries with incomplete answers over semistructured data. In *Proc. of Symposium on Principles of Database Systems*, pages 227–236. 752
3. M. Liu, T.W. Ling, and T. Guang. Integration of semistructured data with partial and inconsistent information. In *Proc. of International Database Engineering and Applications Symposium*, pages 44–52. 752
4. J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with cupid. In *Proc. of International Conference on Very Large Data Bases*, pages 49–58. 752
5. D. Rosaci, G. Terracina, and D. Ursino. A semi-automatic technique for constructing a global representation of information sources having different formats and structures. In *Proc. of International Conference on Database and Expert Systems Applications*, pages 734–743. 752, 754

6. G. Terracina and D. Ursino. Deriving synonymies and homonymies of object classes in semi-structured information sources. In *Proc. of International Conference on Management of Data*, pages 21–32. 752, 753