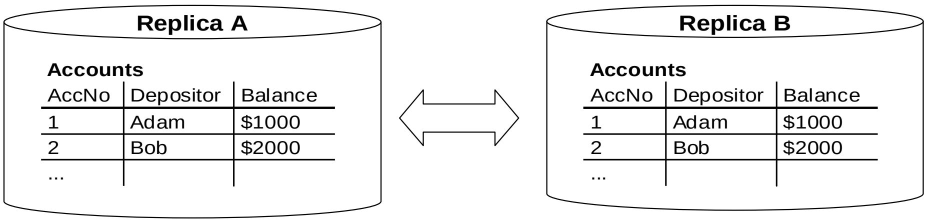
AG Heterogene Informationssysteme
Prof. Dr.-Ing. Stefan Deßloch
Fachbereich Informatik
Technische Universität Kaiserslautern

Middleware for Heterogeneous and Distributed Information Systems – Exercise Sheet 10

Wednesday, January 14, 2009 – 10:00 to 11:30 – Room 48-379

Data Replication



Replica A

Accounts

AccNo	Depositor	Balance
1	Adam	$1000
2	Bob	$2000
...		

Replica B

Accounts

AccNo	Depositor	Balance
1	Adam	$1000
2	Bob	$2000
...		

Figure 1: Sample Replicas

Data is replicated to multiple network nodes to improve query response times and increase data availability. Consider the sample replicas depicted in Figure 1. Each replica provides a relation storing bank account information. Assume that the replicas can be accessed concurrently (*update anywhere* model); hence a mechanism to enforce the consistency of the replicas is required.

1. In class two replication approaches have been distinguished, namely eager replication and lazy replication. Briefly explain each of these approaches and describe the differences!

2. Say, eager replication is used to synchronize the sample replicas depicted in Figure 1. Describe the progress and the outcome of the following operations.

   a. The amount of $100 is transferred from account 1 to account 2 at replica A.

   b. The amount of $100 is transferred from account 1 to account 2 at replica A. At the same time the amount of $200 is transferred from account 1 to account 2 at replica B.

   c. The amount of $100 is transferred from account 1 to account 2 at replica A. At the same time the amount of $200 is transferred from account 2 to account 1 at replica B.

3. Say, lazy replication is used to synchronize the sample replicas depicted in Figure 1. Describe the progress and the outcome of the following operations.

   a. The amount of $100 is transferred from account 1 to account 2 at replica A.

b. The amount of $100 is transferred from account 1 to account 2 at replica A. At the same time the amount of $200 is transferred from account 1 to account 2 at replica B.

c. The amount of $100 is transferred from account 1 to account 2 at replica A. At the same time the amount of $200 is transferred from account 2 to account 1 at replica B.

d. How can replication conflicts be detected? How can reconciliation be performed?

4. Replication conflicts can be avoided if serializability is abandoned for the convergence property (using so called non-transactional replication schemes). That is, if no new transactions arrive all replicas will converge to the same state after exchanging replica updates.

   In general this property cannot be achieved unless global serialization techniques are used. However, adding and subtracting constants from numeric values, for instance, are so called commutative updates, i.e. they lead to the same result irrespective of the order in which they are applied. How can this property be exploited to achieve convergence in the sample scenario? What problems are incurred by this non-transactional replication scheme that can be prevented using eager replication?

5. Compare eager replication, lazy replication, and non-transactional replication schemes! What are the advantages? What are the drawbacks?

Schema Matching

Schema Matching aims at identifying semantically related elements across different schemas. The result of schema matching is referred to as *mapping*. The Cupid schema matching algorithm[1] discovers mappings between schema elements based on their names, data types, constraints, and schema structure, using a broad set of techniques. Cupid is an attempt to address the schema matching problem in a generic way.

1. Initially, Cupid performs *linguistic matching*. Linguistic Matching proceeds in three steps, namely normalization, categorization, and comparison. Explain these steps and give examples!
2. Figure 2 depicts two sample schemas and their linguistic similarity coefficients computed by linguistic matching. What weak points of linguistic matching approaches become here?
3. As a second step, Cupid performs *structure matching*. For that purpose, Cupid applies the so called TreeMatch algorithm. Apply TreeMatch to the sample schemas depicted in Figure 1 with th_{accept} = 0.5, w_{struct} = 0.7, th_{high}= 0.7, c_{inc} = 1.2, th_{low} = 0.4, and c_{dec} = 0.5!
4. The last step of Cupid's schema matching process is the *mapping generation*. What mapping is created using the acceptance threshold above?

[1] Jayant Madhavan, Philip A. Bernstein, Erhard Rahm: Generic Schema Matching with Cupid. VLDB 2001, pages 49-58

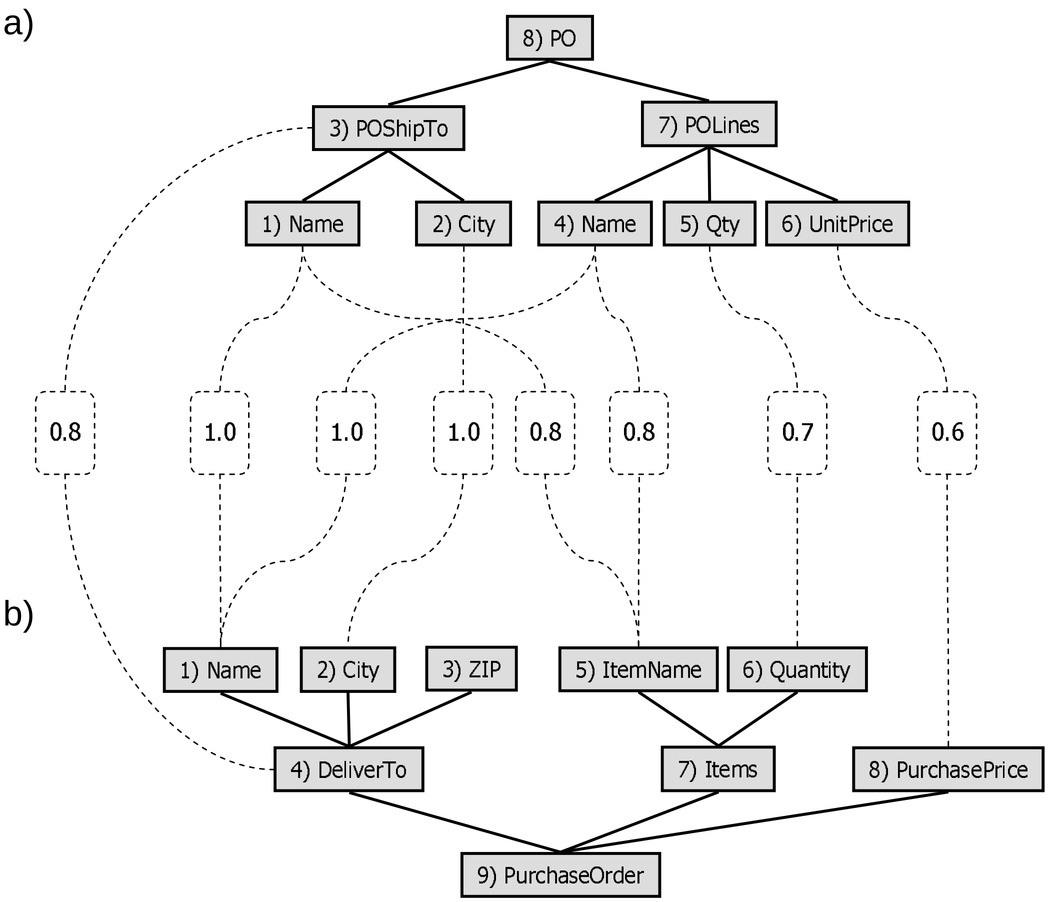


Figure 2: Sample schemas after linguistic schema matching (matches with confidence 0.0 omitted)