VALUE NAME CONFLICT WHILE INTEGRATING DATA INDATABASE INTEGRATION

GHULAM ALI MIRZA[1]

[1]Department of Computer Science , Bahria University, Islamabad, Pakistan
E-MAIL: alimirza@bahria.edu.pk,ghulamali84@hotmail.com

Abstract:

Integration of data is performed to provide user a unified view of heterogeneous data sources residing on different places. Major challenge in data integration process is; first identify and then resolve the data level conflicts. Ignoring identification and resolution of these conflicts produces inconsistent and erroneous data. In this paper we identify and resolve a new data level conflict. Data in different data sources may have similar semantics with different values. This difference in data may create inconsistent results when queried. Value of one concept may be present with different name in two different databases. These types of conflict need to be resolved for consistent and error free data. We named this conflict as Value Name Conflict. We provide the formal definition of this conflict using Z-Notation. Formal resolution function of this conflict is also presented.

Keywords:

Database Integration; Data Integration; Data Merging; Data Level Conflicts; Data Fusion; Data Conflicts; Data Heterogeneities; Value Name Conflict

1. Introduction

Database integration is a process that provides unified view over the data present in heterogeneous databases [1]. Heterogeneous data located on different locations is retrieved in many applications [2]. Different data sources located on different places across the world and many developed organizations need these data sources for their applications for accessing various types of information [3]. This information is useful when it is merged according to the requirement of user and when theses databases are integrated, the information gets complete. Requirement of end user is error free data with appropriate representation. End user of these systems does not know about the local sources of data. Successful integration can only be achieved when conflicts are resolved that arise during the integration of databases. So the core part of integration is identification and then resolution of these conflicts.

Database integration has two core steps; integration of schema and next step is integration of data. Integration of schemas also known as global schema. After successful schema integration the next phase is integration of real data that resides under the schema in data sources, called the data integration. Figure 1 show how heterogeneous data is accessed in database integration system is shown. Steps of query passing and retrieval of results are also described in this figure. Query is posed on global schema considering that the results will be gathered and integrated from local databases underplaying the global schema. The global schema, in fact is just a mapping of the local database. It does not have any data.

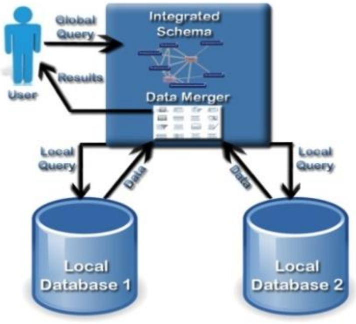


Fig.1 Heterogeneous Database Integration System

When user poses a query on global databases it divided into local queries and posed on local databases where it executes and results from both the databases are sent back to the global schema. Global schema processes the retrieved data and presets to the user. When data is being merged for user, different data conflicts arise and need to be resolved for a consistent view for user.

Major challenge in database integration is identification and resolution of conflicts [4]. These conflicts may arise while integration of schema and data separately. Identification and integration of similar schema elements is performed during the schema integration which has its own complexities, and the integration of data retrieved from local databases becomes problematic during the data integration process [5]. That is why the integration of schema and data require separate processes. Major

978-1-4799-7208-1/14/$31.00 ©2014 IEEE

contribution is reported for integration of schemas among heterogeneous data sources but less significance is given to identification and resolution of conflicts that arise during the data integration process [6][7][8]. The resolution of these conflicts is treated by the researchers along with the schematic conflicts during integration of schema. But not all data level conflicts are identified and resolved during this phase. Deeper look in to the literature also reports that the researchers work on integration systems and algorithms but data conflicts are not been focused that may arise during the integration of data [3][9][10][11][12][13].

Some of the data level conflicts arise when query is posed and result is retrieved and merged in data integration phase but the integrated data may contain erroneous, inconsistent or incomplete information because of data level conflicts. One of these conflicts that may cause this type of abnormality in information after integration of data is Value Name Conflict which is focus of our discussion in this paper. We define two types of this conflict and present a scenario in which this conflict arises. We also present this conflicts and its type formally using Z-Notation. We derived an algorithm that integrates the data from two different data sources and resolves this conflict. Resolution function of this conflict is also presented using Z-Notation. This paper consists of section II with related work done until now for data level conflicts, section III the definition of Value Name conflicts and its types with examples. Formal representation and resolution of this conflict is presented in section IV. In section V, we conclude the work and discuses the future perspectives.

2. Related Work

Data level conflicts have not been focus of researchers. These are discussed along with schema integration conflicts or just mentioned in interoperability among the database systems. Less mature work is found that directly focus on data level conflicts. Bleiholder et al. resolved the data uncertainty problem [14]. Uncertainty means tuples have null values. Resolution is performed using operators and algorithms. Another operator is proposed by them in [15] for resolving the same problem. Improvement of these operators is presented in [16] for elimination of null value conflict in data integration. Similarly, null value conflict is also handled in [7] by Luna et al. Mutual inconsistencies in different data sources is resolved by Luna et al in [2]. Another way of resolution of null value conflict is presented in [17] by Naumann et al. Their developed tool is presented in [18] called HumMer. This Tool works for duplication detection and uses an algorithm proposed in their work in [19].

Few data level conflicts such as Data-value conflicts, Data representation conflicts, Data-unit conflicts, Data precision conflicts are presented in [20]. Semantic Conflict Resolution Ontology (SCROL) is presented in [21]. SCROL mainly capture the following data level conflicts: Data value conflict, Data representation, Data unit conflict, Data precision conflict, Known data value reliability conflicts and Spatial domain conflict. Identification of data level conflicts with examples is presented in [22]. Identified data level conflicts are: Data value concepts, Data representation conflicts, Data unit conflicts, Data precision conflicts and Granularity of the information unit. Resolution of these conflicts is also implemented in their system called PEPS. Semantic taxonomy of schema and data level conflicts is also presented in [4]. Taxonomy included the following data level conflicts: Data Representation Conflicts, Data Scaling Conflicts, Data Precision Conflicts, Default Value Conflicts, and Attribute Integrity Constraint Conflicts. Classification of semantic conflicts is also discussed in [23]. Detailed survey on data level conflicts is in [24]. Formalization of Null Value Conflict is also presented in [27].

Deeper look into the literature shows that less concentration is given to data level conflicts. Survey shows that most of the researchers have discussed similar conflicts. These conflicts are mostly discussed when schema integration techniques are discussed for heterogeneous data sources. In this section we presented the data level conflicts identified by researchers, algorithms and system they developed to resolve them. In next section we present a new Value Name Conflict, its types and situations in which this conflict occurs.

3. Value Name Conflict

Value Name Conflict arises when user poses a query on global schema and data from local databases is not retrieved or retrieved incorrectly. Now question is why this conflict arises? It arises because semantically similar data is present in local databases with different values or data has same values and different semantics and when user poses a query with specific criteria, data having similar semantic is not retrieved or semantically different data with same value is retrieved. For better understanding we divide this conflict into two part, synonym conflict and homograph conflict.

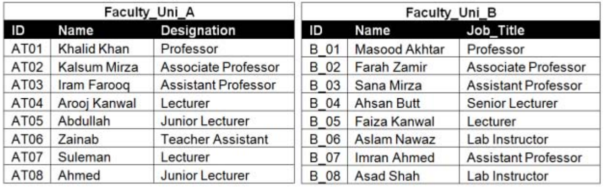
Synonym Conflict: In synonym conflict, a value is present in two different databases with different names but these values present same concept. When query is posed on global schema, data from one of the local databases will not be retrieved.

Consider the following schema tables given below (Faculty_ Uni_A and Faculty_ Uni_B) of Faculty Designations in two different universities as local databases.

These two schemas are integrated and their mapping is given in global table (Global_Faculty).

Faculty_Uni_A (ID, Name, Designation)
Faculty_Uni_B (ID, Name, Job_Title)
Global_Faculty (ID, Name, Designation)

Data tables of above given schemas are shown in Figure 2. Suppose a query is posed on global table to fetch the name of all Lab Instructors; query is transformed to the local databases and names are obtained from only table Faculty_Uni_B which is in this case are Aslam Nawaz and Asad Shah. Faculty_Uni_A also has the lab instructors but in Uni A lab instructor is called as Junior Lecturer which should have also is retrieved when query is posed but matching is not found in query. This conflict is because of value name of a concept lab instructor.



Faculty_Uni_A		
ID	Name	Designation
AT01	Khalid Khan	Professor
AT02	Kalsum Mirza	Associate Professor
AT03	Iram Farooq	Assistant Professor
AT04	Arooj Kanwal	Lecturer
AT05	Abdullah	Junior Lecturer
AT06	Zainab	Teacher Assistant
AT07	Suleman	Lecturer
AT08	Ahmed	Junior Lecturer

Faculty_Uni_B		
ID	Name	Job_Title
B_01	Masood Akhtar	Professor
B_02	Farah Zamir	Associate Professor
B_03	Sana Mirza	Assistant Professor
B_04	Ahsan Butt	Senior Lecturer
B_05	Faiza Kanwal	Lecturer
B_06	Aslam Nawaz	Lab Instructor
B_07	Imran Ahmed	Assistant Professor
B_08	Asad Shah	Lab Instructor

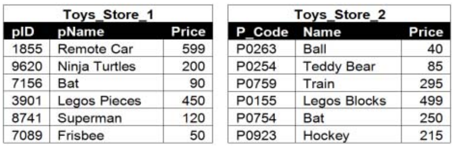
Fig.2 Data Tables with Synonym Conflict

Homograph Conflict: In homograph conflict, a value is present in two different databases with same names but have different concept. When query is posed on global schema, data from any local database will be retrieved from both local databases but one of the values will be incorrect.

Consider the following schema tables given below (Toys_Store_1 and Toys_Store_2) of two different toy stores as local databases. These two schemas are integrated and their mapping is given in global table (Global_Toys_Store).

Toys_Store_1 (pID, pName, Price)
Toys_Store_2 (P_Code, Name, Price)
Global_Toys_Store (pID, pName, Price)

Data tables of above given schemas are shown in Figure 3. Suppose user poses a query on global schema to retrieve a toy called Bat with its price. Requirement of user is a cricket bat. When query is executed on local databases, two records were found; Bat with price 90 and Bat with price 250. Bat with the price 250 is a cricket bat and bat with the price 90 is an animal bat toy. In this case, the animal toy is retrieved from local databases and merged for final result in response of query which is actually incorrect information. Bat is a homograph and it generates conflict when data is integrated.



Toys_Store_1		
pID	pName	Price
1855	Remote Car	599
9620	Ninja Turtles	200
7156	Bat	90
3901	Legos Pieces	450
8741	Superman	120
7089	Frisbee	50

Toys_Store_2		
P_Code	Name	Price
P0263	Ball	40
P0254	Teddy Bear	85
P0759	Train	295
P0155	Legos Blocks	499
P0754	Bat	250
P0923	Hockey	215

Fig.3 Data Tables with Homograph Conflict

In next section, formal representation of these conflict types is presented using Z-Notation.

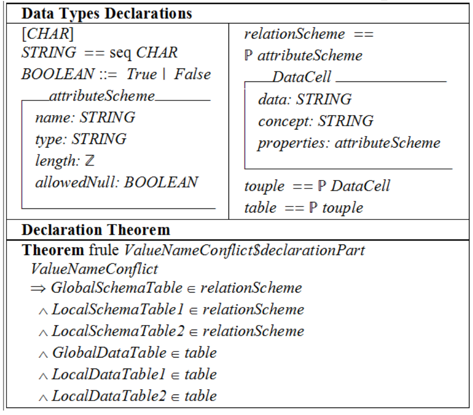
4. Formal Representation and Resolution

Until now, researchers have discussed the data level conflicts textually. The description of the conflicts is provided using text only. We presented the conflict formally for the first time using Z-Notation. Z language has enough functions to describe the state of schemas and well as the situations in which the conflicts arise. State schemas of Z language for representation and resolution of Value Name Conflict is used.

Z/EVES [25] is used for writing the Z notations. This tool is used for writing the definition and resolution of this conflict.

In Figure 4, the formal representation of Value Name Conflicts is presented. First part of the figure shows data type declarations used in conflict representation, attribute Scheme is actually attribute of table that exists at schema level and relation Scheme is set of these attributes. Relation Scheme is actually representation of schema table. DataCell is cell holding the data, touple is defined as set of DataCell and table (data table) is defined as set of touple.

In the second part of the figure, Declaration Theorem is presented for Value Name Conflict Definition which declares the belonging of global and local schema and data tables with relationScheme and table data types. Figure 5 describes the formal representation of Value Name Conflict with both synonym and homograph types. In first portion of the figure, global and local schema and data tables are declared. Second portion has the definition of DataCell belongs to Tuples and Tuples belongs to DataTables. At the end of the figure scenario of synonym and homograph is shown. In synonym conflict, data in data cells has same concept but the value name of data is different. Therefore, data form one cell will be retrieved. In homograph conflict, data in data value has same name but concepts are different.



Data Types Declarations

$[CHAR]$
$STRING == \mathrm{seq}\ CHAR$
$BOOLEAN ::= True \mid False$

attributeScheme
$name: STRING$
$type: STRING$
$length: \mathbb{Z}$
$allowedNull: BOOLEAN$

$relationScheme ==$
$\mathbb{P}\ attributeScheme$

DataCell
$data: STRING$
$concept: STRING$
$properties: attributeScheme$

$touple == \mathbb{P}\ DataCell$
$table == \mathbb{P}\ touple$

Declaration Theorem

Theorem frule *ValueNameConflict$declarationPart*
$ValueNameConflict$
$\Rightarrow GlobalSchemaTable \in relationScheme$
$\land LocalSchemaTable1 \in relationScheme$
$\land LocalSchemaTable2 \in relationScheme$
$\land GlobalDataTable \in table$
$\land LocalDataTable1 \in table$
$\land LocalDataTable2 \in table$

Fig.4 Data Types and Theorem Declarations

Therefore, both data cells will be merged and incorrect information is retrieved.

We represent the resolution of this conflict formally using Z-notation as well. Figure 6 is consists of resolution. First part of resolution has the declaration of GlobalDataTable,andLocalDataTable1and LocalDataTable2. In second part, touple and DataCell are declared and their relations with the declared items are also presented. At the end, for synonym conflict resolution part explains that, if data in data cells is different and concepts are same then merge both data cells with global table. For homograph conflict, if data in data table is same and their concepts are different then one of the cells will be merged with global table.

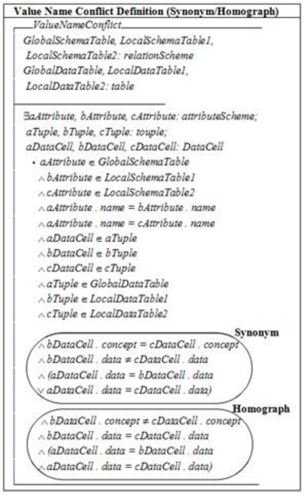
Now a question arises, how we can implement the identification and resolution of this type of conflict in integrated database system. One of the possible solutions to this problem is use of WordNet [26] thesaurus.

WordNet can be used to find the synonyms as well as the homographs of a word and after parsing this information from thesaurus we can decide whether the data from both local databases should be merged or not. But not all synonym concepts can be found in WordNet. In section III, we discussed the issue arise in concept of Lab Instructor and Junior Lecturer, this types of synonym concepts cant not be found in WordNet. We therefore plan to implement the resolution of this function through the help of ontology.

DHResol is our tool having the implementation of this concept. This system takes XML converted integrated schema and local schemas. User poses query on system, system converts the query in to the local queries, fetches the data from local databases, and checks the conflict and integrates the resultant data. Currently, this system resolves the null value conflict among the data columns. System creates the dynamic connections with SQL Server databases after parsing the information from input XML files. New version of this system will be able to take the ontology containing the mapping of concepts present in databases. System will parse the information of concepts from ontology and will resolve the conflict.

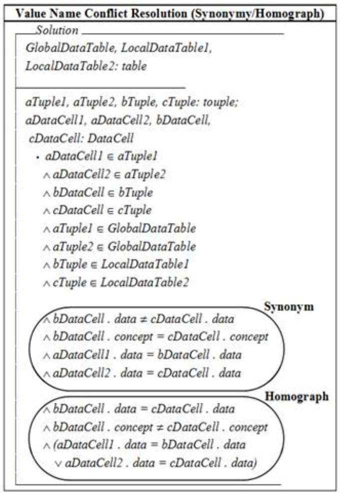
5. Conclusion and Future Work

Identification and Resolution of data level conflicts are necessary for integration. In this paper, we presented a new data level conflict in data integration. Formal definition with resolution function is presented using Z-Notation for the first time. We are implementing this solution in our system DHResol. We plan to present the complete solution of this conflict in our system in future. We are also working on identification of more data level conflicts with formal representation and resolution using Z-Notation. We are defining more conflicts and implementing them in DHResol.



Value Name Conflict Definition (Synonym/Homograph)

```
ValueNameConflict
GlobalSchemaTable, LocalSchemaTable1,
 LocalSchemaTable2: relationScheme
GlobalDataTable, LocalDataTable1,
 LocalDataTable2: table
──────────
∃aAttribute, bAttribute, cAttribute: attributeScheme;
 aTuple, bTuple, cTuple: touple;
 aDataCell, bDataCell, cDataCell: DataCell
  • aAttribute ∈ GlobalSchemaTable
   ∧ bAttribute ∈ LocalSchemaTable1
   ∧ cAttribute ∈ LocalSchemaTable2
   ∧ aAttribute . name = bAttribute . name
   ∧ aAttribute . name = cAttribute . name
   ∧ aDataCell ∈ aTuple
   ∧ bDataCell ∈ bTuple
   ∧ cDataCell ∈ cTuple
   ∧ aTuple ∈ GlobalDataTable
   ∧ bTuple ∈ LocalDataTable1
   ∧ cTuple ∈ LocalDataTable2
                                              Synonym
   ∧ bDataCell . concept = cDataCell . concept
   ∧ bDataCell . data ≠ cDataCell . data
   ∧ (aDataCell . data = bDataCell . data
   ∨ aDataCell . data = cDataCell . data)
                                              Homograph
   ∧ bDataCell . concept ≠ cDataCell . concept
   ∧ bDataCell . data = cDataCell . data
   ∧ (aDataCell . data = bDataCell . data
   ∧ aDataCell . data = cDataCell . data)
```



Value Name Conflict Resolution (Synonymy/Homograph)

```
Solution
GlobalDataTable, LocalDataTable1,
LocalDataTable2: table
──────────
aTuple1, aTuple2, bTuple, cTuple: touple;
aDataCell1, aDataCell2, bDataCell,
 cDataCell: DataCell
  • aDataCell1 ∈ aTuple1
   ∧ aDataCell2 ∈ aTuple2
   ∧ bDataCell ∈ bTuple
   ∧ cDataCell ∈ cTuple
   ∧ aTuple1 ∈ GlobalDataTable
   ∧ aTuple2 ∈ GlobalDataTable
   ∧ bTuple ∈ LocalDataTable1
   ∧ cTuple ∈ LocalDataTable2
                                              Synonym
   ∧ bDataCell . data ≠ cDataCell . data
   ∧ bDataCell . concept = cDataCell . concept
   ∧ aDataCell1 . data = bDataCell . data
   ∧ aDataCell2 . data = cDataCell . data
                                              Homograph
   ∧ bDataCell . data = cDataCell . data
   ∧ bDataCell . concept ≠ cDataCell . concept
   ∧ (aDataCell1 . data = bDataCell . data
     ∨ aDataCell2 . data = cDataCell . data)
```

Fig.5 Value Name Conflict Definition

Fig.6 Resolution Function for Value Name Conflict

References

[1] M. Gagnon, "Ontology-based Integration of Data Sources", 10th International Conference on Information Fusion, Quebec,QC,Canada, 2007.

[2] X. L. Dong , F Naumann, "Resolving Data Conflicts for Integration", VLDB '09, 2428, Lyon, France, 2009.

[3] N. Tatbul, "Streaming Data Integration: Challenges and Opportunities", IEEE ICDE International Workshop on New Trends in Information Integration (NTII'10), Long Beach, CA, 2010.

[4] A. Sheth , V. Kashyap, "So Far Schematically yet So Near Semantically", IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5), North-Holland Publishing Co, Amsterdam, Netherlands, 1993.

[5] A. Halevy, A. Rajaraman, J. Ordille, "Data integration: The teenage years", VLDB `06, Seoul, Korea, 2006.

[6] A.Radwan,L.Popa,I.R.Stanoi,A.A.Younis,"Top-k generation of integrated schemas based on directed and weighted correspondences",SIGMOD Conference, 2009.

[7] X. L. Dong, L. BertiEquille,D. Srivastava, "Integrating Conflicting Data: The Role of Source Dependence", VLDB '09, 2428, Lyon, France, 2009.

[8] K. Ahmad, H. K. Chiew, R. Samad, "Intelligent Schema Integrator (ISI): A Tool to Solve the Problem of Naming Conflict for Schema Integration",

International Conference on Electrical Engineering and Informatics, Bandung, Indonesia, 2011.

[9] S. Biffl, W. D. Sunindyo, T. Mose, "Semantic Integration of Heterogeneous Data Sources for Monitoring Frequent-Release Software Projects", International Conference on Complex, Intelligent and Software Intensive Systems, 2010.

[10] A. Rajabifard, "Data Integration and Interoperability of Systems and Data", 2nd Preparatory Meeting of the Proposed UN Committee on Global Geographic Information Management, New York, USA, 2010.

[11] L. BertiEquille, A. D. Sarma, X. L. Dong, A. e. Marian, D. Srivastava, "Sailing the information ocean with awareness of currents: Discovery and application of source dependence", CIDR, 2009.

[12] A. L. Guido, R. Paiano, "Semantic Integration of Information Systems", International Journal of Computer Networks & Communications (IJCNC), vol. Vol. 2, No. 1, 2010.

[13] T. Hao, C. Hao, L. Ying, S. Hongzhou, "Online application of science and technology program oriented distributed heterogeneous data integration", Computer Research and Development (ICCRD), 2011 3rd International Conference on Computer Research and Development, Beijing, China 2011.

[14] J. Bleiholder, S. Szott, M. Herschel, F. Naumann, "Complement union for data integration", Data Engineering Workshops (ICDEW), 2010 IEEE 26th International Conference on Digital Object Identifier: 10.1109/ICDEW.2010.5452760, 2010.

[15] J. Bleiholder, S. Szott, M. Herschel, F. Kaufer, F. Naumann, "Subsumption and complementation as data fusion operators", EDBT '10: Proceedings of the 13th International Conference on Extending Database Technology 2010.

[16] J.Bleiholder,M.Herschel,F.Naumann,"Eliminating NULLs with Subsumption and Complementation", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,vol.34,pp.18-25, 2011.

[17] F. Naumann ,M. Häussler, "Declarative Data Merging with Conflict Resolution", International Conference on Information Quality, 2002.

[18] L. Bilke, J. Bleiholder,C. Bohm ,K. Draba, "Automatic Data Fusion with HumMer", 31st VLDB Conference, Trondheim, Norway, 2005.

[19] M. Weis ,F. Naumann, "DogmatiX tracks down duplicates in XML", ACM International Conference on Management of Data (SIGMOD), Baltimore,MD, 2005.

[20] J. Park,S. Ram,"Information systems interoperability: What lies beneath?",ACM Transactions on Information Systems, Vol 22, No. 4, pp. pp. 595-632, 2004.

[21] S. Ram ,J. Park, "Semantic conflict resolution ontology (SCROL): An ontology for detecting and resolving data- and schema-level semantic Conflicts", IEEE Trans. Knowl. Data Eng.16, vol. 2, pp. 189-202, 2004.

[22] Peristeras, V, Loutas N, Goudos S, Tarabanis K, "Semantic Interoperability Conflicts in Pan-European Public Services", 15th European Conference on Information Systems, St. Gallen, Switzerland, 2007.

[23] C. E. Naiman ,A.M. Ouksel, "A Classification of Semantic Conflicts in Heterogeneous Database Systems", journal of organizational computing, vol. 5(2), pp. 167-193, 1995.

[24] G.A. Mirza, N. Masood, S. Asghar, “A Survey of Data Level Conflicts in Database Integration”, 4th SKIMA Conference, Paro, Bhutan, 2010.

[25] M. Saaltink, "The Z/EVES System", ZUM'97: The Z Formal Specification Notation — 10th International Conference of Z Users Reading, UK, 1997.

[26] G. A. Miller, "WordNet: A Lexical Database for English", Communications of the ACM, vol. Vol. 38, No. 11, pp. 39-41, 1995.

[27] G.A. Mirza, N. Masood, S. Asghar, “Formal Representation and Resolution of Null Value Conflict in Database Integration”, The 2nd IEEE Conference on Computer and Management, March 9-11,Wuhan, China, 2012.