Unified Approach for Structural Coherence in Data Layer of E-Government Services

[1]Amar Jeet Singh, [2]Rajesh Chauhan, [3]Balvir Thakur

Department of Computer Science,

Himachal Pradesh Univesity, Summer hill, Shimla-5 (India)

aj_singh_69@yahoo.co.uk, rajesh63673@yahoo.com, balvir.Thakur@gmail.com

Abstract: *Multiple view points during design phase about entities that are holding data and independent analysis by different organizations lead to different versions of same entity that are structurally incoherent or inconsistent. Hence centralized publishing of minimal definition of entities play useful role in reducing the duplication of effort involved in e-Government projects. E-Government solutions deal with multiple databases that may store data about different versions of same entity. Present paper is an effort to present language based solution to bring structural coherence in data layer of e-Government solutions. Schema based approach for Entity Definition has been proposed to define structure (vocabularies) for government entities and their metadata. Its adoption by authorities will ensure structural coherence among federal databases and allow the organization to harness the power of reusability and facilitate the rapid development of services based on similar entities*

Keywords: *Interoperability, Structural Inconsistencies, Entity Definition, e-Government, Data Layer.*

1. Introduction

A large number of e-Government services and solutions are currently being developed under the National E-Governance Policy, aiming to enhance access and delivery of government services. Core of all these services are well designed entities which eventually developed into data layer of solutions

that can be implemented in any vendor specific databases. Meticulously designed entities that conform to some standard not only bring interoperability among the applications, but also bring interoperability at data layer by maximizing the structural coherence among entities defined by different organizations.

e-Government Solutions can be tiered in to two layers; one is data layer, which deals with the identification of structure of data, its attributes, constraints, relationship etc and other is application layer, which deals with the user interface and services to connect with the data layer. At data layer, data about entities and their interaction is stored in terms of attributes, constraints & relationship and application layer deals with the behavior aspect of solution which largely depends on the operations/methods defined for the manipulation of entities. Identifying entities, their relationship, scope, hierarchies, role they play in organization, their attributes and the methods through which entity perform its tasks, are crucial tasks before designer. E-Governance is the area where similar services/solutions are expected in different context (depends on the provincial limit of cities and departmental administration) and thereby increases probability of dealing with similar kind of entities. Analyzing & designing of data layer consumes significant time & efforts on the part of analyst & designer and well designed entities, if not conform to standards become useless in long run and affect sustainability of solution. Since 70% of the time government deals with the similar kind of entities with different interaction context hence framework for single consistent definition of entity reduces the duplication of effort in designing data layer every time and also reduces structural inconsistencies (different structure of same entity) among data layer of similar solutions. These structural inconsistencies may hinder data integration, migration and transformation from one data layer to another. Rest of the paper focuses on the structural inconsistencies, data layer in e-Government context and related works and finally introduces Language for Entity Definition vocabulary, which is a tailored XML implementation, suggesting a common structure for defining similar entities.

2. Related Work

Data layer, which starts with the development of conceptual schema, eventually lead to database consumes significant amount of time in whole development process. Either data driven (Korth & Silberschatz, 2006; Ullman, 1997; Elmarsr and Navathe, 2007) or functional (Senn, 1989) approach is used to for designing. Good literature has been found on the major factors that increase the overall effectiveness of database Davidson, 2007; National Research Council, 2002). Conceptual designing of database is mainly based on Entity-Relationship (ER) model or Unified Modeling Language (UML) based class diagrams using either indirect or direct method. Indirect method is used for designing complex databases and is evolved over the time after integration of schemas (Lukovic, I., Ristic, S., Mogin, P., Pavicevic, J., 2006) and need advance knowledge of designing and even then outcomes are uncertain. Direct method is used to design small independent databases. These independent databases are called data silos (Andersen & Dawes, 1991 and is prevalent in e-Government. Independent design of schema in small databases lead to collisions in expressing real world constraints & rules (Lukovic, I., Ristic, S., Mogin, P., Pavicevic, J., 2006) and are found full of structural anomalies (Singh & Chauhan , 2011).

Fundamental principal of database is to present unified representation of organizational data and can be achieved by various approaches of schema integration (Batini, C., Lenzerini, M. & Navathe, S. B., 1986) and data integration (Yugang,Z., Luying,L., 2006; , Bednárek, D., Yaghob,J. & ZavoralData, F. ,2005; , Brazhnik, O. & John F. Jones, 2007). E-Government data layer is federated (Sheth, A. & Larson, J., 1990) and lacking single global view (Joseph Fong, Kamalakar Karlapalem, Qing Li, Irene Kwan, 1999). Federated databases are encountered with problems of maintenance of a federated schema to cope with local schema evolution in a tightly coupled federation (Motz, R., 2002). Most of schema integration approaches that work with heterogeneous databases also work with federated databases. Problems of integration of heterogeneous legacy databases (Schmalz M. S., Hammer J., Wu M., Topsakal O, 2003) and approaches towards global schema derivation by integrating existing databases (Joseph Fong, Kamalakar Karlapalem, Qing Li, Irene Kwan, 1999) are worked upon by researchers. (Olteanu, A. M., Mustière, S & Ruas, A., 2006) discusses the

knowledge requirement for integration of different independent databases to guide the matching process during integration of databases. Some more prominent work that reflect development and implementation in related area are (Govedarica M ,2002 ; Govedarica M., Lukovic I., Mogin P, 2004; Mogin P., Lukovic, I., 1995; Mogin P., Lukovi I., Govedarica M., 2004 ; Pavic evi ì J., 2005; Risti ì S., Lukovi ì I., Mogin P., Pavi ìevi J., 2005).

Data layer should be meticulously designed (Davidson, 2007; National Research Council, 2002) and must be free from inconsistencies (Elaasar, M. & Briand, L. 2004). In government, inconsistencies are not only found in data layer but found in documents also (Freitas, F. et. al., 2011). Various semantic (Freitas, F. et. al., 2011) and metadata (Gail Hodge, 2001) based approaches (Siegel,M., Madnick, Stuart E, 2010), based on analyzing the actual instances of real data, can be followed to eliminate consistency problem (Rahm, E. & Hai, H. Do, 2000). GIF of various countries (UNDP, 2007; Pankowska, M. ,2008; Chauhan, R. & Singh, A., 2011) explores the idea of consistent interoperable data standards that are helpful in exchange of data but these standards lack basic guidelines for defining the data layer through which interoperable definition of entities are possible for similar entities and structural inconsistencies (Singh & Chauhan , 2011) can be reduced. Language based approach (Pitoura, 1997; Helander, 2005) and idea of description of object and use embedded metadata with the object to define data layer using object oriented languages is worked upon by (Zeng, 1999). Responsibilities of objects in data layer are studied by (Shaw & Garlin, 1994, 1996; Nebbe, 1996). Some efforts in this direction can be seen in (OGCIO, 2006).

Cited literature reveals a gap for possibility of development of a language through which entities can be easily & readily described. Such language provides vocabulary for defining entities so that other similar projects can use those definitions and develop structurally consistent data layer for all similar projects. Most of literature in this context is available under the ambit of schema integration and deals with providing global view of schema where as structural coherence aiming at providing facility to define entities that eventually lead to data layer which automatically become compatible with similar solutions and hence increase reusability of existing data layer and

lessens the impedance mismatch (Draper, D., 2004; Mary A. Finn ,2001), and other related problem of data conversion (Kavadias, G and Tambouris, E., 2003) and is also eliminate the need of schema integration for similar solutions across organizations.

3. Structural Inconsistency in Data Layer of e-Government Services

Meticulously designed data Layer in e-Government applications contributed towards long term sustainability, reusability, data sharing, and integration of e-Government services. Higher level of interoperability & coherence is desired not only for better integration but also desired for rapid development of services among different government agencies thus allowing them to exchange data as well as knowledge about data (Loukis, E. & Xenakis, A., 2011; Abbott & Daisy, 2009). Proliferations of independent e-Government projects that have limited coherence and are largely uncoordinated often have inconsistent structure of similar entities at data layer (Singh, A. J. Chauhan, R., 2010). Multiple view points about similar entities during design phase further increases structural inconsistencies which mean similar entities working in similar context for similar solutions may have different structure for same entity. Besides, communication gap among solution developers/designer, involvement of independent designers/ developers for every solution, and Lack of common guidelines/framework to follow for entity design are cause of structural inconsistency.

Structure of the data layer, in the form of entities, their attributes and relationships among entities, represents the overall design for the storage of contents. During design of data layer entity name, attributes, constraints, roles, relationship are considered (Korth and Silberschatz, 2006; Ullman, 1997; Elmarsr and Navathe, 2007) but due to differences in the view points during modeling and design phase, structural inconsistency, paves its way to data layer and limits its capability in other similar contexts. Although existing design technologies (Shaw, M. & Garlan, D., 1996) are good for non-technical users, But Boxes, Lines etc don't have consistent meaning across the system structure diagrams and in practical, solution have quite sophisticated rules about the interaction and shared representations. On the other hand E-R, UML modeling requires an advanced knowledge of designing and even

then outcomes are uncertain (Lukovic, I., Ristic, S., Mogin, P., Pavi, J. evic, 2006).

Limited coherence and lack of coordination makes the data layer of similar solutions structurally inconsistent. With the passage of time these data layers or data islands evolves as federal databases or federation, which is very common phenomena in government sector e-government solutions (Andersen & Dawes 1991). Federation is seen as collection of possible heterogeneous, interoperating but autonomous component databases. Federation needs preservation of local database autonomy, in the sense that the component databases can not be modified for the purpose of integration, and their instances and schemas may evolve independently (Motz, R. ,2002). Federation makes data silos (small independent local databases) structurally different and limits its use in other similar environment. Increased distribution and heterogeneity among information sources in observed in government sector which is further made complex by independent addition of new resources and changing of old resources and thus make the process of data layer development incremental. Autonomy of data layer (Structure & Data) at local level also needs to be preserved, otherwise various consistency problems arise on modification of local schemas in the process of conforming them to global schema (Motz, R., 2002).

Process of integration and transformation of structure or adoption of structure to new but similar context is common and arises need to establish framework based on successful solutions of data layer that enhances reusability and easy transformation of data layer to vendor specific databases. Various inconsistency and impedance mismatch problems can also be mitigated by defining data layer in platform independent manner. The basic idea is that shared data definitions are stored only once and maintained by the producer. In this way, data definitions in use are always up to date and no redundant versions need to be stored (Winter, 2002) and is known as data interoperability (Kasunic, M. & Anderson, W., 2004), which requires a single data definition for all similar services. When a single set of definitions is mandated for all applications, definitions are no longer locally optimal, and therefore the successful implementation of such mandates, centralized

agreeable initiative is needed. Data layer needs to be structurally interoperable to reuse existing definitions.
Following are some common approaches to achieve data interoperability

(i) ***Object orientation:*** This is a technically promising approach for developing data definitions by encapsulating the internal details of the data.

(ii) ***Extensible data model:*** This approach uses an extensible data model and standardized interface. The Simple Network Management Protocol is an example.

(iii) ***Extensible Markup Language (XML):*** This approach requires agreement on the contents and meaning of the XML schema for entities. Schema can be extended for the structure of database to provide structural consistency. This requires meticulous design effort and centralized control over the entity definitions. This approach can be combined with the "Object Orientation Approach" where Extensible Markup Language is used to define object which will widen the scope of conventional entities (Korth & Silberschatz, 2006; Ullman, 1997; Elmarsr & Navathe, 2007) and include methods (Constructor, General Method) and other object oriented feature to give versatility to entities and also define their scope and behaviors more precisely within application context.

4. Entity Definition: Language Based Approach

Database design in organization for specific projects starts with the requirement (Data & Functional) gathering (Elmarsr & Navathe, 2007) and conceptual schema thus designed, responsible for reliable and consistent data layer. Fulfillment of functional requirements becomes the responsibility of language in which application is going to be developed. Conceptual schema is generally implemented in commercial databases that mostly use relational technology and have limited functionality to store complete object (Attributes, Methods and Metadata) (Gornshtein, D. & Tamarkin, B., 2004). Any solution for organization generally need to store following type of data at basic level.

1) Data Related to Entities: This is the data held in columns of any relational data base i.e. values of attributes. This data is permanent non-transactional in nature.

2) Meta-Data about Entities: This is the extra information about the entities by which the entities can be better understood and used. This information also tells about the scope and usability of entities.

3) Master data of organization: This refers to the static data about the organization and frequently needed to develop application /services.

4) Relational Data:: Refers to the data generated during the interaction of entities, which is contextual in nature but depends on the structure and role of entities in an organization.

In this context, data can be categorized as General data, which represents the data that solution need from database and held in its data layer and eventually contribute to building the database, and other is administrative data that provides assistive information to solution and helps in integrating the services/solution in new environment. At least four conditions must be met to harness the full power of interoperability (Chauhan, R., Singh, A.J., 2011) at data layer

1) Entities must be consistently defined within or across the organizations with e-governance point of view.
2) Entities should publish their minimal interfaces, which include minimal set attributes with data type and most common operations performed by entities with signature so that other similar entities in other organizations take advantages of successful designing, for if the interfaces are well defined, the chances of interoperability increases (Lewis & Wrage, 2004).
3) Interaction among the entities must generate data independent of specific solution for sharing it with other solutions.
4) There must be platform independent administrative data available to tell the scope of entities in better way.

Language based solution need entities and their behavior, defined in platform independent manner and also needs vocabulary to hold administrative data (meta-data) as well as master data. Centralized registry of organization wise government specific entities and associated definitions,

that developer or organization can consult when building or modifying solution needs to be established. Without such a registry, developers are less likely to build solution using compatible entity definitions. With this arrangement, developers have the incentive to reuse entity structures found in the registry because doing so reduces costs and brings about structural coherence and interoperability with other existing systems (GAO, 2002). Successfully designed and implemented entities in organizations are helpful in developing similar projects in other organizations. Such language based solution for entity definition is able to solve some challenges of data sharing addressed in (Seligman.L & Rosenthal .A, 2001); semantics can easily be defined for frequently evolving entities through such language. Entity definition language resolves the issue of conceptual schema diversity, which increases duplication of work among developers community, by defining the entity and its role in the whole conceptual schema so that the same elements will not be semantically differ within the different conceptual schemas and hence produce more consistent, robust and scalable base for solution. For interoperability point of view, Language has vocabulary for incorporation of embedded metadata within the entity for better semantics and data value reconciliation (Seligman.L & Rosenthal .A, 2001).

5. Language for Entity Description: Overview

Language for Entity Definition is an orchestration of set of elements responsible for providing the Interoperability and coherence among entities at data layer. To achieve the goal of faster & sustainable e-Government development, it is important to have a common format for specifying the structure for data at data layer and reuse the definition of entities that are successfully implemented. Idea is that the entities can be standardized and the more structurally designed entities are available, it is easier to design better database. Language for Entity Definition data vocabulary is presented as a proposed document and these vocabularies are tailored XML implementation suggesting a common XML structure for describing entities and their behavior. eXtensible Markup Language (XML) is widely recognized as a key technology for interoperability and act as meta language in the development of language for Entity Description. More specifically, in order to serialize Entity Definition Language document in XML format, an XML

schema was implemented for the validation of their structure. XML schema validation mechanism was preferred from Document Type Definition (DTD) because it provides a richer set of data types and allows user to derive their own data types and take advantages of inheritance of elements, attributes and definition of data-types (Williams, K ,2000; Moller, A. & Schwartzbach, M.,2006), Document describing the entity emerges from the XML schema Language for Entity Definition and appropriate transformations can be applied to look the entities previously described by other organizations. XML schema can be easily extended, modified and maintained in the future according to the need.

One aim of Language for Entity Definition, an open XML document structure aiming at support definitions of entities and this will lead to lessen the backend overhead to design and create database for implementation of one stop government. If all entities (responsible for the data in database) could speak the same language, they could talk directly to each other in ways natural to the application without artificial technical barriers. Entity Repository is centralized facility, under the supervision of group of experts, responsible for holding the minimal definition of entities that are registered. Entities are defined by the central authority with the minimal interface (abstract entities) and published from centralized location. Centralized authority and some doctrine or policy is must for interoperability (Hamilton & Murtagh, 2000). Using unique, nonstandard data definitions and Structures, by organizations will make the data non-shareable externally and defeating the purpose of defining the entity without having centralized repository.

GovML, a language for describing the life events and public services (Kavadias, G and Tambouris, E, 2003) has been developed, where every life event triggers public service. Accessing public service eventually triggers access to database to cater the need of life event (Kavadias, G and Tambouris, E, 2003). Quality, reusability and sustainability of public service depends upon how well database is designed and which depends on how well the entities and their relationships are designed and what attributes and functionality is induced in those designed entities. Language for Entity Definition is based on three types of elements 1) Entity Data 2) Entity's Meta

data 3) Organization's Master Data which is more related to organization than entity. In the context of entities, metadata represents the detail of entity used to collaborate with another entity like scope of entity, synonym, primary repository name and intended applications etc. Domain specific knowledge can be induced in the entities and thus makes them more useful and scalable for future applications. Entities are defined with the Entity Specific Metadata and Entity Definition. Entity Specific Metadata contains the extra information about the entity like author of entity, publisher of entity, use of entity etc. and Entity Definition contains minimal abstract definition of entity i.e. attributes in the entity, methods in entity with the signature of methods etc. Purpose of entity repository is to provide consistent & coherent definition of entities to act as a guideline for entities going to be developed in other similar project and hence plays important role in creating coherent database for storing the data related to entities, which leads to the unified structure of entities and hence easier exchange of information without changing the format at data layer is possible without worrying about impedance mismatch about the number of essential attributes, their data types, primary keys, methods and their behavior. This not only makes the structure of database coherent and consistent but also makes it easy to create and deploy already created services in new context. Along with the definition of entity, links of customized concrete successful implementation also maintained with the entity Metadata. Power of the entity lie in how better the entity is capable in capturing the real scenario i.e. most usable attributes, behavior of entity and their relationship with other entities. Core of services is almost always a data layer, and rapid development of data layer is only possible if already designed entities are available to developers. To successfully implement the solution in large multi-site organization like government, solution must be based on some defined set of standards and interoperable in nature. At data level, entities are the basic structures in which the data resides. Language for Entity Definition will provide unambiguous definition of entities involved in the system and their behavior, it is better to use the published entity and use it rather than to reverse engineer the entity and customize the solution.

6. Data Vocabulary of Language for Entity Definition

The proposed Language for Entity Definition addresses the common structure and interoperability challenges. Entity description includes Metadata, Master data and actual description of entity. Elements of Language for Entity Definition are identified based on studying the actual structure of databases of various organizations in government (Singh, A.J, Chauhan. R, 2011), reviewing the government Interoperability Framework (GIF) of various countries available online and studying literature (Pankowska, M., 2008; Chauhan, R. & Singh, A., 2011). Table-1 shows the Data vocabulary of Entity Description language.

Main Node	Sub Node	Description
Entity	Meta Data	**Author Name:** Refers to the name of individual or group or organization who creates the Entity and to whom this entity originally belongs.
		Primary Repository Name: Name of the central organization that holds the definition of entities
		Secondary Repository Name: List of repositories of similar entities that are successfully implemented in some projects or where the concrete implementation is found.
		Domain: Broader area of organization or application e.g finance, inventory, central diary etc. Every domain also has numeric representation which is used in the version of entity.
		Sub-Domain: Refers to the specific area of main domain. Every sub-domain also has numeric representation which is used in the version of entity.
		Date of Creation: This refers to the date on which Entity is created.
		Date of Last Updation: Refers to the Date on which Entity's definition is last modified
		Candidate Organizations: Refers to the list of other organizations which are the prospective candidate for the use of this entity.

Main Node	Sub Node	Description
		Candidate Applications: Refers the list of application in which entity can be used.
		Referenced Entities: List the other entities use to create current entity.
		IntraEntityRelationship: Refers to the one-to-one, one-to-many, many-to-many, many-to-one relationship with other entities. This feature is discouraged because every entity has its own identity apart from the identity it gained from having the relationship with other. Moreover the relationship is more related with how they are analyses rather than having fixed status.
		Constraints: Refers to the constraints imposed on the entity by parent organization.
		Description: This refers to the details description of entity.
		Version Number: Refers to the version number of entity.
		Synonyms: List of alias names by which entity is known
		Concrete Implementation: List the organization where the concrete implementation of the entity can be found.
		Role: Refers to the role of entity in the application for which entity is originally designed and created.
		RightUseCondition : Refers to rights and conditions under which the entity can be reused by other
		Remarks: Other secondary information about the entity.
	Master Data	This is organization wide static data which is used by the service/application to be developed for the particular organization
	Entity Description	**Name:** Refers to the name of entity
		Attributes: Refers to the constituents of attribute like attribute name, data type, whether data type is array or not, if data type is an array then size of array, whether attribute act as primary key or not, default value of attribute, description of attribute, constraint (null, check , unique) on attribute.
		Methods: Refers to the name of methods, return type, name and data type of argument it will take. if data type is an array then size of array, type of method (normal or constructor).

Table-1 (Data Vocabulary)

Production of instance document starts with analyzing and identifying entities during analysis phase. After identifying the initial requirements, organization may adopt definition of existing entities and develop data layer based on previous successful data layer or the organization simply starts defining entities by producing new instance document, which can be validated against language schema. Newly instance document is published from central location so that other similar projects take advantages of its designing. Language document hold all the information related to the entities like Meta Data, Entity Definition etc. apart from this, facility to store Master Data is also provided in language Schema. The Whole database part of application can be developed rapidly by reusing definition and design effort of previous entities.

7. Conclusion

In E-Governance where large numbers of applications are being developed, saving analysis time enhance the productivity and contributes to shorter solution development time. Language for entity description not only takes care of describing full entity (Attributes, methods, constraints, Meta data etc.) but also take care of management of Meta data and Master data. Using entities that are defined by entity description language make the structure of database consistent across organizations that provide similar services. This also makes the information sharing easier without worrying about impedance mismatch and format conversion. Such language contributes towards the rapid development of consistent, coherent, and interoperable data layer in E-Government services.

References:

1. Abbott & Daisy (2009), DCC Briefing Paper: Interoperability, Digital Curation Centre, https://www.era.lib.ed.ac.uk/handle/1842/3363.
2. Andersen, David F., Sharon S. & Dawes. (1991). Government Information Management: A Primer and Casebook. Englewood Cliffs, NJ: Prentice Hall.
3. Batini, C., Lenzerini, M. & Navathe, S. B. (1986), A comparative analysis of methodologies for database schema integration, ACM Computing Surveys (CSUR), Volume 18 Issue 4, Dec. 1986.

4. Bednárek, D., Yaghob,J. & ZavoralData, F. (2005), Integration Using DataPile Structure, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.142.6687&rep=rep1&type=pdf.
5. Brazhnik, O. & John F. Jones (2007), Anatomy of Data Integration, Journal of Biomedical Informatics, Volume 40, Issue 3, June 2007, Pages 252-269, http://www.sciencedirect.com/science/article/pii/S1532046406000967.
6. Chauhan, R. & Singh, A., (2011), Generalizing Meta-Data Elements: Global Scenario and Indian Perspective, JOAAG, Vol. 5. No. 1.
7. Chauhan, R., Singh, A.J. (2011), AKGEC JOURNAL OF TECHNOLOGY, Vol. 2, No. 1, ISSN 0975-9514, Theoretical Categorization of Interoperability in E-Government, http://www.akgec.org/akg_journal_jan-june11.aspx
8. Davidson, L. (2007): Ten Common Database Design Mistakes, http://www.simple-talk.com/sql/database-administration/ten-common-database-design-mistakes. [On-line], Accessed on : 22 July, 2010
9. Draper, D. (2004), "Mapping between XML and Relational Data", article at http://www.informit.com/
10. Elaasar, M. & Briand, L.(2004), An Overview of UML Consistency Management,Technical Report SCE-04-18, http://www.citeseerx.ist.psu.edu/viewdoc
11. Elmasri,R. & Navathe, S. (2007), Fundamental of database systems, ISBN 978-81-317-1625-0, Published by Pearson Education, Third edition.
12. Freitas, F., Candeias, Z. & Stuckenschmidt, H.(2011), new Usage for Semantic Technologies for eGovernment: Checking Official Documents' Consistency, Electronic Journal of e-Government Volume 8 Issue 2, 2010 (pp121-134), ISSN 1479-439X, http://www.ejeg.com
13. Gail Hodge (2001), Metadata Made Simpler, Published by NISO Press, National Information Standards Organization,ISBN: 1-880124-50-5, http://www.ncsi.iisc.ernet.in/raja/is214/214-2001-2002/Metadata_Bklt.pdf

14. GAO (2002) , Electronic Government : Challenges to effective adoption of the Extensible Markup Language, United State General Accounting Officer, Retrieved from http://www.gao.gov/new.items/d02327.pdf
15. Gornshtein, D.& Tamarkin, B. (2004), Features, strengths and weaknesses comparison between MS SQL 2005 (Yukon) and Oracle 10g databases, Revision 1.3, WisdomForce Technologies, Inc, http://www.wisdomforce.com
16. Govedarica M., Lukovic I., Mogin P(2004)., Generating XML Based Specifications of Information Systems. Computer Science and Information Systems (ComSIS), Belgrade, Serbia and Montenegro, Vol. 1, No. 1 2004, pp. 117-140.
17. Govedarica M(2002)., An Automated Development of Information System Application Prototypes. PhD Thesis, University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia and Montenegro, 2002.
18. Hamilton & Murtagh (2000), Enabling Interoperability Via Software Architecture, Technical Reports ,AD Number: ADA458021, Retrieved from http://www.dtic.mil/dtic/
19. Helander (2005), deeply embedded XML communication: towards an interoperable and seamless world, International Conference on Embedded Software, Proceedings of the 5th ACM international conference on Embedded software. Jersey City, NJ, USA, Pages: 62 – 67, ISBN: 1-59593-091-4.
20. Joseph Fong, Kamalakar Karlapalem, Qing Li, Irene Kwan(1999), Methodology of Schema Integration for New Database Applications: A Practitioner's Approach, Journal of database management, http://www.irma-international.org/viewtitle/51209/
21. Kasunic. M & Anderson W (2004), Measuring Systems Interoperability: Challenges and Opportunities, Software Engineering Measurement and Analysis Initiative, Technical Note CMU/SEI-2004-TN-003, http://www.sei.cmu.edu/library.
22. Kavadias, G & Tambouris, E (2003) , GovML : A markup laguage for describing public services and life events , Knowledge Management in Electronic Government, 4th IFIP international Working Conference, KMGov 2003 Rhodes, Greece, May 26-28 2003 Proceedings published by Springer.

23. Korth, H. & Silberschatz, A. (2006)., Database System Concepts. 5th ed., McGraw-Hill, ISBN 007-124476-x
24. Lewis & Wrage (2004), Approaches to constructive interoperability, Carnegie Mellon Software Engineering Institute, Pittsburg, Technical report, CMU/SEI-2004-TR-020, ESC-TR-2004-020, Retrieved from http://www.citeseerx.ist.psu.edu
25. Loukis, E. & Xenakis, A. (2011), Knowledge Interoperability of Parliaments' and Government Agencies' Information Systems, DOI: 10.4018/978-1-61520-887-6.ch016, http://www.irma-international.org/viewtitle/45795/
26. Lukovic, I., Ristic, S., Mogin, P., Pavic, J. evic (2006), Database Schema Integration Process: A Methodology and Aspects of its Applying, Novi Sad J. Math. Vol. 36, No. 1, 2006, 115-140, http://www.kurims.kyoto-u.ac.jp/EMIS/journals/NSJOM/Papers/36_1/NSJOM_36_1_115_140.pdf.
27. Mary A. Finn (2001), Fighting Impedance Mismatch at the Database Level, http://www.intersystems.com/cache/whitepapers/matchwp.html.
28. Mogin P., Lukovicì I.(1995), A Prototyping CASE Tool. XXVIII International Symposium on Automotive Technology and Automation, Stuttgart, Germany, September 18-22, 1995, Proceedings for the Dedicated Conference on Rapid Prototyping in the Automotive Industries, pp. 261-268.
29. Mogin P., Lukovic ì I., Govedarica M.(2004), Database Design Principles, 2nd Edition. University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia and Montenegro, 2004.
30. Moller,A. & Schwartzbach, M.(2006), An Introduction to XML and Web Technologies, Pearson Education Ltd., ISBN 978-81-317-2607-5.
31. Motz, R. (2002), Problems in the Maintenance of a Federated Database Schema, Proceedings of the XXII International Conference of the Chilean Computer Science Society (SCCC'02). 1522-4902/02 IEEE, se-pubs.dbs.uni-leipzig.de/files/Motz2002Problemsinthemaintenanceof.pdf

32. National Research Council (2002), Information Technology Research, Innovation, and E-Government, National Academy Press, Washington. D.C. ISBN: 0-309-50031-1, Retrieved from http://www.nap.edu.
33. Nebbe R.D (1996) , Semantic Structure: a Basis for Software Architecture, Software Composition Group, Retrieved from http://www.scg.unibe.ch/archive/famoos/Nebb98m/nebbe_oosa.pdf
34. OGCIO (2006), HKSARG, XML Schema Design and Management Guide v1.3, [Online], Retrieved from http://www.ogcio.gov.hk/eng/infra/eif.htm
35. Olteanu, A. M., Mustière, S & Ruas, A. (2006) Matching imperfect spatial data, 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences. http://www.citeseerx.ist.psu.edu
36. Pankowska, M. (2008), National frameworks' survey on standardization of e-Government documents and processes for interoperability, Journal of Theoretical and Applied Electronic Commerce Research, ISSN 0718-1876 Electronic Version, Vol 3/Issue 3, http://www.jtaer.com.
37. Pavic evic, J.(2005), Development of A CASE Tool for Automated Design and Integration of Database Schemas. M.Sc. Dissertation, University of Montenegro, Faculty of Science, Podgorica, Serbia and Montenegro, 2005.
38. Pitoura(1997), Providing Database Interoperability through Object-Oriented Language Constructs , Journal of Systems Integration, 7, 99–126, Kluwer Academic Publishers, Boston. Retrieved from http://www.springerlink.com/content/m0087636781550j2/
39. Rahm, E. & Hai, H. Do (2000), Data Cleaning: Problems and Current Approaches, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, IEEE Computer Society, and Vol. 23 No. 4. http://citeseerx.ist.psu.
40. Ristic , S., Lukovic ì I., Mogin P., Pavic evi, J.(2005), Integrating a Database Schema Using IIS*Case Tool. 13th Scientific Conference on Industrial Systems IS'05, Herceg Novi, September 07 - 09, 2005, Proceedings pp. 331-340.

41. Schmalz M. S., Hammer J., Wu M., Topsakal O(2003)., EITH – A Unifying Representation for Database Schema and Application Code in Enterprise Knowledge Extraction. Proceedings of the 22nd International Conference on Conceptual Modeling, Chicago, IL, November 2003.
42. Seligman.L & Rosenthal .A (2001), The Impact of XML on Databases and Data Sharing, MITRE Corporation, Retrieved from http:// www.reference.kfupm.edu.sa
43. Senn. James A. (1989), Analysis and Design of Information Systems, McGraw-Hill Inc.,US. ISBN 9780070562875.
44. Shaw, M. & Garlin, D. (1994), Characteristics of higher-level Languages for Software Architecture, CMU-CS-94-210 , Retrieved from http:// www.cs.cmu.edu/afs/cs/project/vit/ftp/pdf/salang-reqts-tr.pdf
45. Shaw. M & Garlan. D (1996), Software architecture: perspective on an emerging discipline, Prentice-Hall of India, ISBN-81-203-1470-0.,
46. Sheth, A. & Larson, J. (1990). Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases. ACM Computing Surveys, 22(3).
47. Siegel,M., Madnick, Stuart E(2010)., Schema Integration Using Metadata, http://web.mit.edu/smadnick/www/wp2/1989-06-SWP%233092.pdf.
48. Singh, A. J. Chauhan, R. (2010), Indian Journal of Computer Science and Engineering, E-Government Databases: A Retrospective Study, Vol 1 No 2, 66-73, ISSN : 0976-5166.
49. Ullman, J.D.,& Widom, J. (1997), A First course in database systems, Prentice Hall International, Inc. ISBN 0-13-887647-9
50. UNDP (2007), e-Government Interoperability: A Review of Government Interoperability Framework in Selected Countries, ISBN: 978-974-13-1624-3, Retrieved from http://www..ibm.com/ibm/ governmentalprograms/undp-gif-review.pdf
51. Williams, K. (2000), Professional XML Databases, Wrox Press Ltd, ISBN-1861003587.

52. Winter (2002), Chapter on Interoperability, European Territorial Management Information Infrastructure (ETeMII), http:// www.ec-gis.org/etemii/reports/chapter3.pdf
53. Yugang,Z., Luying,L. (2006), Architecture and Implementation of an XML-based Heterogeneous Database Integration System, http:// en.cnki.com.cn/Article_en/CJFDTOTAL-JSJC200602029.htm.
54. Zeng (1999), Metadata Elements for Object Description and Representation: A Case Report from a Digitized Historical Fashion Collection Project, Journal of the American Society for Information Science. 50(13):1193–1208, Retreived from http:// polaris.gseis.ucla.edu/gleazer/260_readings/Zeng.pdf.