Journal of Computer Science 7 (7): 1027-1032, 2011
ISSN 1549-3636

Comparing and Choosing Appropriate Database for Storing Context in Context Aware System

B. Vanathi and V. Rhymend Uthariaraj
Ramanujan Computing Centre, Anna University, Chennai, India

Abstract: Problem statement: The databases introduce the digital era the fact that requirement to maintain large and ample data. The growth of information development and improvement that pursued it transformed the databases into a global reality. This study enquired and described the vital role of storing the context data to an appropriate database in the area of context aware computing. It also adds-on limitations of the existing relational and object relational databases with respect to context aware computing. **Approach:** This article examines traditional database like relational database, object relational database and spatial database. It helps to make an optimal decision while choosing the database for context aware applications. **Results:** All the traditional databases are compared with context aware features like semantic support, ease of transaction of large data, query optimization, reasoning support, formality, scalability, spatial data and xml support. **Conclusion:** This study confirmed that using object relational database along with ontology is more apt for storing the context data for context aware computing applications.

Key words: Pervasive computing, context aware computing, Database Management System (DBMS), relational database, object-relational database, spatial data, ontology, User Defined Types (UDT), query optimization, context data, Data Manipulation Language (DML), object identity

INTRODUCTION

The continuing technical progress in computing and communication lead to an all-surrounding use of networks and computing power called ubiquitous or pervasive computing (Mattern and Sturn, 2003). Pervasive computing system directs at steady adaptation to meet the requirements of users within every changing physical, social, computing and communication context. Context awareness is at the heart of pervasive computing problems. Context can be defined as an operational term whose definition depends on the intension for which it is collected and the interpretation of the operations involved on an entity at a particular time and space rather than the inherent characteristics of the entities and the operations themselves according to Winograd (2001) and Dey and Abowd (2000).

Data required for modeling are obtained from the applications using sensors and it has to be represented in a suitable way for processing. Sensors can be physical, virtual or logical sensors (Baldauf, 2007). Various context management and modeling approaches are introduced to present context for reasoning in different application area. Data from the sensors are presented using any of the following modeling approaches like key-value-pair modeling, Graphical modeling, Object Oriented modeling, logic based modeling, Mark up scheme modeling and Ontology modeling (Strang and Linnho-Popien, 2004). These models support standardization of techniques to present context for productive reasoning in different application area. Among all the modeling approaches, ontology based context model is more suitable for context aware computing (Strang and Linnho-Popien, 2004).Ontology is formally defined as “an explicit specification of a shared conceptualization” (Dey and Abowd, 2000). The term is obtained from philosophy, where ontology is a systematic account of existence. Context is modeled as concepts and facts using ontology.

Context aware systems are based on ad-hoc models of context, which causes lack of the desired formality and expressiveness. Existing models do not separate processing of context semantics from processing and representation of context data and structure. Ontology representation tools are suitable for statically representing the knowledge in a domain. They are not designed for capturing and processing constantly changing information in dynamic environment in a scalable manner. Existing ontology languages and serialization formats are test based (xml/rdf/owl) and not designed for efficient query optimization,

Corresponding Author: B. Vanathi, Ramanujan Computing Centre, Anna University Chennai, India

processing and retrieval of large context data. The main drawbacks of pure ontological approaches are low performance and data throughput (Vanathi and Uthariaraj, 2010a; 2010b; 2010c; 2011a; 2011b; 2011c).

The proposed context aware system has context acquisition layer, context middleware (representation layer, context management layer and decision layer) and application layer. Context acquisition layer gathers the context from the environment using sensors. Context representation layer represents context as entity relation hierarchy form (Ejigu *et al*., 2007). Set of entities for which context is to be captured is called as Entity. Set of binary relations that form an inversely Directed Acyclic Graph (inverted DAG) on entities is hierarchies. Nodes of the graph represent entities and arcs of the graph represent hierarchical relations.

Union of the sets of binary relations R_e and R_a stands for Relation. R_e is a set of binary relations having both its domain and range from the set of entity. R_a is set of binary relations defined from set of entities to set of literals representing entity attributes. Domain of the relation R_a is the set of entity while its range is set of literal values. Axiomatic relation is a relation about relation. Relation can be generic or domain based. Relation can be transitive, symmetry or inverse property. Meta data is a data about data. It is important in context modeling to associate quality, precision, source, time stamps and other information to the context data. Such information is important to prepare the context data for reasoning and decisions. Some of the generic axiomatic relations are sameAs, inverse, symmetric and transitive.

In the context management layer context is further classified as inactive context and active context. Predicates are used to decide the inactive context. For example context defined using predicates like owned by are inactive and context defined using predicates like located in are active. Inactive context are stored in database and active context are stored in Ontology. In the middleware, rules learned or rules derived from other rules are also maintained. Using the rules, relevant context from database and ontology are forwarded to the reasoning component. From reasoning layer appropriate context is sent to user in the application layer.

In this study, inactive context obtained from the context acquisition must be stored in a database. To choose the appropriate database for context aware system is the objective of this study.

MATERIALS AND METHODS

A software system that allows creating and manipulating the huge amount of data in a structured way is known as a Database Management System (DBMS).Group of data managed by a DBMS is called database. A database management system alleviates the following.

Using Data Definition Language (DDL) users can create a database and can assign their data types and structures called as database schema. Data Manipulation Language (DML) is used to insert, delete, update and query data from database by the user. A multilevel certification control allows security system for the user. Consistency of data is preserved through an integrity system. A recovery control system helps to restore the database to a previous consistent state if any hardware or software failures called transaction and recovery control. A user-accessible catalogue, called the schema that contains meta-data of the data in the database is furnished. A higher-level description language for schemas is required known as the data model for the database management system to describe the data requirements of an organization in a readily understandable way by the users. Data base management system use different kind of data models. The evolution of DBMS follows development of new data models as shown in Fig.1.

The first commercial database management system was formulated employing hierarchical and network data models in the late 1960s.These models highly supports data arrangement and storage of data. High-level query languages are not supported by these models. Only possible way to retrieve data is through a graph or tree of data elements. So users must know the detailed knowledge about the physical data arrangement.

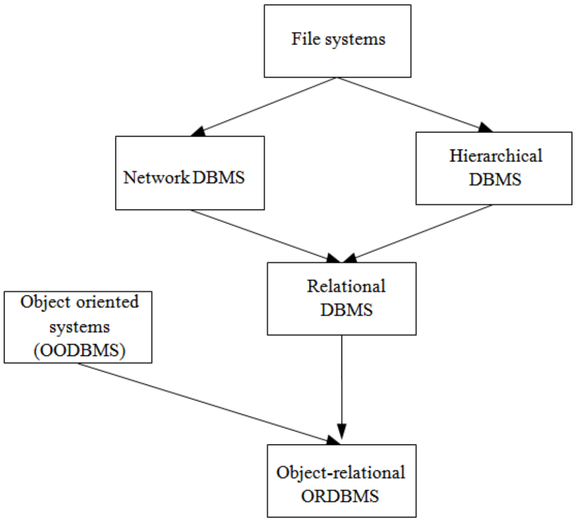

Fig. 1: Evolution of DBMS technology

In the early 1970s, the relational data model was introduced by Codd (1970). It simplifies the process of accessing data and relational database management systems. Here data to the user is presented in a two dimensional tables. A data value with different atomic types like number, character and string are stored in each cell of the table. Data storage is analogous to traditional tabular data representation. It is complex in the internal storage structures to allow for efficient data manipulation. High level query language for efficient data base programming is supported.

The major primitives to specify a query are functional relational algebra and declarative relational calculus. Number of query languages came out with these formalisms. Among them Structured Query Language (SQL) became the effective language. Declarative queries are transferred into an execution plan using SQL query processing modules. This module is a program which specifies in detail the way data is retrieved. Views are created. Views resemble virtual relations defined through query expressions. They do not exist physically but can be queried as they exist physically. Views can be modified by insertion, deletion or update. Data independent in the relational model is provided by dividing the high level query language from the low level implementation details. Physical and logical are the two kinds of data independence. The ability to change the physical structure of data without disturbing the applications is called physical data independence. Invulnerability of the conceptual changes to the applications programs is known as logical data independence.

New application from areas like geographic information system, computer aided engineering Internet, spatial applications and multimedia needed complex data representations highlighted the restrictions of the relational model and insisted the need for a new generation database management system. This leads to Object Oriented Database Management Systems (OODBMS) based on Object-Oriented (OO) data model.

The objects are classified in classes. A class consists of a type and methods that can be executed on objects of the class. A powerful type system is represented with primitive atomic types, record structures, collectiontypes (sets, bags, arrays) and reference types (pointers).Repeatedly apply record-structure and collection operators are used to define complex types. An Object Identity (OID) is used to uniquely identify object. Classes are arranged according to a class hierarchy. Each class can be defined as a sub class of another and inherits all properties from some other classes with overloading and overriding characteristics. Extended object-oriented languages like C++ or Java with database capabilities such as persistence, concurrency control and recovery are used to implement Object oriented database management system.

The object-oriented model improves the database with features to become more powerful in modeling real world objects for the new applications. Any declarative querying facilities is not supported in early object oriented database management systems. Queries were specified through navigating the graph structures where arcs are defined by OIDs stored as attribute values of other objects. The ODMG (Object Data Management Group) developed a standard query language for object oriented database management systems that consists Object Definition Language (ODL) and Object Query Language (OQL). Applications using complex data with minimum important query requirements like computer-aided design packages, object oriented database management are more appropriate.

Combining the declarative power of relational database with the modeling power of object oriented database a new kind of database called Object Relational Database Management System (ORDBMS) was put in. This database is broadly used in commercial products.

The object-relational model comprises extensions of the relational model with the following features: Extensible base type system by User Defined Types (UDT) that can be introduced along with user defined functions, operators and aggregates operating on the values of these types; Complex type support via type constructors for rows (records), collections (sets, bags, lists and arrays) and pointer types; Special operations, methods, can be defined for and applied to, values of user-defined types; Unique OIDs identify each object and its data values. User defined query optimization rules gives cost information about user-defined functions. User defined index structures provide a generic template index structure, like Generalized Search Trees (Hellerstein *et al*., 1995).

The major advantages of extending the relational model is come from reuse and sharing. To enable object-relational features these extended features are implemented SQL: 1999 (Eisenberg and Melton, 1999). It is taken as a step to capture as many as possible of the object-oriented concepts.

The Database view (Stonebraker and Moore, 1996) in Fig. 2 suggests how complex data and complex queries influence each other. It is a two-dimensional graphical representation: the abscissa refers to the ability to define complex data types and the ordinate presents the ability to query databases.

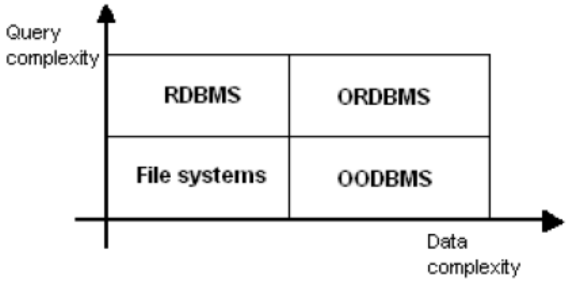

Fig. 2: DBMS types classification

The bottom left quadrant contains the types of applications that process simple data types and requires no data query and use directly the file system of the operating system for persistent data storage.

RDBMS processes simple data types, but allow complex queries. OODBMS handles with complex data types and resolving queries is quite difficult, since for each query must be provided the necessary relationships in the structure of objects. ORDBMS allow complex processing and resolving complex queries. Object-relational model is obviously the most complete, because allows both user-defined data types and complex queries. For this reason, the object-relational data model is also called extended-relational or universal and it can be seen as an attempt to capture as many as possible of the object-oriented concepts.

In current years of rapid development some new ways to store and manipulate spatial data are introduced. Since relational databases have limitations in the case of spatial data, the most effective way proves to be the use of object-relational databases (Huibing, 2008).

A spatial database is a database system that is optimized to store and query basic spatial objects. For example a house or a moving car specifies Points. A line is specified as a road segment or a road network and a polygon can be a county or a voting area. Spatial objects consisting of lines, surfaces, volumes and objects of higher dimensions are commonly used in computer aided design applications, cartography or geographic information systems. These are described both by spatial attributes (length, pattern, perimeter, area, volume) and by non-spatial attributes (the time, the owner, membership of a superior structure). The values of objects' spatial attributes represent spatial data.

Spatial data can be divided into specific data and regional data. The specific data is a point, which is completely characterized by its location in a multidimensional space. It can be obtained directly from measurements or by converting for it to be more easily stored and retrieved.

The regional data is characterized by location (one fixed point in a region) and destination (a line in 2D or a surface in 3D). Spatial data can also have a certain rank (e.g., "Cities in Chennai"), approach (e.g., "Three nearby lakes in Coimbatore"), junction (e.g., "Pairs of city, in Romania, located 50 km apart) .The queries performed for spatial data can be: local, in the area or in the neighborhood (most difficult because it requires the evaluation of proximity – For example, determining the road which passes closest to a specified region).

Post-database management system support user defined abstract data types and spatial data types (e.g., polygon) can be added. The choice of post-relational database management system can be either Object Oriented (OO) DBMS or Object Relational (OR) DBMS. A spatial database is a collection of spatial data types, operators, indices, processing strategies and can work with many post-relational DBMS as well as programming languages like Java, Visual Basic.

Considering ORDBMS, the offer is very generous and covers a wide scale of cost and performance. DBMS can either use free or unlicensed or with public license, such as PostgreSQL. Oracle 10g, DB2 UDB 8 and SQL Server 2005 are commercial ones. They all have adopted the object-relational data model to supports relational tables, abstract data types and functions on abstract data types.

There is a need for spatial database due to the following reasons: Queries to databases are posed in high level declarative manner (usually using SQL). SQL is the "lingua-franca" in the commercial database world. Standard SQL operates on relatively simple data types.

Additional spatial data types and operations can be defined in spatial database using CREATE TYPE statement. A database management system is a way of storing information in a manner that enforces consistency, facilitates access and allows users to relate data from multiple tables together.

RESULTS AND DISCUSSION

Relational models provide standard interfaces and query optimization tools for managing large and distributed context database or receive and send notification on context changes. Relational models are not designed for semantic interpretation of data. Relational database alone cannot be used to represent context in a pervasive environment. For semantic interpretations, ontology is used along with relational database. The Table 1 shows the appropriateness of both approaches in relation to the necessary features (Vanathi and Uthariaraj

2011b; 2011c; Ejigu *et al*., 2007; Codd, 1970; Hellerstein *et al*., 1995; Latip *et al*., 2011). Both approaches have strong and weak sides with respect to features for context management modelling. Best of both sides are combined. From the Table 1 below both relational approach and object relational approach are in the same level. Object relational database is more suitable than Relational approach because of the following advantages.

Object relational approach ensures large storage capacity, which is an important part in web based development. The access speed is fairly quick. Object relational database have a massive scalability compared to relational approach. It boasts excellent manipulation power of object databases. It supports rich data types by adding a new object-oriented layer.

Popularity of eXtensible Markup Language (XML) is used for information interchange on the web. It is important to effectively and efficiently manage XML documents. Two main database designs namely relational and object-relational databases are available. They are suitable for storing and retrieving the data-centric XML documents. Native XML databases or content management systems are better suited for storing and retrieving both document-oriented XML data. To promote performance native data stores are tailored to XML requirements and supports specific XML requirements. Both databases support XML and give good scalability. The disadvantage of using relational database is that it can lower performance. This is because many relations are formed as a result of mapping from XML data to relational data. Queries on the XML data when translated into SQL queries may potentially have many joins, which would make the queries expensive to evaluate (Seng and Yu, 2004).

The advantage of using object relational database for managing XML data are as follows: universal database products are offered by the database vendors which merge relational database and object oriented database as a single product. So Object relational database products have all the positive features of a relational database.

Table 1: Compare RDBMS/ORDBMS/Ontology

Necessary Feature	RDBMS approach	ORDBMS approach	Ontology approach
Semantic support	No	No	Supports
Ease of transaction (large data)	Supports	Extensive supports	No
Query optimization	Supports	Extensive supports	No
Reasoning support	No	No	Supports
Formality	Supports	Extensive supports	Supports
Scalability	Supports	Extensive support	Supports
Spatial data	No	Extensive supports	No
XML support	Supports	Extensive supports	Supports

ORDBMS have more expressive type system which results in more efficient mapping from an XML data model to ORDBMS. For storing and querying XML documents that use a richer set of data types ORDBMS is more suitable (Seng and Yu, 2004).

CONCLUSION

This study deals with the study of evolution and limitations of different traditional databases like relational database, object oriented database, object relational database and spatial database. All the databases are compared with necessary features for context aware computing like semantic support, ease of transaction of large data, query optimization, reasoning support, formality, scalability, spatial data and xml support.

From the comparison it is clear that both relational database and object relational database supports the necessary features. But object relational database provides extensive support for ease of transaction of large data, query optimization, formality, scalability, spatial data and xml. As context aware computing deals with spatial data like location and object oriented database supports spatial data extensibly than relational database , object relational database management system is more suitable for context aware application than relational approach.

ACKNOWLEDGEMENT

This research is partially supported by the All India Council for Technical Education, New Delhi,India.

REFERENCES

Baldauf, M., 2007. A survey on context aware systems. Int. J. Ad hoc Ubiquitous Comput., 2: 263-277. DOI :10.1540/IJAHUC.2007.014070

Codd, E.F., 1970. A relational model of data for large shared data banks. Commun. ACM, 13: 377-387. DOI: 10.1145/362384.362685

Dey, A.K. and G.D. Abowd, 2000. Towards a better understanding of context and context awareness. Proceedings of the CHI Workshop on the ‘What, Who, Where and How of Context-Awareness, Apr. 1-6, Springer-Verlag, The Hague, Netherlands, pp: 304-307. DOI: 10.1145/328595.328614

Eisenberg, A. and J. Melton, 1999. SQL: 1999, Formerly known as SQL3. ACM SIGMOD Record, 28: 131-138. DOI: 10.1145/309844.310075

Ejigu, D., M. Scuturi and L. Brunie, 2007. An Ontology based approach to context modeling and reasoning in pervasive computing. Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications – Workshops, Mar. 19-23, IEEE Computer Society, White Plains, New York, USA., pp: 14-19. ISBN: 978-0-7695-2788-8

Hellerstein, J.M., J.F. Naughton and A. Pfeffer, 1995. Generalized search trees for database systems. Proceeding of the 21st International Conference on Very Large Data Bases Conference (VLDB'95), Zurich, Switzerland, Morgan Kaufmann, pp: 562-573. ISBN: 1-55860-379-4

Huibing, W., 2008. Extending object-relational database to support spatio-temporal data. In :The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXVII. Part B2. Beijing. ISSN: 1682-1750

Latip, R., H. Ibrahim and A.F. Al-Hanandeh, 2011. Scientific data sharing using clustered-based data sharing in grid environment. Am. J. Econ. Bus. Admin., 3: 146-149. DOI: 10.3844/ajebasp.2011.146.149

Mattern, F. and P. Sturn, 2003. From Distributed Systems to Ubiquitous Computing- State of the Art. Trends and Prospects of Future Networked systems. Fachtagung_Kommunikation in Verteilten Systemen(KiVS), Leipzig, Springer-Verlag, Berlin, pp: 3-25. ISBN: 978-3540003656

Seng, J.L. and J. Yu, 2004. Towards a more generic desing of XML benchmark workload. Int. J. Comput. Syst. Sci. Eng., 5: 95-114. http://www4.nccu.edu.tw/ezkm11/ezcatfiles/cust/img/img/67.pdf

Stonebraker, M. and D. Moore, 1996. Object-Relational DBMS-The Next Great Wave. 2nd Edn., Morgan-Kaufmann, ISBN: 155-860-397-2, pp: 216.

Strang, T. and C.A. Linnho-Popien, 2004. Context modeling survey. Proceedings of the the 6th International Conference on Ubiquitous Computing, Sep. 7-10, Nottingham, England, pp: 1-8. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.2.2060&rep=rep1&type=pdf

Vanathi, B. and V.R. Uthariaraj, 2010a. Context Representation and Management in a Pervasive Environment. In: Communications in Computer and Information Science, 1, Volume 101, Information and Communication Technologies, Part 3, Springer, Heidelberg, pp: 543-548. DOI: 10.1007/978-3-642-15766-0_92

Vanathi, B. and V.R. Uthariaraj, 2010b. Context representation using hierarchical methodand heuristic based context selection in context aware computing. Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India (A2CWiC'10), Coimbatore. DOI: 10.1145/1858378.1858410

Vanathi, B. and V.R. Uthariaraj, 2010c. Hybrid hierarchical context representation in a context aware system. Proceeding of the 2nd International Conference on IT and Business Intelligence (ITBI'10), IEEE and IEEE Computational Intelligence Society, Nagpur. http://www.eurojournals.com/EJSR_50_4_14.pdf

Vanathi, B. and V.R. Uthariaraj, 2011c. Collaborative Context Management and Selection in Context Aware Computing. In: Communications in Computer and Information Science, 1, Volume 133, Advanced Computing, Part 4, Springer-Verlag Berlin Heidelberg, pp: 348-357. DOI: 10.1007/978-3-642-17881-8_33

Vanathi. B. and V.R. Uthariaraj, 2011a. Context Management Using Ontology and Object Relational Database (ORDBMS) in a Pervasive Environment. In: Communications in Computer and Information Science, 1, Volume 132, Advances in Networks and Communications, Part 1, Springer-Verlag Berlin Heidelberg, pp: 79-87. DOI: 10.1007/978-3-642-17878-8_9,

Vanathi. B. and V.R. Uthariaraj, 2011b. Towards a Hierarchical Based Context Representation and Selection by Pruning Technique in a Pervasive Environment. In: Communications in Computer and Information Science, 1, Volume 131, Advances in Computer Science and Information Technology, Part 1, Springer-Verlag Berlin Heidelberg, pp: 199-208. DOI: 10.1007/978-3-642-17857-3_20

Winograd, T., 2001. Architectures for context. Hum. Comput. Interact. 16: 401-419. DOI: 10.1207/S15327051HCI16234_18