Proceedings of the 10th IASTED International Conference
INTELLIGENT SYSTEMS AND CONTROL
November 19–21, 2007, Cambridge, MA, USA
ISBN Hardcopy: 978-0-88986-707-9 / CD: 978-0-88986-708-6

DESIGN OF PETROLEUM COMPANY'S METADATA AND AN EFFECTIVE KNOWLEDGE MAPPING METHODOLOGY

Shastri L NIMMAGADDA,
Wafra Joint Operations,
Kuwait Gulf Oil Company,
Al-Ahmadi, Kuwait 61003,
Email: NIMM@chevron.com

Heinz DREHER
Curtin Business School,
Curtin University of Technology,
Perth, WA, Australia
Email: h.dreher@ curtin.edu.au

ABSTRACT
Success of information flow depends on intelligent data storage and its management in a multi-disciplinary environment. Multi-dimensional data entities, data types and ambiguous semantics, often pose uncertainty and inconsistency in data retrieval from volumes of petroleum data sources. In our approach, conceptual schemas and sub-schemas have been described based on various operational functions of the petroleum industry. These schemas are integrated, to ensure their consistency and validity, so that the information retrieved from an integrated metadata (in the form of a data warehouse) structure derives its authenticity from its implementation. The data integration process validating the petroleum metadata has been demonstrated for one of the Gulf offshore basins for an effective knowledge mapping and interpreting it successfully for the derivation of useful geological knowledge. Warehoused data are used for mining data patterns, trends and correlations among knowledge-base data attributes that led to interpretation of interesting geological features. These technologies appear to be more amenable for exploration of more petroleum resources in the mature gulf basins.

KEYWORDS
Petroleum exploration, heterogeneous data, schemas, data integration, warehouse modeling, data mining and knowledge mapping

1. Introduction

Design of an integrated information system in the oil and gas industry depends on the design ([8] and [9]) of conceptual schemas of oil and gas industry's individual entities. Integration of schemas belonging to various operational sub-systems ([9] and [10]) is a requirement for an oil and gas industry to accomplish the legality and validity of data. Intelligent and expert data systems [1] are used in geophysical exploration and prospecting. Issues relevant to computer applications in geosciences [5] have been discussed, with demonstrations of their applicability and feasibility. Geophysical methodologies applicable to oil and gas exploration and prospecting have been discussed in [14]. Industry situations with varied data models are described and demonstrated in [4]. Data warehouse has been designed, using metadata conceptualized by entity/object relational data structuring [6] and [7]. Each sub-system corresponds to a conceptual schema for each operational activity such as *exploration*, *drilling*, *production* etc. The conceptual schema communicates [12] with an integrated system encompassing a multilevel database management system. Sub-schemas, known as views, can be integrated in an oil and gas company to arrive at a complete conceptual schema. The integrated conceptual schema design process is shown in Fig. 1. Views, such as *exploration*, *drilling* and *production*, and how they can be integrated in the design schema process, are shown in Fig. 1. The conceptual schema design process is typically an iterative process ([13], [15] and [16]) of refinement and integration of views (shown as a feedback refinement loop in Fig. 1 with arrows) and involves:

1. Decomposition and/or synthesis of entity sets
2. Redefinition of relationships
3. Redefinition of relationship types
4. Redefinition of mapping constraints
5. Redefinition of higher-level abstractions (e.g. generalization or specialization)
6. Rearrangement of attributes among object classes (i.e. entity and relationship sets)

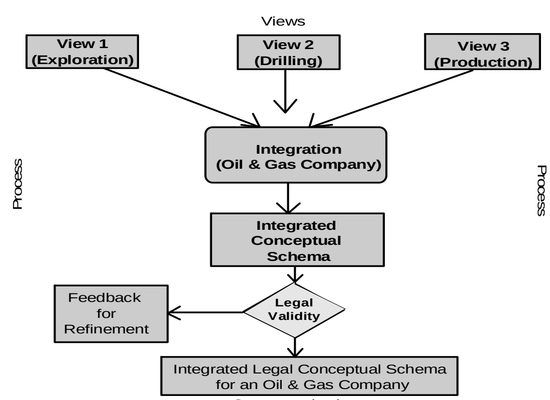

Fig. 1: Integrated conceptual schema-design process

A conceptual schema is said to be legal [12] if it leads to a lossless, normalized, dependency-preserving relational schema at the implementation level. The refinement feedback loop is iterated until a relational schema of required quality is achieved. Methodologies of integrating sub-schema views are given in the following section.

2. Methodologies

Schemas and sub-schemas are integrated after their validation. The process of integration and critical factors associated with development of integration process are narrated in the following sections.

2.1 Schema (View) Integration

A conceptual schema that describes an enterprise of an oil and gas company is the result of an association, bridging various functional sub-systems or divisions. It is conceivable to have each functional sub-system (*exploration* as an example) developing its own schema. When schemas of different oil and gas data entities or objects are disjointed, a union operation is invoked, constructing the overall conceptual schema. However, schemas overlap at several points and it is quite a challenge to draw semantic boundaries between subsystems of a complex nature such as found in the oil and gas industry, with system-wide functional interactions.

Components of centralized information systems are described and each part, referred to as a schema, corresponds to a part of the overall conceptual schema which is called a view. Only when individual views are put together, or integrated, an overall conceptual schema is obtained. Diversity in semantics causes conflicts and variations in modeling, which have to be taken care of systematically. Distributed information systems add a further dimension to the problem. Integration of views is different from database integration where databases of all individual schemas of oil and gas data items are merged centrally, or combined in a distributed database by constructing schemas of schemas, i.e. global schemas. View integration is at a lower level; views are combined into a conceptual schema that represents a database.

2.2 Reasons for Integration Methodology

Data found in the oil and gas industry are often very diverse and heterogeneous. A methodology for data view integration is necessary [12] because it is impossible for people working independently (with individual operations) on data modeling to consistently arrive at the same representation with similar concepts. People have different viewpoints in perceiving data semantics. The richer the abstract model in providing alternatives in representation, the more diverse, is the views explaining the same concepts. A given concept with different names, possibly causing naming conflicts as well as other problems, can be represented with different types. Moreover, incomplete information in the conflicting parts gives rise to inconsistencies, such as differing cardinalities for the same entity or relationship. This often happens in the case of heterogeneous data, where similar data attributes and corresponding values have a common influence in the different operational environments. Besides semantics and naming conventions of data attributes, integration of these attributes of different company entities (with semantics applied), can facilitate the logical and physical data organization of heterogeneous petroleum resources data as a metadata, ultimately permitting the building of knowledge of petroleum systems previously hidden among the data.

2.3 Views can relate

Views can contain anything from totally disjoint events to closely related events. When two views are brought together, enlarging the semantic context, the following possible interactions may be discovered:

1. Inter-schema connections or inter-schema relationships. A geologist object class in one schema and an exploration object in the other, when merged together result in "explorationist" relationship between the geologist and exploration objects (or entities);
2. Common parts of the views may be found to be identical; in which case, the merged schema contains a single copy of the identical representation.;
3. Common parts modeled by two views are not identical but equivalent to each other. In other words, one view can be mapped to the other by some algorithmic transformation, such as when the same concept is an entity in one view and an attribute in the other. Depending on the direction of transformation, the views are merged after one representation is converted to the other.

2.4 Integration process

In this process, N views are combined by pair-wise merging at one extreme or by an all-at-once n-ary merge at the other extreme. In either case, the schemas (or views) are examined in a pre-integration process to determine the degree of conformity, and for conflicts among the views. When schemas are compared, the following are considered:

1. *Naming*: Problems will be discovered due to homonyms and synonyms in the industrial data. An example of a homonym is the use of the same name for different concepts. Such as the use of "*exploration*" as an entity in the "*geologist*-<qualified>-*exploration*" schema and also as an entity in the schema "driller-<belongs>-*exploration* (angle brackets denoting relationships). An example

of a synonym is that between the schemas "*employee*-<*assignedto*>-*exploration* and "*surveyor*-<*conducts*>-*survey*," the same concept being described by two different names, "*exploration*" and "*survey*";

2. *Attribute Correspondences:* Industrial data such as exploration, drilling, mining or production, possess different attributes and their correspondences. The same attributes between schemas may have different data-types (integer vs. real), units etc., which must be converted to a common form. If the keys and underlying domains of attributes are identical, they can be unified with a union operation. Another possibility is containment between sets of attributes. For example, the domain of "*engineers* (or *geologist*)" is a subset of "technical personnel" or "surveyor". Integrating the schemas containing these domains, one per schema, yields a generalization hierarchy between the two entity-sets in the integrated schema. If the underlying domains of the attributes are different, however, the decision rests with the data analyst. The data analyst may choose to create a generalization hierarchy under "*employee*" for *Secretary* (*Eno, Name, Sal, Type-speed*) and *Engineer* (*Eno, Name, Sal, Degree*) even though the domains of engineer and secretary do not intersect. The case *explorationist* (*Dept, Name, Age*) and *secretary* (*Dept, Name, Age*) are not integrable, despite these entities having common attributes;
3. *Structural Correspondence:* These may involve conflicts in types, keys, types of relationships, etc.. Type conflicts arise when different types are chosen for the same concept in different schemas. For example, *Geologist* (*ID, Ssn, Name*)-<worksin>-*Dept* is used in one schema, yet *Geologist* (*Ssn, ID, Name, Dept*) is in another schema. *Dept* is an entity in the first schema, and the *Geologist*'s connection to it is represented by the <working> relationship. In the other more compact representation, *Dept* is expressed as an attribute of the *Geologist* entity. In key conflicts, the same concept may be represented with different primary keys in different schemas, such as the case of *Geologist* having the primary key ID in one schema yet *Ssn* in the other, as shown. Where relationships conflict, for example, a relationship "*Geologist*-(n)<worksin>(1)-*Dept*" can be represented as a weak relationship in one schema (where n, 1 are mapping constraints), making the existence of *employee* dependent on the existence of *Geologist* dependent on the existence of *Department*. Yet, in another schema, the relationship can be a (strong) relationship focused on the department side, meaning that there cannot be any department without any geologists assigned to it.

2.5 Integration of Objects (Entities)

The objects ([6], [7]) under different schemas can be similar due to their key and domain similarities. This similarity can range from identical to a commonality in some domain containment relationships, and to totally dissimilar with disjoint domains (object instances are not common despite being conceptually alike). In the case of identical objects, a single copy of schema and union of the attributes (some may differ) and instances are kept. For example, two different functional units of an enterprise may keep the object "*employee*" between the schemas. Even though the two representations mostly share the same domains, there may be attributes unique to individual schemas such as between *Driller* (*Ssn, Name, Sal, Commission*) and *Driver* (*Ssn, Name, Sal, Overtime*), where commission and overtime are schema-specific. Yet the newly integrated schema contains the concept *Employee* over the entire enterprise as *Employee* (*Ssn, Name, Sal, Commission, Overtime*), which may introduce the use of 'nulls'.

In all the other cases of object integration, the major contribution of integration is identifying generalization hierarchies among similar and dissimilar object classes. As mentioned in the attribute correspondences, the examples for these are as follows: for similar objects, the integration between *surveyor*, and *geologist*; for dissimilar objects, *explorationist* and *secretary* in one case and *driller* and *driver* in the other. Figs.2 (a, b) depict the integration process among these concepts that are placed between two schemas.

As mentioned earlier, dissimilarity here is in the un-commonality of instances, i.e. an *explorationist* is not a *secretary* even though they may not be described similarly. A *driller* is not a *driver*, but in this case there is an opportunity for them to be integrated into a generalization hierarchy as shown in Fig. 2. However, there is a difference between the IS-A hierarchies of Fig. 2 in the sense of domain containment. In *Explorationist*, an inclusive union has to be made (of generalization), whereas in *Employee*, one can have an exclusive union due to the common and disjoint domains of the former and the latter, respectively.

2.6 Integration of relationships

Integrating relationships is the next most difficult type of process. Various aspects must be considered, such as the degree of a relationship, roles of entity sets or objects participating in the relationship, and structural features such as type of relationship and mapping constraints. The degree of the relationship refers to the number of entities a relationship involves. For example, a *Driller-WellSite-Rig* relationship involves the entity sets *driller, rig*, and *well-site* and therefore has a degree of 3. When two relationships are compared for conformity, there may be many combinations among the variables considered. Two relationships can be exactly identical in degree, roles of participating objects, and structural features, or there may exist large differences. In between, there may be several partial matches.

Two relationships with respect to two views (schemas) are highlighted in Fig. 3 (a). Both of these relationships share the same degree, roles, and structural features; hence they are identical. Therefore, in the integrated schema, only one relationship is placed, as shown in Fig. 3 (b). The domains of entity-types *technical personnel* and *Engineer* are related by containment. The integrated schema is combined in a generalization hierarchy, under *driller* which is more generic.

However, this generalization hierarchy would overlap in its specializations. When relationships differ in degree, these may be merge-able if they correspond to each other conceptually and one is a more detailed view of the other. Putting it differently, the relationship of lower degree must be derivable from the relationship of the higher degree by projecting or creating a subset of the latter.

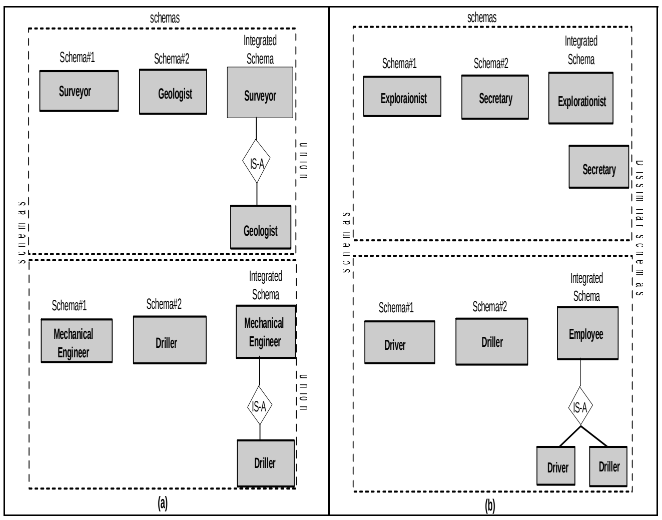

Fig. 2: Objects integration; (a) similar objects and (b) dissimilar objects

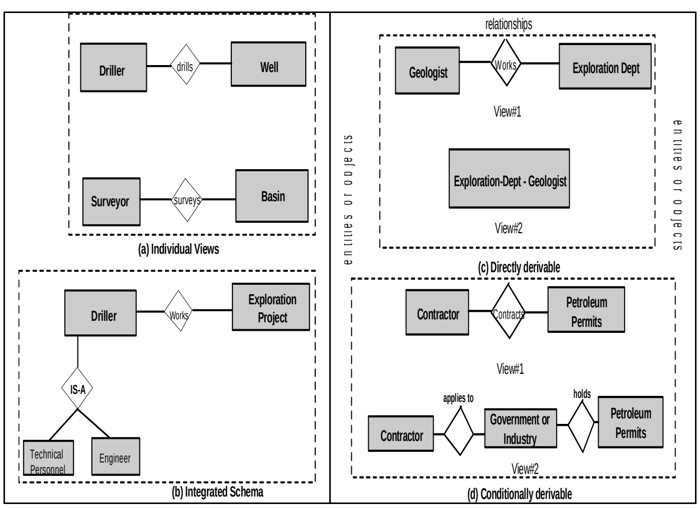

Fig. 3: Identical relationships (a) Individual views (b) integrated schema; Relationships of different degrees (c) directly derivable (d) conditionally derivable

In integrating such schemas, the one belonging to higher degree and representing both is retained. Fig. 3 shows two views that correspond to relationships of differing degrees again. Fig. 3 (c) shows a directly derivable relationship.

View#1 in Fig. 3c is a relationship of degree 2, whereas View#2 represents the whole entity (i.e., in a way, a relationship of degree 1) and is unrestricted in mapping, as opposed to the 1:n mapping constraint of View#2. Therefore, these views are equivalent and View#2 can be derived from View#1 directly. The integrated schema is represented by View#1, i.e., the relationship of higher degree. In Fig. 3 (d), View#2 can represent the integrated schema if View#1 can be derived from it. For this to happen, all the attributes of the *contracts* relationship must be contained in *applies_to*, *government/industry* and *holds*. This case represents a conditional derivability such that if the condition holds, then the schemas are integrable. Fig. 4 shows an extreme case where even though the entity sets are common to the relationships, their roles, semantics, and degrees are different; hence their views are un-integrable. Notice the roles explicitly added for clarity on the connections to the ternary relationship.

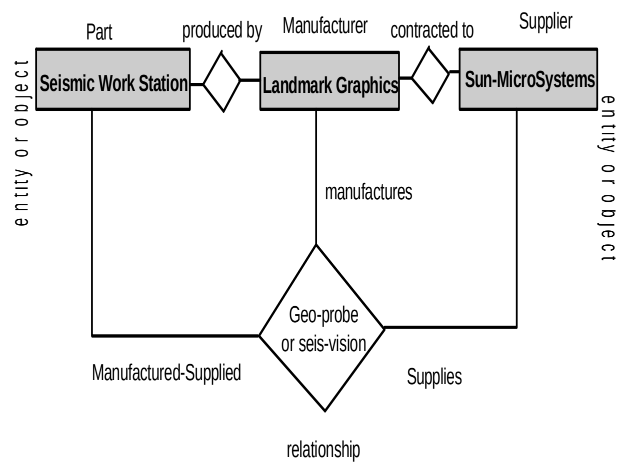

Fig. 4: Un-integrable relationships

So far, different data schemas and their integration have been discussed. Authors attempt to implement them in one of petroleum exploration cases.

3. Oil and Gas Industry (Exploration) Case

There are numerous data entities and attributes in the exploration case study. These are briefly described here.

3.1 Data types and descriptions

The need for warehousing and data mining technologies in the petroleum companies has been explained in [8, 9]. Relational and hierarchical data structures are popularly used, conceptualizing all the data entities and their relationships. Petroleum data from several heterogeneous sources [9] are conceptualized including their relationships. These are intelligently stored in a warehousing environment. Entities are used in conceptual ER modeling and objects in object oriented data modeling approaches. Analogous to entities and objects, dimensions are conveniently used in multidimensional data structuring.

In the past, petroleum companies have typically stored data consisting of only text and numbers, but today, graphics, drawings, photographs, video, sound, voice mails, spreadsheets and other complex objects are stored. Relational database management systems store these data objects and types with certain limitations. The concept of object, which is the core of all OO systems, is some unit of data along with actions, affecting its behavior [6]. A *reservoir* object, for example, could consist of the data relevant to *wells* object (*reservoir* name, type, quality and *production rate*) together with the actions that can take place on *reservoir* object (multi-reservoirs, predicting qualities, *reservoir* extents and thickness for computing the *geological* and *recoverable* reserves from petroleum prospects). Dimensions involved in data structuring are explained in the following sections.

3.1.1 Non-geometric spatial dimension

This dimension contains only non-geometric data. For example, *survey names*, *survey line numbers*, *well numbers*, *survey IDs* and *well IDs* and *permit IDs* and numbers etc, are constructed for the geo-scientific warehouse as a dimension containing only nominal data to locate a phenomenon in space. Such a dimension could start with the names of *exploration permit* names, *survey line* numbers and *basin names*, *state* and *country* names. Such a solution can be implemented as long as navigation representation is not required.

3.1.2 Geometric-to-non-geometric spatial dimension

An *exploration* aggregation component has several dimensions or entities, can further be represented by several object elements and characterized by several patterns. This is a dimension whose prehistoric level data is geometric but whose generalization, starting at a certain high level, becomes non-geometric. For example, a *survey line* represented by a polygon, that is geometric data, is the finest granularity level of this spatial dimension. However, each *survey line* is generalized to some value which is solely nominal and its further generalization remains nominal, thus playing a similar role to a non-geometric dimension at coarser granularities [9] of this spatial dimension.

3.1.3 Fully geometric spatial dimension

This is a dimension whose primitive level and all of its high-level generalizations are geometric elements. For example, polygons of equi-type or value in onshore survey *regions* data (such as equal property of *gravity, magnetic* or *seismic survey* data) or for offshore *regions* data are geometric shaped object classes. There could also be polygons of equal elevations or altitude regions, and every generalization, such as elevations covering 0-700m, 700-1000m. Geometric elements are presented in different object classes. These are demonstrated with the conceptualization of the real world spatial objects [5] through object oriented data modeling reducing the complexity of object data structures. For example, *geological field* samples are represented in *point* objects. "*Seismic survey lines*" is another spatio-temporal view showing occurrence of spatial events in different *basins* represented in *line* objects. *Navigational* data with different polygons and *net oil plays and thickness* contours (2D surface, type of object), describe maps as *region* objects. In case of multidimensional data modeling, measures that can be distinguished within a cube are explained in [8]. These dimensions are narrated as entities or objects are appropriately used in the data schema integration process. A few entity relationships and their schemas have been demonstrated.

Fig. 5 shows two schemas that convey semantics about contractors and their petroleum permits. A problem is discovered with critical examination of two views. They refer to the same concept and *contractor*, *government* or *upstream_company* is selected as the common name. Further, a structural conflict in *contractor* is observed; while it is an entity in View#2, it is used as an attribute in View#1.

Because integration retains the higher degree relationship and View#1 is derivable from View#2, a change can be made in View#1 for conformity, by adding an entity set for contractor and tying it to the *licensed area for exploration* with the applicable relationship while deleting the attribute contractor. The result is shown in Fig. 5(b). Now the schemas conform with each other conceptually, they are merged as shown in Fig. 6. As can be seen, identical objects *contractor* and *government/upstream_company* are copied only once and connected to the other entities with their appropriate relationships.

These relationships however are duplicates, and their simplification can only be possible with a transformation on the merged schemas. Such a transformation is possible since *licensed area for exploration* and *petroleum permits* have common domains and are related to each other through containment. That is, *petroleum permit* is a subset of a licensed area for exploration. This is shown by an IS-A hierarchy. *Permit* inherits all the relationships to its superset, *licensed area for exploration*. Fig. 6(b) shows the transformation and the resulting elimination of the relationships in the integrated schema. Fine-grained refinement of data schemas, combined with data integration process is effective in knowledge mapping [9]. Briefly, refinement of schemas has been discussed in the following sections.

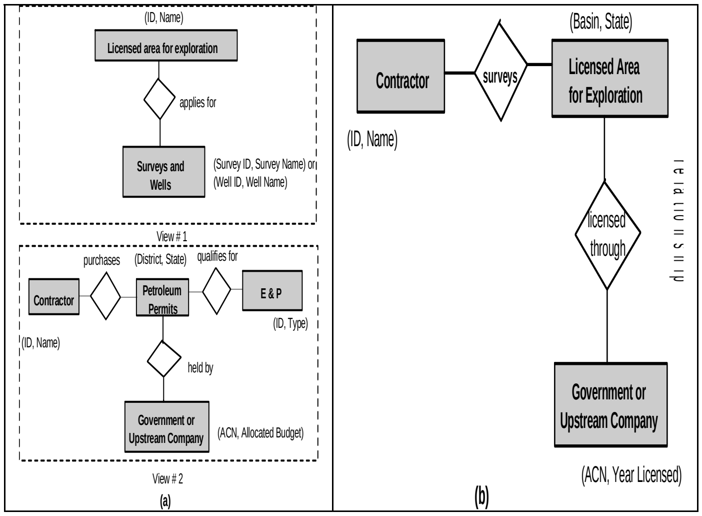

Fig. 5: (a) Original schemas (b) Conforming views

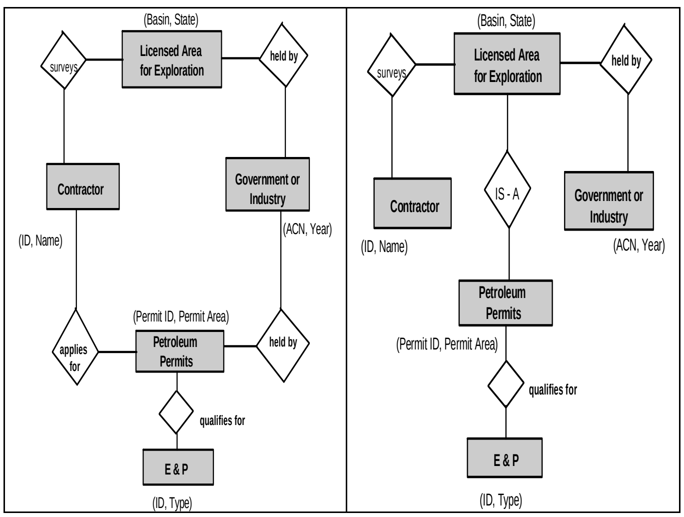

Fig. 6: (a) Merged schemas (b) totally integrable schemas

3.2 Conceptual Schema Refinement

As shown in Fig. 1, a conceptual schema is specified in an enterprise description, and must be refined until it meets the criteria of a legal conceptual schema. Relational schema, which is developed based on relational theory, is the integral target of a conceptual schema. Here, a manual refinement procedure is given citing an example from the resources industry. For deriving a legal schema, possible refinements to the conceptual schema have been demonstrated. This refinement process involves design, information requirements analysis, enterprise description, transaction analysis, schema analysis, normalization and lossless joins.

3.3 A Manual Refinement Process

Manual refinement process is discussed in this section.

3.3.1 Design Topic

The design topic is a surveyor-activity information system. A subset of activities is encompassed which occurs between surveyors and members of the oil and gas company. The project can be divided into the following steps:

1. *Requirements Analysis:* Acquire data in order to describe what is needed and desired by the user;
2. *Enterprise Description:* Begin a rough conceptualization of the model;
3. *Schema Analysis:* Manually refine the system;
4. *Database Description:* Populate the database, apply security controls, and execute queries.

Requirements Analysis: In our Oil and Gas Exploration Data case, data available from several survey documents in multimedia format are gathered for modeling purposes (Tables 1 and 2). Consider the enterprise to be the *surveyor* community with the following sources of information:

Table 1: Documents Narrating Survey Data

Documents in the survey enterprise	***Activities in the survey enterprise***
1. All previous technical and financial reports 2. Maps and geological cross sections and their descriptions	1. Recruitment 2. Training and development 3. Procurement of raw materials 4. Stores & purchases 5. Attendance 6. Book keeping 7. Records & transcripts processing

Table 2: Survey Data Attributes and Description

Functional description of enterprise functions	***Enterprise description:*** Entities and their attributes have explicitly been identified:
1. Surveyors are employed by companies 2. Surveyors are qualified for survey work 3. Surveyors possess skills 4. Surveyors are allotted to field work 5. Surveyors are given responsibility of handling equipment 6. Surveyors are instructed with terms and conditions of working 7. Surveyors acquire exploration data 8. Surveyors discuss the	1. Surveyor: 2. Company: 3. Incentives: 4. Department: 5. Survey Activity: 6. Survey Type: 7. Survey Name: 8. Survey Line: 9. Survey Coordinates: 10. Survey Documents: 11. Survey Budgets:

well positions with drillers 9. Surveyors are offered incentives for acquiring quality and quantity exploration data	

Fig. 7 shows the functional-dependency diagram corresponding to the semantics of the application. Most FDs result from the dependencies on keys. Fig 8(a) displays the preliminary E^2/R conceptual schema of the surveyor-activity information system.

3.4 Transaction Analysis

This process identifies transactions needed in the system. Two examples given are as follows and are shown in Fig. 7.

Transaction: List the surveys and the production rate for a contractor:

Entity types: surveyor, geology
Relationship types: survey or exploration

1. Retrieve the surveyor entity
2. Retrieve geology related to the surveyor entity via survey relationship

Transaction: List all surveyors assigned to a particular basin

Entity types: department, surveyor
Relationships: Approved for survey in a basin

1. Retrieve department entity
2. Retrieve a surveyor entity related to the department entity via approved relationship

3.5 Schema Analysis

The analysis is carried out in the following areas:

Normalization: No MVDs are present; therefore, achieving BCNF will automatically provide 4NF.
Dependency-preserving decomposition: Decompositions to improve normalization must preserve all functional dependencies.
Lossless joins: All decompositions to improve normalization must be lossless join decompositions.

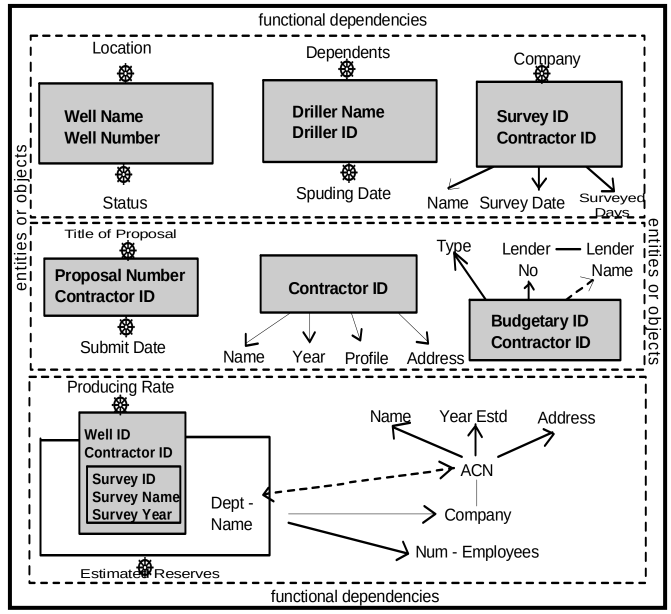

Fig. 7: Functional dependencies in exploration data object

3.6 Normalization

Functional dependency (FD) observes the property of functional mapping in which semantics and integrities among attributes are described. Analysis of relations (NF as described in Table 3) and FDs shows that all relations are in BCNF except *inspection* and *expert*, which are in 2NF. After applying the membership algorithms and the algorithm for dependency-preserving 3NF decompositions, the following can be obtained:

Table 3: Normalized Relations

QC-ID-no: Expert-name is redundant so it is removed	The three relations are in BCNF: Geologist ID-no; well-drilling is also redundant so it is removed
Quality control: ID-no, type Expert: ID-no, name Inspects: Inspect-no, Expert-ID	Geologist becomes: Geologist: ID-no, name, sex, salary Approved: ID-no, Well name Well-drilling: name, basin

Since a relation department (Well-name, basin, num-surveys) already exists, it is used instead of the relation "well-drilling" above. All relations are in BCNF. 2nd normal form (2NF) is one that embodies two disjoint facts together. 2NF is converted into 3NF relational schema, by separating disjoint facts by decomposing "well-drilling" relations. This 3NF is further converted into Boyce-Codd normal form (BCNF) without having dependency of primary attributes (well-name, well-ID) on non-prime attributes.

3.7 Lossless Joins

Because all entities and relationships are linked via their primary keys, all join paths yield lossless joins. In this example decomposition of entities and relationships is not performed and hence no further lossless-ness check is required. As a result of schema analysis, the FD diagram and E^2/R conceptual schema can be refined. As shown in Fig.7, FDs connected to the dashed lines are ignored, and the required FD diagram is obtained. As shown in Fig.8 the refined E^2/R conceptual schema of the *surveyor-activity* enterprise is displayed.

3.8 Rule Based Refinement

The design of a good (or refined) conceptual schema that yields a legal relational schema is an iterative process and cannot be easily accomplished in an ad hoc manner. A legal relational schema is one that contains relations that are lossless, dependency-preserving, normalized, and free of unnecessary redundancies and anomalies. With the advent of knowledge-based systems and rule-based programming, it is feasible to encapsulate a substantial amount of the database administrator's knowledge into a system capable of performing normalization operations and making database-design decisions.

Methodologies used in the knowledge based systems for conceptual schema conversion have been discussed in [2]. The system is programmed in PROLOG and it is an iterative design session. Design methodologies that use the concepts in relational theory are discussed here. The algorithms can easily be programmed using recursion. The refinement flowchart is shown in Fig. 9, and the process consists of E^2/R DDL information. The main ingredients of DDL are entities, relationships, aggregates and generalizations. The knowledge based systems convert these objects into their relational equivalent before applying the refinement procedure. The input may also contain functional and multi-valued dependencies. Normalization is the main highlight of this process. The output of the refined relational schema is synthesized back into E^2/R DDL to complete the cycle. The feedback for refinement process as seen in Fig. 1, in reality takes place during the normalization phase.

The main body of refinement process is therefore in the normalization phase, which may consist of 3NF, BCNF as discussed in [6]. It is not a simple normalization filtering process, but application of the complete know-how to produce lossless, dependency preserving, and normalized schema. Know-how can be provided with algorithms developed for normalizing relations in the different stages. These algorithms have been extensively discussed in [12]. Briefly, algorithms which can easily be programmable [3] have been discussed in [12].

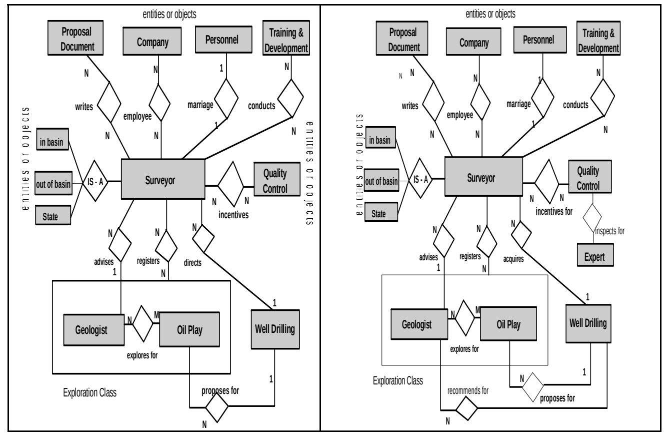

Fig. 8: (a) Preliminary E^2/R conceptual schema (b) refined E^2/R conceptual schema

4. Mining of Warehoused Exploration Metadata

The resources data are complex in nature with several entities, dimensions, objects and attributes. The data integration is crucial for industry managers to make technical, financial and human resources procurement decisions. Methodologies discussed in this paper narrate all the conceptual schemas simplifying the logical data models and facilitate implementation for oil and gas companies. Sub-schemas interpreted as views can easily be added to the existing schemas, so that data integrated are current and allow data warehouses to extract user defined views more precisely. Implementation of conceptual and logical data structures for knowledge mapping has been discussed. Logical and physical data are organized for one of the sedimentary basins in the Arabian Gulf region. This demonstrates the data integration procedure (as described in Fig. 10a,b) which is a prerequisite to explore and exploit interesting geological features attributable to the petroleum prospects in Arabian Gulf offshore basins.

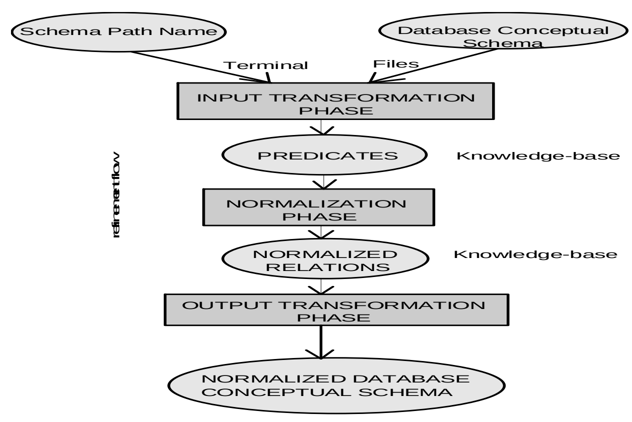

Fig. 9: Conceptual schema refinement flow for integration

Process of integrating exploration data from different sources (such as exploration and drilling) is described in Fig.10. This exhibits and describes cross-clustered petroleum exploration data shared by different users. Besides data integration, sharing of exploration data among different clusters or sources is identified as another important issue. In a typical exploration and production petroleum company, geologists, geophysicists, petrophysicists, reservoir engineers and production engineering professionals work in a team environment. Data are extracted from different sources for data integration.

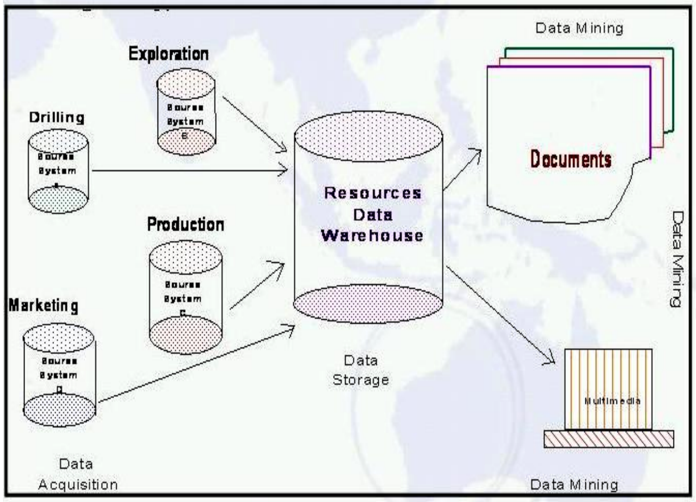

Fig. 10a: Data integration process, indicating various data entities and attributes involved

The classification of the data sources into surface and sub-surface based, seismic, well, reservoir and production data types is documented. As a part of integration process, 3D seismic data (though surface-based, but interpreted as sub-surface) in one of the producing areas (Figs. 11-12) have been calibrated with number of wells and thus mapped a producing geological horizon. Several attribute maps are generated for making crucial technical decisions on drillable exploratory and development locations in the offshore basin.

As an example, structure attribute map of a geological horizon has been integrated with formation tops derived from wells. The exploration area has indicated "clusters" of sinusoidal (or sine-wave shaped) channel features (Fig. 13), a potential development of strati-structural traps associated with this geological horizon. Good porosity attributes have been extracted on top of this geological horizon from many wells. The authors take advantage of the development of good seismic signature of formation in the oil field producing area for better well data integration.

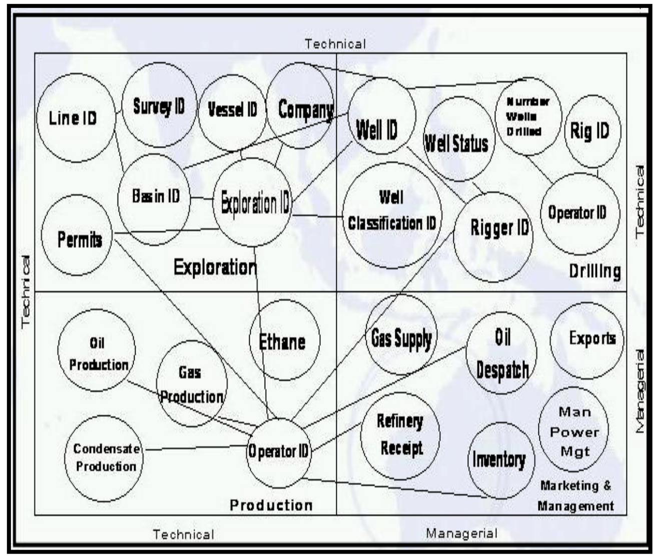

Fig. 10b: Clusters of entities - Data integration and sharing

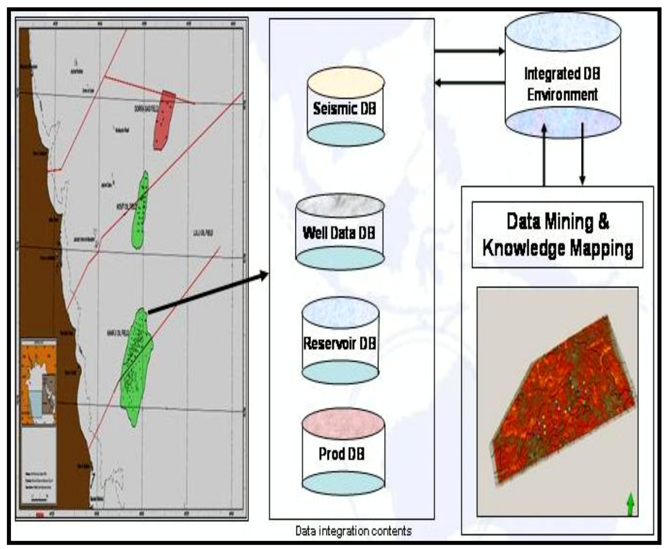

Fig. 11: Process of scaling the structure and well data

As shown in Fig.11, resources data warehouse that captures the exploration, drilling, and production data from operational environments are metadata which are logically and physically organized for data mining purposes. In other words, data from different layers, such as surface and sub-surface are properly organized and integrated to get a good image of sub-surface structure and reservoir of petroleum provinces in the Gulf regions. Many more petroleum resources can be generated if the exploration and production datasets and their structures are logically modeled and better understood. Thus, the knowledge built from data integration process is represented in the form of maps. These maps also characterize in the form of data visualization and data mining, interpretable for drillable exploratory and development locations in the Arabian Gulf region.

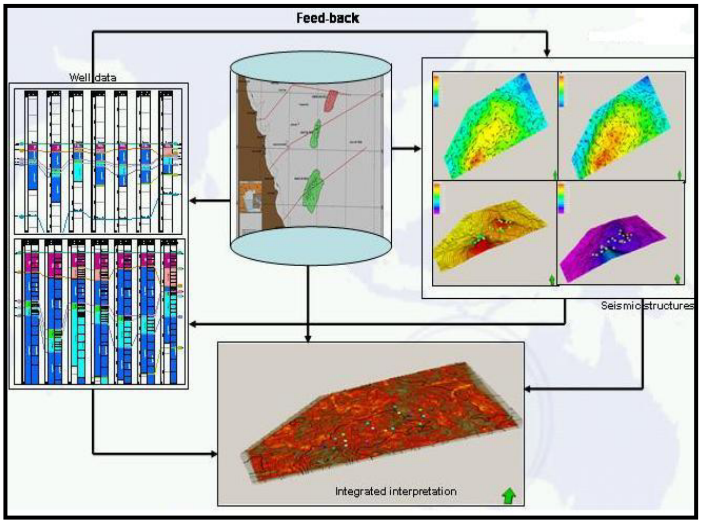

Fig. 12: Mapping of knowledge features for geological interpretation

Similar data integration and knowledge mapping is under study among several other database entities in the producing basins in the middle-east region. There is further scope of connecting onshore and offshore data entities and their possible *area extents* by these data integration methodologies.

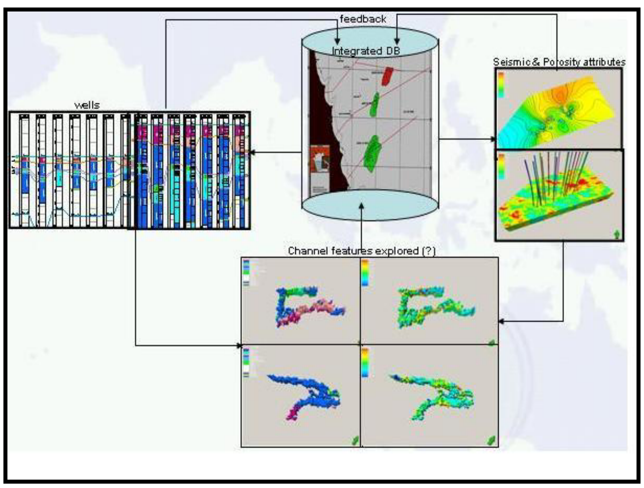

Fig. 13: Interpreting of knowledge features identified (in Fig 12) for drillable locations

5. Conclusion

Data integration process using petroleum metadata is a significant feature when oil and gas industry's functions and activities are viewed as a total conceptual schema. The examples cited here are from exploration data objects. Similar design methodology can be attempted for other entities of oil and gas company's data, such as *drilling* and *production* data classes. Refinement process must be carried out taking into account all the entities and/or objects that have been interpreted during the conceptual schema and sub-schema designs, incorporating the semantics of data attributes. Conceptual schemas are finally converted into logical and implementation data models. An integrated metadata model is generated after integrating all individual metadata sub-schemas.

Authors have attempted to interpret interesting geological features from knowledge from similarities and classifications of integrated exploration metadata. There is a high degree of confidence between *well* and *seismic* entities and their associated attributes, derived porosity attributes at well locations. This infers that a seismic derived *porosity* map depicts exactly the actual porosity of *reservoirs* throughout the mapping area. Stratigraphic and combinational trap attributes (strati-structural) are interpreted in an offshore area in the Gulf basin where hydrocarbon distribution is controlled by both structure as well as reservoir attributes within this basin. Log-derived (sub-surface definition) *porosity* of the *reservoir* attribute is mapped using well data from nearby exploration or development wells. Seismic *structure* attribute map that is superimposed with the respective amplitude data characteristics, indicate definite channel shaped features that follow the structural trends, such as *nosing*, *saddle* and *anticline geological features* (interpreted attributes and their instances) for determining good *reservoir* distribution patterns in the present exploration area. Interpretation of stratigraphic trap feature is considerably increased with effective use and integration of data warehousing and data mining technologies. Data mining facilitates bringing different data of multiple scales to an interpretable scale, especially in the Gulf basins where there are multiple *reservoirs* systems.

References

[1] Aminzadeh, F. and Simann, M. (1991) Expert Systems in Exploration (Geophysical Development Series, Vol.3), Society Exploration Publishers.

[2] Bayle, A. and Ozkarahan, E. (1988) Knowledge based system for relational normalization of GDBMS Conceptual schemas, "Proceedings of the fourth IEEE conference on artificial intelligence applications", pp. 159-165.

[3] Deitel, H.M. and Deitel, P.J. (2001) C++ How to Program (Introducing Object-Oriented Design with the UML), 3rd Edition, Prentice Hall Publishers.

[4] D'Orazio, R., and Happel, G. (1996) Practical Data Modelling for Database Design. The information Technology Series, John Wiley & Sons Australia Ltd, 180-280p.

[5] Green, W.R. (1991) Exploration with a computer: Geoscience Data Analysis Applications (Computer Methods in Geosciences, Vol, 9), 1st Edition, Pergamon Press.

[6] Hoffer, J.A, Presscot, M.B and McFadden, F.R (2002) Modern Database Management, Sixth Edition, Prentice Hall.

[7] Huynh, T.N. Mangisengi, O. and Tjoa, A.M., (2000) Metadata for Object-Relational Data Warehouse, *Proceedings of the International Workshop on Design*

and Management of Data Warehouse, Stockholm, Sweden, June 5-6.
[8] Nimmagadda, S.L, and Dreher, H. (2006b) Mapping of Oil and Gas Business Data Entities for Effective Operational Management, a paper presented and published in the *proceedings of the 4th International Conference of IEEE Industry Informatics*, held in Singapore, August.
[9] Nimmagadda, S.L. and Rudra, A. (2004d) Data sources and requirement analysis for multidimensional database modeling – an Australian Resources Industry scenario, *published in the proceedings in the 7th international conference* on IT, held in Hyderabad, India.
[10] Nimmagadda, S.L, Dreher, H. and Rudra, A. (2005b) Warehousing of object oriented petroleum data for knowledge mapping, a paper presented and published in the *Proceedings of the 5th International Conference of IBIMA*, Cairo, Egypt.
[11] Nimmagadda, S.L. and Rudra, A. (2005c) Data Mapping Approaches for Integrating Petroleum Exploration and Production Business Data Entities for Effective Data Mining, a paper presented and published in the *Proceedings of 3rd Kuwait International Petroleum Conference and Exhibition (KIPCE2005)*, Kuwait City.
[12] Ozkarahan, E., (1990) Database Management, Concepts, Design and Practice, Prentice Hall Publications.
[13] Silberschatz, A, Korth, H.F. and Sudarshan, S. (2002) Database System Concepts, 4th Edition, McGraw Hill.
[14] Telford, W.M. Geldart, L.P. and Sheriff, R.E. (1990) Applied Geophysics, 2nd Edition, Cambridge University Press.
[15] Rob, P. and Coronel, C. (2004) Database Systems, Design, Implementation and Management, 4th Edition, Thomson Learning Series.
[16] Watson, R.H (2002) Data Management: Databases and Organizations, 3rd Edition, John Wiley & Sons, Inc.