Pipeline Production Data Model

Jitao Yang, Yu Fan, Yinliang Liu, Hui Deng, Yang Lin

Pipeline Production Research Lab
PetroChina Company Limited
yangjitao@petrochina.com.cn

Outline

Long-distance Pipelines

- With the rapid construction of long-distance oil and gas pipelines, a cross-country and cross-region oil and gas pipeline network was gradually formed in China and its surrounding areas.
- CNPC owns about 50,000 kilometers long-distance pipelines, of which approximately
  - 33,000 kilometers of natural gas pipelines,
  - 10,000 kilometers crude oil pipelines, and
  - 6,700 kilometers refined oil pipelines.
- This makes the production management of oil and gas pipeline network become increasingly complicated and difficult.

Pipeline Production Management System

Pipeline Production Management System (PPMS) is an information system managing the storage and transportation business of long-distance oil and gas pipelines.

In particular, PPMS focused on the business of: planning management, scheduling and operation, transportation and distribution, professional computing, statistics and analytics, etc.

A pipeline production management system will be used to:

- standardize the management of pipeline production and operation,
- safeguard the accuracy and the completeness of data,
- improve the efficiency and effectiveness of pipeline production management, as well as
- reduce the burden of grass-roots staffs.

Outline

Content

By content of a resource r we mean the set of other resources that make up r; each such resource is called a part of r.

Some important **concepts** in pipeline production:

- Station, is the industrial place supporting the operation, maintenance, and monitoring of pipelines. For example, there are pump station, compressor station, metering station, valve chamber, and oil depots etc. along the pipelines.
- Pipe Segment, is the pipe between two stations. Pipe segment is represented by a directed line segment.
- Pipeline, is the pipe connected by a plurality of directed pipe segments.

Content

- Equipment, is the device used in pipeline production, such as Compressor, Chromatographic Analyzer, Flow Meter, Pump, Air Cooler, and other oil and gas pipeline monitoring and control devices.
- Organization, is the oil and gas pipeline production business-related company and the administrative department affiliated to the company.
- Customer, is the buyer of Natural Gas, Crude Oil, Compressed Natural Gas (CNG), or Liquefied Natural Gas (LNG).
- Supplier, is the company of supplying Natural Gas, Crude Oil, CNG, or LNG.
- Transport Location, is the location where the gas or oil was transported away by train, boat, lorry or the other carriers except by the pipelines.

Content

In pipeline production, some of the resources are closely **related**, for instance:

- Pipe Segment and Station: each pipe segment is defined by the starting station and the ending station.
- Pipeline and Pipe Segment: pipeline is composed by a plurality of pipe segments in a certain order.
- Station and Organization: an organization can manage multiple stations.
- Pipeline and Organization: a pipeline can be managed by multiple organizations.
- Equipment, Station and Organization: an equipment can be associated with multiple stations or multiple pipelines. By default, an equipment belongs to the organization that manages the station where the equipment is located at; if an equipment is associated with multiple stations, then the equipment belongs to the least upper bound organization that can manage all the stations that the equipment is associated with.

Description

- A description of a resource *r* is the descriptive information of the resource.
- For example,
  - The description of a station will include the location and the type of the station.
  - The rotation speed of compressor (*RotationSpeed*) is a very important operating parameter used for monitoring the status of compressor; *RotationSpeed* could be associated with *CompressorSN*, *Station*, *Time*, *Date* and the other related properties by descriptions.
  - A pipeline, however, can be described from different points of view, each leading to a different description. As a result, a pipeline might be associated to a set of descriptions (and the same goes for a station).
- In order to accommodate several descriptions for the same resource, we will treat descriptions as resources in their own right.

Description Reuse

There are thousands and tens of thousands of data acquisition points along the pipelines, some of the data points are the same type.

Descriptions should be reusable so that to reduce the workloads of describing similar resources. Generally, we have two ways of reuse:

- reuse without any customizations, which means a description will be reused directly without any modifications. For example, a station could have multiple compressors, the data acquisition points of compressor rotation speed are the same type, these data points could be described by reusing a same description.
- reuse with customizations, which means a description will be reused by extending the description with new properties. For example, to further describe the rotation speed of a compressor *RotationSpeed*, we could attach new properties for instance *InletPressure* and *OutletTemperature* to the description for *RotationSpeed*.

Outline

The Language $\mathscr{L}$ of Our Model

The language that we propose for our data model is a function-free first-order language, the predicate symbols are listed as follows:

- Class(s, c), expresses the fact that c is a class defined in schema s.
- Property(s, p), expresses the fact that p is a property defined in schema s.
- Domain(s, p, c), expresses the fact that in schema s class c is the domain of property p.
- Range(s, p, c), expresses the fact that in schema s class c is the range of property p.
- IsaCl(s, c_1, c_2), expresses the fact that in schema s class c_1 is a sub-class of class c_2.
- IsaPr(s, p_1, p_2), expresses the fact that in schema s property p_1 is a sub-property of property p_2.

The Language $\mathscr{L}$ of Our Model

- Description(d, s, p), expresses the fact that property p over schema s belongs to description d.
- PrVal(i, s, p, j) expresses the fact that i has a resource identified by j as value of property p from schema s.
- ClInst(i, s, c) expresses the fact that i is an instance of class c from schema s.
- PartOf(i_1, i_2), expresses the fact that i_1 identifies a resource which is part of the resource identified by i_2.
- DescOf(d, i), expresses the fact that d is a description of i.

We denote the the above predicate symbols defined in first-order logic as $\mathscr{L}$.

The Axioms $\mathscr{A}$ of Our Model

The axioms of our model will be used to capture the meaning of the predicate symbols, as well as capture the implicit knowledge in pipeline production.

1. If property p has class c as its domains in a schema s, then p and c must be defined in s:

$$\text{Domain}(s,p,c) \to (\text{Property}(s,p) \wedge \text{Class}(s,c))$$

2. If a property p has class c as its ranges in a schema s, then p and c must be defined in s:

$$\text{Range}(s,p,c) \to (\text{Property}(s,p) \wedge \text{Class}(s,c))$$

3. If c_1 is a sub-class of c_2 in a schema s, then c_1 and c_2 must be defined in s:

$$\text{IsaCl}(s,c_1,c_2) \to (\text{Class}(s,c_1) \wedge \text{Class}(s,c_2))$$

4. If p_1 is a sub-property of p_2 in a schema s, then p_1 and p_2 must be defined in s:

$$\text{IsaPr}(s,p_1,p_2,) \to (\text{Property}(s,p_1) \wedge \text{Property}(s,p_2))$$

The Axioms $\mathscr{A}$ of Our Model

5. If i is an instance of class c over schema s, then c must be defined in schema s :

$$\mathrm{ClInst}(i, s, c) \rightarrow \mathrm{Class}(s, c)$$

6. If a description d contains a property p over schema s, then p must be defined in schema s :

$$\mathrm{Description}(d, s, p) \rightarrow \mathrm{Property}(s, p)$$

7. Sub-class is reflexive:

$$\mathrm{Class}(s, c) \rightarrow \mathrm{IsaCl}(s, c, c)$$

8. Sub-class is transitive:

$$(\mathrm{IsaCl}(s, c_1, c_2) \wedge \mathrm{IsaCl}(s, c_2, c_3)) \rightarrow \mathrm{IsaCl}(s, c_1, c_3)$$

9. Sub-property is reflexive:

$$\mathrm{Property}(s, p) \rightarrow \mathrm{IsaPr}(s, p, p)$$

The Axioms $\mathscr{A}$ of Our Model

10. Sub-property is transitive:

$$(\mathrm{IsaPr}(s, p_1, p_2) \wedge \mathrm{IsaPr}(s, p_2, p_3)) \to \mathrm{IsaPr}(s, p_1, p_3)$$

11. If i is an instance of class c_1 from schema s, and c_1 is a sub-class of c_2 in s, then i is also an instance of class c_2 from schema s :

$$(\mathrm{ClInst}(i, s, c_1) \wedge \mathrm{IsaCl}(s, c_1, c_2)) \to \mathrm{ClInst}(i, s, c_2)$$

12. If p_1 is a sub-property of p_2 in s, and j is a p_1-value of i, then j is also a p_2-value of i:

$$\begin{aligned}&(\mathrm{IsaPr}(s, p_1, p_2) \wedge \mathrm{PrVal}(i, s, p_1, j))\\ &\to \mathrm{PrVal}(i, s, p_2, j)\end{aligned}$$

13. If d describes i that is part of j, then d describes j too:

$$(\mathrm{DescOf}(d, i) \wedge \mathrm{PartOf}(i, j)) \to \mathrm{DescOf}(d, j)$$

This axiom transfers descriptions from parts to the whole.

The Axioms $\mathscr{A}$ of Our Model

We denote the above set of axioms as $\mathscr{A}$, and we denote the theory defined by $\mathscr{L}$ and $\mathscr{A}$ as $\mathscr{T}$.

An interpretation I is created by the contents and the descriptions manually inserted by users or gathered from the relevant systems. The resulting pipeline production is then given by applying the axioms to these facts. In order to make this concept more precise, we re-write the axioms $\mathscr{A}$ in the form of a positive datalog program $D_{\mathscr{A}}$ as follows:

$$
\begin{array}{lll}
\mathsf{Property}(s,p) & :- & \mathsf{Domain}(s,p,c) \\
\mathsf{Class}(s,c) & :- & \mathsf{Domain}(s,p,c) \\
\mathsf{Property}(s,p) & :- & \mathsf{Range}(s,p,c) \\
\mathsf{Class}(s,c) & :- & \mathsf{Range}(s,p,c) \\
\mathsf{Class}(s,c_1) & :- & \mathsf{IsaCl}(s,c_1,c_2) \\
\mathsf{Class}(s,c_2) & :- & \mathsf{IsaCl}(s,c_1,c_2) \\
\mathsf{Property}(s,p_1) & :- & \mathsf{IsaPr}(s,p_1,p_2) \\
\mathsf{Property}(s,p_2) & :- & \mathsf{IsaPr}(s,p_1,p_2) \\
\mathsf{Class}(s,c) & :- & \mathsf{ClInst}(i,s,c) \\
\mathsf{Property}(s,p) & :- & \mathsf{Description}(d,s,p) \\
\mathsf{IsaCl}(s,c,c) & :- & \mathsf{Class}(s,c) \\
\mathsf{IsaCl}(s,c_1,c_3) & :- & \mathsf{IsaCl}(s,c_1,c_2),\ \mathsf{IsaCl}(s,c_2,c_3) \\
\mathsf{IsaPr}(s,p,p) & :- & \mathsf{Property}(s,p) \\
\mathsf{IsaPr}(s,p_1,p_3) & :- & \mathsf{IsaPr}(s,p_1,p_2),\ \mathsf{IsaPr}(s,p_2,p_3) \\
\mathsf{ClInst}(i,s,c_2) & :- & \mathsf{ClInst}(i,s,c_1),\ \mathsf{IsaCl}(s,c_1,c_2) \\
\mathsf{PrVal}(i,s,p_2,j) & :- & \mathsf{IsaPr}(s,p_1,p_2),\ \mathsf{PrVal}(i,s,p_1,j) \\
\mathsf{DescOf}(d,j) & :- & \mathsf{DescOf}(d,i),\ \mathsf{PartOf}(i,j)
\end{array}
$$

Pipeline Production

Given an interpretation I which can be seen as a set of facts of pipeline production, then the above rules in $D_{\mathscr{A}}$ will be applied in order to derive the minimal model of $D_{\mathscr{A}}$ containing I.

The minimal model will be a larger set of facts containing I as well as all the consequences of I according to $D_{\mathscr{A}}$.

Based on the logical programming, the minimal model exists and is unique.

Definition

(Pipeline Production) Let I be any interpretation of $\mathscr{L}$, we call pipeline production over I, denoted PP, the minimal model $\mathscr{M}(D_{\mathscr{A}}, I)$ of $\mathscr{A}$ that contains I.

Outline

Query the Data of Pipeline Production

When searching the data of pipeline production, an intuitive and straightforward way of expressing the user's information need is to relate description to the sought resource.

For example, to search all the compressors in the station "UZS", the user could use a query like:

$$(\exists\ ?\mathit{csn}) \quad \mathrm{PrVal}(?\mathit{csn}, s, \mathit{LocatedIn}, \text{“UZS”})$$

For allowing queries to state simple conditions on property values, we consider any completion of PP endowed with six built-in relations, namely the $=$, $\neq$, $>$, $<$, $\leq$ and $\geq$ relations; we consider these relations as built-in predicates, therefore not subject to the completion of PP.

Query the Data of Pipeline Production

Definition

(Query over Pipeline Production) A *query over* pipeline production PP is well-formed formula $Q(x_1, \ldots, x_n)$ of $\mathscr{L}$, in which $x_1, \ldots, x_n$ are free variables and $n \geq 1$.

The answer of a query with n ($n \geq 1$) free variables is the set of resources $\langle r_1, \ldots, r_n \rangle$ such that, when every variable x_i is bound to the corresponding resource r_i, the resulting formula of $\mathscr{L}(r_1, \ldots, r_n)$ is true in PP. Formally, we have the following definition for the answer of a query.

Definition

(Answer of A Query) The *answer* of a query $Q(x_1, \ldots, x_n)$ over pipeline production PP is given by:

$$answer(Q, \mathrm{PP}) = \{\langle r_1, \ldots, r_n \rangle \mid Q(r_1, \ldots, r_n) \in \mathrm{PP}\}$$

Outline

- Introduction
- The Basic Concepts of Pipeline Production
- Pipeline Production Definition
- Query the Data of Pipeline Production
- **Pipeline Production System**
- Conclusions and Future Work

Relational Implementation

A simple strategy for implementing the model could then consist of the following steps:

- Store the initial interpretation I of pipeline production to a relational database $RDB(I)$; the mapping from I to $RDB(I)$ is straightforward.
- Compute the completion of $RDB(I)$ to obtain the database $RDB(\mathscr{M}(D_{\mathscr{A}}, I))$ via a datalog engine; this requires adding tuples to the tables in $RDB(I)$ using the inference mechanism that we have described in $\mathscr{A}$.
- Map each query Q against the pipeline production to an equivalent SQL query $SQL(Q)$.

By choosing an implementation strategy based on relational technologies, we can benefit at a minimal effort of the scalability and the optimized query evaluation of relational database management systems (RDBMSs).

RDF based Implementation

In order to implement our model in RDF, we must provide means for:

- mapping the relations of $\mathscr{L}$ into equivalent RDF graphs,
- mapping the inference mechanism of our data model into that of RDF, and
- translating the query language of our model to SPARQL.

As we have already observed, implementing our model using RDF is more complicated, while implementing our model using relational technologies is straightforward.

Function Architecture of Pipeline Production Management System - Business

- Planning Management, manages yearly agreement intake and transportation volume of gas/oil,subdivided monthly production plan, etc.
- Scheduling and Operation, manages station daily pipeline production operating data, production logs, unit operating data, flow-meter operating data, chromatographic analyzer operating data.
- Transportation and Distribution, manages the daily gas/oil intake volume, daily gas/oil distribution volume, gas/oil quality information and etc.
- Professional Computing, provides pipeline production professional computing services. For instance, pipe deposit calculation according to the ISO 12213-2:2006.
- Statistics and Analytics, generates different kinds of reports including production daily report, energy consumption report, etc for the group company headquarters, the regional companies, and the vendors, as well as provides decision support.

Function Architecture of Pipeline Production Management System - System

- System Management, configures the system functions, such as the user group creation based on users' authority layers, user role configuration based on user's duties, work-flow management based on business process, and etc.
- Basic Information Management, manages the basic information of the system such as pipeline basic information maintenance, pipeline network and station visualization, pipeline running dynamic visualization, supplier and customer information management, and etc.
- Application Management, manages the configurable applications such as the configuration of different form templates for different stations, configuration of different reports based on different companies' requirement, and etc.
- Working Assistant, enhances production management efficiency by the functions of job alerts, work-flow process, system information publish, short message service, personal information management, and the other auxiliary functions.

Function Architecture of Pipeline Production Management System

BUSINESS FUNCTIONS

Planning Management	Scheduling and Operation	Transportation and Distribution	Professional Computing	Statistics and Analytics
1 Annual Budget Management 2 Transportation and Distribution Plan (Annual, Monthly, Weekly, Daily) 3 Holiday Gas Supply Plan 4 Winter Gas Supply Plan	1 Process Technology Chart Management 2 Scheme and Plan Management 3 Operating Parameter Confirm 4 Equipment Status 5 Duty Management 6 Operation Scheme and Analysis Report	1 Natural Gas Transportation and Distribution Daily Report 2 Transition Measuring Certificate 3 Daily Gas Intake Volume 4 Gas Quality Information	1 Pipe Deposit Calculation 2 Turnover View 3 Pipe Deposit Computing Parameters Adjustment 4 Material Balance Computation	1 Daily Report 2 Monthly Report 3 Quarterly Report 4 Annual Report 5 Decision Support 6 Fixed Report Form 7 Dynamic Report Form 8 Flexible Analysis

SYSTEM FUNCTIONS

Working Assistant	Application Management	Basic Information Management	System Management
1 Job Alert 2 Reporting Progress 3 Publish Information 4 Profile 5 Change Password 6 Send SMS 7 Process Workflow	1 Data Point Template 2 Entity Management 3 Benchmark Management 4 Workflow Template 5 Form Template 6 Report Template 7 SCADA Management 8 Interface Management	1 Pipeline Management 2 Station Management 3 Customer Management 4 Equipment Management 5 Pipeline Segment Management 6 Supplier Management 7 System Dictionary	1 Reporting Progress Checking 2 Pipeline Network Dynamic Information 3 User Management 4 User Group Management 5 User Role Management 6 Organization Management 7 Page Management 8 Function Management 9 Operation Management 10 Manual Management 11 Log Management 12 Workflow Monitoring 13 Workflow Design 14 Report Configuration 15 SCADA Data Acquisition

Figure 1 : Function Architecture of Pipeline Production Management System

System Architecture of Pipeline Production Management System

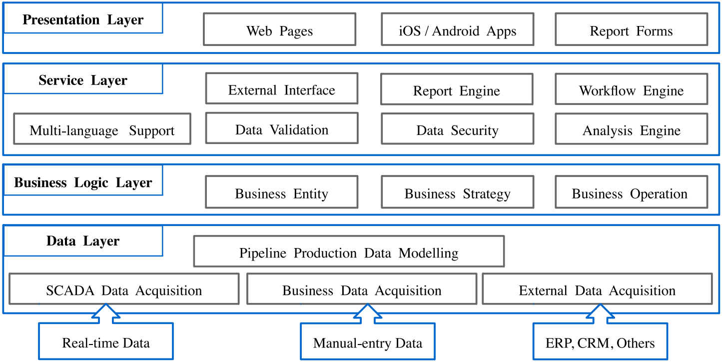

Figure 2 : System Architecture of Pipeline Production Management System

Software Architecture of Pipeline Production Management System

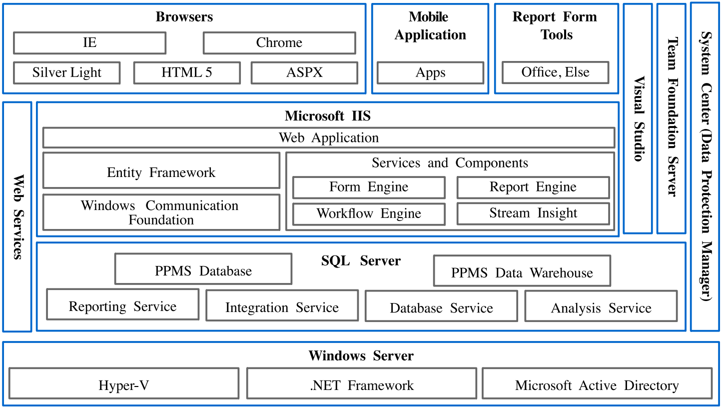

Figure 3 : Software Architecture of Pipeline Production Management System

The Security of Pipeline Production Management System

A few ways were used to ensure the security of pipeline production management system:

- communications security: strict access control was provided by a firewall that isolates the enterprise internal network from the outside world; Demilitarized Zone (DMZ) and HTTPS encryption were used to protect external access to PPMS; in addition, hardware encryption (USB key) was used for PPMS user authentication.
- data security: local and off-site data backups were performed regularly (in seconds).
- system application security: hot standby mechanism was established for important services; virtualization technology was used to enhance the high availability of web servers.

Outline

Conclusions

- We have defined a data model for pipeline production based on two basic concepts: content and description.
  - The two concepts are expressed by a certain set of predicates using a first-order logical approach,
  - the axioms of the model are defined to fix the semantics of the predicates and to capture the implicit knowledge in pipeline production.
- Descriptions are placed as first class citizens with their own identifiers and are defined independently of the resources they might be attached to, additionally descriptions are flexible for extension and reuse.
- We also present a query language for discovering resources of interest in pipeline production based on the descriptions attached to the resources.
- Following our model, we implemented a PPMS. The system received very high appraisal from the users of the group company, the regional companies, and the overseas companies.

Future Work

- The group company has many different information systems, which are categorized into
  - ERP Systems,
  - Petroleum Exploration and Development Systems,
  - Refining and Marketing Systems,
  - Service and Support Systems,
  - Infrastructure and Security Systems, etc.
- We are currently working towards the extension of our model to manage the big data across these above mentioned systems to allow all information systems of the enterprise to operate together in a cooperative manner, so that to maximize the overall data management and analysis benefit to the enterprise.

Thanks

QUESTIONS