Data integration technologies in exploration and production

Steve Hawtin of UK company Oilfield Systems produced this review of key data management issues earlier this year for the Geoshare User's Group. In view of the interest expressed in his assessment, *First Break* is publishing a slightly edited (for space reasons) version of the original.

Does XML mean that SEG-Y, LIS and LAS are redundant? Does DAEX replace the need for Geoshare? Will OpenSpirit mean that information transfers are no longer needed? These are the types of questions that are confronting data management professionals in Exploration and Production (E&P) departments throughout the Oil Industry.

Data integration

Before Data Integration Technologies can be addressed, there are some questions that must be answered. What does this paper mean by Data Integration? Is there a distinct set of technologies that address the questions of Data Integration or is the category of Data Integration Technologies more apparent than real? In order to answer such questions it is best to start at the very beginning.

One of the most important roles of Information Technology (IT) departments in Oil Companies is to support a process like that shown in Fig. 1. If this is used as our model, Data Integration occurs in the Some Process phase. Measurements taken of the 'real world' use a variety of techniques, each of which has its own special peculiarities. The information that is gathered, while it is all consistent with reality, requires effort to bring it together to contribute to a single Mental Model in the geoscientist's mind. This 'bringing it together' effort is what Data Integration is all about.

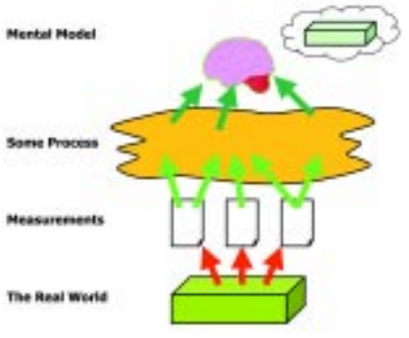


Figure 1 Understanding the real world.

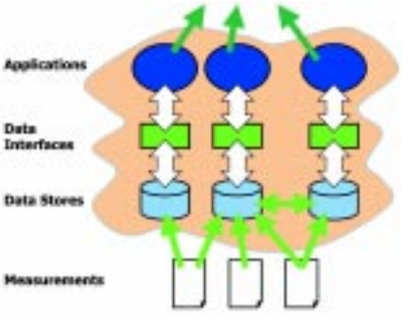


Figure 2 Some process.

The typical types of steps that occur in the Some Process stage are shown in Fig. 2. Measurements are loaded into Data Stores where the relations between them are encoded into data structures. These pieces of information are then presented to a series of applications via some well defined interfaces. The users explore, manipulate and interpret the information using these applications. There is a key role in this process for well-designed applications.

The point in this process where information is harmonised has a major influence on the end result. The two extremes would be Early Integration and Late Integration.

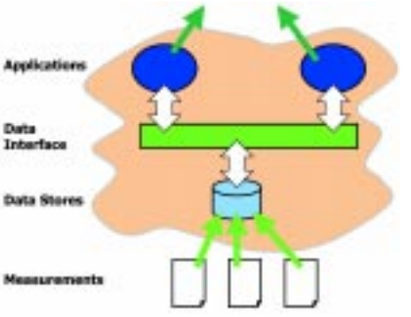


Figure 3 Early integration.

In Early Integration, information is tied together as it is gathered. An extreme example is shown in Fig. 3. In this case all the data is loaded into a single data store. This ensures that all the information presented to the user is consistent. Unfortunately, this technique disallows any information that does not 'fit'. Some years ago exactly this approach was universally advocated. Despite much effort to define a structure that is able to hold all the information no generally used 'ultimate' data store is likely to appear. The diversity of data is too great.

An Early Integration bias results in a consistent but inflexible set of information. Such characteristics are perfect for highly constrained problem domains, not so good for the flexible world that E&P data lives in.

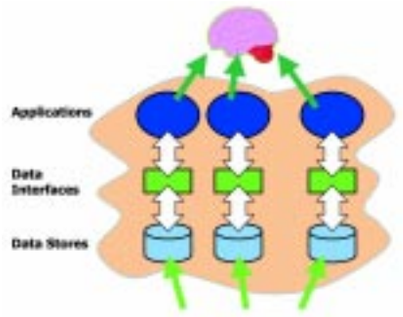


Figure 4 Late integration.

In Late Integration information is tied together closer to the user. An extreme example is shown in Fig. 4, where the user's mind is the first place where reconciliation of the information takes place. This approach is completely flexible, the data is not constrained in any way. However, it does not exploit the computer's ability to sift through mountains of facts, picking out the interesting inconsistencies.

Neither of these two extremes, of early and late integration, are appropriate in today's world. Usable work flows require a mixture of strategies to achieve their goals.

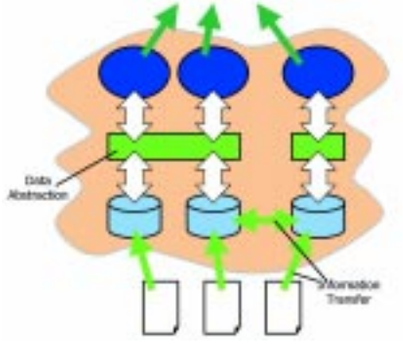


Figure 5 Two tactics to integrate data.

There are two tactics that are in current use to integrate data. These are Information Transfer and Data Abstraction. Information Transfer copies data from one place to another, usually cleaning up the data, checking for quality, and consolidating with existing data in the target. Data Abstraction creates a consistent interface that provides access to information, possibly held in a variety of locations.

Information transfer

Information Transfer is the most widely used approach to data integration. It allows complex, time consuming processes to be carried out such as data clean up, auditing and reconciliation. In today's data management world it is the most important way that data integration is done.

Figure 6 A high level view of Information Transfer

At one level all Information Transfer mechanisms perform the same function. As shown in Fig. 6 they extract a copy of the data from some source and insert it into some target.

There are many working data transfers that are built using exactly this philosophy, with no attempt to break the transfer down into reusable steps. These 'hand crafted' transfers are matched to exactly fit the needs of their customers. They are efficient and customisable, but expensive to create and maintain.

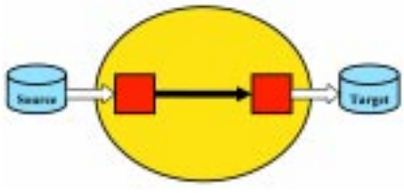


Figure 7 Building a transfer from half-links.

The monolithic approach has been shunned by professional link builders for many years. The half-link concept, illustrated in Fig. 7, has become familiar since it was pioneered by Geoshare. Under this design the transfer is carried out in two steps, one of which reads from the source while the other writes to the target.

Links are easier to build if they are split into reusable steps. Steps can be independently created by the data store vendor, then they can be combined to create complete links when they are required.

Figure 8 There are any number of ways to combine steps.

In practice it was soon found that allowing users to combine steps in flexible ways simplified the construction of links. In fact, all serious link building frameworks are able to join steps in a variety of ways. Geoshare, for example, has been used in this mode for a number of years, despite its reputation for maintaining the purity of the half-link approach.

So modern transfers are constructed by combining a series of reusable steps, each one of which performs part of the overall task. The most distinctive aspects of different links are not to be found in the way that steps are arranged, but rather in the ways that steps are connected.

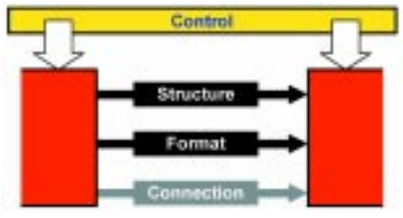


Figure 9 The way steps are connected.

There are two main aspects of this interaction: how the steps are controlled and how the data flows between steps. The *control* of steps, how they are invoked, how they are connected, how they interact with users and so on, is often hand crafted for each link. There are also tools that can automate this task.

The way that data flows between steps has been the subject of much discussion in recent years. It is useful to consider this connection at three different levels:

The *connection* level tells the steps how to establish a reliable stream of bytes. A typical *connection* defines the mechanism for reading and writing at the byte level. It also describes how different locations are identified.

The *format* level defines how entities and attributes are represented. A typical *format* level will provide the ability to send or receive objects and the mechanisms for asserting and retrieving attributes from them. Once a *format* level has been selected the user is not concerned with how the entities are encoded in bytes, this is defined by the *format* level.

The *structure* level determines which entities and attributes are allowed within a particular data flow. In order for the target step to correctly process the data it is important that the source supplies the objects that are required and does not send unexpected data.

At the *connection* level modern mechanisms use a combination of files and TCP/IP sockets. These have become ubiquitous and are not further considered in this paper.

So, for the purpose of reviewing Information Transfer technologies the differences can be thought of as occurring in the data flow at the *format* and *structure* levels and in the *control* of steps. The Technology Categories section provides more information about these interrelationships between steps.

Data abstraction

It could be argued that Data Abstraction is exactly the opposite of Data Integration, the whole ethos of this approach is to avoid having to gather the data together at all.

For some time Development Interfaces have been available to access the information maintained in data stores. These have been very successful at hiding the complex structures that are required to store today's data. In addition they isolate the application developer from the differences between versions and information located on different media. The availability of high quality Development Interfaces to third parties has been one of the main factors responsible for the success of data stores such as OpenWorks and GeoFrame.

The success of interfaces that isolate applications from the peculiarities of data storage has encouraged the creation of Common Interfaces that are completely independent of the data stores being addressed. These access the data via the Development Interfaces as shown in Fig. 11. The most well known example of this approach is the OpenSpirit initiative.

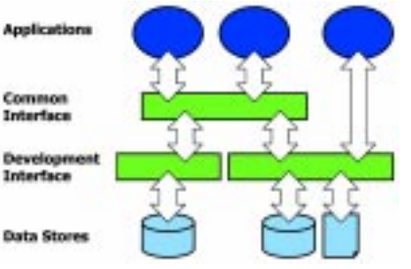


Figure 11 A common interface.

One might imagine that a Common Interface has no advantages over creating a usable comprehensive E&P data store, a goal that has proved elusive despite much expenditure of effort in recent years. This is, however, not the case, by ensuring that *all* the data is held in mature data stores, such as OpenWorks and GeoFrame, the existing data management tools and techniques can be applied. The fact that Common Interfaces do not store data locally ensures that the significant issues of data loading, consistency checking and synchronisation can be left to the source data stores, where tools and techniques are well tested and widely understood.

Data integration technologies

Based on the preceding background for classifying data integration technologies this section reviews the technologies currently in use in E&P. The list here attempts to avoid technologies that are only available from a single source. Either software is available from a selection of vendors, or, in the case of frameworks, a toolkit is easily available to any vendor that wishes to use it.

Each of the technologies has been assigned an 'area of influence' from the two pictures shown in Fig. 12. This provides a guide as to the role that the technology plays. Where possible the notes have been based on publicly available descriptions from the standard's maintainers.

Technology categories

The technologies listed above fall into a small number of categories, each of which has common issues and properties: fixed representations, format standards, structure definitions, controls, and data interfaces.

Fixed representations

There are a number of technologies that define a complete format and structure: LAS, LIS, RESCUE, SEG-Y and UKOOA. These tend to be older file formats that are focused on one particular category of data.

All these representations define the complete structure of the data, from what is encoded to how all the elements are depicted. Mostly these standards originated as file formats. The better documented standards amongst them however, are also used extensively in

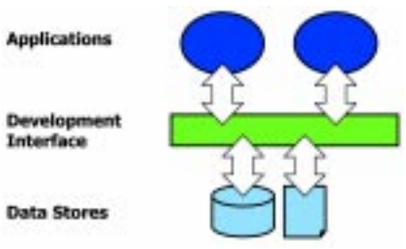
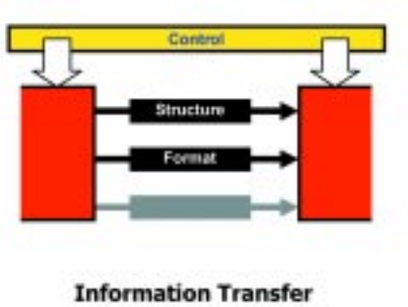


Figure 10 Development interfaces



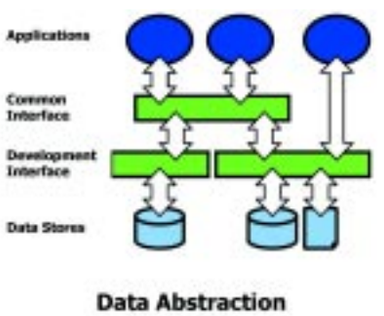
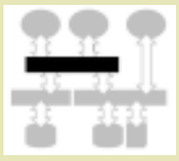
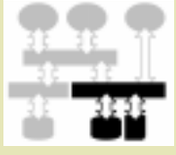
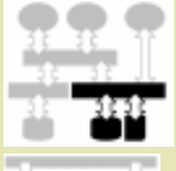
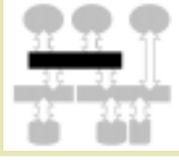
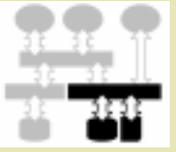
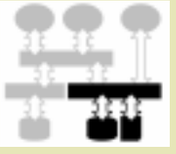
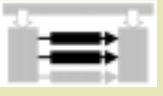


Figure 12 Two approaches to integration.

Name	More Details	Category	Where	Notes
BizTech 4energy/ Com4 Energy	http:// biztech4energy. org	Common interface		Was 'COM4Energy'. Based on COM/DCOM from MicroSoft. An open, multi-platform software interface standard focusing initially on exploration and production data.
DAEX	http://www. oilfield-systems.com	Control, structure & format		A complete system for constructing information transfers. Can construct transfers based on XML format or Geoshare structure.
DEX	http://www.lgc.com/ solutions/DEX/	Structure		Landmark sponsored open data exchange technology for disparate drilling applications that do not share a common data model. Based on XML format.
DLIS	http://www.api.org	Structure		A standard structure for well log information maintained by the American Petroleum Institute (API). Based on RP66 format.
Epicentre	http://www.posc.org	Development interface		The most comprehensive data model in E&P. More commonly used as the basis for structure standards than as a pure data store.
GeoBASIC	http://web2.airmail.net /b0001476/products/ geobasic.htm	Control		A language for building data transfer steps. Can use Geoshare or DLIS structures Based on RP66 format
GFDK	http://www. geoquest.com	Development interface		The GeoFrame Development Kit is the definitive mechanism for accessing information held in GeoFrame.
Geoshare	http://www. geoshare.org	Structure		A standard for the exchange of data used in the oil and gas industry. Based on RP66 format.
LAS	http://www.cwls.org/ las_info.htm	Structure & format		An ASCII based standard for well logs
LAS 3.0	http://www.cwls.org/ las_info.htm	Structure & format		An ASCII based standard for well information
LIS		Structure & format		A standard format for well logs on tape, well understood but widely varying. Created by Schlumberger.
OpenSpirit	http://www. prismtechnologies. com/products/ openspirit	Common interface		A CORBA based application integration framework for the upstream energy sector leveraging component-based middleware.

Name	More Details	Category	Where	Notes
OpenWorks DevKit	http://www.lgc.com/solutions/OpenWorks_DevKit/OWDevKit.asp	Development interface		The definitive mechanism for accessing data held in OpenWorks. Available as a free software download.
Production ML	http://www.posc.org/ebiz/ProductionML/	Structure		A proposal for a standard for web-based exchange of production data. Based on XML format.
PPDM	http://www.ppdm.org	Development interface		A vendor-independent standard petroleum data model based on SQL. Also used as the structure for data interchanges.
RESCUE	http://www.posc.org/rescue/	Structure & format		Format for exchanging reservoir characterisation information in binary files.
RP66	http://www.posc.org/technical/data_ exchange/RP66/V2	Format		A format designed as an efficient encoding for E&P data. Originally defined by the API now maintained by POSC. RP66 requires a structure to make sense.
SEG RODE	http://www.rodecon.demon.co.uk/rode_rv2/RODE-v2.html	Structure		The RODE schema is a collection of object types specifically designed for geophysical formats. Based on RP66 format.
SEG-P	http://www.seg.org	Structure & format		A set of formats for describing Seismic Navigation information, closely related to UKOOA.
SEG-Y	http://www.seg.org	Structure & format		The most widely used of the SEG (Society for Exploration Geophysics) standards. A format for encoding seismic data.
SPS		Structure & format		A set of formats for describing Seismic Navigation information
STEP	http://www.ukcic.org/step/step.htm	Format		Standard for the Exchange of Product model data (also known as ISO 10303, PDES and IGES)
UKOOA	http://www.ukooa.co.uk/ukooa	Structure & format		A set of formats for describing Seismic Navigation information
WellLog ML	http://www.posc.org/ebiz/WellLogML/	Structure		A standard for web-based exchange of well log data. Based on XML format.
XML	http://www.xml.org	Format		A language for encoding documents containing structured information. Defines how entities are described, but cannot be used without a structure definition (called a 'schema' in XML)

other ways, for example to move data between processes.

Recent standards efforts have tended to avoid specifying complete formats, preferring to define structures that sit on top of RP66 or XML.

Format standards

There are two standards that purely address format issues: RP66 and XML. Both of these standards describe how elements are encoded but do not specify which elements are valid or how they are related. This means that neither one provides a complete representation of any piece of data.

The motivation behind the two standards is different. RP66 was designed to be efficient while XML was designed to be easy for humans to understand. In today's environment the need for efficiency has been reduced by the ready availability of large disks and fast processors. Given the wide support enjoyed by XML, both inside and outside E&P, it is hardly surprising that recent structure definition effort has used XML as its underlying format.

At a high level the similarity between the format definitions is much greater than the differences. This means that the criteria for selection which one to use will be based on other factors such as the availability of tools, the ease of use and the quality of documentation. At the moment XML has a clear lead in these areas.

XML vs. LAS

In the initial paragraph one of the questions was 'Does XML mean that SEG-Y, LIS and LAS are redundant?' Clearly the answer to this question is no. XML provides a standard way to describe entities and attributes, it does not attempt to provide a complete representation, such as that encoded in LAS since the legal entities are not specified.

XML is an important part of the puzzle but it is only a part of any complete solution. Effort is still required to define which elements are allowed.

Structure definitions

The definition of abstract formats such as XML and RP66 has allowed modellers to create standards without being concerned about the details of representation. The practice of building on an independent format standard was started with the RP66 based definitions: DLIS, Geoshare and SEG RODE. Recent definitions, such as DEX, ProductionML and WellLogML, have been based on the more recently adopted XML standard.

One of the advantages of separating the structure from the representation is that structures defined for one standard format can be adapted to another. An example of this process was presented at the 3rd Petroleum Data Integration and Management Conference in 1999, when Jim Theriot of POSC illustrated how Geoshare structures could be represented in CORBA.

Control

It is noticeable how few Information Transfer technologies reach up to the control level. The only ones listed are GeoBASIC from ICS and DAEX from Oilfield Systems. These two have quite different philosophies.

GeoBASIC provides a simple language for constructing steps in the overall data transfer process. GeoBASIC has been used to construct steps using Geoshare and DLIS structures. It can use any structures defined on the RP66 format.

DAEX provides a comprehensive environment for constructing data transfers. This includes building the steps, defining the ways that steps are combined and overseeing the running exchanges.

It could be argued that Geoshare imposes some aspects of the control level, since the 'half-link' concept is so deeply ingrained. This element of the Geoshare standard has been skipped over, mostly because Geoshare is actually used in a wider variety of ways.

DAEX vs. Geoshare

In the first paragraph one of the questions was 'Does DAEX replace the need for Geoshare?' The answer is no. DAEX is a complete system for constructing links. The data transfer format in DAEX can be its own, XML or RP66, the structure can be Geoshare or Epicentre. The main advantage that DAEX provides is at the control level. DAEX automates the way that the steps, or components as DAEX calls them, are controlled. This is valuable regardless of the structure and format selected.

Data interfaces

The technologies that were classified as interfaces were: BizTech4energy, Epicentre, GFDK, OpenSpirit, OpenWorks Development Kit and PPDM.

An examination of the interfaces illustrates that the division between Development Interfaces and Common Interfaces is actually artificial. PPDM's data interface may be closely tied to the data structures and OpenSpirit's is clearly independent of them, however GeoFrame's and Epicentre's are somewhere in between. The OpenWorks interface describes 'business objects' and the latest version of OpenSpirit has its own private local data store. There really is no sharp dividing line between Development Interfaces and Common Interfaces.

There are, of course, major differences between the implementation details of these data interfaces. However, from the viewpoint of the application developer, these implementation details are only important where they restrict the way that software is developed. More important differences are to be found in the maturity of the documentation, the flexibility of the data access routines and the complexity of installation.

Data interfaces and structure definitions

There is a close relationship between data interfaces and the structure definitions. Both of these define entities and attributes, and, the occasions when they are appropriate. There has been some effort to bring these data models together, for example creating structure definitions based on data interfaces.

It is to be hoped that future data interfaces will continue to adopt some of the constructs from the structure definitions and vice versa. For example using the Geoshare structures to define elements of the OpenSpirit interface and using Epicentre subsets to define XML schemas.

There is one interesting difference between data interfaces and the type of structure definitions that are allowed by interfaces such as OpenSpirit. OpenSpirit business objects are not just passive data objects with attributes but are dynamic objects that have methods. This makes it possible to realise interoperability through a common set of methods without being forced to agree on a common, encapsulated, data representation. This 'late time' binding to objects can give clients run-time interoperability with diverse object implementations that encapsulate widely varying and changing implementations. This approach has not yet been fully explored and may turn out to be a critical future offering.

Data abstraction and data transfers

In the first paragraph the question was posed 'Will OpenSpirit mean that information transfers are no longer needed?' This is an important question that directly impacts on Geoshare.

It is anticipated that Common Interfaces will always be more general than the specialised development interfaces. This is because, in practice, common interfaces will tend to only support constructs that are similar in *all* the underlying data stores. There is little benefit to be gained otherwise. For this reason applications that address niche domains will continue to access certain categories of data directly through the development interface. This will, of course, mean that such applications will depend on having the appropriate data available in the correct store.

In the current environment the data repositories in use are, to a large extent, determined by the applications that are available. For example, an oil company will store stratigraphic information in a certain data store because the available software matches the requirements of their geoscientists. When a majority of critical applications are capable of running on any data repository this restriction will have been relaxed. However, oil companies will still have good business reasons for constraining where data is stored. The data manager will still want to control where stratigraphic information is to be found, to make it easier to find, easier to maintain and easier to manipulate.

When Common Interfaces are generally available to access all the relevant information, data will still need to be reformatted and duplicated in order to:

- Allow data management to impose business constraints, such as having a single trusted version of key data elements
- Make off-line data, such as that found on tapes, available to the applications
- Enable the import of partner's data from alternate systems
- Perform complex quality checks on the data as it is loaded into the company's systems
- Resolve ambiguities in identifiers when relating entities in different stores
- Transform between the different views of the world held in different disciplines
- Audit the processes that are applied to information

Common Interfaces with their emphasis on 'late' integration do not remove the requirement to construct reliable, maintainable and efficient information transfers.

Conclusion

This paper has presented a simplified overview of an important and complex area. One aim has been to disseminate information and provide a common basis for future discussion. In addition, the Geoshare User Group wished to review the impact of current technologies on the future of Geoshare.

The transfer of data, either as part of populating data repositories or when exchanging data between repositories, will remain a crucial task for the foreseeable future. Geoshare is one of the most well tested and widely used approaches for constructing such transfers. It can be predicted that it will continue to be an important technology.

The future

What will data integration look like in five years' time? We can say that a typical E&P company will continue to store its data in a number of locations. Each company will continue to define a short list of preferred data stores and a slightly longer list of actually used data stores. No two companies will maintain their data in precisely the same set of data stores.

A significant number of applications will be capable of running on any one of a selection of stores. Within each company these applications will extract each type of data from just a single location since business and technical drivers will prevent the chaos that would result from a lack of clarity about where information is located. There will also be a number of niche applications that are tied to particular data stores, taking advantage of capabilities that will continue to make those data stores indispensable.

There will still be a requirement to transfer information between stores. A significant quantity of this interchange work will still be carried out by hand crafted specialised transfers, often written by people who have long since departed the organisation. A rising proportion of transfers will be carried out within frameworks that define the format, the structure and the control mechanisms.