A Knowledge-Driven Approach to PDM Interoperability Projects

S.Buswell, T.Holtham, Stilo UK, United Kingdom

1. Introduction

Central to adding value to the enterprise from the activities inherent in a PDM migration project is 'Corporate Memory Preservation'. This could be characterised as “Think about what you are going do; do it; remember what you have done so you can re-use it”. The added value is in the re-use (both by humans and computers) of the formal knowledge gained by undertaking such a project. In fact, the knowledge model itself can be used to drive the interoperability and migration processes.

XML is used to represent the formalised knowledge, as it is easily processable by computers and can be read by humans in its native format or rendered in a more visually pleasing way in a web browser.

2. Corporate Memory Preservation

This consists of Knowledge Assets together with the management of those assets.

Knowledge Assets

Most people think that knowledge assets are explicit (i.e. the actual data for an object and supporting documentation), however, there is another type of asset – implicit or tacit knowledge. Implicit knowledge is the information required by domain experts over a number of years and can be lost to an organisation in an instant (i.e. when the expert leaves the organisation). These assets are used to create informal and formal models of the knowledge. The diagram below shows the relationship between the models and corporate memory.

Migrating from one system to another involves the identification and transformation of corporate knowledge assets. This is a very time-consuming and therefore expensive and as such, is an opportunity to look for added value from this stage of the project.

Explicit and Implicit Knowledge

Traditional migration strategies have relied on a relative simple mapping of the explicit meta-data from one system to another. However, data is created and modified through processes, often embedded deep within the PDM system. Equally, the new system will have its own set of processes. SophX allows the definition and management of very complex transformations. To define the transformation of a business object completely, it is necessary to understand what logic was used to create it and how this can be reflected in the new system. Only when this implicit (tacit) knowledge is understood can the transformation be fully defined.

SophX uses knowledge engineering to capture this tacit knowledge. This knowledge is represented in a model depicting the semantics of the objects and the relationships between them. This knowledge model is used to derive the conversion rules for the data transformations and forms the foundation of a comprehensive knowledge base that can be used at a later date.

Knowledge Management

This is the "Phrase of the Decade" and can be split into two areas: hard and soft. Soft Knowledge Management consists of document management and the knowledge obtained from domain experts/consultants. Hard Knowledge Management consists of re-useable processes, methodologies and tools.

The benefits of hard Knowledge Management are that it reduces the cost of capturing knowledge and that there are well defined processes that focus on clarifying goals (the benefit to the organisation and the needs of the end user). It helps experts remember what they now and more importantly, facilitates agreement between those experts. Finally, it ensures that the knowledge is validated and reduces duplications and omissions.

MOKA – The Methodology Oriented to Knowledge based Engineering Applications

MOKA was a project part funded by the European Commission within the AIT (Advanced Information Technology in Design and Manufacturing) initiative that is part of the ESPRIT projects covering Information Technology. It provides a framework for representing and storing knowledge. The MOKA consortium consists of Airbus, Aerospatiale, BAE Systems, Daimler-Chrysler, PSA Peugeot Citröen, Knowledge Technologies International and others.

MOKA is a methodology for supporting design engineers in the development of KBE applications. It is particularly aimed at capturing and applying knowledge within the aeronautical and automotive industries for the design of complex mechanical products. MOKA aims to provide a methodology that

- Reduces the lead times and associated costs of developing KBE applications by 20-25%
- Provides a consistent way of developing and maintaining KBE applications
- Forms the basis of an international standard
- Makes use of a software tool to support the use of the methodology

The number of KBE systems used in the aeronautical and automotive industries has increased in recent years. Experience has shown that employing a systematic methodology that covers the development and maintenance of such systems can reduce long-term risk.

The steps involved in this methodology are:

1. Capture the raw knowledge from the domain experts
2. Create an informal model of this knowledge using knowledge acquisition techniques and software
3. Create a formal model from the informal one
4. Populate the Knowledge Base using the information contained within this formal model
5. Publish the knowledge

MOKA Analysis and Modelling

MOKA identifies two models to be used in the KBE development lifecycle:

- **Informal Model**: A structured, natural language representation of engineering knowledge using pre-defined forms, including
    - Entities
    - Constraints
    - Activities
    - Rules
    - Illustrations

- **Formal Model**: A graphical, object-oriented representation of engineering knowledge at one level of abstraction above application code.

Within each of these models, various knowledge representations are used to capture, analyse and structure the knowledge required for the generation of downstream KBE applications.

Modelling and Managing the Knowledge Base

Engineering knowledge is complex and inter-related. Using SophX-Pack, complex concepts and tasks are broken down into simple objects and knowledge is structured using various knowledge models such as trees, diagrams, grids and hypertext links. Knowledge can be quickly extracted from standard text-based information by marking-up knowledge objects and classifying them through the simulated use of multi-coloured highlighter pens.

Complex inter-relationships between objects, including product models and process models, can be represented using SophX-Pack's easy to use drag and drop capabilities.

In addition to providing a fully integrated suite of knowledge management tools, SophX-Pack provides knowledgebase configuration management, integrity checking, version control and concurrent working on a client server standard operating environment architecture. Engineers can also work offline from the main system as required.

Use of Semantic Web Technologies in SophXPack

The knowledge model captured in SophXPack is captured internally in a form known as a description logic. This is a formal language with known properties which allows us to perform such tasks as automated consistency checking and merging and conflict detection / resolution between parts of knowledge bases captured in different acquisition and modelling sessions.

A subset of this internal language has semantics that align with OWL and RDF, the W3C Languages for Ontologies and Resource Description on the Web. We have developed OWL/RDF import/export capabilities that will allow us to exchange knowledge bases with other Semantic Web aware systems. The semantic extensions beyond the scope of OWL relate primarily to support for logical predicates and quantifiers and mathematical expressions. We currently use MathML to encode these, however the logical structures will probably be covered in the future by some Semantic Web rules language, perhaps an evolution of RuleML.

Publishing Information

SophX-Pack stores information in XML format, enabling technical information to be easily published to the web, incorporated into printed training manuals, or published to other types of media including CD-ROM and mobile devices. It also enables engineering knowledge to be rapidly integrated with content management systems, CAD, CAE and PDM applications, and made available to groups of engineers across the enterprise and throughout the supply chain.

Stilo's content engineering technology, OmniMark, complements SophX-Pack and is used extensively by many of the world's largest organisations to overcome the complex tasks of content transformation and integration in large scale technical publishing and data migration projects.

Completing the Knowledge Cycle

Having successfully acquired, modelled and published engineering knowledge, it needs to be continually updated and refined, thereby establishing a dynamic and evolving knowledge base within the organisation.

SophX-Pack provides the capability for engineers to annotate knowledge objects online, and immediately share updated information with other team members.

Using the electronic equivalent of 'post-it' stickers, engineers can attach notes to almost any piece of technical information. This not only enables a vast amount of knowledge to be accumulated quickly, but also provides essential details to better understand the use and traceability of knowledge objects within the engineering process.

Implementation

Stilo, in conjunction with its expert partner Epistemics, provides customers with a structured approach to implementing KBE methodologies, fully supported by SophX-Pack. Key steps for a successful project implementation include:

- Scoping
    - Scoping the project for success with lead engineer
    - Understand the needs of the organisation
    - Identify resources and documentation available

    - Agree interview plans, time-scales and deliverables

- Knowledge Acquisition
    - Each interview's objectives clearly defined
    - State responsibilities and commitment of both customer and knowledge engineers

- Knowledge Structuring
    - Product information
    - Process definition
    - Document knowledge

- Knowledge Validation
    - Deliverables to be approved by authorities within the business

- Publication & Maintenance
    - Specifying multiple media outputs & information exchange with other systems

Benefits of SophX-Pack

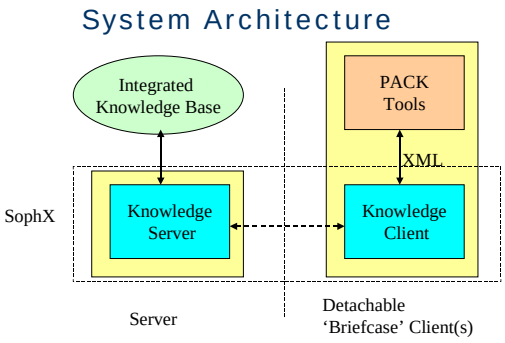
By implementing SophX-Pack, engineering companies can fully realise the benefits of implementing KBE, including:

- The creation of a knowledge base from information which previously did not exist in a traceable way, preserving a valuable corporate asset
- New and experienced engineers will have a reliable and available source of information, significantly reducing training time
- Building a web site which provides links to existing documents saves time in searching known documents and raises awareness of unknown or misused company procedures or documents at the relevant time
- Radically reduces design cycle time
- Allows re-use of best practices
- Improves quality of designs and minimises rework
- Reduces costs

Tools that utilise/implement MOKA is critical to the acceptance of such a methodology. One such tool is SophX-Pack. SophX-Pack consists of two tools, SophX and PCPack.

SophX provides corporate memory preservation, intelligent information navigation, and meaning driven data transformation. At the heart of SophX is the formal knowledge model that actually preserves corporate memory. This is known as the Knowledge Base and is actually a collection of formal models. SophX handles the management of all the knowledge contained within it by providing tools to manipulate the knowledge base allowing ease of integration of the separate models and conflict management and resolution.

PCPack is used for knowledge acquisition and modelling. It has a diagrammatic interface presenting different views of the knowledge held in the knowledge base.



3. Using Sophx-Pack for PDM migration

Background

The PDM (Product Data Management) system is often referred to as the intellectual backbone of the organisation. This is because it contains and controls critical information regarding the organisation's products and is at the heart of every engineering and manufacturing company. This information is typically in the form of data representing the engineering designs and product build standards created and modified over many years, together with associated processes surrounding the creation, approval and use of that data.

The successful migration of data and processes from one PDM system to another in a timely manner is on the critical path to the successful deployment and acceptance of the new system.

Typically, vendor migration projects are fraught with uncertainties, as the complexities are often underestimated or overlooked. As a result, a traditional technology approach is adopted (point-to-point), leading to escalating costs with expanding and unpredictable timescales. This introduces a danger of the project being forced onto an exponential growth curve for the required costs and resources, leading to the inevitable failure of the project. Furthermore, any unforeseen and increased expense incurred results in no residual value to the business.

Data migration is a process consisting of:

- A strategy utilising tried and tested components that can capture data and processes reliably;
- A solution that analyses source and target applications and supports an iterative method of mapping the complex transformations needed;
- A platform providing a fast, scalable and reliable way to capture and process all the data held in the source system (including any system related info such as a list of users and their rights);
- An approach that forms the foundation for an on-going methodology for all future migration requirements and adds lasting value to the organisation.

Complexity

The complexity of product data leaves many businesses with little confidence that their data migration will be a total success. This means that a big-bang approach is usually not considered forcing the adoption of a project-by-project approach. This results in the existing PDM system being active for a considerable amount of time. Even when the migration has been successfully completed, the system is often left on 'just in case' some of the data was missed or mishandled.

SophX addresses these issues by applying proven technology in a number of discrete and manageable steps. The first step is to make sure that all data held in the source system is captured into a neutral, non-propriety format – XML. This ensures longevity of access and the ability to use a range of commercial tools.

With all the data stored in a neutral format in a XML repository, the subsequent steps can be undertaken in complete confidence that all the data is ready for migration. Should any of this data not be migrated to the new system, it will always be available for future use.

XML Repository

Stilo UK, together with SOLASS, have developed a complete enterprise migration framework solution for the transfer of existing PDM data to any replacement system. This approach focuses on, and addresses the business problems associated with migrating critical business systems, and includes securing the original source PDM system data external to the application, providing ongoing access to it, throughout and beyond the migration activity. This repository will therefore contain all source PDM system data in a format that preserves the structure of the data (XML). At any time in the future the data owners will be able to use the repository to supply any incomplete or missing information. The repository therefore provides the longevity to the source data, which the business requires, storing it in a neutral format and providing continued ease of access to it via standard industry browsers.

The repository also provides the platform and source data for the ongoing migration to the target application, facilitating any data cleansing requirements prior to loading the data. This then provides a highly efficient mechanism to enable the population of the target application, with the correct data and within a timeframe, which the business allows.

Key Benefits

- **Uses of Native Application Tools reduce cost and risk.**

The use of native application tools to export and import data improves data quality, reduces risk and reduces costs.

- **Data Cleansing**

Data Cleansing is supported by combining the source PDM system and target Data Schemas with an industry-standard XML schema checker that can be used to check the validity and integrity of the XML objects, prior to loading into the target system.

- **Big-Bang and Incremental Data Loads**

The data is extracted and stored in the repository as business objects. Therefore, either all or a subset of the objects can be processed at any point in time. The data load framework may subsequently track the Business Objects that have been loaded into Target, hence supporting both incremental and big bang data migration.

- **Source PDM system data archived in standards-based neutral format.**

Existing users will be able to access their source PDM system data even when the source PDM system has been shut down. Therefore, they will be able to accept that some of the source PDM system data has not been migrated to the target PDM system, hence speeding up the analysis and test phases.

- **Reduce time taken to perform Big-Bang Data Migrations**

The full migration process requires:
  - Object export from Source PDM system
  - XML manipulation
  - Object import into Target PDM
  - The use of the repository as part of the above process reduces the elapsed time to complete a big-bang data migration.

- **Avoid repetitive Source PDM system Exports**

Data Loads often require multiple passes of the data to address the rules embedded in the applications within the target system. The Repository removes the need to perform multiple repetitive object exports.

- **Additional Incremental Loads do not need Source PDM system**

Once the main data load has been completed additional data can be loaded. An incremental load can be completed without the need to access the Source PDM system application. This also allows for the Source PDM system to be retired, while access to the data is maintained, in readiness for the Target application to match the functional requirements of the business.

- **Secure Intellectual Capital**

  The organisation should be concerned about turning the Source PDM system off and losing the intellectual capital stored within it. The Repository combined with the application rules, ensure that the intellectual capital of the organisation is secured. All object data contained within source PDM system will be available as part of the XML version of the object. This means that any reports generated by the source PDM system could be replicated from information contained within the repository unless such reports rely on data external to the original source PDM system objects.

3 tier architecture

The basic architecture of the system comprises:

- The Repository;
- The Repository Server – the front-end to the Repository;
- The User Interface (a web browser).

The Repository

The Repository itself resides in an Oracle database.

Meta-data will be used to facilitate the extraction and delivery of a specified object from the XML objects within the repository and will contain any supporting information. Although this supporting information is available within the XML version of the source PDM system object, extraction of this information from the XML and storing as meta-data during the repository load phase will reduce later search times.

This information could be dynamically created when a specific object is requested, however this is a very time and processor intensive task since it requires searching through all objects within the repository and extracting a list of those that reference the current object.

The Repository Server

The Server encompasses the processing rules that are needed to serve the requested XML object. All linking will be handled within the server together with the code to find the latest revision of an object where this is needed.

The User Interface to the Repository

A web browser should be used as the repository client, the repository Server will deliver the XML object from the repository to the client together with its supporting XSL and JavaScript files.

The main client interface to the repository is through a web browser, however, when requesting an object to convert to the target PDM system, a non-browser client will be used.

The user will be able to determine which view will be used to display the record:

- **Raw XML** The XML representation of the data will be displayed. Programmers who are using the data to populate next generation systems will typically use this view.

- **Default View** The default or object view uses internal object attribute (Source PDM system) names to display the data.

- **Styled View** The styled view uses the screen names or user names to display the attributes. To improve usability the reference attributes will be displayed as a separate screen that is accessed using a hyperlink.

Parent to Child navigation

Within the Styled View the user will be able to traverse to any child record by using the hyperlinks that are displayed for each child record. The resolution of the relationship will take account of the "Latest" and "Specific" reference resolution rules.

Child to Parent Navigation (Where used)

Within the Styled View the user will be able to traverse to any parent record by using the hyperlinks that are displayed for each parent record.

4. Example of using the Repository to facilitate data cleansing

Data cleansing could be performed on the repository. It is suggested that should data cleansing be performed on the repository that a new "cleansed" repository be created. The reason behind this is that as soon as you actually change the data held within the repository, it is no longer an exact representation of the original data that was held in the source PDM system.

Data cleansing can take a number of forms. For example:

- Renaming of objects
- Modification of the unique identifiers of the objects to conform to another format
- Removal of objects that are not referenced by any other objects
- Insertion of default values into fields within the object
- Modification of the data held within each object to facilitate its conversion to the target format

Data cleansing is facilitated by the fact that the object is in XML. XML is highly structured and it is easy to obtain the contents of an element for further processing.

Data cleansing could also be performed on the source PDM system object dump before conversion to XML but again the argument above applies – the fact that as soon as you change the data coming out of Source PDM system, it is no longer a true representation of that data.

5. Conclusion

The migration methodology has a number of key benefits over and above a more traditional ‘point to point’ migration approach. These are summarised below. In addition to the benefits identified, the approach adopted utilises technologies that will provide ongoing benefit to the organisation after the migration activity has been completed, thus turning a necessary cost of migration into an investment for the future. In addition, the resources and expertise used for the development of the migration methodology, and the delivery of the project itself, is recognised as the leading domain expertise within the industry.

- A faster migration is achieved, meeting the timeframes required by the business, enabling a quicker deployment of the new PDM application.
- Cost savings are achieved by earlier decommissioning of the source system.
- The architecture of the SophX data migration framework allows for the migration of data and processes involving multiple source and target systems.
- All the source data is secured at a reasonable cost in a Standards compliant neutral format.
- Source processes, and hidden ‘tacit’ knowledge, are captured in a Knowledge Management system that makes the processes available for future use.
- All types of data may be migrated to the new system including system-generated dates.
- The Application and Conversion Rules support configurable and extensible data mappings.
- Both incremental and big bang data loads are supported.
- Incremental data loads allow for a piecemeal migration of the data, in line with the business drivers and requirements.