OILFIELD SYSTEMS
Data Integration Solutions

Practical E&P Data Mapping using XML

Oilfield Systems Limited

April 2001

OILFIELD SYSTEMS
Data Integration Solutions

This presentation is about...

- Oilfield Systems' experience…
  - *of building data exchange solutions over eight years*
- …of using XML…
  - *eXtensible Markup Language…a language for encoding documents containing structured information*
- ...for Mapping E&P Data
  - *issues faced when moving between alternative representations of E&P data*

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Content

- Design choices for XML
- Mapping issues
- Practical examples
- Conclusions

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Let's put XML in Context

- XML is a data "Format"
- To define the "Structure" we need to specify which entities and attributes are allowed
- Since "vanilla" XML defines only a "string" data type we need ways of specifying more complex data

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

This is what we advocate...

From our work in E&P data exchange using XML we advocate...

- adopting an XML standard for an extended set of data types *(e.g. time, coordinate)*
- adopting an XML standard for efficiently specifying bulk data *(i.e. data arrays)*
- encouraging specialists and organizations such as POSC to publish schemas to prevent duplication and drive towards standards

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Content

- Design choices for XML
- Mapping issues
- Practical examples
- Conclusions

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Design Choices for XML

- Data Types
- Metadata
- Bulk Data

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Aggregate Data Types

- Schema
  - Schemas allow easy definition of compound data types
- NDT
  - POSC Exchange Format “Named Data Types”
  - uses DTD format

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Example Aggregate Data Type

```
<datatype name="coordinate">
  <basetype>real</basetype>
  <min Inclusive="1000.0"/>
  <max Inclusive="9000.0"/>
</datatype>

<ElementType name="x" type="coordinate"/>
```

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Example NDT Aggregate Data Type

Location: A location is a reference to a previously defined coordinate system, an optional reference to a previously defined vertex, and a list of quantity values. The list order is determined by the order in which they are listed in the XML file. The dimensionality of the location is determined by the dimensionality of the coordinate system.

```
DTD: <!ELEMENT location (coordinateSystem-ref, vertex-ref?,
                coordinates)>
     <!ELEMENT coordinates (quantity+)>
     <!ELEMENT coordinateSystem-ref EMPTY>
     <!ATTLIST coordinateSystem-ref refTo IDREF #REQUIRED>
     <!ELEMENT vertex-ref EMPTY>
     <!ATTLIST vertex-ref refTo IDREF #REQUIRED>

XML Example:
  <data_value>
    <location>
      <coordinateSystem-ref refTo="NAD27">
      <coordinates>
        <quantity uom="dega">30.275993</quantity>
        <quantity uom="dega">-98.931124</quantity>
      </coordinates>
    </location>
  </data_value>
```

DAEX

Float Data Type

- To avoid rounding errors, floating point numbers need to be specified in a notation which maintains the IEEE format
- Oilfield Systems advocates using an “exact” attribute encoding the 4-byte value

```
<elevation exact=”403B8DF">55.70</elevation>
```

Data Types: Conclusions

- Use Schema rather than DTD
- Augment with a set of NDT's
- Also provide an exact IEEE float

OILFIELD SYSTEMS
Data Integration Solutions

Metadata

Some issues:

- How much metadata to include in the XML?
- How to map class and attribute names into XML tags and attributes?
- UML, Schemas, DTD's, or what?

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Example Metadata Issue

```
<name type="string">UTM31N</name>
```

```
<depth_array>
  <depth type="float">3100.0</depth>
  <depth type="float">3200.0</depth>
  <depth type="float">3300.0</depth>
  ...
</depth_array>
```

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Example Metadata Issue

```
<DAEX_DepthValues>
  <depth type="columndef">
    <type type="type">float</type>
    <name type="string">DEPT</name>
    <unit type="string">M</unit>
  </depth>
  <values type="rowdata">
    <row><depth>3100.0</depth></row>
    <row><depth>3200.0</depth></row>
    <row><depth>3300.0</depth></row>
    ...
  </values>
</DAEX_DepthValues>
```

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Metadata: Conclusions

- Start from a base platform that allows some generic processing
- Build your models on some agreed baseline schemas
- Use UML or something better for documenting your data models
- Generate and publish schemas

DAEX

Bulk Data

Some issues:

- Readability vs. processing power
- Multi-dimensional generic index vs. the simple array
- How to represent metadata
- Aiding schema checking

OILFIELD SYSTEMS
Data Integration Solutions

Bulk Data: Conclusions

- Enable Stylesheet Processing
- Keep it simple - 1D heterogeneous arrays organised into collections of columns including an index column
- Get the metadata into the schema EXCEPT for array column types.

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Content

- Design choices for XML
- Mapping issues
- Practical examples
- Conclusions

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

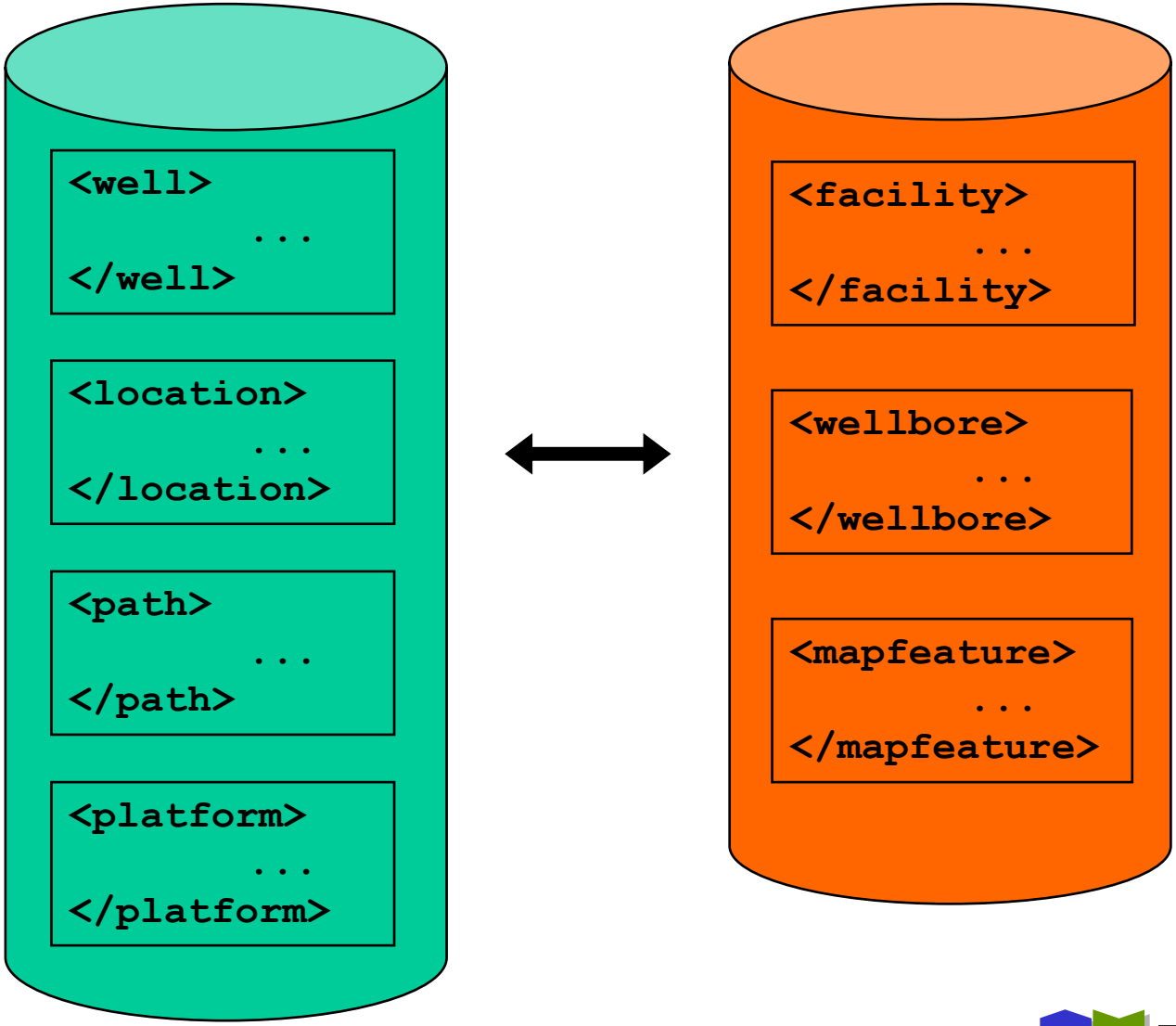
Mapping Issues

- High level object representations may be fundamentally different between source and destination
- Low level objects *(i.e. data types)* may also use alternative syntax
- Bulk data is often organized for access rather than processing
- Bulk data is mixed with metadata

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

High Level Object Representations



DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Mapping Issues: Conclusion

- XML is NOT a silver bullet
  - It helps organize the data but it needs "structure" to make it suitable for data exchange and to deliver interoperability
- We need to move towards industry standards for representing data types and business objects in XML

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

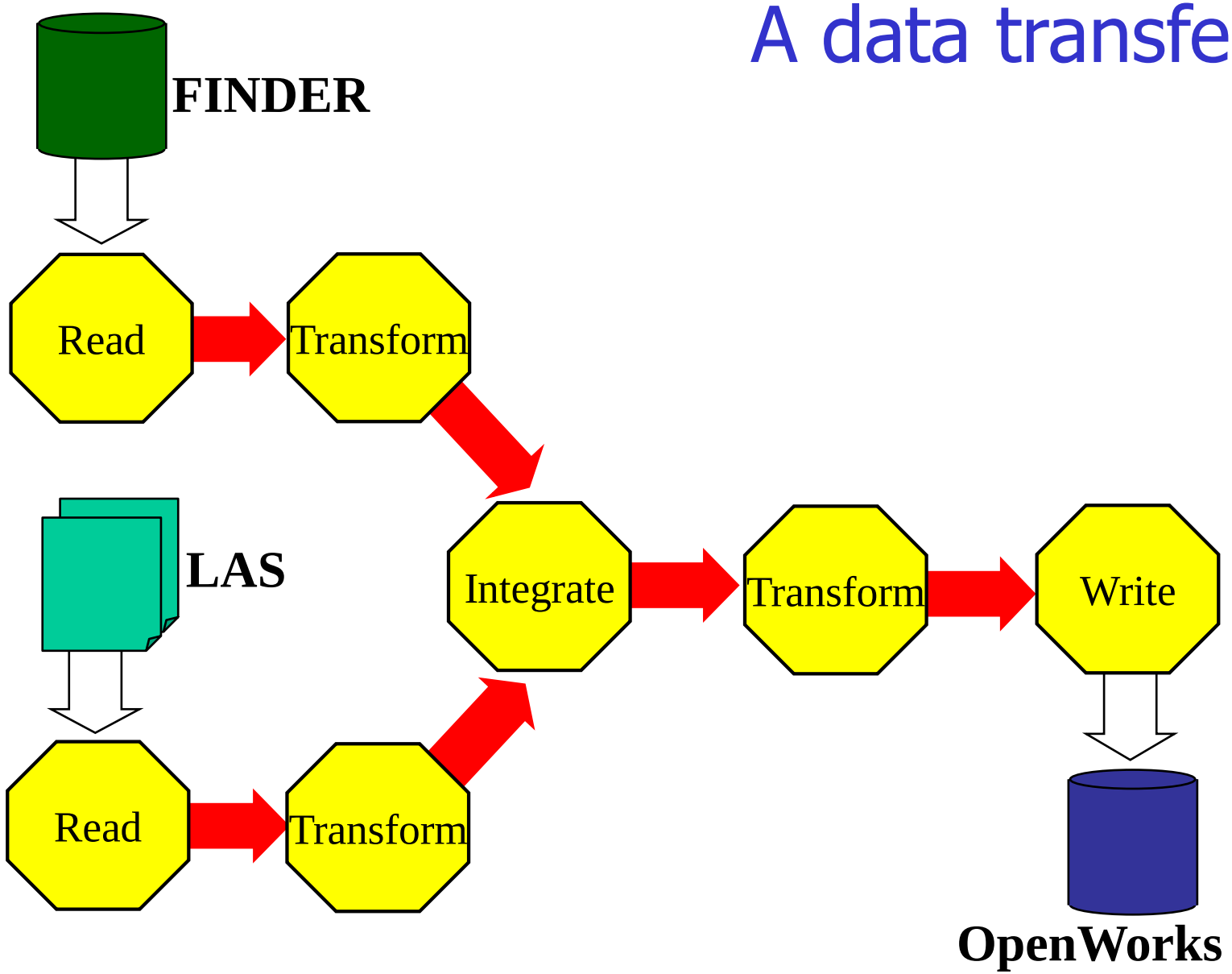
Content

- Design choices for XML
- Mapping issues
- Practical examples
- Conclusions

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

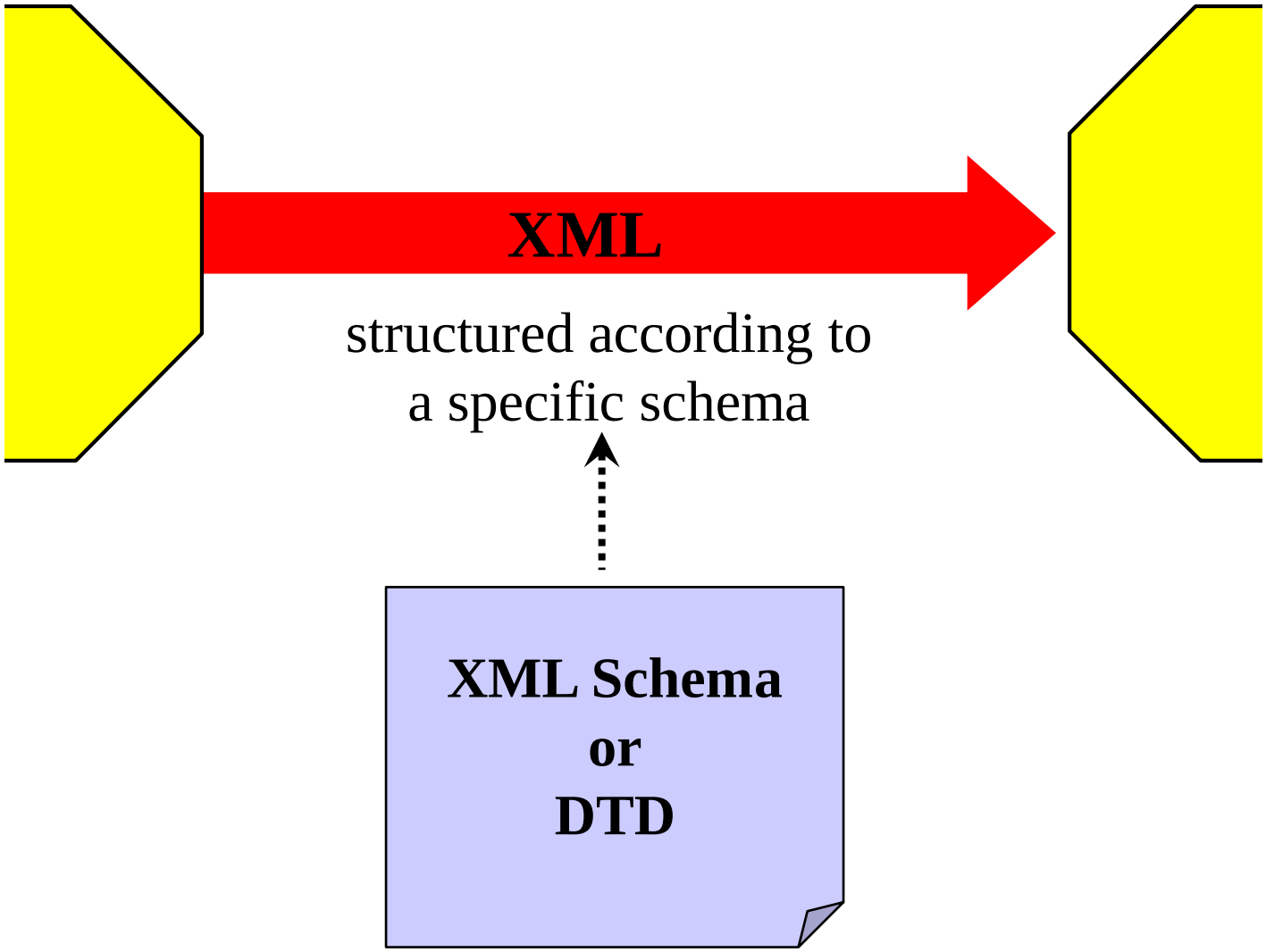
A data transfer



DAEX

OILFIELD SYSTEMS
Data Integration Solutions

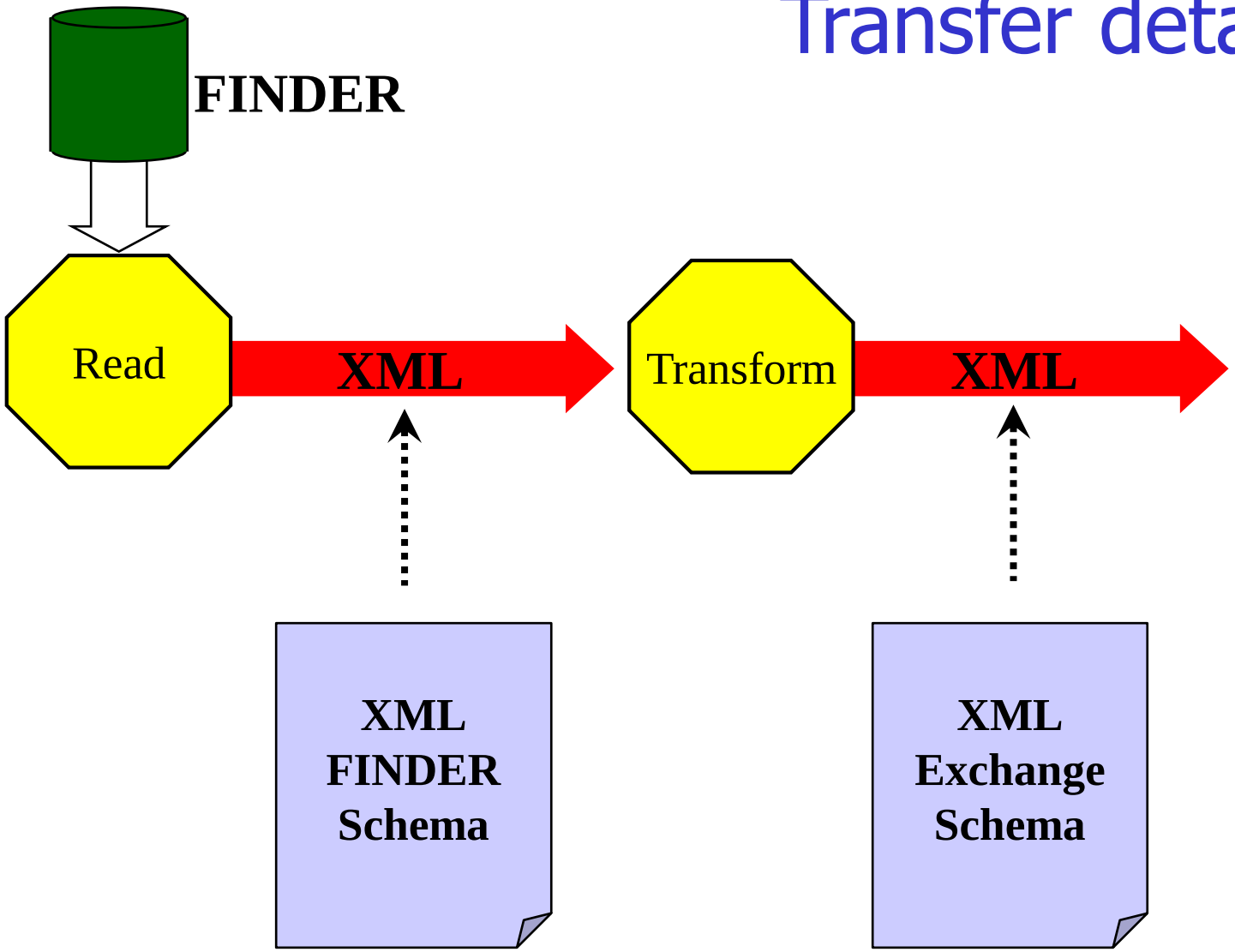
Transfer detail



DAEX

OILFIELD SYSTEMS
Data Integration Solutions

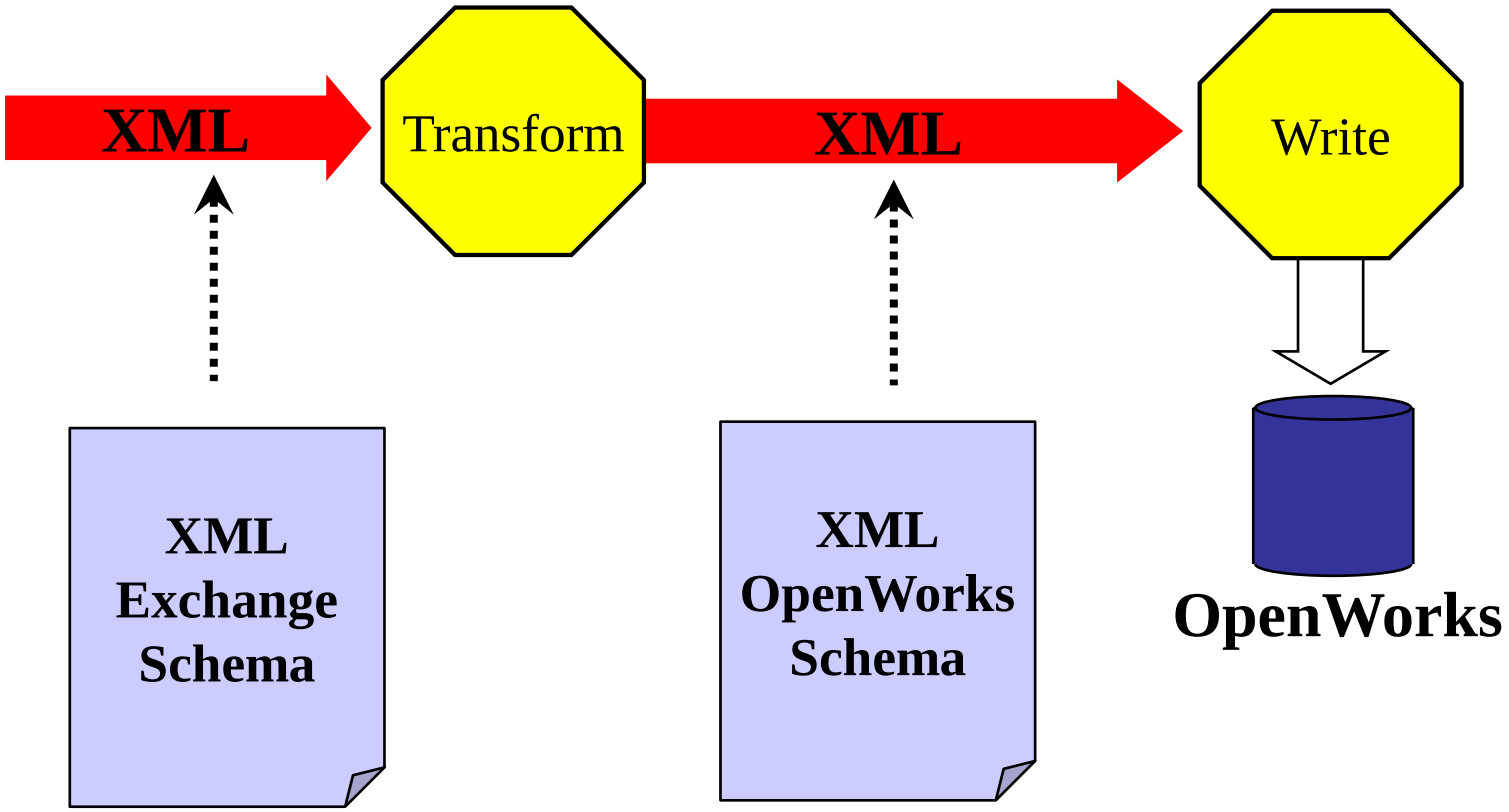
Transfer detail



DAEX

OILFIELD SYSTEMS
Data Integration Solutions

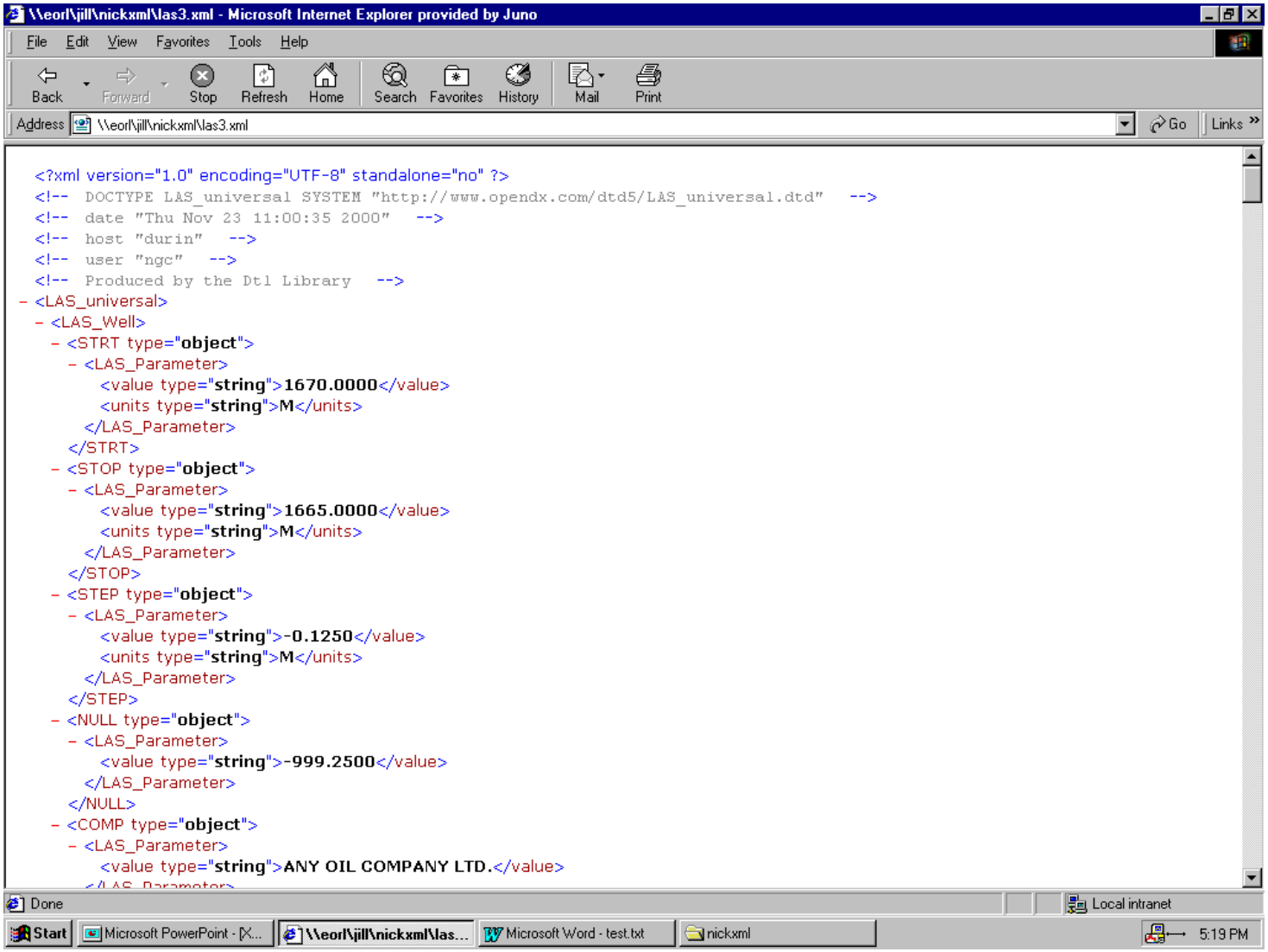
Transfer detail



DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Example LAS 3.0 XML



DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Content

- Design choices for XML
- Mapping issues
- Practical examples
- Conclusions

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

Conclusions

Within E&P we can gain the advantages of XML in our Data Mapping environments through…

- adopting an XML standard for data types
- adopting an XML standard for bulk data
- encouraging specialists to publish schemas

DAEX

OILFIELD SYSTEMS
Data Integration Solutions

What Next?

At Oilfield Systems we are:

- Building an API to work with E&P XML standards
- Contributing to E&P XML standards
- Building tools to generate Java, C++, Perl and C libraries for working with business level objects from data model definitions

DAEX