ESCWA SDMX Workshop

Session: SDMX and a Re-usable Component Architecture
Goals

• Language Independent Architecture
• High Cohesion
• Loose Coupling

• Program to Interfaces – Not Implementations
Use the Model

- **SDMX-ML** is for **EXCHANGE** between systems

- The **SDMX Information Model** is where the power is
Finding the Common Model
What is Data?

• Results of Measurements

Country: UK
Unemployed: 13million
Year: 2010

Country: Australia
State: Perth
Coastline (Km): 12500

Country: France
Unemployed: 5million
Year: 1982
Modelling Data

Country : UK
Basket : Narrow
Time Period : 2000-01
Value : 107.8

Time Period : 1999-10
Value : 106.34

Time Period : 2001-02
Value : 101.73

Type : Real
Value : 107.8

Type : Nominal
Value : 104.8
Simple Data Transformation Model

Compact
2.0 To 2.1
Transformation

Compact
2.0

“Compact”
2.1 To 2.0
Transformation

“Compact”
2.1
Simple Data Transformation Model

- Compact 1.0
- Compact 2.0
- Generic 1.0
- Generic 2.0
- Utility 1.0
- Utility 2.0
- Structure Specific 2.1
- Generic 2.1
- EDI
- CSV
- ???

© Metadata Technology
Decoupled Data Transformation Model

Data Reader Interface
- Read Key
- Read Observation

Data Writer Interface
- Write Key
- Write Observation

Compact
- Generic
- EDI

© Metadata Technology
Summary so far

• Good design protects software from change

• Finding a common model means everything can be reused
Break Down The Problem
Creating Sub-Systems
Make a Generic Solution

• Each aspect of a system can be broken down into a sub-system

• Each sub-system has a specific responsibility
Break down the Problem

- SDMX breaks down nicely into sub systems:
  - Data Transformation
  - Structure Validation
  - Generating a Structure Query
  - Data Reading
  - Data Validation
  - Structure Transformation
  - Generating a Data Query
  - Data Writing
  - Consuming a Structure Query
  - Querying a File
  - Consuming a Data Query
  - Calling a Web Service with a Structure Query
Interfaces can be reused

```java
BeanRetrievalManager

/**
 * Returns a single CodeList , this expects the reference object to contain
 * all the attributes required to uniquely identify the object. If version information
 * is missing then the latest version is assumed.
 *
 * @param ref
 * @return
 */
CodeListBean getCodeList(MaintainableRefBean ref);
```

| File Retrieval | In Memory Retrieval | Web Service Retrieval |