# DIGITARQ
# Creating and Managing a Digital Archive

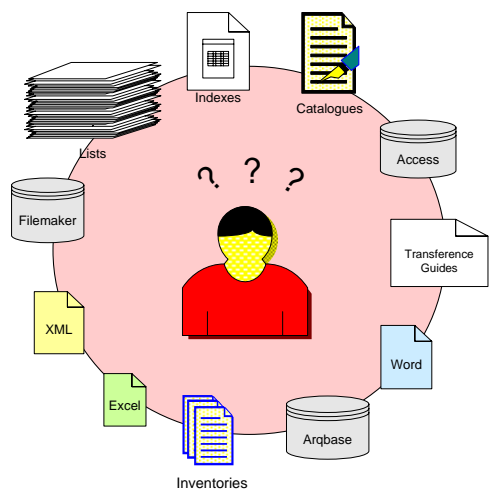**Miguel Ferreira**
mferreira@dsi.uminho.pt

**José Carlos Ramalho**
jcr@di.uminho.pt

---

## Porto District Archive [ADP]



Indexes

Catalogues

Access

Lists

Filemaker

Transference Guides

XML

Word

Excel

Arqbase

Inventories

**2**

## The proposal

- To build an unique repository according to international standards
- To have an unique interface to access and query the repository
- To have the system available through the Web

3

## Contents

- Introduction to the project
- Archive's Metadata
- XML throughout the project
- Storing the metadata
- Model evaluation
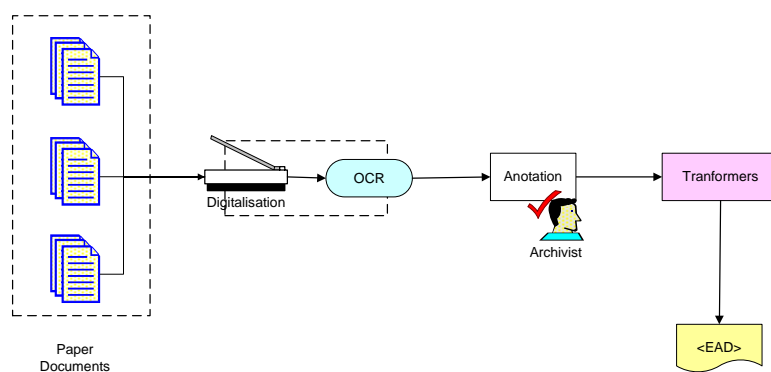- Conclusions and future work

4

## DigitArq Project [main goals]

- Information centralisation
- International Standards
  - □ *International Standard Archival Description* - ISAD(g)
  - □ *Encoded Archival Description* - EAD/XML
- Paper use reduction/elimination
- Publish the archive's catalogue on the Web
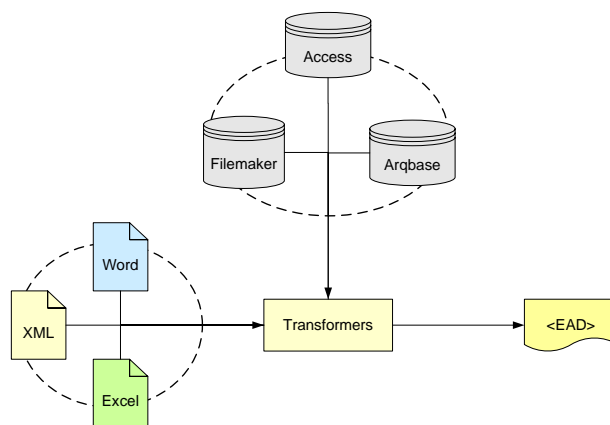
5

## Paper finding aids [digitalisation]
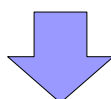


6

## Different databases [migration]



7

## Results of migration

- Hundreds of  EAD/XML files
- The need to manage all this information
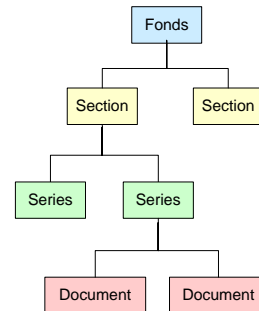- Publish adequately on the Web



How do we store this information?

8

## Archival metadata (EAD)

- Hierarchical structure
- A top-down, general to specific description
- Organised in description levels
- Unit ID, title, unit dates, physical description,scope and content, etc.

```
            ┌────────┐
            │ Fonds  │
            └────────┘
           ┌────┴─────┐
      ┌─────────┐ ┌─────────┐
      │ Section │ │ Section │
      └─────────┘ └─────────┘
       ┌───┴────┐
  ┌────────┐ ┌────────┐
  │ Series │ │ Series │
  └────────┘ └────────┘
           ┌────┴─────┐
      ┌──────────┐ ┌──────────┐
      │ Document │ │ Document │
      └──────────┘ └──────────┘
```
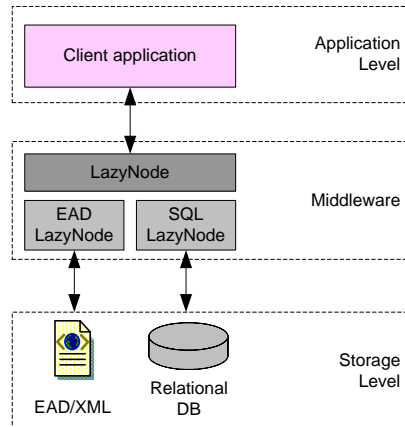
9

## XML Pros and Cons

- Pros
  - The hierarchical model is assured
  - Portability – easy to exchange information with other archives
- Cons
  - Too Low-level for user consumption
  - Difficult to store and maintain

10

## Metadata storage architecture

| | |
|---|---|
| Client application | Application Level |

| | |
|---|---|
| LazyNode | Middleware |
| EAD LazyNode / SQL LazyNode | |

| | |
|---|---|
| EAD/XML / Relational DB | Storage Level |

- Download()
- Upload()
- Children()
- CreateNode()
- AppendChild(child)
- RemoveChild(child)
- HasChildren()
- Clone()
- Parent()

11

## Model implementations

- EAD/XML Files
  - □ DOM
- Relational Databases
  - □ A column for each description property
  - □ Id, ParentId, HasChildren
  - □ Circular relationship
  - □ In optional properties allow NULL values
  - □ Additional tables for list-based data
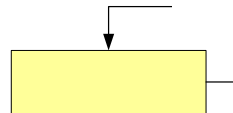
12

## Model evaluation

- **Pros**
  - Simple to use and implement
  - Accounts for the use of Catamorphisms
  - Simplified migration of data between different storage-level types (e.g. relational vs XML)
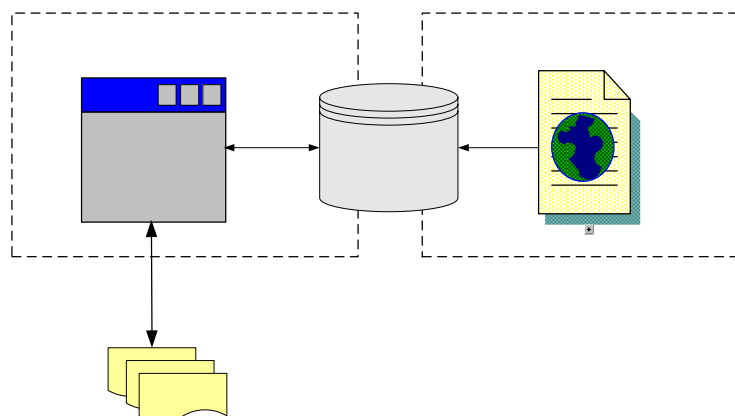- **Cons**
  - Doesn't take full advantage of RDB
  - Finding a particular records usually means traversing the whole tree of description
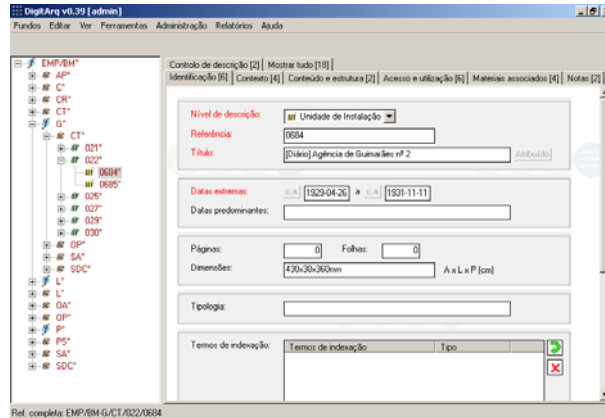
**13**

---

## DigitArq [final product]



**14**

---

## Finding aids management software [innovative features]

- Uses relative references
- Automatic description revision
- Inference mechanisms



**15**

---

## Conclusions

- Abstract model for hierarchical metadata representation
- Simple and transparent
- Independence of storage-level
  - Different types of databases can be used
- Easy to migrate data between different storage-levels

**16**

## Future work

- Model optimisation for Relation DB
  - Stored procedures
- Cache
  - Minimises traffic between application and DB
- Pre-loading
  - Predict what information is going to be requested next

17

---

## Questions?

**DIGITARQ**
**Creating and Managing a Digital Archive**

**Miguel Ferreira**
mferreira@dsi.uminho.pt

**José Carlos Ramalho**
jcr@di.uminho.pt

18