Building a Web Warehouse for Accessibility Data

Christian Thomsen and Torben Bach Pedersen,
Aalborg University,
Denmark

Agenda

- Introduction
  - Accessibility
  - EIAO and EIAO DW
- Architecture for the entire EIAO system
- The EIAO DW data warehouse
  - Conceptual model
  - Logical model
- Source data
- Aggregations/accessibility scores
- Experiences
- Future work

Introduction

- More and more important information is (only) available on the web
- Many web resources are not usable for users with special needs
  - Blind users using, e.g., *screen readers*
  - Elderly users with impaired vision
  - Physically disabled users that cannot use a pointing device

Accessibility

- If, for example, an image with a link does not have a text alternative, a blind user might not be able to navigate
- A web resource is *accessible* if people with disabilities can use the resource
- W3C provides a recommendation about how to create accessible web resources
  - Some of the check points can be checked automatically
  - Others cannot

Accessibility – cont.

- Accessible resources are advantageous to both providers and end users
- Accessibility is recognized as an important field and authorities have policies about ensuring accessibility
- Interesting to be able to check and compare accessibility of (groups of) web sites

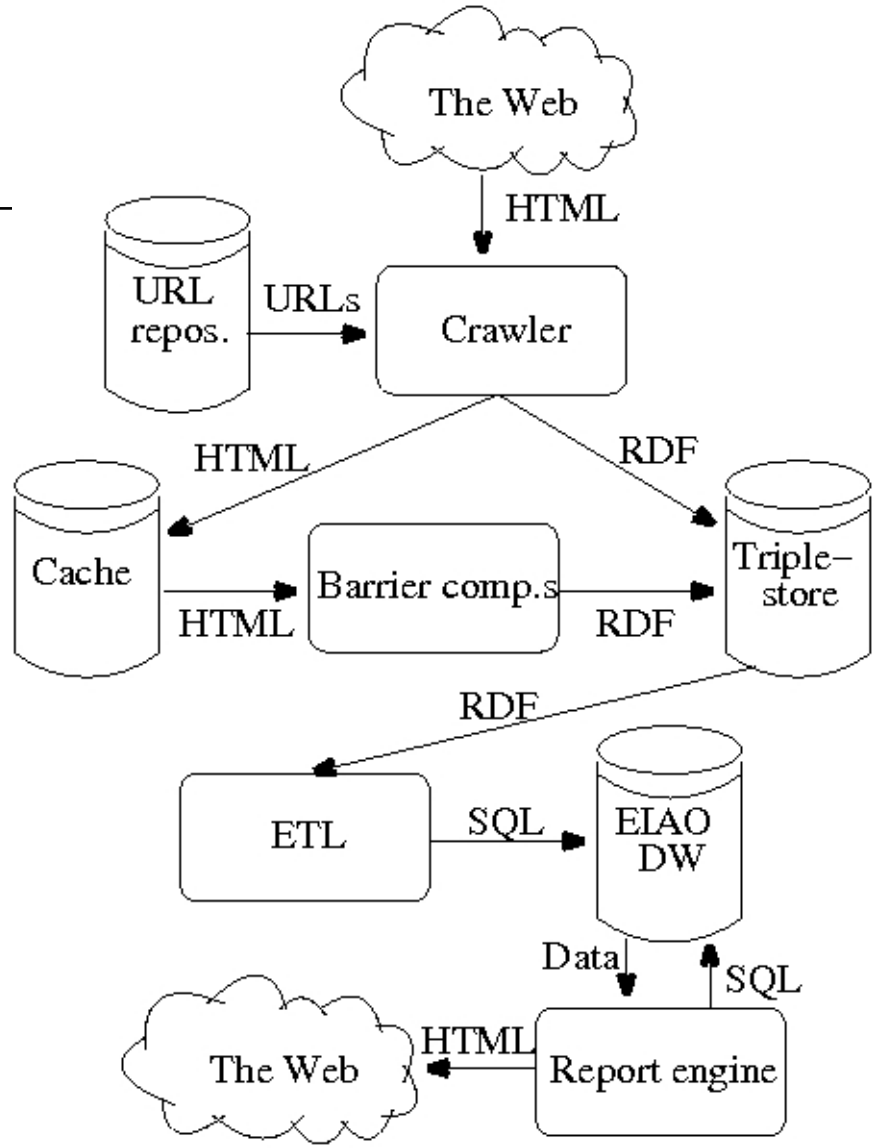
The EIAO project

- The European Internet Accessibility Observatory project develops a large-scale accessibility benchmarking
- The accessibility of 10,000 European web sites will be monitored automatically
- EIAO is based entirely on open source software
- The accessibility results will be stored in a data warehouse (EIAO DW) that will make results available online
- EIAO DW measures properties of the web
  - A web warehouse

Agenda

- Introduction
  - Accessibility
  - EIAO and EIAO DW
- *Architecture for the entire EIAO system*
- The EIAO DW data warehouse
  - Conceptual model
  - Logical model
- Source data
- Aggregations/accessibility scores
- Experiences
- Future work

Architecture



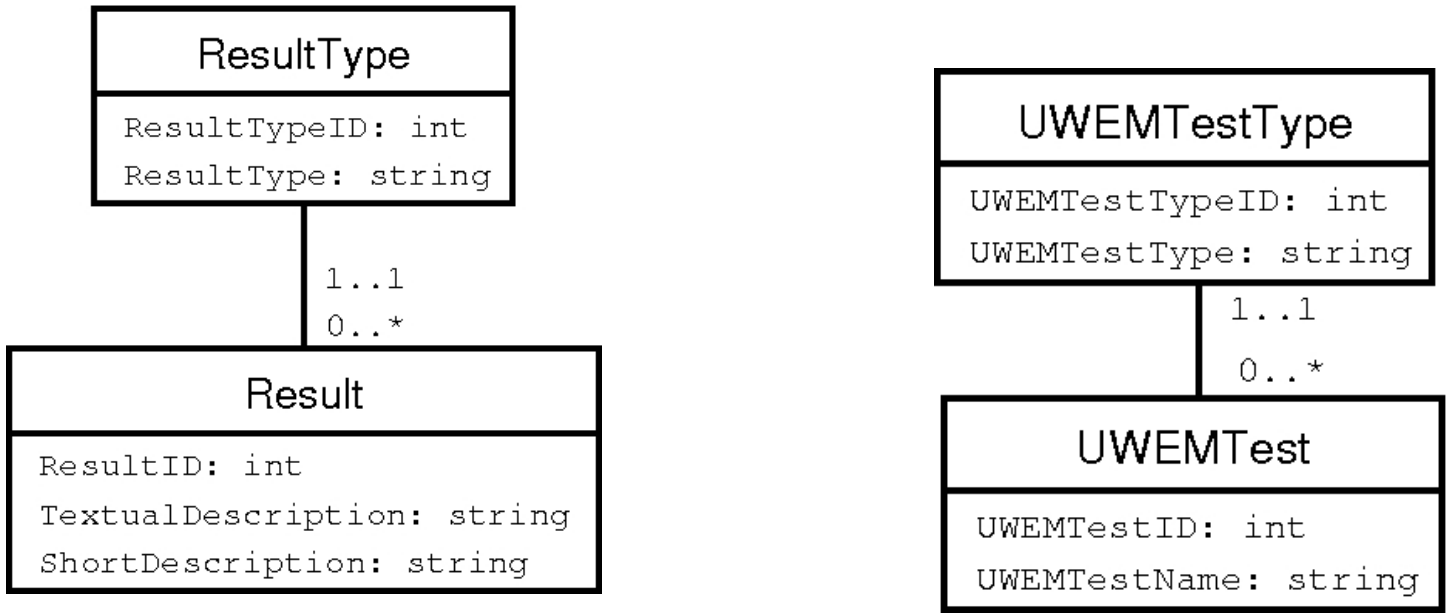
EIAO DW (Release 1)

- Should facilitate easy, efficient, and reliable analysis of the accessibility data
- Preliminary results indicate that
  - 76 pages per site are assessed
  - A page has 247 test subjects on average
  - $\Rightarrow$ 187.7 millions facts for 10,000 sites – each month
- Implemented in PostgreSQL 8.x
- Based on a star schema
  - But not a pure star schema

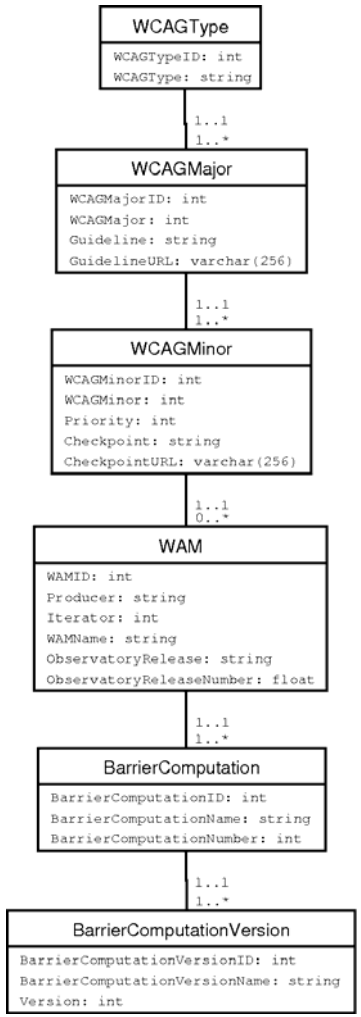
Conceptual model

- Modelled as UML
  - Classes and attributes are shown
  - Associations shown, not foreign keys
- 42 classes (9 unique dimensions in the logical model)
- 113 attributes
- Classes are grouped in dimensions in the descriptions

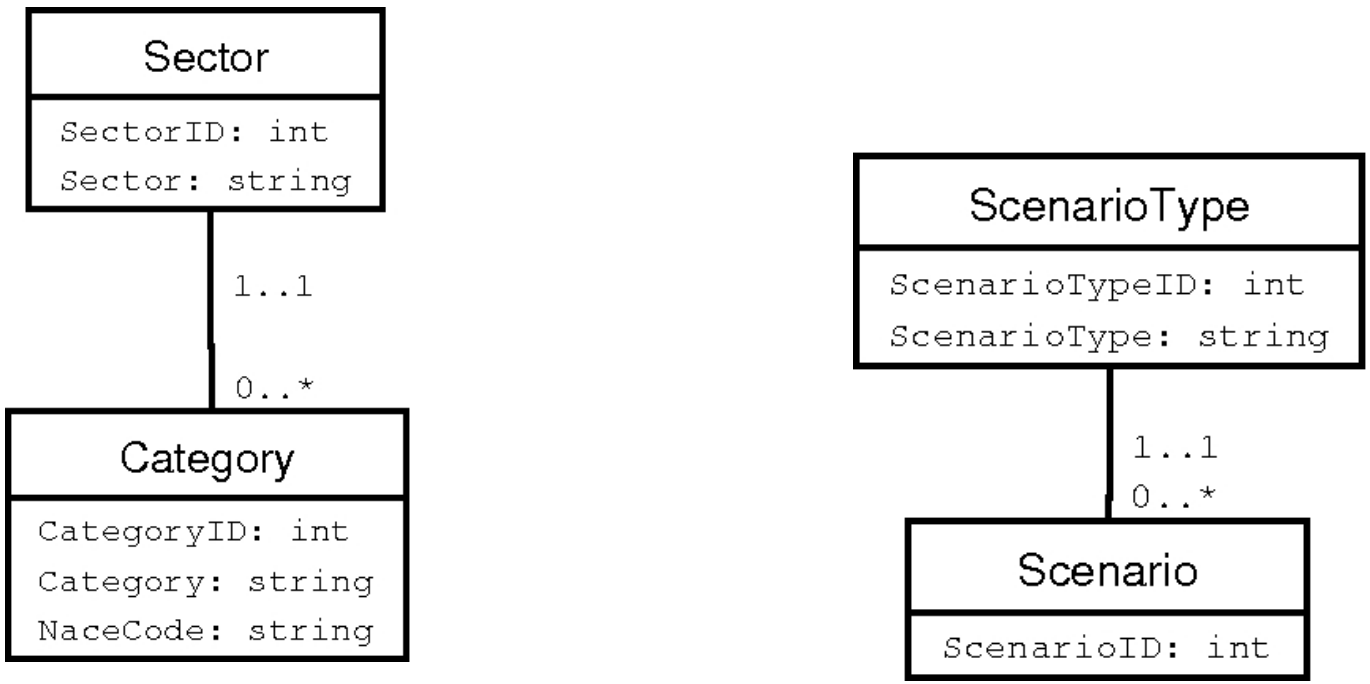
Conceptual model: Result and UWEMTest dimensions



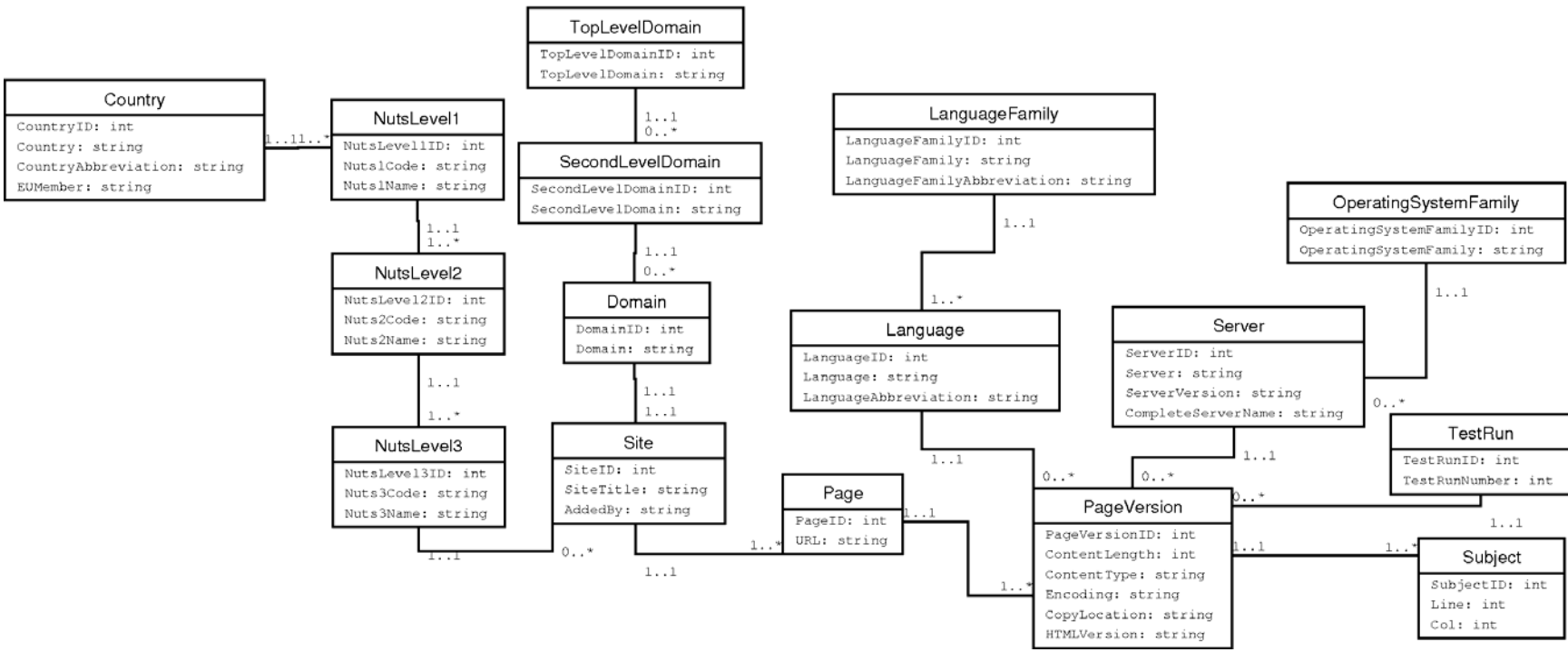
Conceptual model: BarrierComputationVersion dim.



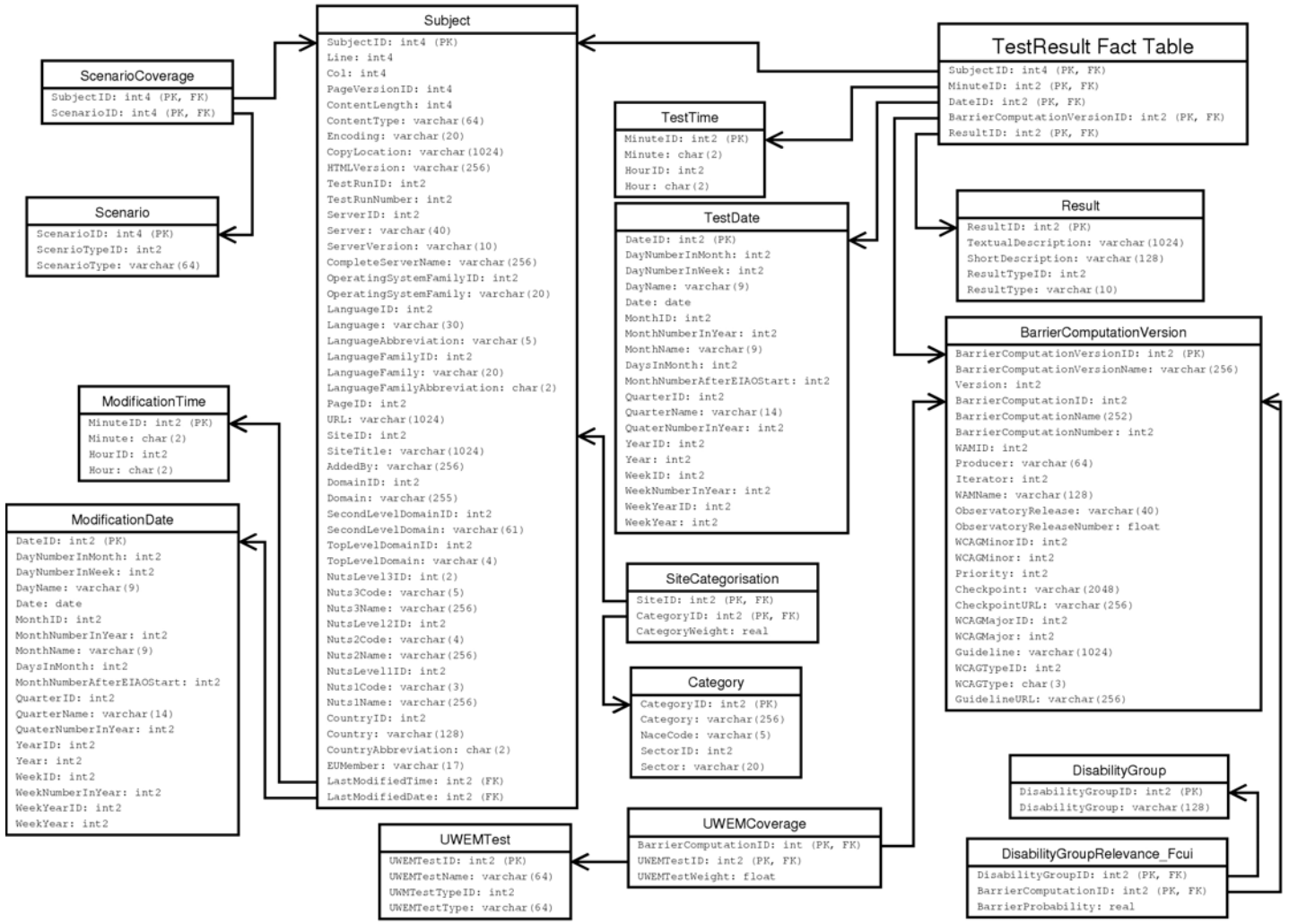
Conceptual model: Category and Scenario dimensions



Conceptual model: Subject dimension



Logical model



Agenda

- Introduction
  - Accessibility
  - EIAO and EIAO DW
- Architecture for the entire EIAO system
- The EIAO DW data warehouse
  - Conceptual model
  - Logical model
- *Source data*
- Aggregations/accessibility scores
- Experiences
- Future work

Source data

- Meta data stored in a Resource Description Framework (RDF) language
- The barrier computations produce test reports in Evaluation And Reporting Language (EARL)
  - EARL is itself an RDF language
- RDF is based on triples
  - (subject, predicate, object)
  - Things are identified with URIs
  - (http://example.org/persons/Eric, http://example.org/hasEmailAddress, mailto:eric@example.org)

Source data – cont.

- An object can be the subject of another triple
- Triples can form a graph
- The Extract-Transform-Load (ETL) tool, traverses the graph to load the data into the DW
- The load is conceptually relatively simple to do

Loading the data

- For each test run *t*
  - Add information about *t*
  - For each web site *w* covered by *t*
    - Add informations about *w*
    - For each scenario *s* within *w*
      - Add informations about *s* and its covered pages
      - Find test results from *s* and add them to the DW

Loading the data

- The EIAO project stores the triples in a *triplestore*
- Performs slowly when the triplestore gets big
- At one point, the EIAO project had more that 75 millions triples
- Between 90% and 99% of the ETL running time was spent on waiting for the triplestore

Loading the data

- The performance of the triplestore does not seem to scale linearly in the number of triples
- Solution:
  - Have many small triplestores instead of one big
  - Have a triplestore for each site in each test run and start the ETL many times

Agenda

- Introduction
  - Accessibility
  - EIAO and EIAO DW
- Architecture for the entire EIAO system
- The EIAO DW data warehouse
  - Conceptual model
  - Logical model
- Source data
- *Aggregations/accessibility scores*
- Experiences
- Future work

Aggregations/accessibility scores

- Pre-defined reports are available from the graphical user interface
- The reports consider a single site or a group of sites
  - For example all sites for radio stations or all sites from EU countries
- All the reports perform some kind of aggregation and presents an accessibility score
  - Based on how likely it is that a disabled used will meet a barrier within the (group of) site(s)
  - Very different from traditional aggregation such as SUM, MAX, and MIN

Aggregations – cont.

For the domain *d* in test run *t* (with key use scenarios $k_1,\ldots,k_m$) and disability group *g*:

$$C(d,t,g) = \sum_{i=1}^{m} C'(k_i, g)/m$$

where *C'* for a key use scenario *k* with the fail reports $r_1,\ldots,r_n$ is given by

$$C'(k,g) = 1 - \prod_{i=1}^{n} (1 - P_b(r_i, g))$$

where $P_b(r, g)$ is the barrier probability for user group *g* for fail *r*

Agenda

- Introduction
  - Accessibility
  - EIAO and EIAO DW
- Architecture for the entire EIAO system
- The EIAO DW data warehouse
  - Conceptual model
  - Logical model
- Source data
- Aggregations/accessibility scores
- *Experiences*
- Future work

Experiences

- EIAO DW is based on open source
- PostgreSQL is chosen as DBMS
  - Well-suited
  - Reliable
  - Support for materialized views is missing
    - Summary tables added manually to hold results of expensive aggregation functions

Experiences – cont.

- Python used for ETL software
  - Easy to put together a simple script using the in-built lists and dictionaries (hash maps)
  - Possible to tune to get good performance
- Some aggregation values can be calculated fast in the ETL process when all test results are seen

Experiences – cont.

- Hard to do develop DW concurrently with the development of the source systems
  - Schema for source data may change
  - Bugs in data-generating tools not found before ETL development starts
  - No realistic test data
  - Late specifications of reports and aggregations
- A lot of coordination needed
  - In the EIAO project, the developers are located in four countries
  - Testers and analysts in other countries

Experiences – cont.

- The used triplestore scales badly when millions of triples are present
  - Use many small triplestores
  - Investigate possibilities for better scaling repositories

Future work

- Work on Release 2.0 of the EIAO in progress
- A new EIAO DW will be released
- Updated schema
  - More meta data
  - (X)HTML and CSS documents considered together
  - Also information about technology usage
- Partitioning to handle huge data amounts
- New aggregations

Acknowledgements

- This work was supported by the European Internet Accessibility Observatory (EIAO) project, funded by the European Commission under Contract no. 004526