

QSS-based development of a web accessibility observatory

Terje Gjørseter, Parastoo Mohagheghi*, Morten
Goodwin Olsen

Faculty of Technology,
Agder University College

* Department of Computer and Information Science,
Norwegian University of Science and Technology

Background

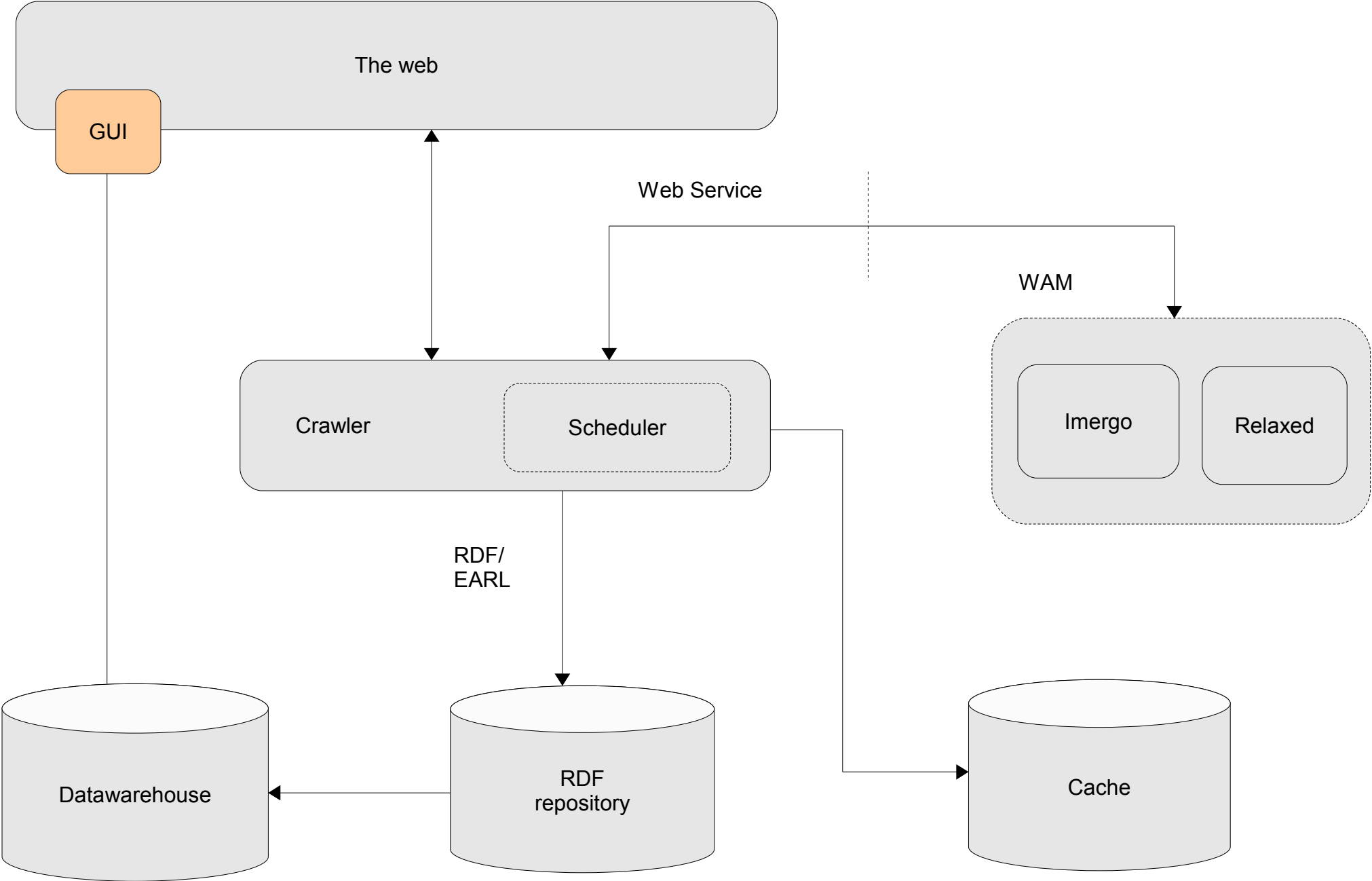
- EU funded project EIAO (European Internet Accessibility Observatory) for large-scale web accessibility monitoring
- Requirements from the EU
 - Scalability
 - Free and Open Source Software (OSS)
 - Interfacing with third party software for accessibility assessment
- Limited budget for software purchases combined with interest in OSS among the project participants

Approach

We have developed an OSS-based heterogeneous architecture with two main types of components:

- New components
 - Developed from scratch
 - Use functionality from other components, but do not extend these
 - Example of a new component: URL repository
- OSS-based components
 - Reused as-is, modified or customised existing OSS components
 - Example of reuse as-is: MySQL
 - Example of modification: HarvestMan

Architecture



Solution

- **Crawler**
 - Customised version of HarvestMan
- **WAMs (Web Accessibility Components)**
 - Relaxed as Framework
 - Schematron to express the formal rules
 - Customised to return EARL/RDF
 - Wrappers for integrating external tools as WAMs, both OSS and proprietary software
- **RDF Repository**
 - Heavily customised version of 3Store.
 - Uses Redland / MySQL
- **Data Warehouse**
 - PostgreSQL to give flexible analysis
- **GUI (Graphical User Interface)**
 - Uses Plone/Zope to present data

Results & Challenges

- Results
 - A functional observatory
 - A stable architecture to build on
- Challenges
 - Integration of OSS software with unknown bugs and missing expected functionality
 - Mix of languages; Python, Java/Jython etc. caused by external components
 - Performance

Conclusion

- Advantages of OSS
 - Good cooperation with OSS developers
 - Openness leads to repeatability
 - Facilitate other uses of the observatory
- Pitfalls
 - Need to test external OSS components extensively for bugs and needed functionality
 - Readability and understandability of external code is very important
- The Observatory is made possible with a limited budget through use and reuse of OSS components