OPEN-ADAS: progress and plans

Allan Whiteford, Martin O'Mullane and Hugh Summers

14th November 2006

Contents

- Background.

- Scope of OPEN-ADAS.

- Implementation:

  – tagging system,
  – relational database,
  – web interface.

- Next steps.

- Conclusions.

Background

- OPEN-ADAS is a joint development between the IAEA and the ADAS Project.

- Main goals are:

  - to index the data contained within the ADAS database,
  - to provide a searching system for these data,
  - to re-work the documentation and data status,
  - to provide access to the data freely via the web.

- With the exception of the last point, all of the above have benefit to ADAS Project members

Scope of OPEN-ADAS

- OPEN-ADAS is limited to a selection of key data classes:

  - key diagnostic data classes for fusion are targeted,
  - opacity (and related data) already available,
  - no point in releasing driver files.

- OPEN-ADAS will not release any of the ADAS code, only data,

  - exception is code necessary for reading the data.

- New developments with flexible partitioning will not be included:

  - still in development,
  - need to be tuned to transport characteristics etc.,
  - best used with close support from ADAS personnel.

OPEN-ADAS data classes

Class	Description	Files	Size
ADF01	Charge exchange Cross sections	118	3.0 MB
ADF04	Resolved specific ion data collections	1078	404 MB
ADF07	Electron impact ionisation coefficients	67	1.8 MB
ADF08	Radiative recombination coefficients	100	2.6 MB
ADF09	Dielectronic recombination coefficients	1531	1.0 GB
ADF11	Iso-nuclear master files	343	50 MB
ADF12	Charge exchange emission coefficients	45	2.0 MB
ADF13	Ionisation per photon coefficients	153	38 MB
ADF15	Photon emissivity coefficients	173	77 MB
ADF21	Effective beam stopping coefficients	218	4.4 MB
ADF22	Effective beam emission coefficients	402	7.6 MB
	Total	4228	1.58 GB

Implementation
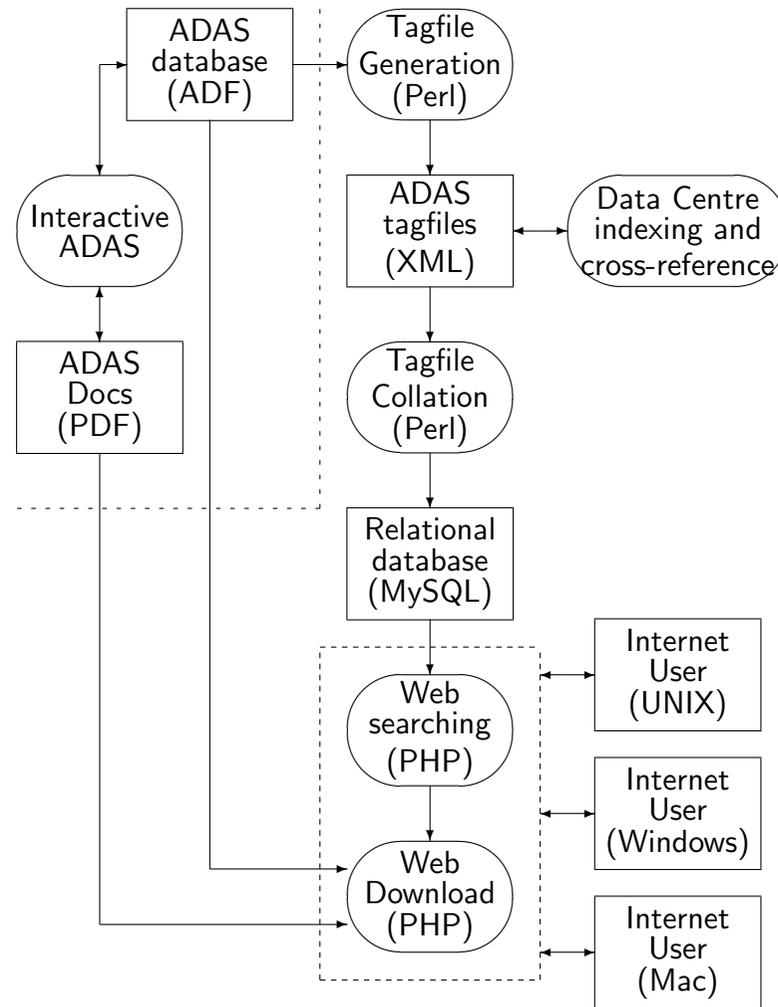
Tagging system

- Perl is used to scan each ADF file:

  - each data class is done in turn,
  - custom code is written for each data class,
  - each part of the file is examined:
    - ∗ filename,
    - ∗ data contents,
    - ∗ comments section.
  - Consistency is checked and warnings are flagged.

- A corresponding '`.tag`' file is generated alongside the original '`.dat`' file.

- Tag files are actually fully compliant XML files.

Example of tag file header

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE ADAS PUBLIC "-//ADAS//DTD ADAS 1.0//EN"
          "http://adas.phys.strath.ac.uk/adas.dtd">
<adf15>
      <file>
             <type>ADF15</type>
             <filename>pec96#c_pjr#c4.dat</filename>
             <directory>adf15/pec96#c</directory>
             <tagged_on>2006-07-21</tagged_on>
             <tagged_by>Allan Whiteford</tagged_by>
             <md5sum>722bda2a220dc12fe8a74c8c227bf725</md5sum>
      </file>
      <ion>
             <z0>6</z0>
             <z1>4</z1>
      </ion>
```

Transition entry in a tag file

```
<transitions>
        <transition>
                <z1>4</z1>
                <lambda units="A">40.7</lambda>
                <upper>
                        <level>3</level>
                        <cfg>1S1 2S1</cfg>
                        <m>1</m>
                        <l>0</l>
                        <j>.0</j>
                </upper>
                <lower>
                        <level>1</level>
                        <cfg>1S2</cfg>
                        <m>1</m>
```

```
                            <l>0</l>
                            <j>.0</j>
                </lower>
                <driving>
                        <level>1</level>
                        <cfg>1S2</cfg>
                        <m>1</m>
                        <l>0</l>
                        <j>.0</j>
                </driving>
                <type>Excitation</type>
        </transition>


    </transitions>
```

Predecessors and contributors

- All ADAS data files contain information on:

  - Predecessors — other files in the ADAS database used in generation,
  - Contributors — people who worked on the generation of the file.

- Up until now these links had to be followed on a case by case basis by hand:

  - no automatic way of knowing where a possible error in, e.g. an ADF04 file, had propagated to elsewhere in the database.
  - In reality, these things were always tracked down by someone.

- The indexing system extracts this information from each file.

- Searchable tree is then built which forms a key part of the database generated from the '.tag' files.

Relational database

- Each '`.tag`' file is re-read by a different Perl script and inserted into a database.

- Using MySQL due to support within Linux and price (free!) but others (e.g. Oracle) could be used.

- Information archived according to relational database type standards:

  - data for each file is split across different tables,
  - links between files are done by unique IDs rather than filenames,
  - names of contributors are stored in separate table.

- Indexing/cross-referencing possible with data centres.

- MySQL also has ease of integration with web based software.

Web Interface

- Web interface written in PHP and outputs XHTML.

- Allows interactive searching of the database via a number of routes:

  - searching by data class,
  - cross-data class searching by ion,
  - cross-data searching by wavelength*,
  - general free-form search*.

- Gives information on contents of each data file.

- Option to download the file, reading routines or documentation.

  $\star$ — Awaiting implementation.

Searching ADF15 files

Results of searching ADF15 files

Information on an ADF15 file

Next steps

- Documentation of code — almost none so far.

- Expansion to remaining data formats.

- Search by wavelength and freeform search.

- Review of the design aspects of the web interface.

- Preparation of software libraries, including documentation.

- Setting up a live site and specific deployment considerations.

Conclusions

- Progress so far has been quite fast and effective.

- A short report (28 pages) has been prepared.

- We think carrying on in a similar way is appropriate.

- Looking for a couple of volunteers who can:

  - try using the beta version of the website,
  - write a short summary of their findings: good and bad,
  - Probably a time commitment of 4–6 hours.

- Demo will be running on a laptop during coffee.