Astro-**W**ise **E**nvironment

Context

D.R. Boxhoorn, E.A. Valentijn

February 8, 2005

Contents

Chapter 1

Context - basic ideas

1.1 Context - introduction

Given the large amount of data hosted by the Astro-Wise Information System and the various ways to process the data, it is useful to filter the sea of stored objects and methods for daily data reduction work on a particular project/task.

Normally, for particular projects/tasks only a subset of all data is used at a time. The concept context facilitates this by means of tools which help to identify existing data and mark newly created data. The concept context also helps in marking and identifying data reduction methods or algorithms (TBC).

It is useful to discriminate the functionality of marking objects and methods when **creating** and when **viewing, reading or querying** objects. In the forthcoming we will discuss these two worlds separately.

The context is an integrated feature of the Astro-Wise environment, and therefor should be seen as an addition to the system, without losing existing functionalities. It is meant to support the work of groups or projects, but could equally well support individual enterprises, like giving a demonstration, for example. Objects created under a given context can still be accessible by the world or other groups when decided to set such privileges.

1.1.1 Properties of the context

In the Astro-Wise Environment, context refers to the context in which raw data, calibration data and reduced science data are **created** and **viewed** (selected). The context is characterized by a project; i.e. a project defines the values of the different properties of the context.

- Name - The name of the project.
- Instrument - The instrument that is used by the project.
- Members - The people that are responsible for the project.
- Managers - The people that can add members to the project, change the read privileges of data, change the project defaults, etc.
- Default privileges - Who can initially read data, that is created as part of the project.
- Derivation and Workflow - Defined by project designers, involving:
  - Methods (TBC - input from Project designers wanted)

  - Qualification

- Category - The category or type of project - Calibration, Science, Survey, Virtual, Maintenance. (TBC)
- Observing Strategy: Standard, Deep, Freq. (TBC should be standard attribute)
- Observing Mode: Stare, Jitter, Dither. (TBC should be standard attribute)
- Status - Processing status. (TBC)

1.1.2 Provided functionality

The available context functionality can be split into two parts. One part is related to the *creation* of objects, the other part is related to the *selection* of objects.

• Context for make

When objects are created, some of their attributes are set according to the current context. Most context attributes cannot be changed because they are defined by the project. The following attributes are set for created objects.

context-user Name of the user. Identifies the user accessing the database. Once an object is created by a user, its user attribute cannot be changed.

context-project Name of the project. Identifies the project within which the user wants to work. The user can change the context-project during a session. The default is undefined.

privileges Sets whether new objects are only readable by the creator, only readable by the members of the currently select project (by definition includes the creator), readable by `awe` users, or readable by anyone outside `awe`. The user can change the default privileges for new objects during a session. If not set by the user, the default privileges are the privileges of the currently selected context-project. To be able to delete objects it is required that only their creator has access to them.

own algorithms/task/CalFile TBD

For existing objects, *only* the privileges can be changed by users. More specifically, the creator of an object can specify if it is also accessible to the members of the project to which the object belongs, or to the world, if allowed by the definition of the project. A tool is provided to perform this change. This ensures that all dependent objects have their privileges changed, which is necessary to keep the database in a consistent state. A project manager is allowed access to all data belonging to a project, even if it was created by one of the other users in a project.

• Context for view

When objects are selected, some of their attributes are checked to see if they comply with the context as set by the user. Part of these attributes just simplify querying, because their specification in queries is no longer required. Other attributes are used to enforce read and delete permissions.

instrument WFI, PDS, OCAM, WFC, ...

project KIDS, VESUVIO, VST16, WHITEDWARFS, MYPROJECT, ...

privileges There are five levels of privileges

user Data are only readable to the user.

project Data are only readable by members of the project.

astro-wise Data are only readable by Astro-Wise users, i.e. everyone with an `awe` account.

world Data are readable by everyone, including an anonymous user.

vo Data can be accessed via VO mechanisms.

A project can be private to a specific group of users. Only these users are allowed to access existing data or to create new data under this project. Unauthorized users, i.e. those not part of the project, will not have any access to a private project.

The privileges can be used to mark data as private or public to that user. Data that are private can be deleted by the user that owns them. However, once data are made public, it can no longer be deleted because that could break history tracking for data that reference it.

The context is used to filter data, but this filtering can also be done in Python queries. It should be noted that the context does not replace the use of general Python queries. The context is more powerful because it can enforce read and write access and authorization. The context is also in effect when directly using SQL from any Oracle client. Hence, the same privileges are taken into account and authorization is required to access data given a certain context .

1.1.3 What Oracle provides

In Oracle it is possible to define a Virtual Private Database (VPD), where access to tables can be controlled at the object-level (row-level) and at the attribute-level (column-level).

In a VPD, access to a table is protected by one or more security policies. A security policy defines a limiting condition, which is automatically applied to each query accessing a table. The condition is given in the form of a standard **where** clause or an **and** clause, which means that attributes of a table can be used freely in the condition. Security policies can be applied to **select**, **insert**, **update**, **delete** and **index** commands. For each type of command a different security policy can be created.

Since Oracle 10g, attribute-level security policies are available, which are only applied when a particular attribute or attributes are accessed. In addition, protected attributes can return **NULL** if selected.

A security policy uses, in Oracle terms, an Application Context. The Application Context makes it possible toy define, set, and access application attributes. These application attributes are used to separate different users and applications. Setting of the application attributes can be enforced during logon of a database user. The predefined `USERENV` Context already contains attributes such as `CURRENT_USERID`, `ISDBA`, `OS_USER` and the like. Security policies are written in PL/SQL, which, for the Astro-Wise Environment, are part of PL/SQL package named `AWOPER.AWSECURITY`.

1.1.4 Building blocks

The context is implemented using a standard Oracle framework, called VPD, as explained in section 1.1.3. First, the definitions of the tables, that are used for internal bookkeeping of projects, are given. These tables are created once, during the installation of the `awe` database. The contents of these tables can be modified by tools, the function of which is explained in 1.1.5.

AWEPROJECTS – Central table with all project definitions

PROJECT	DESCRIPTION	INSTRUMENT	DEFAULT PRIVILEGES
KIDS	...	OCAM	PROJECT
VESUVIO	...	OCAM	PROJECT
MYPROJECT	...	OCAM	USER
PUBLIC-WFI	...	WFI	VO

This table contains the definitions of all projects and has columns for:

PROJECT Name of the project.

DESCRIPTION Description of the project.

INSTRUMENT Instrument that is associated with a project.

DEFAULT PRIVILEGES Privileges for newly created objects. Can have any of the privilege levels: USER, PROJECT, ASTRO-WISE, WORLD, VO.

A project management tool is used to change entries in this table, e.g. to add a project or to change the default privileges. The possible operations on this table are defined in section 1.1.5.

AWEPROJECTUSERS – Table with the users of each project

PROJECT	USER	TYPE OF USER
PET	RSTRAKE	ADMINISTRATOR
PET	KKERCHIVAL	NORMAL
PET	JTIDLEY	NORMAL

This table contains the members of each project and has columns for:

PROJECT Name of a project.

USER Name of the `awe` account.

TYPE OF USER NORMAL, ADMINISTRATOR, READONLY.

A project management tool is used to change entries in this table. The manager of a project can use this tool to add users to the project. The possible operations on this table are defined in section 1.1.5.

---

context attributes that are part of each DBObject

ScienceFrame, as stored in the object tables in Oracle

astrom	*...*	*ZEROPNT*	**USER**	**PROJECT**	**PRIVILEGES**
10.132, -32.434	...	22.15	JTIDLEY	VESUVIO	PROJECT
153.541, -49.592	...	22.21	RSTRAKE	KIDS	USER
...	...	...	...	...	...
...	...	...	...	...	...
...	...	...	...	...	...

Here, *astrom, ..., ZEROPNT* are the attributes of a `ScienceFrame` as defined in Python. The **USER**, **PROJECT** and **PRIVILEGES** attributes are not defined in Python, but are used by the policy functions that are defined for `awe` in Oracle. *All* objects in the database have the `USER`, `PROJECT` and `PRIVILEGES` attributes. It is the responsibility of the policy function to take them into account, or not.

POLICY FUNCTIONS

An Oracle security policy is defined for a complete *class* of objects. A security policy is only defined for certain classes. For RawScienceFrame, ScienceFrame, RegriddedFrame, etc. the security policy has to be defined. However, for classes like Chip, Filter, Instrument a security policy is not useful.

There are separate policy functions for combinations of operations—SELECT=VIEW, INSERT=CREATE, UPDATE=CHANGE, DELETE—and different kinds of tables=persistent classes. Each combination is now discussed separately.

	Raw Cal	Raw Science	Reduced Cal	Reduced Science
SELECT				
INSERT				
UPDATE	Project member only. Super QualityFlag + timestamp			
DELETE	Not allowed		Use Data Deletion Tool	

SELECT Raw Calibration Data Raw calibration data are not tied to specific projects and can always be read by all `awe` users.

SELECT Raw Science Data Use current project to access data.
`WHERE PROJECT=currently_selected_project`

SELECT Reduced Science Data Use current project to access data.
`WHERE PROJECT=currently_selected_project`

SELECT Reduced Calibration Data Use current project to access data.
`WHERE PROJECT=currently_selected_project`

INSERT Raw Calibration Data Raw calibration data are not tied to specific projects.

INSERT Raw Science Data Set project to current project.
`INSERT INTO ARAWSCIENCETABLE SET PROJECT=currently_selected_project`

INSERT Reduced Calibration Data Set project to current project.
`INSERT INTO AREDUCEDCALTABLE SET PROJECT=currently_selected_project`

INSERT Reduced Science Data Set project to current project.
`INSERT INTO AREDUCEDSCIENCETABLE SET PROJECT=currently_selected_project`

UPDATE Raw Calibration Data Only the Super QualityFlag can be changed.

UPDATE Raw Science Data Only the Super QualityFlag can be changed.
`WHERE PROJECT=currently_selected_project`

UPDATE Reduced Calibration Data Only the timestamp and the Super QualityFlag can be changed.
`WHERE PROJECT=currently_selected_project`

UPDATE Reduced Science Data Only the Super QualityFlag can be changed.
`WHERE PROJECT=currently_selected_project`

DELETE Raw Calibration Data Not allowed.

DELETE Raw Science Data Not allowed.

DELETE Reduced Calibration Data Only possible with Data Deletion Tool.

DELETE Reduced Science Data Only possible with Data Deletion Tool.

1.1.5 Functionalities

CREATE PROJECT (name, description, users=[], instrument, privileges)

Performed by User.

Purpose Define a new project.

Description Create a new project *name* for a group of users that is associated with the project. An instrument and default privileges can be defined for the project.

Specifies

- Name.
- A description of the project.
- Project users and their type (NORMAL, ADMINISTRATOR, READONLY).
- Instrument.
- Default privileges.

Restriction None.

DELETE PROJECT (name)

Performed by DBA.

Purpose Delete the definition of an existing project called *name*.

Description Project definitions that are no longer used can be deleted.

Restriction Project definitions can only be deleted if there is no data associated with the project.

CHANGE PROJECT (privileges|instrument|drop user|add user)

Performed by User with appropriate privileges.

Purpose

- Add a user to the project.
- Delete a user from the project.
- Change the type of a user.
- Change the instrument for a project.
- Change the default privileges for a project, i.e. the privileges that are set for newly created objects.

Description After a project has been defined, it is possible to add users to it or remove users from it. It is also possible to change the type of each project user to NORMAL, ADMINISTRATOR or READONLY.

Restriction

- Name cannot be changed.
- Only users of type ADMINISTRATOR can change the project definition.

SET PROJECT

Performed by User.

Purpose Select a project in an interactive `awe` session or script.

Description Inside `awe` this function is used to select the current project. Selecting the project affects all making and querying of data that follows.

Restriction A project can only be selected if the user is a member of the project. If the ANONYMOUS user is member of the project, *all* users can select the project.

GET PROJECT

Performed by User.

Purpose Obtain the definition of a project in an interactive AWE session or script.

Description Find out which group of users are taking part in a project, what the description of a project is, what instrument is being used, etc. This is useful in scripts and webservices, where it is only known at runtime which project is selected.

Restriction None.

GET LIST OF PROJECTS

Performed by User.

Purpose Get a list of all projects in an interactive `awe` session or script.

Description Obtain a list of all currently defined projects. This is useful in scripts and webservices, where it is only known at runtime which project is selected.

Restriction None.

DELETE DATA OF A PROJECT

Performed by User with appropriate privileges.

Purpose Clean up things that are no longer needed.

Description Data can be marked for removal. In that case, all data that belong together have to be removed, because of the requirement that the complete history of any object can be tracked. There are two categories of data that can be deleted:

- Reduced calibration data.
- Reduced science data.

Restriction Data can only be deleted if there are no references to it.

CHANGE THE READ PRIVILEGES OF DATA

Performed by User with appropriate privileges.

Purpose Make data visible to a different (larger) group of users.

Description The read level of the data can be either USER (only readable by the user), PROJECT (only readable by members of the project), ASTRO-WISE (readable by `awe` users), WORLD (readable by anyone) or VO (readable by VO compliant clients).

Restriction The read privileges can only be changed in such a way that they give access to the data to a larger group. Restricting privileges to a smaller group could break history tracking and is therefore forbidden.

1.1.6 MyDB

MyDB is a project with only one member, and which is readable by this member only. MyDB serves two purposes. On one hand it allows the user to *create persistent objects* that are part of the shared `awe` database. On the other hand MyDB allows users to *add persistent classes* for themselves, giving the user flexible database access from Python scripts. MyDB also allows direct creation of tables, views and similar database constructions via SQL.

Persistent classes that are defined in MyDB, can refer to the persistent classes in the shared `awe` database. The reverse is not true, i.e. the persistent classes in the shared `awe` database are not allowed to refer to MyDB persistent classes.

Calibration files and reduced science images, that are created in the shared `awe` database with the MyDB context, are initially only visible to the user that created them. At a later stage, the user can decide to make these data visible to the rest of the community, e.g. the people participating in a project. Each calibration file and reduced science image has a method to perform this action. It is important to know that this operation is done recursively and is irreversible. This keeps the shared data consistent and complete, by ensuring that there are no references from the shared data to any data in MyDB.

1.1.7 Setting and changing **context** properties

The following example shows how the context is set in `awe`

```
awe> from astro.database.Context import context
awe> context.set(instrument='OCAM')   # Operate on OmegaCAM data exclusively
awe> print len(RawScienceFrame.filename != '')
13245
awe> context.set(project='Hercules')  # Only Hercules + OmegaCAM data
awe> print len(RawScienceFrame.filename != '')
2013
awe> context.unset('project')         # Select from any project again
awe> context.unset('instrument')      # Select from any instrument again
awe> print len(RawScienceFrame.filename != '')
34767
```

The following example shows how the context is set in `SQL`

```
SQL> EXECUTE AWOPER.AWSECURITY.SET_INSTRUMENT('WFI');

PL/SQL procedure successfully completed.

SQL> SELECT COUNT(*) FROM "RawScienceFrame";

  COUNT(*)
----------
     10632

SQL> EXECUTE AWOPER.AWSECURITY.UNSET_INSTRUMENT();

PL/SQL procedure successfully completed.

SQL> SELECT COUNT(*) FROM "RawScienceFrame";

  COUNT(*)
----------
     20740
```

1.1.8 Open issues

- algorithms
- specific methods for calibration