ALMA Computing Memo No. 19

	Atacama Large Millimeter Array	ALMA-SW-019
		Revision:1.3
		2003-1-20
		AIPS++ Audit
		Steven T. Myers

Audit of the AIPS++ Package for ALMA Offline Data Processing

AIPS++ Audit

S. Myers, F. Viallefond, K.-I. Morita

Keywords: Requirements, Offline, Software, Science, Calibration	
Author Signature: Steven T. Myers	Date: 2003-1-20
Approved by: B. Glendenning, G. Raffi, K.-I. Morita	Signature:
Institute: NRAO, ESO, NAOJ	Date:
Released by:	Signature:
Institute:	Date:

Change Record

Revision	Date	Author	Section/ Page affected	Remarks
1.0	2002-07-12	S.T. Myers	All	based on SW-018
1.1	2002-09-17	S.T. Myers	All	version after ASAC
1.2	2002-11-22	S.T. Myers	All	final audit grades
1.3	2002-12-05	S.T. Myers	All	draft summary
1.3	2003-01-20	S.T. Myers	All	ok for release

$Id: ALMAoffline_audit.tex,v1.3,2003/01/20 myers Exp myers $

Contents

1   General Considerations

This document presents the results of the auditing of the AIPS++ software Package for the purposes of ascertaining its compliance with the ALMA Offline Data Processing Requirements given in ALMA-SW MEMO 18.

A. We use a descriptive scheme, with a set of grade codes stating how well the Package fulfills a given requirement: Adequate (A), Inadequate (I), Not Available (N), and Unable to Evaluate (U). The latter is used for items could not be properly evaluated at this time (e.g. items related to the tbd ALMA data format). There is an additional qualifier for "adequate" items (A/E) that indicates desired enhancements to the package.

B. For each requirement, the tool or function in the Package fulfilling that requirement is listed, where possible.

C. For items deemed inadequate (I) or missing (N), or which are adequate but enhancements are desired (A/E), the reasons for this are listed. Where possible, a *severity level* for the failure is noted: low, medium, high. This is based upon the importance of the requirement and the margin of failure of the package for the requirement. Note that a subjective choice is made between items at are adequate but could use further improvement (A/E), and those deemed (I) but low severity.

For (U) items that the auditors felt they could not rate, the reason was noted (e.g. unfamiliarity with this part of the package, incomplete documentation). We will try to enlist expert help to evaluate this item, and update the audit to reflect this new information.

D. The priority codes, as given in ALMA-SW MEMO 18 and repeated here, are:

1 = essential
2 = highly desirable
3 = desirable, but not critical

It is intended that Priority 1 items must be present in the Package and work with high efficiency. Priority 2 items should be in the Package, though there may have to be sacrifices in performance or availability may be delayed. We expect that the Package will fulfill all Priority 1 and 90% or more of Priority 2 requirements. Priority 3 items should be considered for upgrades or development.

E. The considerations during grading include the *functionality* (whether a feature is there and whether it does what is required), the *usability* (ease-of-use, efficiency, reliability), and the *documentation*. In this audit, grades are not given separately for these factors. Note that depending on the wording of the requirement, not all of these will apply to a given requirement. For example, some requirements only indicate the existence of a feature (functionality), while some in addition explicitly address the look and feel or ease of use (usability). Also, some of the performance issues will have to be deferred until a later cycle of auditing, when detailed testing and benchmarks will be available.

F. This first round of auditing is based upon the July 2002 release, plus any patches available. The the daily/weekly updates which are available at AIPS++ consortium sites can be used to follow up on problem areas where fixes are in the works. The audit reflects the best possible assessment of the AIPS++ package as of September 7, 2002. Note that the links to the AIPS++ documentation tree are for the July 2002 Release v1.7, and will likely be broken with the next release due to the way the AIPS++ documentation tree is built.

G. *Proposed use of this document:* The auditing team, with the SSR, will generate a synopsis of the results of this audit, which will include an executive summary of the audit, and proposed focus for the next cycle(s) of AIPS++ development. We suggest that this audit, along with the synopsis, will be given to the AIPS++ project as input to their development, and will form the basis for ALMA's future acceptance testing of the Offline Data Processing deliverables. For example, a list of I and N Priority 1 and 2 items from this audit could be used as a checklist. We expect that in response to this audit, the AIPS++ Project will generate an approximate costing and proposed

development schedule, which will then be negotiated with the ALMA Computing IPT. Note that the audit will be revisited at (TBD) intervals (e.g. annually, or at designated milestones), and progress toward fulfillment will be monitored. Note that in these later cycles, more extensive testing and benchmarking will be folded into the auditing process.

H. *Acknowledgements:* In addition to the contributions from the three primary authors of this document, extensive use was made of auditing and comments from a number of people. Additional auditing in critical areas was carried out by Crystal Brogan (NRAO), Michel Caillat (Obs. Paris), and Alain Coulais (Obs. Paris). Comments from Wim Brouw (ATNF), Tim Cornwell (NRAO), Brian Glendenning (NRAO) and Robert Lucas (IRAM) were extremely helpful, and were incorporated into the summaries. The authors thank these individuals for their contributions to this effort.

2 Offline Data Processing Requirements

2.1 General Requirements and Interaction with Other ALMA Elements

2.1.1 Goals of the Offline Package and Relation to the Pipeline

Note: An ALMA Offline Data Processing Package (or "the Package") is primarily intended to enable end-users of ALMA (e.g. observers or Archive users) to produce scientifically viable results that involve ALMA data products. The secondary use is to enable ALMA staff to assess the state of the array and derive calibration parameters for the system, although it is anticipated that the Pipeline will be the primary engine for Online Data Processing and system monitoring.

Also note that not all offline analysis tools will necessarily be in the Pipeline Package. For example, one of the important differences between Pipeline and Offline reduction path is that Offline one should have extensive interactive capabilities to merge and compare data with different resolution, coordinate system, data grid, and so on.

The SSR does feel that there is a great deal of synergy between the Offline Package and Pipeline that should be exploited by the designers of the respective software.

OL–1.1–R1 There must be an Offline Data Processing Package that fulfills the requirements laid forth in this document.

> *Priority: 1*
>
> *Grade: A*

OL–1.1–R2 All standard observing modes supported by ALMA must be processable by the Package.

> *Priority: 1*
>
> *Grade: U*
>
> *Summary:* ALMA standard modes are not finalized at this time.

OL–1.1–R3 The Package shall be installable at the users home institution and available at ALMA regional centers (both locally and remotely). It shall be portable to **supported platforms designated by the ALMA Project**, including systems without network connections and laptops.

> *Priority: 1*
>
> *Grade: U*
>
> *Summary:* The list of supported platforms is unavailable at this time.

OL–1.1–R4 The performance of the Package shall be quantifiable and commensurate with the data processing requirements of ALMA output and the scientific needs of users at a given time. The timing and reproducibility of results for a fiducial set of reduction tasks on specified test data will be benchmarked (e.g. "AIPSmarks") and compared against other packages and **a list of benchmark specifications provided and maintained by the Project**.

> *Priority: 1*
>
> *Grade: U*
>
> *Summary:* The benchmarking process will commence at a later date.

OL–1.1–R5 All standard processing functionality available in the Pipeline shall be available in the Package also as an offline analysis option.

> *Priority: 2*
>
> *Grade: U*
>
> *Summary:* The pipeline functionality is undefined at this time. A separate audit will use the updated SW-11 Pipeline requirements as a template.

2.1.2 Operational Issues

OL–1.2–R1 Installation of the Package by an end user must be straightforward, preferably without special (e.g. root) user permission.

Priority: 1

Grade: A

Summary: The AIPS++ installation from CDROM, ISO image, or inhale process is straightforward.

OL–1.2–R2 Error reporting and handling shall be user-understandable and non-destructive at all levels in the Package:

> **OL–1.2–R2.1** Error reporting messages shall be written for end users, not programmers.
> *Priority: 1*
> *Grade: I Severity:* low
> *Summary:* Many error messages, especially those generated by glish failures such as glisk-tk widgets, are unduly cryptic and are alarming to users.
>
> **OL–1.2–R2.2** There must be provision for job control such as interrupt and abort.
> *Priority: 1*
> *Grade: A/E*
> *Summary:* GUI tools have Abort buttons and cntl-c gives the option to abort from glish. However, this does not always work well, and more graceful job control is desired.
>
> **OL–1.2–R2.3** Error handling shall be non-destructive (data shall not be left corrupted, and recovery of recent changes be available). Common failure modes (as enumerated: invalid application parameters, exceeding of resource limits (disk/memory), and algorithm failure modes (e.g. no convergence) should be handled gracefully.
> *Priority: 2*
> *Grade: I Severity:* high
> *Summary:* There are still failure modes (e.g. glish-tk problems) where aips++ crashes or hangs, sometimes crashing or hanging the windowing system of the OS in addition. Sometimes, files are left corrupted, though it is ususally possible to reproduce processing to the point of crash (though not always). Memory usage of AIPS++ can be high, and problems can occur when using a default setup on a machine with less memory — the should be transparent to the user. These kind of severe errors are high priority with the AIPS++ programmers, and improvement in this area should be steady over the next releases.
>
> **OL–1.2–R2.4** Code traceback of execution errors shall be available. This should be geared to the effective reporting of failure modes and bugs by the user.
> *Priority: 3*
> *Grade: A*
> *Summary:* There seems to be traceback on errors, though they are nearly impossible for non-programmers to figure out.

OL–1.2–R3 There shall be session logging, including the following features:

> **OL–1.2–R3.1** logging of commands and user inputs shall be provided
> *Priority: 1*
> *Grade: I Severity:* medium
> *Summary:* The `scripter` currently does not log all commands, only (most) GUI operations (bug AOCso03842), while the `logger` has the option to log the CLI commands but does not do the GUI commands. A full logging (in correct order), perhaps by adding the GUI commands to the `logger`, is needed.
>
> **OL–1.2–R3.2** logging of tool results such as success or error, files written, time of completion shall be provided

Priority: 1
Grade: A
Summary: The `logger` seems to record the pertinent information.
OL–1.2–R3.3 logging of tool output such as summaries of results shall be provided
Priority: 2
Grade: A
Summary: The `logger` seems to record the pertinent information.
OL–1.2–R3.4 the session log shall be readable by the user (i.e. in a text file, not in a binary format)
Priority: 2
Grade: A
Summary: The `scripter` output is a glish script (ascii), though the `logger` data is stored in an AIPS++ table.
OL–1.2–R3.5 session logs shall be executable by the package UI, reproducing the entire session
Priority: 3
Grade: I Severity: medium
Summary: The `scripter` output is an executable glish script, though the scripter output is currently incomplete (see above).

OL–1.2–R4 There shall be comprehensive handling of multiple users and multi-tasking with access and process control.

Priority: 1

Grade: A

Summary: The aips++ server system (at least as its installed at NRAO) is able to deal with multiple users (more than a dozen during the recent summer school). You can have multiple tools running concurrently, though you must beware of interactions.

OL–1.2–R5 The Package providers shall provide support for the Package, with bug-fixing on timescales appropriate to the severity level of the defect (e.g. 1 week or less for catastrophic bugs with no work-around). **These defect levels and timescales should be delineated by the ALMA project.**

Priority: 1

Grade: U

Summary: The levels and timescales have yet to be provided by ALMA. Note that the DDTS system aips++ uses should make bug-tracking straightforward.

OL–1.2–R6 The Package providers shall provide timely improvements and updates based on user feedback. There shall be a path for the ALMA Project, for its own use and as a proxy for the users, to influence the development cycle of the Package. It is the responsibility of the Project to negotiate needed improvements with the Package providers.

Priority: 1

Grade: A

Summary: The DDTS defect tracking system is in place, and there are a number of AIPS++ advisory and user groups functioning (e.g. the NAUG at NRAO). There is an ALMA representative on the AIPS++ executive. This document is a step in the direction of AIPS++ development towards ALMAs needs.

OL–1.2–R7 Backward compatibility of core package components should not be broken without compelling scientific reasons. Tools should be provided to parse user scripts and warn of package changes.

Priority: 2

Grade: U

Summary: The AIPS++ package is not yet mature enough for us to gauge whether updates are compatible. So far, there are often changes which are not compatible (particularly in glish scripts, not in data formats), but this is expected in quickly-developing code. This issue will be resolved when a long-range development plan is in place for ALMA AIPS++ deliverables, and this requirement should be revisited in one year.

OL–1.2–R8 Source code for the astronomical routines in the Package shall be available to the user.

Priority: 2

Notes: This does allow for the use of some proprietary data handling, output formatting, and special processing (e.g. Pixon) routines.

Grade: A

Summary: Source code, both C++ and glish, is available in the distribution.

OL–1.2–R9 User installation of the basic package shall not be restricted by other issues such as expensive or unduly restrictive licenses. The Package license should convey all other necessary licenses (such as GNU) or they should be available to the user for only a nominal fee.

Priority: 2

Comment: This allows for the use of commercial components. However, a balance must be struck between excessive costs to the project and requiring some cost to be borne by the user, should this issue arise.

Grade: A

Summary: Currently, it is free with no extra licenses necessary except in one case (IDL for Pixons), and this is not a high priority feature.

OL–1.2–R10 There shall be the provision for the development and incorporation of user-supplied code.

Priority: 2

Grade: A

Summary: Incorporation of glish-level code (scripts) is straightforward. Incorporation of C++, C or Fortran is possible by users.

OL–1.2–R11 The application of successive stages of calibration, correction, flagging and editing shall not be destructive to the data. The Package should be able to recover and revert to earlier stages without resorting to saving of the entire dataset at intermediate states.

Priority: 2

Grade: A

Summary: Some operations change columns in the ms, others make tables. For the most part, the original data is untouched, and the process can be repeated to recreate the data. This is not perfect, but is getting better and is adequate for users.

2.2   Interface

2.2.1   General User Interface Requirements

OL–2.1–R1 User must be able to choose from a variety of interface styles, including:

> **OL–2.1–R1.1** A Command Line Interface (CLI) must be provided, with access via both an interactive input and via script.
> *Priority: 1*
> *Grade: A*
> *Summary:* The glish interface is a suitable CLI environment.
>
> **OL–2.1–R1.2** A Graphical User Interface (GUI) must be provided for interactive processing. Actions taken under the GUI must be loggable, editable, and executable by the CLI.
> *Priority: 1*
> *Grade: A*
> *Summary:* There is a GUI interface to glish in aips++, and GUI actions can be sent to the `scripter`.

OL–2.1–R2 The user shall be able to interact with the host operating system with command sequences invoked from the UI.

Priority: 1

Grade: A

Summary: The `misc.os` tool has OS-like functionality. There is also the glish `shell` command (which does not log the output or break lines however), and also the `misc.shellcmd`. Though technically not the same as interacting with the host OS, in my estimation this functionality is probably good enough for our needs.

OL–2.1–R3 Multitasking for all interfaces shall be available where appropriate. It must be possible to run one or more long-running calculations in the background. While background tasks are running normal interactive activities must be possible.

Priority: 1

Grade: A

Summary: Multiple tools and windows can be run concurrently, as well as multiple instances of aips++ (memory permitting).

2.2.2   Graphical User Interface (GUI)

Note: The GUI is intended to be the primary interactive mode for users, especially neophyte users, and thus should be tailored for clarity and ease of use. Use of the Package GUIs should be a pleasurable, not a frustrating, experience!

OL–2.2–R1 The GUI shall provide real-time feedback via standard compact displays:

> **OL–2.2–R1.1** Window updating must be fast (less than 0.1s on same host).
> *Priority: 1*
> *Grade: U*
> *Summary:* Benchmarking at this time is not ideal. However, it does appear that updating of windows (when GUIs are constructed in glish) can be significantly slower than this. This should be quantified within the next year.
>
> **OL–2.2–R1.2** Windows shall not take up excessive screen space, with full GUI controls visible on a window one-third the size of a standard view surface (or approximately 800 × 800 pixels).
> *Priority: 1*

Grade: I Severity: low

Summary: Most GUIs are compact enough, though some can be large (e.g. the `viewer` Adjust Panel can be too long (section Axis label properties) for the 1024 × 780 screen of a typical laptop), and some tools generate many windows. Some of these are being changed. Questions like this should be addressed by an interface focus group (with ALMA participation).

OL–2.2–R1.3 Users shall have the choice of cascading windows or re-use of a single window for new operations.

Priority: 2

Grade: I Severity: low

Summary: The manner of windowing is not under user control.

OL–2.2–R1.4 Moving and resizing of all windows must be available, robust, and easy.

Priority: 2

Grade: A

Summary: Windows can be dragged and resized as normal X windows. However, once resized the automatic sizing capability in the GUI is lost. Note: this has been submitted as a defect, and is being worked upon.

OL–2.2–R1.5 There shall be a master control GUI for process control which keeps track of sub-windows and tools.

Priority: 3

Grade: A/E

Summary: There is a *Tool Manager* GUI which keeps track of tools, but not of windows per se (must use the OS window manager. Could use some improvement.

OL–2.2–R2 It must be easy to run GUIs remotely from the host machine (e.g. via X displays).

Priority: 1

Grade: A

Summary: This seems straightforward.

OL–2.2–R3 The use of the GUI shall not entail an excessive learning curve. Average users, with experience with the current generation of packages (e.g. AIPS, GILDAS, IRAF, MIRIAD) shall be able to become proficient in GUI use in a timescale of approximately 12 hours dedicated use, and truly neophyte users (e.g. graduate students) should be reach proficiency with an investment not exceeding 40 hours of dedicated use.

Priority: 2

Grade: A/E

Summary: The intent of this requirement is unclear as stated — as written it seems to refer to the mechanics of the GUI operation rather than ease of use following astronomical tasks which was probably the intent of the writers. From a mechanical standpoint, the use of the GUIs from the *Tool Manager* is straightforward, and the `viewer` operation is also easy to learn. In hindsight, the learning time requirements should refer to astronomical processing specifically, rather than GUI mechanics, and a new requirement should be written as such, though in practice these issues are covered under other requirments (e.g. OL–2.2–R6 and in section 2.2.5).

OL–2.2–R4 GUI-based tools shall be available for reduction of data taken in all standard ALMA observing modes.

Priority: 2

Grade: U

Summary: ALMA standard observing modes are undefined at this time. Currently, the GUI has access to all glish-based functions and thus all of aips++, so fulfillment of the rest of these requiements will fulfill this one.

OL–2.2–R5 The look and feel of the GUI will be uniform throughout the entire package.

> *Priority: 2*
>
> *Grade: A*
>
> *Summary:* This is currently the case, as there are few custom GUIs (e.g. `viewer`, `msplot`, parts of the `dish` package) and those are acceptable.

OL–2.2–R6 The look and feel of the GUI must be acceptable to both novice and more advanced users. The GUI mode might be customizable (perhaps through menu selection for "novice" or "advanced" mode) or a different simpler set of GUI tools might be available for beginners. The GUI features for beginner mode shall include:

> **OL–2.2–R6.1** built-in help facility with access to novice-oriented help documents (e.g. sections in the cookbook)
> *Priority: 2*
> *Grade: A*
> *Summary:* There are buttons and "wrench" menu items available for help which drive a browser to the (hopefully) appropriate entry.
>
> **OL–2.2–R6.2** sensible defaulting of values for parameters, with guidance for user choices where needed
> *Priority: 2*
> *Grade: A/E*
> *Summary:* Many, if not most, menu boxes have ways of setting defaults from the ms, and global defaults are for the most part sensible. This could use some further improvement, but is quite good now and is getting better.
>
> **OL–2.2–R6.3** integrated functionality built around common data analysis tasks (e.g. single-field spectroscopic observations with fast switching, OTF mosaicing in continuum mode, snapshot observations of a large number of targets in single-field continuum mode)
> *Priority: 2*
> *Grade: I Severity:* high
> *Summary:* There are few integrated tools at this time available from the GUI. Many of the early attempts at "wizards" (e.g. `imagerwizard`, `mosaicwizard`) or integrated tools (`map`) have been abandoned or have not been updated for a long time. Some progress has been made in building interactivity in some tools (e.g. *interactive=T* in `imager.clean`). NOTE: The development of specialized tool for ALMA could reasonably be the topic of a special ALMA-AIPS++ focus group study. It might be appropriate that the ALMA project, via the SSR, give input to AIPS++ in this area.

OL–2.2–R7 It is shall be easy for users to develop and include their own custom GUIs in the Package.

> *Priority: 3*
>
> *Grade: A*
>
> *Summary:* There is a substantial set of GUI building tools (the `guiutils` module).

2.2.3   Command Line Interface (CLI)

Note: the CLI is the primary mode for automatic reduction for ALMA, and it is anticipated that there will be a suite of "standard" scripts developed to help users in data reduction tasks. Thus it is important that the Package support all of its critical modes in the CLI.

OL–2.3–R1 The interface must have the facility to read in command files for batch processing of a sequence of CLI commands.

> *Priority: 1*

Grade: A

Summary: Glish scripts can be read in and executed easily.

OL–2.3–R2 The CLI shall have command-line recall and editing, with name completion where appropriate.

Priority: 1

Grade: A/E

Summary: There is command-line recall (with editing) in glish. However, there is no name completion, probably due to the general nature of glish syntax. Given the ability to introduce and overload variable names, it might be argued to be never appropriate, but should be investigated.

OL–2.3–R3 All functionality of the GUI must also be available in CLI mode (although possibly with loss of simplicity in instances where the graphical selection is important).

Priority: 1

Grade: A

Summary: Nearly all GUI tools are direct implementations of the CLI equivalents. There are a few GUIs (e.g. msplot, viewer) that have no CLI level equivalents (invocation from the CLI brings up the GUI), but these are necessary.

OL–2.3–R4 The CLI shall be usable remotely over low-speed (14400 baud) modem lines or network connections, with ASCII terminal emulation.

Priority: 2

Grade: A

Summary: This works but is infrequently used.

OL–2.3–R5 A CLI mode menu for display and editing of input parameters and execution of tasks is desirable.

Priority: 3

Grade: N Severity: low

Summary: This can be done through custom scripting (see examples in the Recipes repository on the AIPS++ web page. It may be available in some tools, but is not well developed or documented.

2.2.4 Interface Programming, Parameter Passing and Feedback

OL–2.4–R1 The UI must have basic programming facilities such as:

OL–2.4–R1.1 variable assignment and evaluation
Priority: 1
Grade: A

OL–2.4–R1.2 array handling
Priority: 1
Grade: A

OL–2.4–R1.3 conditional statements
Priority: 1
Grade: A

OL–2.4–R1.4 control loops
Priority: 1
Grade: A

OL–2.4–R1.5 string manipulation
Priority: 1
Grade: A

OL–2.4–R1.6 user-defined functions and procedures with argument or parameter passing
   Priority: 1
   Grade: A

OL–2.4–R1.7 process control, interrupts, error handling
   Priority: 1
   Grade: A
   Summary: A strong point of glish. Used by Tony Willis for ACSIS (and in fact by all the GUIs and tools).

OL–2.4–R1.8 standard mathematical operations and functions
   Priority: 1
   Grade: A

OL–2.4–R1.9 efficient special vector and matrix operations
   Priority: 2
   Grade: A
   Summary: Glish handling of vectors and arrays is efficient programming-wise, though benchmarking will be necessary to evaluate the speed.

OL–2.4–R1.10 user-defined data structures
   Priority: 2
   Grade: A
   Summary: Glish records are somewhat arbitrary, but are probably adequate for what users would want.

OL–2.4–R2 Commands executed will be logged, with provision to re-execute the session.

   Priority: 1

   Grade: A

   Summary: The GUI user can choose to send commands to the `scripter`, which currently does not log all commands, only GUI operations (bug AOCso03842), while the `logger` has a script generator which does not log GUI commands. This requirement overlaps with OL–1.2–R3.1.

OL–2.4–R3 Input parameter and syntax checking shall be effected upon function calling or parsing with reporting of incorrect, suspicious or dangerous choices before execution where possible.

   Priority: 1

   Grade: I Severity: medium

   Summary: The parameter checking is inconsistent across tools, and seems to depend on whether the programmer bothered to implement this or not. There does seem to be basic type checking (which can give confusing results for more complicated measures), but few user-friendly warnings for suspicious or dangerous choices. There is room for improvement here, though many cases have been submitted as bugs and are being fixed.

OL–2.4–R4 Parameters shall be passable between applications in as transparent a manner as possible. However, global variables should not be the default, unless designated specifically by the user-programmer.

   Priority: 2

   Grade: A

   Summary: All glish applications are function calls with arguments. By default, function variables have *function scope* only, but can be defined to have *local, wider*, or *global scope*. What this requirement means for GUIs is unclear, though cut and paste of values seems to suffice.

OL–2.4–R5 Application variables shall be named consistently and as clearly as possible indicating their intended use using astronomical terms where appropriate.

   Priority: 2

Grade: I Severity: low

Summary: For the most part, variables have sensible names. However, there are cases where naming between tools is not consistent, in particular the way in which channels, etc. are selected in tools such as `ms`, `flagger`, `imager`. Many of these have been noted and are being worked on.

OL–2.4–R6 Parameter inputs to tools shall be stored for later recall:

> **OL–2.4–R6.1** Tool inputs shall be saved on closure, and reinstated on the next instantiation of the tool.
> *Priority: 2*
> *Grade: A*
> *Summary:* The GUI tools accesible from *Tool Manager* menus remember previous inputs, though in script mode inputs are not remembered and parameters default to the original defaults.

> **OL–2.4–R6.2** It shall be possible for the user to save the state of the parameters for the entire package, as well as for individual tools, as a named set (e.g. SAVE/GET in AIPS), and to recall these when desired.
> *Priority: 3*
> *Grade: I Severity:* low
> *Summary:* The *Tool Manager* level GUI parameters are saved in a cache (see above), and some GUI tools (e.g. `viewer`) allow saving of internal parameters, but this is not consisently implemented.

2.2.5 Documentation and Help Facility

OL–2.5–R1 There shall be a variety of help levels and documentation formats accessible from the UI and over the Internet, applicable to novices, experts, and technical users. These shall include:

> **OL–2.5–R1.1** user cookbooks with extensive examples
> *Priority: 1*
> *Grade: I Severity:* high
> *Summary:* Although much progress has been made in the past year on the telescope-specific "cookbooks" in Volume 3 of the Getting Results document of AIPS++, there are still notable deficiencies. In particular, much more GUI-oriented information, including screen shots, is required. I anticipate that this will improve steadily over the next few years with regards to VLA, GBT, and eventually VLBA, processing. When ALMA comes along, then a significant effort will need to be made to have the relevant cookbooks ready. Although not currently available as part of the AIPS++ documentation tree, there is a BIMA Cookbook available from NCSA, which is a step in the right direction, and is more millimeter-wave oriented than the official AIPS++ documentation. Note that examples in the Reference manual should probably include the module dependencies (e.g. `include ''simulator.g''`) so that a novice cutting and pasting will not encounter unnecessary problems.

> **OL–2.5–R1.2** application descriptions and reference manual (with all inputs to functions and tools);
> *Priority: 1*
> *Grade: I Severity:* medium
> *Summary:* The AIPS++ Users Reference Manual describes each tool, but many of the descriptions are too terse to be of help to users (e.g. just saying what type the entry is), and the examples tend to be trivial.

> **OL–2.5–R1.3** online help, FAQ, email contacts;
> *Priority: 1*
> *Grade: A*
> *Summary:* The AIPS++ website is extensive, and fairly easy to navigate. Contact links are provided.

OL–2.5–R1.4 release history, bug reports and tracking, patch descriptions;

Priority: 1

Grade: I Severity: low

Summary: There are some changelogs available from the Documentation and Programming links, and there are Release Notes available with each release (I could not find this online). However, these are not well maintained or easily navigable. I expect these to improve as the releases become more mature and functional.

OL–2.5–R1.5 programmer references and guides;

Priority: 2

Grade: I Severity: low

Summary: There are a number of programmer references available under the Programming AIPS++ doc tree. The programmers that I canvassed say that much of the documentation is out-of-date and the quality varies wildly across the package (depending on the documentation skills of the programmer of a given function). The system does seem to be set in a reasonable way such that documentation is put in the headers to the code itself, and thus the "reference" part of the documentation of the classes and methods is nearly complete. However, there does not seem to be an up-to-date comprehensive programmer's guide, and there is a lack of complete examples with comments describing how to use the main concepts presented in the reference part. Some effort in the direction of updating and improving programmer documentation would be very beneficial to new programmers. We suggest that the AIPS++ website offers a page where the user will find links to the appropriate section(s) depending on her/his kind of activity, e.g. full end user, glish programmer, AIPS++ programmer.

OL–2.5–R1.6 data format descriptions

Priority: 2

Grade: A

Summary: There is an updated description of the MeasurementSet definition V2.0 in Notes tree which is fairly comprehensive and sufficiently detailed for developer use.

OL–2.5–R1.7 algorithm descriptions

Priority: 2

Grade: A/E

Summary: There are some descriptions of what the tools do in the Users Reference Manual and in the Volume 1 - Basic Tools and Volume 2 - Generic Processing of Getting Results, but the algorithm descriptions are of variable quality. This could be improved.

OL–2.5–R1.8 newsletters, email exploders, notes series

Priority: 3

Note: these would be maintained by the Package providers, with help from the ALMA project.

Grade: A

Summary: There are newsletters, memo and notes series, and other docs available under the Learn More tree. Extensive email reflectors are maintained and are available under the Contact Us link.

OL–2.5–R2 The Package creators must ensure the documentation is up-to-date and complete for all parts of the Package.

Priority: 1

Grade: I Severity: high

Summary: There are still significant gaps in both quality and comprehensiveness of the documentation (particularly in the Users Reference Manual). This should be a high priority area as AIPS++ reaches maturity. Documentation will be an even higher priority with ALMA, as we hope to bring in users who have less experience with radio interferometry packages.

OL–2.5–R3 Help materials shall also be available in printable formats, including

OL–2.5–R3.1 standard document formats (pdf, postscript)

Priority: 1
Grade: I Severity: medium
Summary: Postscript is available for all documentation. No PDF is available.

OL–2.5–R3.2 printer-friendly versions of HTML pages

Priority: 2
Grade: A
Summary: The HTML pages print out reasonably well, though the quality seems to be mostly a function of the browser.

OL–2.5–R3.3 popular proprietary formats (MS-Word)

Priority: 3
Grade: N Severity: low
Summary: not available, but very low priority.

OL–2.5–R4 Help shall be context-sensitive where relevant. In GUI mode, fly-over banners should indicate use of buttons and fields, and clickable help buttons should be available on all pages.

Priority: 2

Grade: A

Summary: There are fly-over pop up help text in the GUI, as well as ways to be directed to web pages (see next item).

OL–2.5–R5 In GUI mode, help functions may direct a browser to a Web page. In CLI mode, the Package must support in-line text based help also.

Priority: 2

Grade: A/E

Summary: The GUI has a *Function help* button which drives a browser to the relevant page in the Users Reference Manual. For CLI mode (glish level) there is the `help` function. Note however that the textual help is minimal, consisting of lists of functions for
- `help('general.images.image')`
and a usage parameter list with no description for
- `help('general.images.image.set.function')`
which may be inadequate for a user. Probably will not be used much, but could be improved.

OL–2.5–R6 Full search capability must be built into the documentation library.

Priority: 2

Grade: A

Summary: The Search link in the documentation tree seems to work well.

2.3 Data Handling

2.3.1 General Data Requirements

OL–3.1–R1 The Package must support data taken in any of the standard ALMA hardware modes.

Priority: 1

Grade: U

Summary: The standard ALMA hardware modes are undefined at this time.

OL–3.1–R2 The Package shall be able to handle the integrated data objects corresponding to the observational programs carried out by ALMA. These objects may be implemented in any manner appropriate, though relations between the components of the object must be maintained through some mechanism. These include:

OL–3.1–R2.1 Program header information;

Priority: 1

Grade: A

Summary: The ms table system has provision for storing header information in special tables, e.g. the `OBSERVATION` and `HISTORY` tables.

OL–3.1–R2.2 Observation status information (and schedules themselves);

Priority: 1

Grade: A

Summary: These would be included in the `OBSERVATION` table using the `SCHEDULE_TYPE` keyword and the `SCHEDULE` column.

OL–3.1–R2.3 Field information;

Priority: 1

Grade: A

Summary: There is a `FIELD` table in the ms.

OL–3.1–R2.4 Data from interferometer and single-dish organized by:

OL–3.1–R2.4.1 position;

Priority: 1

Grade: A

Summary: The `FIELD_ID` column in the main table of the ms tracks the position. There is provision for moving sources.

OL–3.1–R2.4.2 subreflector state;

Priority: 1

Grade: A

Summary: This could be done in the `STATE_ID` column ms, with definitions in the `STATE` table. There are other possible ways of doing this, I am not sure how this is done with GBT data.

OL–3.1–R2.4.3 polarization products;

Priority: 1

Grade: A

Summary: There is a `POLARIZATION` table in the ms which describes what the correlation products are in the data.

OL–3.1–R2.4.4 spectral channels;

Priority: 1

Grade: A

Summary: The channel information is stored in the `SPECTRAL_WINDOW` table in the ms, each visibility in the ms has N channels.

OL–3.1–R2.4.5 frequency bands;

Priority: 1

Grade: A

Summary: These are specified in the SPECTRAL_WINDOW table, e.g. as frequency groups.

OL–3.1–R2.4.6 IFs;

Priority: 1

Grade: A

Summary: There is a SPECTRAL_WINDOW table in the ms which describes what the IFs are.

OL–3.1–R2.4.7 subarray;

Priority: 1

Grade: A

Summary: There is an ARRAY_ID column in the ms corresponding to subarrays defined in the ARRAY table.

OL–3.1–R2.5 Coherence function (visibility) data from interferometer, including:

OL–3.1–R2.5.1 cross-correlations;

Priority: 1

Grade: A

OL–3.1–R2.5.2 auto-correlations;

Priority: 1

Grade: A

Summary: These are there by default, with (columns) ANTENNA1 equal to ANTENNA2.

OL–3.1–R2.5.3 uncorrected and/or online WVR corrected (if chosen by user);

Priority: 1

Grade: A

Summary: There could in principle be a WVR_CORRECTION flag with a corresponding column in the data, or a separate column for corrected data.

OL–3.1–R2.5.4 phased-array data (if chosen by user);

Priority: 1

Grade: A

Summary: Can be treated as another (sub)array or telescope. Explicitly mentioned in the MS memos (used e.g. for WSRT and VLA as VLBI element).

OL–3.1–R2.6 Total power (autocorrelation) data from antennas in single-dish mode;

Priority: 1

Grade: A

Summary: The autcorrelations are handled by setting (columns) ANTENNA1 equal to ANTENNA2 (see above).

OL–3.1–R2.7 Weights and/or data uncertainties;

Priority: 1

Grade: A

Summary: There are WEIGHT and SIGMA columns in the ms. Later processing can add columns such as IMAGING_WEIGHT.

OL–3.1–R2.8 States indicating special modes (such as ON/OFF positions), for:

OL–3.1–R2.8.1 OTF scanning;

Priority: 1

Grade: A

Summary: This is in the STATE table.

OL–3.1–R2.8.2 subreflector switching;

Priority: 1

Grade: A

Summary: This is in the STATE table.

OL–3.1–R2.8.3 frequency switching;

Priority: 1

Grade: A

Summary: We presume that this would also go into the `STATE` table, probably with a new set of columns and ids for frequency switching, or could use the `FREQ_OFFSET` table. Used successfully for GBT in *dish*.

OL–3.1–R2.9 Flagging data or masks;

Priority: 1

Grade: A

Summary: There are `FLAG`, `FLAG_CATEGORY` and `FLAG_ROW` columns in the data. There is also a `FLAG_CMD` table presumably as a record of flagging.

OL–3.1–R2.10 Diagnostic data and errors;

Priority: 1

Grade: A

Summary: System temperatures are in the `SYSCAL` table. Weather information is in the `WEATHER` table. ALMA specific M&C data would probably go into a special purpose table.

OL–3.1–R2.11 A priori calibration data, including:

OL–3.1–R2.11.1 bandpasses;

Priority: 1

Grade: A

Summary: This is currently in a separate calibration table not directly associated with the ms (Jones matrix type **B**). It would be possible to link or add the table to the ms.

OL–3.1–R2.11.2 source flux densities;

Priority: 1

Grade: A

Summary: There is a `SOURCE` table though it currently does not contain columns for flux density or spectral info (as in the FITS SU table). At this time, source flux densities are included as part of the model that fills the `MODEL_DATA` column of the main ms table (computed in the `imager.setjy` function).

OL–3.1–R2.11.3 antenna polarization leakages;

Priority: 1

Grade: A

Summary: This is currently in a separate calibration table not directly associated with the ms (Jones matrix type **D**). It would be possible to link or add the table to the ms.

OL–3.1–R2.12 Derived calibration data, including:

OL–3.1–R2.12.1 gain tables;

Priority: 1

Grade: A

Summary: This is currently in a separate calibration table not directly associated with the ms (Jones matrix type **G** or **T**).

OL–3.1–R2.12.2 flux bootstraps;

Priority: 1

Grade: A

Summary: At this time, derived flux densities are included as part of the model that fills the `MODEL_DATA` column of the main ms table (computed in the `calibrater.fluxscale` function).

OL–3.1–R2.13 Images and/or models produced from data;

Priority: 1

Grade: A

Summary: These currently would be separate files, though links to these could be entered into a table attached to the ms.

OL–3.1–R2.14 Processing history;

Priority: 1

Grade: A

Summary: There is a `HISTORY` table in the ms, so the mechanism is there. However, it currently does not get filled in with the AIPS++ processing history info. It seems to contain only the AIPS history if you used `fitstoms` to read in an AIPS FITS file. See OL–3.1–R17.

Summary: See the MeasurementSet definition V2.0 document for details on the ms. We find the ms format to be general enough to cover all proposed instances, though it may end up bloated if we ask for too much. Therefore, we grade all of the above as *Adequate* even though we arenot sure of how exactly they will be finally implemented. Note that description of the calibration table formats is given in Calibration table definition version 2.0.

OL–3.1–R3 There must be a selection mechanism integrated within tools to choose between the various available data subsets such as:

OL–3.1–R3.1 sourcenames, and user specified data qualifiers (with wildcarding);
Priority: 1
Grade: I Severity: med
Summary: These are currently specified by `fieldid` parameters, or `FIELD_ID` keywords in queries. Some tools allow specification by source name (e.g. '0137+331') — this should be possible in all cases. Wildcarding would work on (probably complicated) queries using TaQL, but should be built into the more general source name specification.

OL–3.1–R3.2 mosaic or scanning pointing centers;
Priority: 1
Grade: A
Summary: These are specified by `fieldid` parameters, or `FIELD_ID` keywords in queries.

OL–3.1–R3.3 polarization products or channels;
Priority: 1
Grade: A
Summary: These are specified by `POLARIZATION` ids (e.g. LL,RL,LR,LL) in most tools.

OL–3.1–R3.4 bands (frequency bands, IFs);
Priority: 1
Grade: A
Summary: The IFs are usually specified by the `spwid` numbers in most tools, or using TaQL strings on `SPECTRAL_WINDOW_ID`. I would guess frequency bands would be specified as a frequency group with its own spwid or `SPECTRAL_WINDOW_ID`.

OL–3.1–R3.5 spectral channels;
Priority: 1
Grade: A
Summary: The channels are not always set in the same way, usually through `nchan`, `start`, and `step` parameters in the selecting function, but also sometimes as a vector of channel numbers (e.g. in the `flagger.setchan` function). You can also select by velocity instead of channel in many cases.

OL–3.1–R3.6 frequency-switched data;
Priority: 1
Grade: A
Summary: Handled by keywords (in the `STATE` table?) for GBT processing.

OL–3.1–R3.7 interferometer: subarrays;
Priority: 1
Grade: A
Summary: This would be specified with the `arrayid` parameter or `ARRAY_ID` keyword in a query.

OL–3.1–R3.8 interferometer: WVR-corrected or uncorrected baselines;
Priority: 1
Grade: A

Summary: As of the daily release 04Sep02 (v1.8 build 173) there is now an `ALMA` package with the `iramcalibrater` tool, which has a function `iramcalibrater.phcor` to do this.

OL–3.1–R4 Multiple pointing centers for mosaics must be supported.

Priority: 1

Grade: A

OL–3.1–R5 For polarization products, transformation must be provided to the desired Stokes output parameter(s).

Priority: 1

Grade: A

Summary: This is available in the context of imaging.

OL–3.1–R6 Averaging of data over time, bands and spectral channels shall be possible.

Priority: 1

Grade: I Severity: med

Summary: There are parameters that control the averaging of data over time and frequency for imaging and calibration purposes, but this requirement mostly refers to the possibility of averaging upon reading in of data or making an ms containing averaged data. There does not appeare to be a simple way to do this.

OL–3.1–R7 Data taken in arbitrary (but parameterized) scanning patterns must be dealt with.

Priority: 1

Grade: A

Summary: While it depends on what one means by "arbitrary", given that the system can deal with polynomial motions (per axis), this requirement is probably fulfilled.

OL–3.1–R8 All standard time systems shall be supported, including:

OL–3.1–R8.1 Universal Time (UT), also UT1;
Priority: 1
Grade: A

OL–3.1–R8.2 Coordinated Universal Time (UTC);
Priority: 1
Grade: A

OL–3.1–R8.3 International Atomic Time (IAT);
Priority: 1
Grade: A

OL–3.1–R8.4 Local Sidereal Time (LST);
Priority: 1
Grade: A

OL–3.1–R8.5 Greenwich Mean Sidereal Time (GMST);
Priority: 1
Grade: A

OL–3.1–R8.6 Julian Date (JD), also Modified Julian Date (MJD);
Priority: 1
Grade: A

OL–3.1–R8.7 Dynamical Times (TDT, TDB).
Priority: 1
Grade: A

Summary: The tools (e.g. the `measures.epoch` function) support the following epochs: UTC, TAI, LAST, LMST, GMST1, GAST, UT1, UT2, TDT, TCG, TDB, TCB. MJD (and JD) is supported as a time format.

OL–3.1–R9 All standard coordinate systems shall be supported, including:

> **OL–3.1–R9.1** equatorial (RA, DEC);
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R9.2** ecliptic (ELON, ELAT);
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R9.3** helioecliptic (HLON, HLAT);
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R9.4** galactic (GLON, GLAT);
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R9.5** supergalactic (SLON, SLAT);
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R9.6** terrestrial (AZ, EL);
> *Priority: 1*
> *Grade: A*

Summary: Allowed directions include J2000, JMEAN, JTRUE, APP, B1950, BMEAN, BTRUE, GALACTIC, HADEC, AZEL, SUPERGAL, ECLIPTIC, MECLIPTIC, TECLIPTIC, MERCURY, VENUS, MARS, JUPITER, SATURN, URANUS, NEPTUNE, PLUTO, MOON, SUN, COMET. Allowed epochs are J2000, B1950, GALACTIC, SUPERGAL, ECLIPTIC, with positions HADEC, AZEL.

OL–3.1–R10 All standard coordinate reference frames and equinoxes shall be supported, including:

> **OL–3.1–R10.1** J2000 (and other FK5 equinoxes);
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R10.2** B1950 (and other FK4 equinoxes);
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R10.3** geocentric apparent place, coordinates of date;
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R10.4** topocentric coordinates;
> *Priority: 1*
> *Grade: A*
> **OL–3.1–R10.5** International Celestial Reference System (ICRS);
> *Priority: 1*
> *Grade: A*

Summary: Frames for directions include J2000, JMEAN, JTRUE, APP, B1950, BMEAN, BTRUE, GALACTIC, HADEC, AZEL, AZELSW, AZELNE, AZELGEO, AZELSWGEO, AZELNEGEO, JNAT, ECLIPTIC, MECLIPTIC, TECLIPTIC, SUPERGAL, ITRF, TOPO plus planets etc. (obtained using `dm.listcodes(dm.direction())`).

OL–3.1–R11 All standard velocity definitions shall be supported, including:

> **OL–3.1–R11.1** radio;
> *Priority: 1*
> *Grade: A*

OL–3.1–R11.2 optical;
  Priority: 1
  Grade: A
OL–3.1–R11.3 redshift;
  Priority: 1
  Grade: A

Summary: The doppler frames RADIO, OPTICAL, Z, RATIO, RELATIVISTIC, BETA, GAMMA are supported (e.g. in the `measures.doppler` function).

OL–3.1–R12 All standard velocity frames shall be supported, including:

OL–3.1–R12.1 topocentric;
  Priority: 1
  Grade: A
OL–3.1–R12.2 geocentric;
  Priority: 1
  Grade: A
OL–3.1–R12.3 barycentric;
  Priority: 1
  Grade: A
OL–3.1–R12.4 heliocentric;
  Priority: 1
  Grade: A
OL–3.1–R12.5 kinematic LSR;
  Priority: 1
  Grade: A
OL–3.1–R12.6 dynamic LSR;
  Priority: 1
  Grade: A
OL–3.1–R12.7 galactocentric;
  Priority: 1
  Grade: A
OL–3.1–R12.8 local group;
  Priority: 1
  Grade: N Severity: low

Summary: The velocity frames LSRK, LSRD, BARY, GEO, TOPO, GALACTO (`measures.radialvelocity`) are supported (obtained using `dm.listcodes(dm.radialvelocity())`). This seems to cover all but possibly Local Group.

OL–3.1–R13 Coordinates and locations (e.g. of the antennas, or the array center) defined with respect to the standard frames shall be supported, including:

OL–3.1–R13.1 topocentric;
  Priority: 1
  Grade: A
OL–3.1–R13.2 geocentric;
  Priority: 1
  Grade: A

Summary: Frames for positions include the the WGS84 and ITRF formats (obtained via `dm.listcodes(dm.position())`).

OL–3.1–R14 Any existing flagging mask or table must be maintained and associated with the data it refers to during any subsequent operations (such as splitting of data sets).

Priority: 1

Grade: A

Summary: Currently, flags are applied in columns of the ms, though there are flag tables generated by some tools (e.g. `msplot`), and thus are preserved.

OL–3.1–R14.1 The flagging mask or table shall be transferable to other equivalent data sets (e.g. flags derived for a continuum dataset should be transferable to a line dataset derived from the same observations).
Priority: 2

Grade: I Severity: med

Summary: It is not possible to transfer flags between ms (except perhaps for identical ms, and possibly exactly isomorphic ms), but there are fixes in the works to allow more flexible scoping of flags in the editing tools (e.g. `msplot`).

OL–3.1–R15 Calibration and ancillary monitoring data (e.g. weather information, WVR data, pointing) that are provided in standard ALMA formats must be preserved, if requested by the user.

Priority: 1

Grade: U

Summary: There are no ALMA standard formats yet, but there are appropriate tables and ms columns for this data to go into.

OL–3.1–R16 Correlation products accumulated at multiple bit depths (16-bit,32-bit) or compressed data must be supported transparently (see ALMA-SW Memo 11 3.2.3-R5).

Priority: 1

Grade: U

Summary: It is not clear whether this is possible — we suspect not (other than having the filler write everything to maximum 32-bit depth). The intent of this requirement is that compression be applied on read-in and maintained. In any event, this will have to await the ALMA filler.

OL–3.1–R17 Comprehensive and understandable processing history information for the data must be maintained and be exportable (both as tables and as plain text).

Priority: 1

Grade: I Severity: med

Summary: There is a `HISTORY` table in the ms, however it currently contains only the contents of the AIPS history file if you use `fitstoms` to read in an AIPS FITS file. The contents of this table can be printed to an ASCII file using `table.toascii` though better control of the format would be nice. Currently one would have to rely on the *logger* output for history.

OL–3.1–R18 Users shall have access to, and the ability to change, all aspects of the data including the header.

Priority: 1

Grade: A

Summary: Full access to all aspects of the ms is available, and relatively straightforward, in AIPS++.

OL–3.1–R19 The Package must support locking data files so that there is no possibility of one process corrupting a file that is also being written to by another process in the Package. The default model should be: "one writer, multiple readers."

Priority: 1

Grade: A

Summary: Table locking is implemented in AIPS++.

OL–3.1–R20 Merging (e.g. concatenation) and splitting of datasets shall supported:

> **OL–3.1–R20.1** Extraction of specified subsets of data (e.g. by source, time, subarray) shall be supported.
> *Priority: 1*
> *Grade: A/E*
> *Summary:* It appears that you can do this with the `ms.command` function which makes a 'new' ms that is referenced to an old ms (so you cannot delete the old one). You might be able to use some `table` commands to then copy this (e.g. `table.copy`). It should be made simpler.
>
> **OL–3.1–R20.2** Merging and reinsertion of data subsets (e.g. combination of different configurations, different epochs, mosaic pointings) shall be supported.
> *Priority: 2*
> *Grade: A/E*
> *Summary:* Concatenation passing through uvfits is possible using `ms.concatenate` (see next subreq). It should be made more widely applicable.
>
> **OL–3.1–R20.3** The merging and splitting process, including selection of data to be merged (e.g. merging of frequency channels which may be differently labeled) shall be straightforward and not overly complex.
> *Priority: 2*
> *Grade: I Severity:* high
> *Summary:* Concatenation of ms is still difficult (`ms.concatenate` must go through FITS first). This needs improvement (and is on the development list).
>
> **OL–3.1–R20.4** The use of merged or split data in the Package shall be robust and transparent to the user. Subsequent operations shall work the same, whether the data is in its original form or built from merged data subsets, where possible.
> *Priority: 2*
> *Grade: A*
> *Summary:* This appears to be true, as long as it is a valid ms.
>
> **OL–3.1–R20.5** The appropriate calibration and ancillary monitoring data for the merged or split data (e.g. keeping only the data relevant to sources split out) shall be preserved.
> *Priority: 3*
> *Grade: I Severity:* low
> *Summary:* It appears that the current splitting does not deal with the subtables completely (it may copy whole rather than selecting relevant parts).

OL–3.1–R21 Distinctions between "single-source", "multi-source", single-dish, and interferometer datasets shall be avoided with context built into the dataset or header.

> *Priority: 2*
>
> *Grade: A*
>
> *Summary:* This seems to be the case for measurement sets.

OL–3.1–R22 Tools for access to header information, scan summaries (e.g. LISTR in AIPS), antenna locations (e.g. PRTAN in AIPS) shall be available, with printable output.

> *Priority: 2*
>
> *Grade: A*
>
> *Summary:* This is possible with the `ms.summary` function.

OL–3.1–R23 When sorting or indexing is needed for performance enhancement, it shall be carried out in a manner transparent to the user.

> *Priority: 3*
>
> *Grade: A*
>
> *Summary:* This seems to be the case (e.g. in `imager`).

OL–3.1–R24 Compression of the data, with a selectable level of loss, shall be possible at various stages of the processing path.

Priority: 3

Grade: I Severity: low

Summary: Some compression, such as scratch column compression in `imager`, is available this release. However, it does not appear that user controlled compression is fully available after filling. There should be a way to generally compress columns (e.g. in `ms`) later on. See also OL–3.1–R16 and OL–3.2–R5.

2.3.2   Data Import and Export

OL–3.2–R1 A variety of data formats must be supported by the Package:

> **OL–3.2–R1.1** The **ALMA standard archival data format** must be supported for both input and output without loss of functionality or information.
> *Priority: 1*
> *Grade: U*
> *Summary:* These have not yet been designated.

> **OL–3.2–R1.2** Other standard formats **as designated by the ALMA Project** shall also be supported for both input and output without loss of functionality or information.
> *Priority: 1*
> *Grade: U*
> *Summary:* These have not yet been designated, though it is likely that these will consist of FITS (e.g. FITS-IDI) and measurement sets, both of which are currently supported by AIPS++. Note ALMATI-FITS should be supported for the test data.

> **OL–3.2–R1.3** Other popular data formats may be supported as the Package sees fit, possibly with some loss in information content. Flexibility and wider use beyond ALMA is to be encouraged.
> *Priority: 3*
> *Grade: A*
> *Summary:* There are fillers for most popular formats used by AIPS++ consortium members. We need not specify beyond this.

OL–3.2–R2 Disk and offline data storage (e.g. DAT, DDS, DLT) must be supported. **The project will maintain a list of media which the Package must support.**

Priority: 1

Grade: A

Summary: There is not yet a designated list from ALMA, but AIPS++ currently supports tape devices, and it is probably adequate to buffer to disk and use system I/O to devices when this does not work.

OL–3.2–R3 The Package must be able to handle, efficiently and gracefully, datasets larger than main memory of the host system.

Priority: 1

Grade: U

Summary: We will require a very large dataset to benchmark this, though part of the test would be to try making a very large many-faceted image cube.

OL–3.2–R4 The ability to drop flagged data on export shall be included.

Priority: 2

Grade: I Severity: low

Summary: It does not appear there is a way to throw away flagged data, the flags are carried along.

OL–3.2–R5 The Package internal data format, which may be independent of other supported formats, must not be "bloated" and the required storage should not exceed by more than 1.5× the raw data format.

Priority: 2

Grade: I Severity: med

Summary: This is dataset dependent — it appears that for continuum data the ms to fits filesize ratio is significantly worse: 1.6 and greater for the raw data versus 1.1 or so for line data. However, when the MODEL_DATA, CORRECTED_DATA, and IMAGING_WEIGHT scratch columns are added by imagr this jumps to 2.4 for continuum or 1.8 for line data with scratch column compression turned on in imagr, and 2.6 or more without compression in both cases. Since ALMA will operate mostly in line mode (even continuum data will have multiple channels), we can expect to get a bloat factor of about 1.8 with compression. This is only slightly above our limit of 1.5 and thus may be acceptable as is. Compression of other columns or tables would probably easily bring this in line. The capability is built into low-level routines but has not yet made its way up to the user visible level (e.g. in vlafiller or fitstoms. One issue that must be addressed is whether this level of compression will compromise the subsequent data processing (like it can in AIPS with weights).

2.3.3 Images and Other Data Products

OL–3.3–R1 Standard multi-dimensional images and arrays must be supported, including:

> **OL–3.3–R1.1** Spectra and image slices (1D);
> *Priority: 1*
> *Grade: A*
> **OL–3.3–R1.2** Planar images (2D);
> *Priority: 1*
> *Grade: A*
> **OL–3.3–R1.3** Spectral and Time Cubes (3D);
> *Priority: 1*
> *Grade: A*
> **OL–3.3–R1.4** Higher-dimensional Arrays (4D+).
> *Priority: 2*
> *Grade: A*

Summary: AIPS++ images, e.g. as supported in the images module, can have arbitrary dimensionality or shape. I/O to and from FITS, ASCII, and the AIPS++ image formats is provided.

OL–3.3–R2 Other standard derived data products must be supported, including:

> **OL–3.3–R2.1** Point (CLEAN) models;
> *Priority: 1*
> *Grade: A*
> **OL–3.3–R2.2** Pixel (gridded image) models;
> *Priority: 1*
> *Grade: A*
> **OL–3.3–R2.3** Elliptical Gaussian models;
> *Priority: 1*
> *Grade: A*
> **OL–3.3–R2.4** Uniform (optically thick) disk models;
> *Priority: 2*
> *Grade: A*
> **OL–3.3–R2.5** Optically thin disk models;
> *Priority: 2*

Grade: N Severity: low

OL–3.3–R2.6 wavelets;

    Priority: 3

    Grade: N Severity: low

OL–3.3–R2.7 Pixons;

    Priority: 3

    Grade: I Severity: low

Summary: Pixel images are supported in the `images` module. A number of component types are supported in the `componentmodels` module, which are manipulated through the `componentlist` tool. Currently, three shapes are supported: point, Gaussian or disk. Only uniform disks are available as far as I can tell.

Pixon imaging, as described in AIPS++ Note 242 is available in the `imager.pixon` function. General support for pixon components etc. beyond image deconvolution is sketchy. This is currently under development, and should be viewed as an area for ongoing experimentation.

OL–3.3–R3 Blanking of pixels shall be maintained through the processing of images.

Priority: 1

Grade: A

Summary: A *pixel mask* is included as part of an image, see the documentation for the `image` tool.

OL–3.3–R4 The package shall support all standard projections, including:

    OL–3.3–R4.1 sine or slant orthographic (SIN);

        Priority: 2

        Grade: A

    OL–3.3–R4.2 tangent or gnomonic (TAN);

        Priority: 2

        Grade: A

    OL–3.3–R4.3 cylindrical plate carrée (CAR);

        Priority: 2

        Grade: A

    OL–3.3–R4.4 Mercator (MER);

        Priority: 2

        Grade: A

    OL–3.3–R4.5 stereographic (STG);

        Priority: 2

        Grade: A

    OL–3.3–R4.6 Hammer-Aitoff (AIT).

        Priority: 2

        Grade: A

Summary: The support for coordinate systems in the `images` module is extensive and compliant with the WCS standards. For example, see the documentation for the `coordsys` tool for details.

2.3.4 Foreign Data

OL–3.4–R1 Data produced by other interferometers and single dishes in similar observing modes shall be importable and processable if provided in **ALMA standard archival data format** or an **ALMA supported data format**

Priority: 2

Grade: U

Summary: ALMA format currently undefined. A substantial set of fillers for different telescopes is currently available.

OL–3.4–R2 Imaging data in standard formats (e.g. FITS) from astronomical instruments at different wavelengths shall be importable, with the ability to combine (coadd) these with ALMA data where appropriate. This should be through a set of widely used formats, with a **minimal list of supported standards established by the project**.

Priority: 2

Grade: U

Summary: List of standards not available. FITS is currently supported.

2.3.5   Interaction with the Archive

OL–3.5–R1 Access from the archive (when such access is granted, e.g. when Package is run by ALMA staff) must be supported.

Priority: 1

Grade: U

Summary: Archive not available

OL–3.5–R2 The interface between the Package and archive must be able to provide data access (when such access is granted) without interfering with other access to the archive.

Priority: 1

Grade: U

Summary: Archive not available

OL–3.5–R3 Security and integrity of the archive must be ensured during these operations.

Priority: 1

Grade: U

Summary: Archive not available

2.4   Calibration and Editing

2.4.1   General Calibration and Editing Requirements

OL–4.1–R1 The Package must be able to handle reliably all **designated ALMA standard calibration modes** , possibly including but not exclusive to: temperature controlled loads, semi-transparent vanes, apex calibration systems, WVR data, noise injection, fast-switching calibration transfer, planetary observations.

Priority: 1

Grade: U

Summary: These have not been defined yet. This item will require input from the ALMA Science IPT and the SSR.

OL–4.1–R2 Calibration, editing, flagging, and correction of data shall be easily reversible within the Package (i.e. not requiring re-reading of the data from the archive).

Priority: 1

Grade: A

Summary: Calibration steps affect only scratch columns in the ms, or create new tables which can be reused. Combined with the *logger* and *scripter*, it is usually easy to reverse calibration and editing steps (barring corruption of the ms).

> **OL–4.1–R2.1** Logging of editing steps will be clearly marked in a history table or data object (possibly distinct from a more readable history).
> *Priority: 2*
> *Grade: I Severity:* med
> *Summary:* The `HISTORY` table is not currently filled by AIPS++ processing (see OL–3.1–R17), though some of this information can be found in the *logger*.

> **OL–4.1–R2.2** Individual edit undo is desirable.
> *Priority: 3*
> *Grade: I Severity:* low
> *Summary:* This is not generally available, though some individual edit undo is available in the *viewer* editing of ms introduced in v1.8 after the July2002 release (e.g. 1.8 build 109 2002-08-09). Note that this is priority 3, and only marked as desirable, and likely has cost impact beyond its utility. However, some ability to at least undo flags entered at a given stage, presuming they can be differentiated (e.g. in a different flag table, or having a flag code) would be extremely useful.

OL–4.1–R3 Data calibration operations shall take into account the scan structure and switching scheme of the data. The user shall be able to request calibration solution intervals that correspond to and reference from scan boundaries, for example.

Priority: 1

Grade: N Severity: high

Summary: This is currently missing from the package, as the solution intervals are locked to an absolute time grid. The ability to carry out solutions on a scan by scan basis will be in place in the next release (and is available already in daily builds after Sep 2002). The ability to count the solution intervals from scan boundaries is in development.

OL–4.1–R4 Data editing and flagging shall be possible based upon array, environmental, astronomical, and calibration monitoring data, including:

> **OL–4.1–R4.1** pointing data;
> *Priority: 1*
> *Grade: N Severity:* med

OL–4.1–R4.2 array tracking information (encoders);
  Priority: 1
  Grade: N Severity: med
OL–4.1–R4.3 weather data (wind speed, temperature, relative humidity, pressure);
  Priority: 1
  Grade: I Severity: med
OL–4.1–R4.4 T_{sys} data;
  Priority: 1
  Grade: I Severity: med
OL–4.1–R4.5 WVR data;
  Priority: 1
  Grade: N Severity: med
OL–4.1–R4.6 RFI monitoring;
  Priority: 2
  Grade: N Severity: low
OL–4.1–R4.7 site-test interferometer (STI) and/or tipping radiometer;
  Priority: 2
  Grade: N Severity: low
OL–4.1–R4.8 array monitoring points (e.g. dewar temperatures) if data provided in standard format;
  Priority: 3
  Grade: N Severity: low
OL–4.1–R4.9 other site instrument (e.g. FTS) providing data in standard format.
  Priority: 3
  Grade: N Severity: low

Summary: These are largely missing from the package. There is provision for T_{sys} data in the `SYSCAL` table and weather in the `WEATHER` table which would nominally be filled in the `vlafiller` or in `fitstoms` from the AIPS `TY` and `WX` tables. These are not currently handled. The other information is not currently provided by the VLA, though developments related to the GBT and the IRAM testing will provide this capability soon.

OL–4.1–R5 Calibration shall involve flexible averaging of data and calibration quantities with user-controllable interpolation, filtering, weighting, and application scope.

Priority: 1

Grade: I Severity: high

Summary: The `calibrater.setapply` appears to have only the provision for default interpolation (two-point linear?), though this will likely change soon. The interpolation types in the AIPS task `CLCAL` (e.g. boxcar, 2pt, polynomial, median window filtering) should at the minimum be available. There is no simple way to edit the solutions before applying, though this could presumably be done using the `table` tool.

OL–4.1–R6 Interactive data editing, calibration, and display of calibration quantities shall be largely graphical and intuitive. Specialized editing display tools should include:

Summary: Interactive editing is carried out by the tool `msplot`, which does X-Y plotting and some `TVFLG`-like raster editing, and the `flagger` tool for CLI based editing. There is also *viewer* based (raster plot) editing of ms introduced in v1.8 after the July2002 release (e.g. 1.8 build 109 2002-08-09) is clearly the way of the future. This is currently undergoing testing and should be available in the next release.

OL–4.1–R6.1 specification of data by selection on observational parameters, and/or plotting versus these parameters, including:
  OL–4.1–R6.1.1 antenna;

Priority: 1
Grade: A

OL–4.1–R6.1.2 baseline;
Priority: 1
Grade: A

OL–4.1–R6.1.3 time range;
Priority: 1
Grade: A

OL–4.1–R6.1.4 uv range;
Priority: 1
Grade: A

OL–4.1–R6.1.5 pointing center;
Priority: 1
Grade: I Severity: low
Summary: Selection and plotting by `FIELD_ID`, scan number or sourcename is available in `msplot` (but not yet in the `viewer`). It is not clear you can easily edit more complicated mosaic scanning patterns easily, e.g. via the `POINTING` sub-table.

OL–4.1–R6.1.6 azimuth, elevation;
Priority: 1
Grade: N Severity: med

OL–4.1–R6.1.7 hour angle range;
Priority: 1
Grade: N Severity: med

OL–4.1–R6.1.8 parallactic angle;
Priority: 1
Grade: N Severity: med

OL–4.1–R6.1.9 slice through data cube;
Priority: 1
Grade: N Severity: low

OL–4.1–R6.2 displays of spectra and spectral cubes;
Priority: 1
Grade: I Severity: med
Summary: `msplot` can plot visibilities as a function of frequency. You can also average the data and view it using start, stop, and increment. It is not yet possible to edit based on either a single channel or averaged spectral data (i.e. CH 0) and have it applied to the whole cube, although it is being actively worked on. The `viewer` now has editing capability also.

OL–4.1–R6.3 display of closure quantities;
Priority: 1
Grade: N Severity: low
Summary: Probably mostly useful for diagnostics rather than end-use for science.

OL–4.1–R6.4 display of and selection on monitor data quantities (e.g. Tatm, Tamb);
Priority: 1
Grade: N Severity: med

OL–4.1–R6.5 the above with baseline, time, band and/or channel averaging;
Priority: 1
Grade: N Severity: med
Summary: In the standard plotting tools there is little averaging capability. Note that `msplot` allows channel averaging only, while the `viewer` does not seem to currently support any averaging for ms data (nor does it allow X-Y plots yet).

OL–4.1–R6.6 for interferometer data, amplitude (phase) vs. time on each baseline (Difmap vplot), vs. time-baseline (AIPS TVFLG) with interactive zoom, selection, and clipping;
Priority: 1

Grade: I Severity: med

Summary: X-Y plot editing is available in `msplot`, though there is still missing functionality (e.g. scope of flagging). Some `TVFLG`-like raster editing is available in `msplot`, though the *viewer* based editing of ms introduced in v1.8 after the July2002 release (e.g. 1.8 build 109) is clearly the way of the future. This is currently undergoing testing and should be available in the next release.

OL–4.1–R6.7 editing of data points based on statistical quantities, including:

OL–4.1–R6.7.1 a data point versus a running mean over a timescale;
Priority: 1
Grade: N Severity: high

OL–4.1–R6.7.2 a data point versus a median over a timescale;
Priority: 1
Grade: N Severity: low

OL–4.1–R6.7.3 an rms scatter in a time range;
Priority: 1
Grade: N Severity: high

OL–4.1–R6.7.4 difference versus a model;
Priority: 1
Grade: N Severity: med

Summary: Control of interactive editing and plotting versus statistical quantities is lacking, especially in `msplot` and the `viewer` (though the latter is in the works). However, there is comprehensive statistical flagging in the `autoflag` tool. I cannot find editing of data versus models (e.g. by dividing data by model) except if one made a custom script. Note that the first two of these are standard modes in AIPS `TVFLG`.

OL–4.1–R6.8 auto-scaling and user-specified scaling of axes;
Priority: 1
Grade: A

OL–4.1–R6.9 auto-scaled and user-specified colormap or greyscale;
Priority: 1
Grade: A

OL–4.1–R6.10 inclusion and marking of flagged data in plots and in auto-scaling;
Priority: 1
Grade: A/E
Summary: This seems pretty much there, except perhaps the ability to plot flagged data but not include in auto-scaling of axes.

Summary: There is still significant missing functionality in the interactive editing tools, but this has been identified as a key area for development in the next cycle(s) and many things on this list are likely to be checked in over the next couple of months. Some of the missing items could in principle be done by putting together some of the low-level tools making a custom script, but this would be beyond the normal user.

OL–4.1–R7 Non-interactive and automated editing tools shall be available in the Package, including:

OL–4.1–R7.1 direct editing of data based user-specified ranges for quantities available for plotting or editing in interactive mode
Priority: 1
Grade: A
Summary: This is the `flagger` tool.

OL–4.1–R7.2 automated editing with tunable criteria for automated selection of parameter ranges
Priority: 2
Grade: A
Summary: This is the `autoflag` tool.

OL–4.1–R8 Data calibration, correction and flagging shall be possible based upon standard or user-defined models in either functional or tabular form. Arbitrary user-specified scaling of data shall be simple.

Priority: 2

Grade: N Severity: high

Summary: Editing based on data versus models (not just editing based in the MODEL_DATA column) seems to be missing from all the tools.

OL–4.1–R9 Access to time history of calibration information (e.g. source catalogs containing flux density histories) shall be built into calibration engines. Output of calibration procedures shall be exportable into similar structures.

Priority: 2

Grade: I Severity: low

Summary: Currently, only the Baars, Perley90, Perley-Taylor95, and Perley-Taylor99 scales for 3C286, 3C48, 3C147, 3C138, and 1934-638 (without polarization) are supported for automatic calculation in the imager.setjy tool. The user would have to enter others by hand or write a script.

OL–4.1–R10 The Package shall support the establishment and verification of the relative calibration of the various component epochs, configurations or other subsets for merged datasets.

Priority: 2

Grade: I Severity: med

Summary: As mentioned elsewhere (see OL–3.1–R20) it is currently not straightfoward to concatenate two datasets. One must first read the data out of AIPS++ and then read it in again with fitstoms, afterwhich it can be concatenated. There is formalism available to handle multiple subarrays based on the msselect OBSERVATION_ID keyword. This parameter is not yet fully implemented in most tasks. The meaning of this requirement is unclear (perhaps clarification by the SSR would help) — one interpretation is that there be a function that will in some way reconcile the scaling of data subsets (e.g. epochs) perhaps by examining points nearby in the uv plane, or something similar.

OL–4.1–R11 Data display and editing shall be effected through generic tools applicable to both single-dish and interferometer modes. These shall, as far as possible, present similar interfaces to the user and have the same look-and-feel.

Priority: 2

Grade: A

Summary: Both interferometer and single dish data are dealt with through the viewer or msplot interfaces. Note that there are additional capabilities in the dish package, which has a different look-and-feel, but this is OK.

OL–4.1–R12 Editing shall be incorporated into most visualization tools where data or data-derived quantities are plotted, such as from calibration solutions, amplitude vs. uv-distance plots, or any number of other plots. A "see-it, flag-it" capability shall be the standard within the tools.

Priority: 2

Grade: I Severity: med

Summary: Plotting is done mostly in the viewer and msplot which are editors. However, editing of calibration solutions in calibrater.plotcal is missing. Custom plots (e.g. made by the user using the PGPLOT routines) are not considered here. Capability analogous to AIPS SNEDT in the calibrater.plotcal function is needed.

2.4.2   Atmospheric Calibration

OL–4.2–R1 Atmospheric modeling shall be available in the Package. **The ALMA Project shall provide a standard model as used in the Pipeline processing of data which the package must support.** There shall be provision for the use of user-supplied models, or the package may have its own built-in models as an option.

Priority: 1

Grade: N Severity: high

Summary: Atmospheric modeling is not yet available in AIPS++, though some advancements have been made during the IRAM testing (but checked in after this audit). ALMA has not defined a standard model, but Juan Pardo's code is emerging as the one to use.

OL–4.2–R2 The Package shall be able to predict the absorption, emission and path length on the line of sight through the atmosphere at all ALMA bands using the model. The prediction will be based on the following data:

    OL–4.2–R2.1 measured atmospheric parameters at the site: temperature, pressure, humidity
    Priority: 1
    Grade: N Severity: high
    Summary: Not available, but should be do-able given that there are slots in the `WEATHER` table, and VLA weather data exists for testing (and given current IRAM testing).

    OL–4.2–R2.2 measured atmospheric emission in the observed ALMA bands
    Priority: 1
    Grade: N Severity: med
    Summary: The ALMA bands are currently known.

    OL–4.2–R2.3 data from site test interferometer or tipping radiometer
    Priority: 2
    Grade: N Severity: low
    Summary: There are STI's at the ALMA site and the VLA, so data (and formats) are available.

    OL–4.2–R2.4 measured FTS data (if FTS available for ALMA)
    Priority: 3
    Grade: U
    Summary: Pending ALMA FTS and format

    OL–4.2–R2.5 measured atmospheric profiles of temperature and water content if available from atmospheric sounders or other devices
    Priority: 3
    Grade: U
    Summary: Pending ALMA scoping of this

OL–4.2–R3 Atmospheric modeling shall be usable to derive the system temperatures corrected for atmospheric absorption in all astronomical bands in use, in order to correct the observed amplitudes at various elevations.

Priority: 1

Grade: N Severity: high

OL–4.2–R4 Atmospheric modeling shall be also usable to provide the conversion factors between WVR data and the water contribution to the astronomical phase in the astronomical bands.

Priority: 1

Grade: N Severity: high

Summary: Note: this will be necessary for dealing with VLA WVR data also (though at 22 GHz instead of 183 GHz).

2.4.3   Interferometer Data

Note: Antenna-based determination of calibration quantities such as gains, polarization leakages, bandpasses, will be the primary form of calibration where appropriate. However, in addition to antenna-based calibration, baseline dependent corrections will be important in some cases. For example, coherence loss due to atmospheric phase fluctuation depends on baseline length (this aspect will be more important at higher frequencies) and orientation, and must be taken into account if some of the WVR corrections are discarded while others are applied. Also, in general, the bandpasses are baseline dependent and contain non-closing terms.

OL–4.3–R1 Antenna-based determination of calibration quantities shall be available, and are the default choice for calibration in tools where appropriate, for quantities including:

> **OL–4.3–R1.1** antenna gains;
> *Priority: 1*
> *Grade: A*
> **OL–4.3–R1.2** polarization leakages;
> *Priority: 1*
> *Grade: A*
> **OL–4.3–R1.3** antenna-dependent bandpasses;
> *Priority: 1*
> *Grade: A*

Summary: These are all standard in AIPS++. Note that mode useful for mm wave (i.e. low SNR) data was introduced (antenna-based gains modelled as cubic splines and polynomials) for the IRAM PdB tests (available in `iramcalibrater`) as of daily v1.8 build 173.

OL–4.3–R2 Baseline dependent corrections shall be supported for quantities including:

> **OL–4.3–R2.1** WVR corrections to subsets of baselines;
> *Priority: 1*
> *Grade: A*
> *Summary:* Now available in the `iramcalibrater` package as of daily v1.8 build 173, though this relies upon having the corrected and uncorrected data available (since the WVR corrections need to be applied online before any averaging).
> **OL–4.3–R2.2** closure errors;
> *Priority: 1*
> *Grade: N Severity:* med
> *Summary:* The machinery for doing closure and baseline-dependent corrections is in the measurement equation, but do not currently appear to be do-able from high-level tools.
> **OL–4.3–R2.3** baseline-dependent bandpasses;
> *Priority: 1*
> *Grade: N Severity:* med
> *Summary:* The machinery for doing closure and baseline-dependent corrections is in the measurement equation, but do not currently appear to be do-able from high-level tools.

OL–4.3–R3 Gain corrections will be made based on differences between observed and modeled data quantities, possibly with iteration (e.g. self-calibration and determination of gains using calibration sources). Where solutions are discrepant or poor, automatic editing shall be possible.

Priority: 1

Grade: A/E

Summary: This is the way that calibration works in AIPS++, though the user should have more control over what constitutes a bad solution and what happens when it tries to apply a table with flagged solutions. For example, a more user-friendly way of reporting progress in the calibration

solver is needed. It is currently difficult to gauge the sucess of a particular set of calibrator solutions beyond displaying the solutions themselves, ie. number of good vs. failed solutions is not reported, nor is the S/N of the fits (the reported chi-squared is difficult to interpret).

OL–4.3–R4 Calibration quantities (possibly stored in tables or data structures) shall be transferable between sources and/or frequency bands, after any necessary interpolation, extrapolation or smoothing. This will be the primary method of phase calibration transfer using fast-switching between source and calibrator.

Priority: 1

Grade: I Severity: high

Summary: Standard calibration transfer between sources works. Calibration transfer between bands is not yet checked in but is being actively worked on and should be checked in soon (IRAM work).

OL–4.3–R5 Determination of the time-variable, complex bandpass using calibration source observations, and transfer to target sources, shall be simple and robust.

Priority: 1

Grade: I Severity: low

Summary: The calibration solution interval can be set when solving for the bandpass so tracking of time variability is possible. Currently, however, only simple interpolation is possible although it is being actively worked on.

OL–4.3–R6 Determination of polarization calibration quantities such as leakage (D-term or Jones matrix) and complex gain difference shall be an integral part of the Package, with the capability of performing full matrix calculations.

Priority: 1

Grade: A

Summary: This works well in AIPS++.

OL–4.3–R7 Determination of, correction for, examination of, and flagging based on closure errors shall be straightforward to carry out.

Priority: 2

Grade: N Severity: med

Summary: These do not appear to be available in the high-level tools, e.g. `calibrater`.

OL–4.3–R8 Incorporation of standard models (e.g. planetary disks, models for HII region structure, known source spectra) shall be easy for calibration operations. User-supplied models in a standard format (e.g. ASCII table) for these quantities shall be supported.

Priority: 2

Grade: A

Summary: Most model types can be dealt with (see OL–3.3–R2), and image models can be created from images or ASCII tables.

OL–4.3–R9 Redundancy (e.g. same, similar, or crossing baselines) shall be used wherever possible to increase accuracy of or to check calibration solutions. Editing based on this comparison shall be possible.

Priority: 2

Grade: N Severity: low

Summary: This is not built into AIPS++, though `autoflag` tool can do uv-plane averaging (essentially what this is) for editing.

OL–4.3–R10 Interferometric pointing, focus, baseline, and beam response fitting shall be available in the Package as a supplement to the on-line calibration.

Priority: 3

Grade: N Severity: low

Summary: Should carry out as an enhancement, perhaps done as a custom script.

2.4.4 Single Dish Data

OL–4.4–R1 Straightforward and flexible fitting of spectral bandpass from calibration source observations is required.

Priority: 1

Grade: A

Summary: This functionality is available in `dish`.

OL–4.4–R2 De-striping and adjustment of scan normalization factors must be available for single-dish OTF observations with overlapping and crossing scans.

Priority: 1

Grade: I Severity: med

Summary: Not available as a specific function or tool, but can be done (fairly effectively, as shown in GBT images) by FFT-ing the data, filtering, and transforming back. Should be bundled into a tool, with more options (e.g. using interlocking scans to solve for scan offsets).

OL–4.4–R3 Calibration of system parameters such as temperature controlled loads and noise sources from observations of celestial sources shall be supported.

Priority: 3

Grade: N Severity: low

Summary: Not currently available in `dish` as the Tcal values are provided by the GBT online system.

OL–4.4–R4 Processing for pointing, focus, tipping, or beam-fitting data must be available for both single-scan calibration observations and for multiple datasets.

Priority: 3

Grade: N Severity: low

Summary: Mostly for staff use, and could probably be done using custom scripts running low-level functions (glish level).

2.4.5 Mosaicing Considerations

OL–4.5–R1 Determination of and correction for pointing offsets and the polarized primary beam is critical to the ability to reliably mosaic using ALMA, and thus must be available in the Package, preferably in several algorithmic forms.

Priority: 1

Grade: U

Summary: There does not appear to be a method for determination of or implementation of pointing corrections to amplitude and phase, though this could be done with special scripting. Development of mosaicing algorithms, including pointing effects, should be a high priority of the ALMA Science IPT, which can then be coded into AIPS++.

OL–4.5–R2 Careful cross calibration of the flux scales between ALMA interferometric data and single dish data is required for high fidelity imaging. There must be tools to cross-check and correct the relative calibration between mosaics and different component observations. *Note: this is particularly important and more difficult for ACA data.*

Priority: 2

Grade: U

Summary: Currently, GBT+VLA combination is not easy nor does it always give adequate results, though the new feathering and joint deconvolution algorithms that are due to be checked in might change that. Thus, further development is needed, and for ALMA's needs some work by the ALMA Science IPT should be done.

OL–4.5–R3 The Package shall allow user-input of scan information given a parameterized pattern or interpolated from a provided list of pointing centers versus time

Priority: 3

Grade: A/E

Summary: You can specify a polynomial scan pattern in the MS, though the way to do this is not totally clear (though it will be the job of the filler, so there will be some work for ALMA ahead). The exact meaning of this requirement is unclear, but it seems to be fulfilled as written.

2.4.6 Ancillary and Diagnostic Data

OL–4.6–R1 Output from the atmospheric monitoring (e.g. WVR, FTS) instrumentation provided in ALMA format shall be importable to the calibration software, for example to be used in flagging.

Priority: 1

Grade: U

Summary: An ALMA format needs to be detailed.

OL–4.6–R2 Derivation of calibration and correction of data based on pointing, focus and subreflector status information shall be available in the Package.

Priority: 1

Grade: N Severity: med

Summary: There is no easy implementation of pointing corrections, e.g. OL–4.5–R1. The ALMA Science IPT should provide guidance on just what is needed here.

OL–4.6–R3 Environmental data such as weather (e.g. wind speed, temperature, dew point) supplied in ALMA format, as FITS extension tables, or as ASCII tables shall be importable for editing or calibration procedures, and easily incorporated into user-specified calibration models.

Priority: 2

Grade: U

Summary: The ALMA formats needs to be detailed, though the current VLA and VLBA weather information (e.g. in the AIPS WX tables) should be importable into the WEATHER table well before this. Furthermore, an interface document spelling out just what info will be delivered by M and C should be provided.

OL–4.6–R4 Engineering monitoring information such as temperature sensor readings and tilt-meter outputs, included in ALMA format, as FITS extension tables, or as ASCII tables, shall be readable and incorporated into the calibration and editing process.

Priority: 3

Grade: U

Summary: The ALMA formats needs to be detailed. The EVLA M&C information will be written as measurement sets (or AIPS++ tables?), ALMA might want to do the same. Furthermore, an interface document spelling out just what info will be delivered by M and C should be provided.

2.5   Imaging

2.5.1   General Imaging Requirements

Because ALMA is inherently a multi-channel instrument, spectral cube mapping shall be built in as the primary mode from the beginning. Also, due to the high volume of data that can be produced by ALMA, it is imperative that the imaging and deconvolution tools in the Package be user-friendly, efficient, and flexible. This is the workhorse of the Package as far as most users will be concerned, and suitability and success of the Package will be judged with this in mind.

OL–5.1–R1 Imaging of data taken from any combination of ALMA exported data, the ALMA archive, or other instruments supporting common export formats must be provided. **A list of supported data and formats will be maintained by the project.**

> *Priority: 1*
>
> *Grade: U*
>
> *Summary:* ALMA must supply a list.

OL–5.1–R2 Efficient selection of subsets of the imaging data must be provided.

> *Priority: 1*
>
> *Grade: A*
>
> *Summary:* A vector of `FIELD_ID` numbers can be supplied to the `imager` tool. Also **msselect** in `imager.setdata` using a TQL string.

OL–5.1–R3 Provision must be made for the utilization and development of a variety of imaging, deconvolution, and analysis algorithms, including:

> **OL–5.1–R3.1** raw ("dirty") images with selectable weighting (natural, uniform, Briggs robust);
> *Priority: 1*
> *Grade: A*
> *Summary:* These are available in the `imager.makeimage` using the **observed** type from the `DATA` column of the ms.
>
> **OL–5.1–R3.2** residual images after model subtraction;
> *Priority: 1*
> *Grade: A*
> *Summary:* These are available in the `imager.makeimage` using the **residual** type from the `CORRECTED_DATA` and `MODEL_DATA` columns.
>
> **OL–5.1–R3.3** single-scale CLEAN (Hogbom, Clark, Cotton-Schwab);
> *Priority: 1*
> *Grade: A/E*
> *Summary:* This is carried out by the `imager.clean` function, with the **Hogbom** and **Clark** options. **Cotton-Schwab** does not appear to be available, if it is really desired it is an enhancement.
>
> **OL–5.1–R3.4** maximum entropy method (MEM);
> *Priority: 1*
> *Grade: A*
> *Summary:* This is carried out by the `imager.mem` function, which uses the Cornwell-Evans maximum entropy or maximum emptiness algorithms in the context of single field or multi-field imaging.
>
> **OL–5.1–R3.5** linear mosaics;
> *Priority: 1*
> *Grade: A*

Summary: This is carried out by the `imager.linearmosaic` function. Cleaning of the mosaic in `imager.clean` can be carried out using the **multi-field** option using Clark clean on the joint fields specified.

OL–5.1–R3.6 non-linear mosaics;

Priority: 1

Grade: A

Summary: `imager.clean` and `imager.mem` have this function.

OL–5.1–R3.7 non-negative least-squares (NNLS);

Priority: 2

Grade: A/E

Summary: This is carried out by the `imager.nnls` function, which works on stokes I. This appears it might also work in multi-field mode on a mosaic, though the documentation seems to indicate that it works on compact images only. Better explanation, at least, is desired.

OL–5.1–R3.8 multi-scale CLEAN;

Priority: 2

Grade: A/E

Summary: This is carried out by the `imager.clean` function, with the **multi-scale** option. Better control facility, and/or documentation of heuristics for choosing scale parameters, is desired.

OL–5.1–R3.9 modelfitting (point, Gaussian, disk);

Priority: 2

Grade: I Severity: high

Summary: Image-plane modelfitting is carried out by the `imagefitter` tool. There is not currently any uv-plane modelfitting available.

OL–5.1–R3.10 multi-frequency synthesis with different spectral models;

Priority: 2

Grade: I Severity: med

Summary: This is not currently available as far as I can tell, though it is on the development list for ATCA using the Sault algorithm. Dealing with pointing, polarization, instrumental changes, and source spectrum will be challenging, and this will be a research topic for ALMA Science IPT.

OL–5.1–R3.11 special function deconvolution (Pixon, wavelet);

Priority: 3

Grade: I Severity: low

Summary: Pixon deconvolution is available in `imager.pixon`, though this is currently only image-plane deconvolution. Further development in this area would be interesting. There is no wavelet deconvolution.

OL–5.1–R4 Image pixel and spectral channel blanking must be supported.

Priority: 1

Grade: A

Summary: A mask can be created or specified using the `imager.mask`, `imager.boxmask`, `imager.exprmask`, and `imager.regionmask` functions. The masks can include intermediate values between 0.0 and 1.0 for downweighting in addtion to blanking (0.0).

OL–5.1–R5 Interactive graphical selection of deconvolution region masks shall be available.

Priority: 1

Grade: A

Summary: This is done as a region in the `regionmanager` tool, e.g. invoked when running `imager.clean` setting `interactive=T`.

OL–5.1–R6 Multiple input datasets shall be supported directly in the tools, rather than requiring previous concatenation of the data.

Priority: 2

Grade: I Severity: med

Summary: This is not currently available, data must be concatenated using `ms.concatenate` (see OL–3.1–R20. This is in the development plan.

OL–5.1–R7 An integrated deconvolution, self-calibration, and editing/filtering tool shall be available, especially for novice users with data taken in commonly used modes.

Priority: 2

Grade: A/E

Summary: There is the `map` tool, which was meant as a prototype for this sort of integrated tool. There is also a `imagerwizard` function, and the `imager.selfcal` function allows a pre-constructed `calibrater` tool to be used from within `imager`. These all have deficiences in the ease-of-use department (covered under OL–2.2–R6.3) but do exist and can be made to work. Integrated tools and GUI look-and-feel issues should be the subject of a focus group from ALMA and AIPS++.

OL–5.1–R8 Images made on different coordinate systems (e.g. OL–3.1–R9), equinoxes (e.g. OL–3.1–R10), or projections (e.g. OL–3.3–R4) shall be transformed, merged and compared appropriately.

Priority: 2

Grade: A

Summary: The `viewer` deals with world coordinates transparently.

OL–5.1–R9 Image cubes using different velocity definitions (e.g. OL–3.1–R11) and velocity frames (e.g. OL–3.1–R12) shall be transformed and merged correctly.

Priority: 2

Grade: A

Summary: It is possible to select the velocity definition within the image analysis tool in the `viewer`. There are also a number of tools to change or manipulate the velocity definition permenantly in the `coordsys` tool.

2.5.2   Interferometer Imaging

OL–5.2–R1 High-fidelity imaging of the entire primary beam in all Stokes parameters is the primary goal — therefore, incorporation of the polarized primary beam response of the array is required.

Priority: 1

Grade: I Severity: high

Summary: There are some tools to create and use antenna voltage patterns and primary beams, e.g. in the `vpmanager` tool which can set patterns to be used in `imager.setvp`. As far as I can tell, these do not include full polarization, though beam squint can be specified (e.g. in `vpmanager.setcpbnumeric`). The current implementation of multi-field clean in `imager.clean` requires the primary beam to be the same in all polarizations, although the Hamaker-Bregman-Sault formalism does not require this. Note that there will have to be some algorithm development to use a polarized primary beam in imaging.

OL–5.2–R2 Imaging of direct polarization products (e.g. RR, LL, RL, LR) or Stokes polarization states (e.g. I, Q, U, V) must be selectable and interchangeable where possible given the data.

Priority: 1

Grade: A/E

Summary: In the `imager.setimage` function, the stokes parameter can be set to any (or all) of IQUV. The `imager.makeimage` function can make complex images of RR, LR, etc. There do seem to be some problems throughout the tools dealing with data where one of the products has been flagged (e.g. use RR for I if LL is flagged), so some work could be done.

OL–5.2–R3 There must be straightforward and seamless integration of data from multiple epochs and configurations.

Priority: 1

Grade: A/E

Summary: The impression is given that once data are concatenated, the ms will not care whether it is multi-epoch or multi-configuration (but this should be tested thoroughly). This has been done before, and indeed the IRAM tests have confirmed that this works. Some work can be done (e.g. allowing unions of MS without concatenation) to streamline this as per the requirement.

OL–5.2–R4 Simultaneous multiple-field imaging and deconvolution must be supported.

Priority: 1

Grade: A

Summary: This is implemented, e.g. `imager.clean` will jointly clean a mosaic using the **multi-field** option.

OL–5.2–R5 Subtraction of continuum level from spectral data is required, in both the Fourier and image domain. In the case of uv-plane subtraction, flexible setting of the frequency channel ranges for the calculation of the continuum level and selection of the type of fitting (polynomial, spline) should be available in both GUI and CLI.

Priority: 1

Grade: I Severity: med

Summary: It is currently possible to do continuum subtraction in the image plane by subtracting a moment 0 map created from a predefined region of channels (the union function within the region tool allows complicated sets of channels to be selected for the moment map). In addition, a function similar to UVLSF in AIPS will be checked in very soon (available after the audit closing date). It will have some improvements like preaveraging the data to improve signal to noise if desired. It is also possible to fit a continuum level for a single spectrum in the profile fitter tool by selecting it fit for the zero level.

OL–5.2–R6 There must be the ability to include "zero-spacing" values and short-spacing data taken in single-dish mode (both ALMA and non-ALMA data), with selectable weighting.

Priority: 1

Grade: I Severity: high

Summary: The `imager.feather` function "feathers" together single-dish and interferometer data using the "imerg" algorithm of AIPS and SDE, or the "feather" algorithm of MIRIAD. However, the weighting does not appear to be selectable, and this has not worked well on VLA plus GBT data. Further development of this is needed. It is possible to enter the zero spacing as a single number in MEM. Joint deconvolution and improved feathering was checked in after the audit closing date.

OL–5.2–R7 There shall be the choice of FFT and DFT imaging (especially for small datasets).

Priority: 2

Grade: N Severity: low

Summary: There does not appear to be this option in `imager`, e.g. in `imager.setoptions` only the gridded Fourier Transform machine gridft can currently be specified.

OL–5.2–R8 There shall be the provision for the near-field imaging of solar-system objects. This can be done through the introduction of phase corrections based upon the sphericity of the incoming wave front.

Priority: 2

Grade: A

Summary: Within the `imager.setimage function` there are parameters called "distance" and "phasecenter" which are supposed to allow nearby objects to be imaged. I haven't tested it so I don't know how well it works. Help documentation for this is minimal. See OL–8.3–R6 also.

2.5.3 Mosaicing and Single Dish Imaging Considerations

OL–5.3–R1 Combination of interferometer and single-dish data into mosaic imaging is essential.

Priority: 1

Grade: A/E

Summary: The mechanisms are there in AIPS++ though development and testing is still needed (see OL–5.2–R6).

OL–5.3–R1.1 The ACA (if and when it is scoped as part of ALMA) must be integrally supported by the Package.
Priority: 1
Grade: U
Summary: The ACA is still not scoped, pending the Japanese entry into ALMA.

OL–5.3–R2 Careful (polarized) primary beam correction and pointing correction is critical for high fidelity mosaic imaging and must be incorporated into the mosaicing algorithms.

Priority: 1

Grade: I Severity: high

Summary: This overlaps with OL–5.2–R1, OL–4.5–R1, and OL–4.6–R2.

OL–5.3–R2.1 The primary beam calculation and correction must take into account the effect of on-the-fly scanning.
Priority: 1
Grade: U
Summary: It is not clear what this means other than possibly treating OTF scanning pointing centers correctly, smearing by the movement in an integration etc. Will need further evaluation to test.

OL–5.3–R2.2 A set of **ALMA standard beam images** will be made available by the project and distributed with the Package, with updates available for download when appropriate.
Priority: 1
Grade: U
Summary: Must be provided by ALMA.

OL–5.3–R2.3 The user shall be able to specify the primary beam in a number of forms, both analytic and tabular, in addition to the ALMA provided primary beam.
Priority: 2
Grade: A
Summary: The `vpmanager.setcpbnumeric` function allows user input of a glish vector for the primary beam, while other functions in the `vpmanager` tool allow various analytic forms or use of telescope specific defaults. Polarization is not yet an option other than squint.

OL–5.3–R3 Image plane and uv-plane gridding parameters and interpolation shall be selectable based on desired image criteria (e.g. field-of-view, accuracy)

Priority: 1

Grade: A/E

Summary: Imaging parameters are set in `imager.setimage`, gridding parameters in `imager.setoptions` and weighting in `imager.weight`. The function `imager.advise` will suggest appropriate parameters such as cellsize, pixels, and facets. In the case of single-dish data, this is selectable in `gbtcalutils.makeimage`, which is not documented anywhere in the User Reference Manual, but is described in the GBT Continuum Single Dish Imaging chapter of Getting Results Vol3 and repeated (as far as I can tell verbatim) in Getting Results Vol2. Needs better documentation, and maybe there should be only one place in Getting Results where this is described.

OL–5.3–R4 Scaling and de-striping of scans on image-plane combination shall be available.

Priority: 2

Grade: A/E

Summary: Currently done by FFT then filtering of the image (see OL–4.4–R2), and manual adjustment. Should have its own tool or function (with some auto-scaling capability), and better documentation.

OL–5.3–R5 The Package must be able to produce an image by combining data observed on different rasters, possibly taken with different (regular or irregular) spacings and image centers.

Priority: 2

Grade: A

Summary: The AIPS++ implementation is general enough to handle this (e.g. with parameterized scanning patterns). Concatenation works, though is not as streamlined as it could be (see OL–3.1–R20).

OL–5.3–R6 Pointing corrections (e.g. as determined by optical cameras or through monitoring data) shall be applicable to the data during imaging

Priority: 2

Grade: N Severity: med

Summary: This overlaps with OL–5.2–R1, OL–4.5–R1, and OL–4.6–R2, and further development is needed on algorithms.

OL–5.3–R7 The imaging tools shall allow the option for the mitigation of the effects of non-coplanar baselines and sky curvature.

Priority: 2

Grade: A

Summary: There are wide-field Hogbom and Clark algorithms in the `imager.clean` function, with aliasing control in `imager.setoptions`.

2.6   Data Analysis

2.6.1   General Analysis Requirements

OL–6.1–R1 Seamless transformation between image-plane and uv-plane analysis is necessary:

> **OL–6.1–R1.1** Analysis based on goodness-of-fit to models, in both uv-plane (for interferometry) and image plane, shall be available.
> *Priority: 1*
> *Grade: A*
> *Summary:* This is the standard implementation in AIPS++.

> **OL–6.1–R1.2** Fourier transform of images between angular and uv domains shall be available.
> *Priority: 2*
> *Grade: A*
> *Summary:* The `imager.fft` function transforms an image or model component list to the uv plane (and puts it in the `MODEL_DATA` column). Obviously, other `imager` routines to the reverse to make images. There is an `fftserver` that can be used for transforms in scripts.

OL–6.1–R2 Spectral and scanning baseline removal facility is required:

> **OL–6.1–R2.1** Polynomial baseline fitting shall be supported
> *Priority: 1*
> *Grade: A*
> *Summary:* This is available for singe dish in `dish.base`. This is now available in `iramcalibrater.rf` as of daily v1.8 build 173.

> **OL–6.1–R2.2** Baseline fitting based on image statistics (e.g. median filtering) shall be supported
> *Priority: 1*
> *Grade: I Severity:* high
> *Summary:* There did not appear to be any tools matching this requirement.

> **OL–6.1–R2.3** Fourier analysis of standing waves and their removal from spectra shall be available.
> *Priority: 2*
> *Grade: N Severity:* low
> *Summary:* There did not appear to be any general tools matching this requirement, though sinusoid baseline fitting is available for single dish data in `dish`. Needed for interferometer (e.g. `iramcalibrater`). Ideally should be the same tool for both.

OL–6.1–R3 Translation between various astronomical quantities and units shall be available, including:

> *Summary:* The `quanta` module handles the unit conversions, and most tool parameters can take quantities and do conversions internally, or can be done from glish with the `quanta.convert` function.

> **OL–6.1–R3.1** flux density — Jy, mJy, μJy, mag;
> *Priority: 2*
> *Grade: A/E*
> *Summary:* The `quanta` tool understands prefixes such as 'm' and 'u' (for micro). See the `quanta.map` function which lists know units and prefixes and constants. It does not appear to understand magnitudes, though the user has the ability to define special units and do mathematical operations (like taking logarithms). I think that using magnitudes (defined in some quantitative system, e.g. AB magnitudes) is tricky enough that it should be left as an enhancement.

> **OL–6.1–R3.2** temperature — K, mK (Rayleigh-Jeans, Planck);
> *Priority: 2*
> *Grade: I Severity:* low

Summary: The `quanta` tool can deal with temperature units, but does not know about radiation temperature because it needs a reference frequency or wavelength. These could be defined as a `measure`.

OL–6.1–R3.3 surface brightness — Jy/beam, Jy/sr, MJy/sr, mag/arcsec^2;

Priority: 2

Grade: A/E

Summary: The units in the image are essentially specific intensity (e.g. Jy/beam) not flux density in any event. The `quanta` tool can understand units like `'1Jy.sr-1'` which converts to `kg.s-2.sr-1` using `quanta.canonical` (awkward units but dimensionally correct). Leave magnitudes for enhancement if ever.

OL–6.1–R3.4 frequency — Hz, MHz, GHz, cm^{-1} (wave number);

Priority: 2

Grade: A

Summary: All but wavenumber are handled by the `quanta` tool. Wave number can be handled (as is any description like energy; momentum etc., i.e.: time, 1/time, angle/s, length, 1/length, energy, impulse.

OL–6.1–R3.5 velocity — km/s, m/s, z (redshift);

Priority: 2

Grade: A

Summary: Note that since doppler velocities are defined with respect to a reference frame, these use the `measures` module.

OL–6.1–R3.6 wavelength — m, cm, mm;

Priority: 2

Grade: A/E

Summary: It is easy to make quantities with units of length, e.g. `'21cm'`, and convert between them. However, the `quanta` tool does not understand the notion of wavelength, e.g. you cannot convert between a frequency and length directly using `quanta.convert` (it would have know the context in any event), though it is simple to write a function to convert. Im not sure how useful this is so this should be an enhancement. Wim claims you can do it anyway, so examples would be useful.

OL–6.1–R4 The astronomer must have the capability to develop their own tools or tasks, with easy access to data and images, and straightforward interface with the Package.

Priority: 2

Grade: A

Summary: The toolkit aspect of AIPS++ is impressive.

2.6.2 Spectral Line Analysis

ALMA will be a spectroscopic machine, and thus the ease and flexibility of spectral line analysis is likely to be a driver for the Package. Automatic and user-controlled fitting routines will need to be included. Note that there is necessarily some overlap with the image cube analysis, but here we concentrate on issues relevant to traditional 1-D spectra.

OL–6.2–R1 Automatic and user-controlled measurement of line parameters shall be available.

Priority: 1

Grade: A

Summary: The `imageprofilefitter` tool has this function, though the meaning of "automatic" is unclear.

OL–6.2–R2 It shall be possible for the user to specify a velocity or frequency windowing region for line fitting.

Priority: 1

Grade: A

Summary: The `imageprofilefitter` fulfills this function.

OL–6.2–R3 Available line fitting parameters and profiles shall include:

> **OL–6.2–R3.1** Gaussian line parameters (central and integrated intensity, line width, line center) for single or multiple lines
> *Priority: 1*
> *Grade: A*
> *Summary:* The `imageprofilefitter` fulfills this function.
>
> **OL–6.2–R3.2** Damping profiles (Lorentzian)
> *Priority: 2*
> *Grade: I Severity:* low
> *Summary:* This does not appear to be supported. This is slated for future development according to the `imageprofilefitter` documentation, including "Line shapes other than Gaussian (e.g. Zeeman, Voigt)".

OL–6.2–R4 A set of **ALMA standard line catalogs** shall be made available by the project and distributed with the Package, with updates available for download when appropriate.

Priority: 1

Grade: U

Summary: The ALMA standard line catalog has yet to be defined.

OL–6.2–R5 User importable line catalogs shall be supported by the Package as an ASCII table and in the **ALMA standard format**.

Priority: 2

Grade: U

Summary: The ALMA standard line catalog format has yet to be defined. Note that the `measures.linelist` function deals with line catalogs, which can be added using the `measuresdata.addline`.

OL–6.2–R6 Setting of fit constraints (e.g. spacing for multiple lines) shall be available and flexible (i.e. by GUI or by parameter list).

Priority: 2

Grade: I Severity: low

Summary: This is not currently available, though there appears to be low-level functions to do this. Propagation to higher-level is in development.

OL–6.2–R7 Export of fit results in ASCII-format is desirable.

Priority: 3

Grade: A

Summary: The `imageprofilefitter.getstore`, `imagprofilefitter.getfit` and `imagprofilefitter.getestimate` return glish records, which could be output to ASCII files with some scripting.

2.6.3 Image Cube Analysis and Manipulation

Because ALMA is inherently a multi-channel instrument, and due to the design of the ALMA correlator, spectral image cubes can be considered to be the fundamental image structure. Single-channel or continuum images can be considered as subsets or instances of cubes. Note that the ability to pull lower-dimensional structures from larger-dimensional cubes is especially important.

OL–6.3–R1 The Package shall support the construction and analysis of image cubes with a variety of axis choices, including:

    OL–6.3–R1.1 time (e.g. OL–3.1–R8);
        Priority: 1
        Grade: A
    OL–6.3–R1.2 position (e.g. OL–3.1–R9 or general X,Y);
        Priority: 1
        Grade: A
    OL–6.3–R1.3 Fourier (u,v);
        Priority: 1
        Grade: A
        Summary: Can be constructed by FFT of an image?
    OL–6.3–R1.4 channel (frequency, velocity, channel number, band, IF);
        Priority: 1
        Grade: A
    OL–6.3–R1.5 polarization;
        Priority: 1
        Grade: A

OL–6.3–R2 Basic cube rotation and transposition operations shall be available, including rotation not orthogonal to cube faces.

Priority: 1

Grade: I Severity: med

Summary: It was expected that `image.regrid` will do rotations or transpositions itself, but it does appear that it will regrid an already rotated or transposed image (note that some rotations or transpositions can be done at display time in the `viewer`). The transpositions appear to be done with `image.coordsys` and `coordsys.reorder` followed by `image.setcoordsys`. Most "rotations" based on coordsys changes (e.g. equinox) can be done on `coordsys`, but it is surprising that general axes rotations are not supported. The documentation on how you would do these operations seems to be absent or too indirect. Note that transpositions should not be necessary so long as the packages data structures allow processing cubes in any axis order (which maybe should have been a requirement).

OL–6.3–R3 It shall be possible to define multi-dimensional regions of the cube for further processing, both interactively and by specifying coordinates manually.

Priority: 1

Grade: I Severity: med

Summary: This can be done laboriously with the `regionmanager` tool, but this should be available interactively in the `viewer`. In the *Region* tool, you seem to have to make a compound region one plane at a time. See also OL–7.5–R1.

OL–6.3–R4 Identification and reporting of image features (e.g. as determined in processing operations) shall be available, and interactive (where appropriate). Features shall include:

    OL–6.3–R4.1 output in pixel coordinates (e.g. row, column);
        Priority: 1
        Grade: A

OL–6.3–R4.2 output in world coordinates (e.g. RA, Dec, Vel);
Priority: 1
Grade: A

OL–6.3–R4.3 convertible to other supported coordinate systems (e.g. precessed to B1950);
Priority: 2
Grade: A

OL–6.3–R4.4 exportable (e.g. as an ASCII file);
Priority: 3
Grade: A

Summary: These are all available, e.g. in the `viewer` or in `image`.

OL–6.3–R5 The ability to extract same or lower-dimensional structures from higher-dimensional data cubes efficiently is required:

OL–6.3–R5.1 Extraction of "cubical" sub-structures aligned with the original cube axes must be straightforward.
Priority: 1
Grade: A
Summary: This is possible using the `image.subimage` function.

OL–6.3–R5.2 User selection of extraction criteria must be possible through the GUI as well as scriptable.
Priority: 1
Grade: I Severity: med
Summary: It appears that it is not possible to select a subimage of a cube with the GUI and then save it as a new image.

OL–6.3–R5.3 User-selectable sub-structures with arbitrary orientation within the parent cube, with appropriate transformation or interpolation, shall be possible.
Priority: 2
Grade: I Severity: med
Summary: Arbitrary orientation not yet supported.

OL–6.3–R5.4 Extraction of data structures based on standard database (e.g. SQL) queries shall be available.
Priority: 3
Grade: A/E
Summary: The `image.subimage` function will extract based on an "OTF mask" (using LEL or boolean expressions) or on a region, which can be constructed using the `regionmanager` tool, which has many options. Note that there does not appear to be a direct way to query using TQL etc., which is surprising. This is probably sufficient for most cases, but query-based subimaging would be an enhancement.

OL–6.3–R6 The ability to collapse or integrate over sub-dimensions of data cubes in order to form "moments" is required. This shall be possible along any direction(s) in the cube aligned with the axes.

Priority: 1

Grade: A

Summary: Available in `image.moment` if this is over 1 dimension at a time.

OL–6.3–R6.1 Moments along arbitrary user-specified directions in the cube shall be possible.
Priority: 3
Grade: N Severity: low
Summary: This is asking that it be possible to do moments over arbitrary directions not necessarily orthogonal to "cube faces", which is not implemented. This is probably an enhancement (low priority, limited use and probably difficult).

OL–6.3–R7 The Package must have the capability of assembling lower-dimensional data structures into higher-dimension cubes.

Priority: 1

Grade: A

Summary: In addition to `image.adddegaxes` (for degenerate 1-pixel axes), there is the `imageconcat` constructor.

OL–6.3–R8 Blanking of pixels must be maintained through the analysis process.

Priority: 1

Grade: A

Summary: Pixel masks are used (see OL–3.3–R3).

> **OL–6.3–R8.1** It must be possible to turn on and off different blanking (mask) levels, when blanking is set within the Package.
> *Priority: 1*
> *Grade: A*
> *Summary:* According to the documentation this is implemented; there are exceptions, e.g. blank pixels are set with zeros before Fourier transforming, which is OK provided that the user get a warning.
>
> **OL–6.3–R8.2** Blanking shall not be destructive, and the original pixel value is retained (if defined).
> *Priority: 2*
> *Grade: A*
> *Summary:* The `image.maskhandlergui` tool allows a fairly easy and efficient way of both selecting different masks and unsetting them.
>
> **OL–6.3–R8.3** Interactive and automatic facilities for setting of any blanking parameters shall be provided.
> *Priority: 2*
> *Grade: A*
> *Summary:* Available interactively via the `viewer` Image Analysis tool Regions function. A number of tasks exist for non interactive masking using expressions or flux limits in the `image` tool, e.g. the `image.calcmask` function.

OL–6.3–R9 A variety of image processing and filtering operations on the cube shall be available, including:

> **OL–6.3–R9.1** Smoothing and convolution, with kernels including:
>
> > **OL–6.3–R9.1.1** Uniform (box-car or top-hat) kernel;
> > *Priority: 1*
> > *Grade: A*
> > *Summary:* `image.arrconvolve` appears to satisfy this requirement (not tested).
> >
> > **OL–6.3–R9.1.2** Elliptical Gaussian kernel;
> > *Priority: 1*
> > *Grade: A*
> > *Summary:* Available with `image.convolved2d` (not tested).
> >
> > **OL–6.3–R9.1.3** Symmetric polynomial;
> > *Priority: 2*
> > *Grade: A/E*
> > *Summary:* Note that `image.convolved2d` supports only Gaussian functions. It is possible to make an array with the polynomial kernel, e.g. using the `polyfitter.eval` function from the `mathematics` module, and then use `image.arrconvolve`. It is not clear that it is worthwhile to bring out a special function for this.

OL–6.3–R9.1.4 User-supplied;
> *Priority: 3*
> *Grade: A*
> *Summary:* `image.arrconvolve` appears to satisfy this requirement (not tested).

OL–6.3–R9.2 Clipping
> *Priority: 1*
> *Grade: A*
> *Summary:* The `image.calcmask` function fulfills this requirement (e.g. using LEL calculations). It is not clear how this requirement differs from masking; we interpret this as masking based on a pixel value like intensity (rather than just a position).

OL–6.3–R9.3 Windowing and boxing, with definitions of regions of interest recordable and passable for subsequent use
> *Priority: 1*
> *Grade: A*
> *Summary:* Available from `regionmanager` tool.

OL–6.3–R9.4 Arithmetical operations, including:

> **OL–6.3–R9.4.1** addition of uniform level;
>> *Priority: 1*
>> *Grade: A*

> **OL–6.3–R9.4.2** multiplicative scaling;
>> *Priority: 1*
>> *Grade: A*

> **OL–6.3–R9.4.3** logarithm;
>> *Priority: 2*
>> *Grade: A*

> **OL–6.3–R9.4.4** exponential;
>> *Priority: 2*
>> *Grade: A*

> *Summary:* All available from the `image` tool via the `image.calc` function.

OL–6.3–R9.5 Statistical operations over a defined region, including:

> **OL–6.3–R9.5.1** mean
>> *Priority: 1*
>> *Grade: A*

> **OL–6.3–R9.5.2** rms
>> *Priority: 1*
>> *Grade: A*

> **OL–6.3–R9.5.3** standard deviation from mean
>> *Priority: 1*
>> *Grade: A*

> **OL–6.3–R9.5.4** median
>> *Priority: 1*
>> *Grade: A*

> **OL–6.3–R9.5.5** mode
>> *Priority: 2*
>> *Grade: A*

> *Summary:* These functions are available in `viewer` and in the `image` tool, as the analysis function `image.statistics`.

OL–6.3–R9.6 Fourier and correlation operations on cube or cubical sub-regions, including:

> **OL–6.3–R9.6.1** Fourier transform
>> *Priority: 1*
>> *Grade: A*

Summary: This (`image.fft`) works, even for non power-of-2 image sizes (confirmed by testing).

OL–6.3–R9.6.2 power spectrum

Priority: 1

Grade: A/E

Summary: Possible with `image.calc`, e.g. using LEL and `image.calc` to square the amplitude of the Fourier transform of an image. Note that what you probably want is something more complicated (like estimating the power spectrum as in the CMB) but this would be a research area. Is this worth investigating (enhancement)?

OL–6.3–R9.6.3 autocorrelation

Priority: 1

Grade: A/E

Summary: Possible with `image.calc`, e.g. using `image.calc` to square the amplitude of the Fourier transform of an image and then transform it back, or using `image.arrconvolve` to convolve with itself after extracting to an array. However, it is surprising that there aren't more (auto/cross) correlation and convolution tools in `image` or in LEL. These are fundamental enough to qualify as an enhancement.

OL–6.3–R9.6.4 structure function

Priority: 2

Grade: A/E

Summary: Possible with `image.calc` after having copied and shifted an array version of the image. However, it is surprising that there aren't shift operations in `image` or in LEL. These are fundamental enough to qualify as an enhancement.

OL–6.3–R9.7 Scalar arithmetic between different regions (including treatment of masked regions and differently shaped regions), including:

OL–6.3–R9.7.1 sum (difference);

Priority: 1

Grade: A

OL–6.3–R9.7.2 product (division);

Priority: 1

Grade: A

OL–6.3–R9.7.3 spectral index;

Priority: 1

Grade: A

OL–6.3–R9.7.4 median;

Priority: 2

Grade: A

OL–6.3–R9.7.5 maximum (minimum);

Priority: 2

Grade: A

Summary: All LEL functions (see Note 223). This can be done using `image.calc` on regions from two images. Regions which do not have the same shape can be dealt with also, e.g. through union, intersection, difference. This can be complicated (probably have to extract into arrays) but it could be done in a few lines of glish. Should this be made easier?

OL–6.3–R9.8 Construction and comparison of vector quantities in the cube, including:

OL–6.3–R9.8.1 polarization (E) vector at each cube pixel;

Priority: 1

Grade: A

Summary: Available via the `imagepol` tool

OL–6.3–R9.8.2 rotation measure between different frequencies;

Priority: 1

Grade: A

Summary: Available via the `imagepol` tool

OL–6.3–R9.9 Vector arithmetic between different regions (including treatment of masked regions and differently shaped regions), including:

OL–6.3–R9.9.1 sum (difference);

Priority: 1

Grade: A/E

OL–6.3–R9.9.2 dot product;

Priority: 2

Grade: A/E

OL–6.3–R9.9.3 cross product;

Priority: 2

Grade: A/E

Summary: These could be done in glish using the `matrix` tool of the `mathematics` package on extracted arrays. This is kind of obscure and probably prioritized too high, but AIPS++ likely covers this sufficiently and special functions or scripts could be written to deal with these.

OL–6.3–R9.10 Interpolation across blanked or masked regions

Priority: 1

Grade: N Severity: med

Summary: Not available, should be put into `image.replacemaskedpixels`.

OL–6.3–R9.11 Fitting of models, shapes, profiles and functions over regions including:

OL–6.3–R9.11.1 polynomials

Priority: 1

Grade: A/E

Summary: In `image.fitpolynomial` for 1D only; should propagate to `image.fitsky` which currently does Gaussians only.

OL–6.3–R9.11.2 exponentials

Priority: 1

Grade: I Severity: med

Summary: Not in `image` tool, should add to `image.fitsky`. Could be done via `functionals` and `fitting` (see below).

OL–6.3–R9.11.3 Fourier modes

Priority: 1

Grade: I Severity: med

Summary: Not in `image` tool, should add to `image.fitsky`. Could in principle FFT the image or region and fit peaks in that plane. Note that `sinusoidfitter` (in the `mathematics` module of the `utility` package) only works in 1D.

OL–6.3–R9.11.4 elliptical Gaussian components

Priority: 1

Grade: A

Summary: In `image.fitsky`.

OL–6.3–R9.11.5 uniform (multi-dimensional) spheres

Priority: 2

Grade: I Severity: low

Summary: Should be built into `image.fitsky`. Could be done via `functionals` and `fitting` (see below).

OL–6.3–R9.11.6 trigonometric functions

Priority: 2

Grade: I Severity: low

Summary: Should be built into `image.fitsky`. Could be done via `functionals` and `fitting` (see below).

OL–6.3–R9.11.7 Lorentzian profiles
Priority: 2
Grade: I Severity: low
Summary: Should be built into `image.fitsky`. Could be done via `functionals` and `fitting` (see below).

OL–6.3–R9.11.8 user-supplied functions
Priority: 3
Grade: A/E
Summary: Can be done using the `functionals` and `fitting` tools from the `Utility` package, as can some of the odd functions above. This can be complicated to do, so recommended as "adequate" only for the user-supplied functions.

OL–6.3–R9.12 Filtering, including:

OL–6.3–R9.12.1 Fourier (Wiener) filtering
Priority: 2
Grade: A/E
Summary: This is a common enough operation that it should be built in to some function (or at least an explicit recipe given).

OL–6.3–R9.12.2 Sobel filtering
Priority: 3
Grade: A/E

OL–6.3–R9.12.3 unsharp mask
Priority: 3
Grade: A/E

OL–6.3–R9.12.4 user-supplied filtering functions
Priority: 3
Grade: A/E

Summary: These all possible with `image.arrconvolve` combined with `functionals`. There should be examples given in the URM.

OL–6.3–R9.13 Deconvolution
Priority: 2
Grade: A
Summary: The `image.deconvolvecomponentlist` function deconvolve a Gaussian componentlist from the restoring beam. The `deconvolver` tool allows deconvolution of images (no MS required).

OL–6.3–R9.14 Resampling (e.g. at lower temporal or spectral resolution) after processing
Priority: 2
Grade: A
Summary: `image.regrid` allows this (not tested).

OL–6.3–R9.15 Calculus on scalar and vector fields in cube, including:

OL–6.3–R9.15.1 differentiation along paths;
Priority: 3
Grade: N Severity: low

OL–6.3–R9.15.2 integration over regions;
Priority: 3
Grade: N Severity: low

OL–6.3–R9.15.3 gradient of scalar fields;
Priority: 3
Grade: N Severity: low

OL–6.3–R9.15.4 divergence of vector fields;
Priority: 3
Grade: N Severity: low

OL–6.3–R9.15.5 curl of vector fields;
Priority: 3
Grade: N Severity: low
OL–6.3–R9.15.6 Laplacian of vector fields;
Priority: 3
Grade: N Severity: low

Summary: None of these are available. Might, with difficulty, cobble something together some finite-difference methods together.

OL–6.3–R10 There shall be the capability to manipulate data cubes as general data structures, so that arithmetical and logical operations can be applied as object methods.

Priority: 3

Grade: A

Summary: The `image.calc` function and the `imagecalc` constructor will combine images using the Lattice Expression Language (LEL). You can also extact subimages into arrays and manipulate these in glish.

2.7   Visualization

This is intended as the purely graphical part of data analysis. There is by necessity some overlap with the functionality discussed under Data Analysis, particularly that for image cube manipulation, and it would in fact be ideal if visualization and analysis were so closely integrated that there were no effective difference. The intention here is that the user is not only able to display pre-calculated images (processed using tools from the Data Analysis suite), but also has the capability of doing some processing and display on-the-fly as an integral part of the visualization.

2.7.1   General Visualization and Plotting Requirements

OL–7.1–R1 Plotting and display capabilities shall be integrated into the GUI tools throughout the Package.

Priority: 1

Grade: A/E

Summary: This is largely available in AIPS++ tools (e.g. `calibrater.plotcal`, `image.view`, `imager.plotvis`), and no glaring omissions are apparent. There are some issues of the ease-of-use, and the exact meaning of "integrated" in this requirement is unclear (it might mean that the viewer windows reside in the tool GUI window, not spawn a new viewer). Some of the display windows (e.g those generated by the `pgplotter`) are more problematic, with less intuitive control options. However, the look-and-feel of the GUI (dealt with in 2.2.2 should be separated from the capability itself (dealt with here). There should probably also be sub-requirements of this item to deal with look and feel, and ease of use, issues. As this is requirment is written, the AIPS++ approach is acceptable, though improvement would be desired.

OL–7.1–R2 Standard type of plots must be supported, such as:

OL–7.1–R2.1 X-Y plots;
Priority: 1
Grade: A
Summary: Available in `msplot`, and eventually in the `viewer`.

OL–7.1–R2.2 histograms;
Priority: 1
Grade: A
Summary: Available in `image.histogram`, and in the `viewer` ImageAnalysis, section Statistics, Plot button after having defined a ROI).

OL–7.1–R2.3 contour plots;
Priority: 1
Grade: A
Summary: Available in the `viewer`, and tested extensively.

OL–7.1–R2.4 vector plots;
Priority: 1
Grade: A
Summary: Available in the `viewer`, and tested extensively.

OL–7.1–R2.5 2D images;
Priority: 1
Grade: A
Summary: Available in the `viewer`.

OL–7.1–R2.6 wireframe 3D surfaces;
Priority: 1
Grade: N Severity: low

Summary: 3D images are not currently available (some development is ongoing to use OpenGL within AIPS++ to render 3D solids on a pixel canvas).

OL–7.1–R3 "Blinking" between two different images, or stepping through a set of ordered images, at a user-selected rate and with adjustable transfer functions must be possible.

Priority: 1

Grade: I Severity: high

Summary: The basic capability to "blink" automatically between two (registered) images is missing from the `viewer`. Animation controls work well (though there are suggested improvements, such as a scroll bar).

OL–7.1–R4 Standard plotting formats shall be supported, both displayed (e.g. X window) and hardcopy:

OL–7.1–R4.1 There must be at least one designated standard output format (e.g. postscript) that can be converted by the user to a variety of formats using easily obtainable tools.
Priority: 1
Grade: A
Summary: There is `.ps` (with optional EPS) and `.xpm` (X11-pixmap) output available from the `viewer`. The `pgplotter` can save as Postscript as well as print, but should eventually incorporate all standard pgplot outputs including portrait and landscape.

OL–7.1–R4.2 The Package shall also support the output of a variety of commonly used formats such as FITS, postscript, pdf, gif and/or jpeg.
Priority: 3
Grade: I Severity: low
Summary: The Canvas Print Manager in the `viewer` can output in `.xpm` format also, and works reasonably well. However, it would be good to see at least JPEG. The `pgplotter` can save only as Postscript but should eventually incorporate all standard pgplot outputs.

OL–7.1–R5 Identification of cursor position shall be available for interactive plots. Where appropriate, this information shall be recordable and exportable. If you "see-it" you should be able to figure out where it came from.

Priority: 1

Grade: I Severity: low

Summary: In `msplot`, you can use the **Locate** button to tell you what visibilities you were looking at. There is also a running report of the X and Y that the cursor is over, but no other info (like fieldname). The `viewer` has a status bar at the bottom which give OK information for images, but needs some improvement for viewing ms. I see no way to record this info (e.g. to the `logger`). The `pgplotter` plots (e.g. `calibrater.plotcal` rely on colors or symbols, there is no way to identify a point under the cursor.

OL–7.1–R6 An extra "axis" of information shall be encodable on the standard plot types using color and/or intensity.

Priority: 2

Grade: I Severity: low

Summary: It is not apparent how to produce from the `viewer` such things as intensity reflecting moment of order 0 and color moment of order 1. In `msplot` with the option "display data as an image" there is no "image pixel" item with amplitude for intensity and phase in color. However there is a method described in the Sec 3.10.2 *Advanced worked examples* of Vol 1 of Getting Results to explain how to load different images into RGB planes. This should be made easier. See also OL–7.2–R6.2.

OL–7.1–R7 The displays shall have similar look and feel to reduce the plotting learning curve.

> *Priority: 2*
>
> *Grade: A*
>
> *Summary:* The tools all use the same `pgplotter` tool, or the `viewer` (which in turn uses the Display Library). See also OL–2.2–R5.

2.7.2 Display Appearance and Interactivity

OL–7.2–R1 Plot selection parameters (axes, limits, colormap) shall be conveniently controllable.

> *Priority: 1*
>
> *Grade: I Severity:* med
>
> *Summary:* The Adjust Panel options for the `viewer` offer extensive, if sometimes laborious to use, control for controlling the display. Currently, the `pgplotter` based display suffer from the use of the arcane PGPLOT command controls (great for FORTRAN-era PG-pundits!). However, the pgplot parts are targeted for eventual replacement with Display Library plotting (ie. the `viewer` toolkit), at which point it will inherit the Adjust Panel controls. Should the Adjust Panel functions be made more accessible (through hotkeys)?

OL–7.2–R2 There shall be interactive display zooming and unzooming capability within plot windows.

> *Priority: 1*
>
> *Grade: I Severity:* med
>
> *Summary:* The zoom/unzoom function in the `viewer` is serviceable, though interaction with panning and the additional capability of in/de-cremental zoom (with hotkey support) is desired. The `pgplotter` based displays have poor implementation of zooming. The `msplot` zoom and unzoom has poor interaction with box drawing for example. This is such a basic function that it should work well. Note that this requirement as written does not specify behavior (the feature is there though we may not like how it is done) — there should be a requirement on behavior if this is desired. However, the pgplotter deficiencies move this from A/E to I.

OL–7.2–R3 The plot update speed shall not be a bottleneck. Speed shall be benchmarked, and should be commensurate with comparable plotting packages.

> *Priority: 1*
>
> *Grade: I Severity:* high
>
> *Summary:* Nominally should awaiting plotting and interface benchmarks, however slowness, particularly running the Adjust Panel, is so obvious that this needs to be flagged. Note that improvements in the glish event handling (such as for interface button pushes) is currently being worked upon.

OL–7.2–R4 Different line styles, sizes, thicknesses and colors must be available. User shall be able to manipulate intensity and color scales, interactively for graphical displays. The setup shall be saveable and reloadable.

> *Priority: 1*
>
> *Grade: A*
>
> *Summary:* Different line styles, sizes, thicknesses and colors are available for graphics; interactive ITT and LUT are also avaible in the `viewer`.

OL–7.2–R5 Basic axis transformations shall be built in to plotting, such as:

> **OL–7.2–R5.1** Logarithmic amplitude and intensity scale
> > *Priority: 1*
> > *Grade: I Severity:* med

Summary: Not available in `msplot`, and in the `pgplotter` plots only by modifying the plot at the pgplot level. In the `viewer` you can set something called the *scaling power cycles* which effectively sets logaritmic scaling after a fashion (see Adjust panel option records).

OL–7.2–R5.2 Different time and coordinate units and formats (e.g. hours, hhmmss, radians, ddmmss.s)
Priority: 1
Grade: I Severity: med
Summary: Ability to select different coordinate units, and and formats is exceptional in the `viewer`. The `msplot` tool needs similar capabilities.

OL–7.2–R6 User shall be able to augment plots and produce overlays of different data sets of standard formats:

OL–7.2–R6.1 Images with same axes, size and orientation shall be superposable directly, with basic control of colors and symbols.
Priority: 1
Grade: A/E
Summary: Available in `viewer`. Note that only a single raster is visible at a given time. You can overlay multiple contour, vector, or marker plots. Should multiple rasters be superposeable?

OL–7.2–R6.2 Overlay layer style shall be selectable, e.g. contours, greyscale, colormapped (RGB or HSV), or single color (i.e. one layer gets assigned intensity scales of red, another one of green, and one of blue).
Priority: 1
Grade: A/E
Summary: RGB, HSV mappings available in `viewer`. The method is described in the Sec 3.10.2 *Advanced worked examples* of Vol 1 of Getting Results to explain how to load different images into RGB planes. This should be made easier.

OL–7.2–R6.3 Overlay of selectable coordinate grids (e.g. J2000, B1950, galactic, ecliptic, pixel number) shall be available. It must also be possible to overlay multiple grids.
Priority: 1
Grade: I Severity: low
Summary: While it is possible to change to a different coordinate system and overlay that coordinate systems grid on an image, e.g. in the `viewer`, you cannot have two (e.g. galactic grid and a RA-Dec grid) simultaneously.

OL–7.2–R6.4 The user must be able to overlay functional fits (e.g. polynomials) or points read in from standard tabular files
Priority: 2
Grade: I Severity: low
Summary: Using a sky catalog it is possible to read in ASCII tables in a standard format and overlay them in the `viewer`. It should also be possible to define more complicated functions to overlay using this method (but not tested). But not available for X-Y plots, currently in `msplot` or `pgplotter`. We await X-Y plots in the `viewer`...

OL–7.2–R6.5 It shall be possible to place data sets in "layers" be which can be interactively colormapped, and switched on and off.
Priority: 2
Grade: A
Summary: The `viewer` has the capability to register, display (and turn off) individual data sets. Colormapping can be done in the relevant Adjust Panel. It is not clear what is meant here by "layers", there is only one raster displayed unless you map to colors (see OL–7.2–R6.2).

OL–7.2–R6.6 It shall be possible to display and overlay data with different coordinate systems, i.e. the coordinate system of the display can be chosen independent of the system the data were observed in and the data transformed appropriately with pre-computation.
Priority: 3

Grade: I Severity: med

Summary: The `viewer` should be able to register images in different coordinate systems on-the-fly, however it will not display a contour image in B1950 over the same raster image in J2000. Nominally one could use `image.regrid` to do the transform before import, but this requirement is here to avoid that indirect method.

med

OL–7.2–R6.7 It shall be possible to shift, rotate and scale the images interactively.

Priority: 3

Grade: I Severity: low

Summary: This is not available withing the `viewer`, for example. There are a number of functions with do this within the images tool, but these are not interactive in the sense of the requirement.

OL–7.2–R7 Users shall be able to synchronize multiple display windows, for example such that zooming to a given pixel in one image window will select the equivalent pixel in the slaved windows.

Priority: 2

Grade: N Severity: low

Summary: Not currently available, but proposed for upgrade.

OL–7.2–R8 Users shall be able to add annotation, both interactively and through scripts, including text with various fonts (including Greek letters), symbols (e.g. all the symbols provided by the LaTeX package with AMSTeX extension), arrows, geometrical figures like boxes and circles, etc.

Priority: 2

Grade: A/E

Summary: Basic annotation is available in the `viewer` with text, lines, arrows, circles of different colors, etc. There are also a number of fonts and symbols available. While there are Greek and ZapfChancery symbols fonts are listed, but they do not produce the advertised characters. Presumably this is a short term bug. There are some ease of use issues, which could use some improvement.

2.7.3   Visibility Data

OL–7.3–R1 Plotting of commonly used basic data and calibration quantities must be straightforward and easily accessible from all relevant tools. Any quantity should be displayable versus any other (or any two plus another encoded as a z-axis or intensity) as long as these quantities are defined for the same visibility or calibration solution interval.

These displayable quantities include:

OL–7.3–R1.1 real and/or imaginary;
Priority: 1
Grade: A
OL–7.3–R1.2 amplitude and/or phase;
Priority: 1
Grade: A
OL–7.3–R1.3 delay and/or rate;
Priority: 1
Grade: N Severity: low
OL–7.3–R1.4 time;
Priority: 1
Grade: A
OL–7.3–R1.5 hour angle;
Priority: 1

Grade: N Severity: med

Summary: Not available in `msplot` or `viewer`

OL–7.3–R1.6 parallactic angle;

Priority: 1

Grade: N Severity: med

Summary: Not available in `msplot` or `viewer`

OL–7.3–R1.7 uv distance;

Priority: 1

Grade: I Severity: med

Summary: Available in `msplot`, not available in `viewer`

OL–7.3–R1.8 u and/or v;

Priority: 1

Grade: I Severity: med

Summary: Available in `msplot`, not available in `viewer`

OL–7.3–R1.9 baseline;

Priority: 1

Grade: A

OL–7.3–R1.10 channel or frequency or velocity;

Priority: 1

Grade: A

OL–7.3–R1.11 azimuth and/or elevation;

Priority: 1

Grade: N Severity: med

Summary: Not available in `msplot` or `viewer`

OL–7.3–R2 Data selection parameters shall be understandable (e.g. by antenna name or number instead of antenna table entry number, polarization name RR or I) and straightforward, using graphical browsers (in GUI mode) and/or standard selection language (e.g. SQL queries) in script mode.

Priority: 1

Grade: I Severity: med

Summary: Use of names instead of AIPS++ or ms internal values is not yet integrated throughout the package. For example, you have to specify a field ID number (corresponding to the `FIELD_ID` keyword in the ms) in most functions rather than as a source name (e.g. '0137+331' or '3C286'). This requirement applies to what the user sees in the GUI and CLI interface, not what a programmer sees in the ms.

OL–7.3–R3 It shall be possible to interpolate or extrapolate any tabulated quantity onto a visibility or calibration solution point, and then manipulate these like extra visibility information.

Priority: 3

Grade: I Severity: low

Summary: It is possible to do things like this at the glish level, but to manipulate like extra visibility info would require new columns in the MS (which is easy) and tools that could understand these (hard).

2.7.4   Other ALMA Data

Although the Package will not likely be the primary vehicle for the ALMA staff to assess the state of the array, it is intended that users (as well as staff) have the full capability of using ancillary data provided by ALMA to aid in the processing and understanding of their data. Therefore, the Package should be able to deal with this data in as user-friendly a manner as possible.

OL–7.4–R1 The Package shall be able to plot standard ALMA-format ancillary data, including:

> **OL–7.4–R1.1** amplitude or single-dish power versus AZ and EL;
> *Priority: 1*
> *Grade: I Severity:* high
> *Summary:* You cannot plot vs AZ-EL in `msplot` or the `viewer`.
>
> **OL–7.4–R1.2** focus data and curves;
> *Priority: 2*
> *Grade: U*
> *Summary:* Not available, though there needs to be some input from ALMA on formats etc, or some VLA or IRAM data to test upon.
>
> **OL–7.4–R1.3** pointing data and offset vectors;
> *Priority: 2*
> *Grade: U*
> *Summary:* Not available, though there needs to be some input from ALMA on formats etc, or some VLA or IRAM data to test upon.
>
> **OL–7.4–R1.4** WVR output data;
> *Priority: 3*
> *Grade: U*
> *Summary:* Requires development of WVR system. Can possibly test when VLA WVR data is available, or using BIMA data.
>
> **OL–7.4–R1.5** holography and beam map data;
> *Priority: 3*
> *Grade: U*
> *Summary:* Requires development from ALMA.
>
> **OL–7.4–R1.6** monitor point values (e.g. temperatures);
> *Priority: 3*
> *Grade: U*
> *Summary:* Requires definition of what ALMA M&C will deliver.

2.7.5   Image-cube Manipulation

OL–7.5–R1 Histograms of pixel values must be easily produced for selected regions of the cube.

Priority: 1

Grade: I Severity: low

Summary: In the `viewer` Image Analysis tool you can get statistics for either full image or a region of the currently shown plane — there needs to be a way to extend the region over which the statistics and histogram plotting to more than one plane. One can make a compound region stepping over each plane (see OL–6.3–R3), or define the regions at glish level. In short, 2D works well, and 3D extension is needed but is probably lower priority.

OL–7.5–R2 It shall be possible to view subsets or slices of data cubes:

> **OL–7.5–R2.1** for axes aligned with cube faces;
> *Priority: 1*
> *Grade: A/E*
> *Summary:* Functionality available in 2D and 1D but too slow for good interactivity (`viewer` ImageAnalysis Position) in comparison to other packages. Note that there currently is not an easy way to mark out a region of the cube purely graphically in the `viewer` (see OL–7.5–R1).
>
> **OL–7.5–R2.2** for arbitrarily aligned axes.
> *Priority: 3*
> *Grade: N Severity:* low

Summary: No (sufficiently easy) way to select arbitrarily shaped regions (ie. not aligned with axes).

These shall be interactively selected in GUI mode.

OL–7.5–R3 It must be possible to plot values of the same pixel in different cube layers, or in different images, against each other.

Priority: 1

Grade: A

Summary: Available for 1D slice, and, for different 2D images, using Canvas manager.

OL–7.5–R4 Data cubes must be viewable as movies with selectable frame rates and layer step directions.

Priority: 1

Grade: A

Summary: Available in the `viewer` tool (the tapedeck controls).

OL–7.5–R5 Interactive display of spectra corresponding to a pixel or region in a displayed image shall be supported.

Priority: 2

Grade: A

Summary: You can select a region to average over with the region button and then display the averaged spectra with the image analysis tool. Note that you can set the plotting to auto and drag the cursor over the image and watch the profile change.

OL–7.5–R6 Interactive display of a 1D slice taken from a 2D image, such as dragging the line on the map to bring up a position-velocity diagram, shall be available.

Priority: 2

Grade: N Severity: med

Summary: Does not appear to be available.

OL–7.5–R7 Plotting of spectra on a pseudo-grid corresponding to position on a raster (e.g. a "stamp map" or "profile map", basically thumbnail spectra in panels corresponding to position) shall be possible.

Priority: 2

Grade: I Severity: low

Summary: Not available in the `viewer`, nor for synthesis data. Might be available in `dish`.

2.8 Special Features

2.8.1 Simulation

Note: Inclusion of at least moderate simulation capability in the Package will benefit the user who may not have access to the more comprehensive simulator built into the Pipeline and Online systems. In the spirit of providing the user with offline options equivalent to those available in the project, it is desirable that some simulation capability also makes it into the Package albeit at low priority. The main goals would be to allow the user to simulate basic ALMA modes based on input models or images and to replace existing data in ALMA format with simulated data.

OL–8.1–R1 The Package shall inherit the basic simulation capability of the ALMA Online Data Processing Pipeline (Levels 1 and 2) as outlined in ALMA-SW Memo 11, as per (OL–1.1–R5).

> *Priority: 2*
>
> *Grade: U*
>
> *Summary:* Pipeline and simulation requirements under review.

2.8.2 VLBI

Note: It is assumed that VLBI data will be reduced in whatever package supports general VLBI observations, as ALMA will not be a stand-alone VLBI instrument. There are no VLBI requirements on the ALMA Offline Data Processing Package, though it may end up being the case that the Package also supports VLBI processing. This would have some advantage to users of the ALMA Package, and thus it is worth including a low-priority requirement to this effect.

OL–8.2–R1 It is desirable that the Package also support general VLBI processing suitable for dealing with ALMA data, in order to allow general ALMA users access to VLBI science without having to learn a completely new software package.

> *Priority: 3*
>
> *Grade: I Severity:* low
>
> *Summary:* Full VLBI processing is not yet available in AIPS++.

2.8.3 Solar System Object Observing

The Sun and planets are likely to be important and interesting targets for ALMA observing. The requirements are likely to be strongest on the actual hardware (e.g. high frequency and time resolution) but software compatibility must also be considered. In particular, solar and planetary observations require a special effort in tracking and handling of ephemerides.

OL–8.3–R1 The Package must be able to calculate and compensate for the position of moving objects in the solar system:

> **OL–8.3–R1.1** Ephemerides must be provided for major solar system objects, including:
>> **OL–8.3–R1.1.1** Sun;
>>> *Priority: 1*
>>> *Grade: A*
>>> *Summary:* The `measures.direction` tool understands the reference codes: SUN.

OL–8.3–R1.1.2 Moon;
  Priority: 1
  Grade: A
  Summary: The `measures.direction` tool understands the reference codes: MOON.

OL–8.3–R1.1.3 planets;
  Priority: 1
  Grade: A
  Summary: The `measures.direction` tool understands the reference codes: MERCURY, VENUS, MARS, JUPITER, SATURN, URANUS, NEPTUNE, PLUTO.

OL–8.3–R1.1.4 major asteroids (all known with 50km dia. greater);
  Priority: 1
  Grade: N Severity: low
  Summary: There do not seem to be any asteroids built in to measures.

OL–8.3–R1.1.5 known short period comets;
  Priority: 1
  Grade: N Severity: low
  Summary: The `measures.direction` tool understands the reference codes: COMET. However, it appears that "comets" are generic user input ephemerides, there was no list of comets.

OL–8.3–R1.2 Ephemerides must be calculated in all available reference frames (e.g. OL–3.1–R10)
  Priority: 1
  Grade: A
  Summary: The `measures` tool appears do all the needed conversions for its supported frames.

OL–8.3–R1.3 The Package shall be able to import a user-supplied ephemeris in tabular form.
  Priority: 2
  Grade: A
  Summary: There are facilities for creating new emphemerides. The `measuresdata.createcomet` will make a table from an input text file in "p120" format, which can then be used as a reference frame with `measures.framecomet`.

OL–8.3–R1.4 The Package shall calculate positions from user-provided orbital elements.
  Priority: 2
  Grade: N Severity: low
  Summary: This does not appear to be available.

OL–8.3–R2 The Package must carry out astrometry for moving sources.

  Priority: 1

  Grade: A

  Summary: The measurement set and reference frame system deals with moving objects.

OL–8.3–R3 The distance visibility amplitude correction for distance (from ephemeris for major objects, or user-supplied) shall be computed.

  Priority: 1

  Grade: N Severity: med

  Summary: The planets should be added to the `calibrater.setjy` list, or another function built in.

OL–8.3–R4 The Package must calculate quantities for the "physical ephemeris" for the enumerated major solar system objects, including:

OL–8.3–R4.1 subearth latitude and longitude;
  Priority: 2
  Grade: N Severity: low

OL–8.3–R4.2 subsolar latitude and longitude;
 Priority: 2
 Grade: N Severity: low
OL–8.3–R4.3 position angle of North Pole;
 Priority: 2
 Grade: N Severity: low
OL–8.3–R4.4 season;
 Priority: 2
 Grade: N Severity: low
OL–8.3–R4.5 phase angle;
 Priority: 2
 Grade: N Severity: low

Summary: These do not appear to be part of the ephemerides.

OL–8.3–R5 The Package shall do coordinate transformations from plane-of-sky to planetocentric for a selection of projections, including those of OL–3.3–R4, and in addition:

OL–8.3–R5.1 3-D backprojection (onto the planetary sphere).

Priority: 2

Grade: N Severity: low

Summary: This does not appear to be available.

OL–8.3–R6 The imaging corrections for the near-field shall be included.

Priority: 2

Grade: A

Summary: A `distance` parameter was added in v1.7 to the `imager.setimage` to do the near-field delay corrections.

OL–8.3–R7 In addition to the standard models available in the data analysis tools, there shall be provision for:

OL–8.3–R7.1 prolate and oblate ellipsoids;
 Priority: 2
 Grade: I Severity: low
 Summary: Disks in the `componentmodels` module are elliptical. Currently only uniform disks are supported.
OL–8.3–R7.2 limb-darkened disks (polynomial, $\cos^n$, Legendre polynomial).
 Priority: 2
 Grade: N Severity: low
 Summary: These are not available.

OL–8.3–R8 The Package shall support the 3-D reconstruction of the emission using observations of the target object at different aspect angles and/or rotational phases.

Priority: 3

Grade: N Severity: low

Summary: This does not appear to be available.

2.8.4 Pulsar Observing

ALMA's pulsar needs will be much easier to satisfy than those of centimeter-wave arrays since dispersion smearing is a non-issue at these frequencies. We expect the main out-of-the-ordinary requirement will

be integration into multiple phase bins. A minimum of 16 phase bins would be scientifically interesting. Pulsars have periods as short as 1.6 ms, so a minimum bin size of 0.1 ms would probably suffice. The main use of this binning capability will likely be measuring the on-pulse and off-pulse brightness which can be used to estimate or place bounds on the temperature of the neutron star. Pulsar timing is unlikely to be done with ALMA, barring the installation of specialized hardware. It is conceivable that ALMA would be a great instrument for pulsar parallaxes, especially if pulsars are detectable at 900GHz, and will likely be the best instrument in the southern hemisphere for years to come. The minimal spec of 16 pulsar phase bins would be sufficient for this work.

Summary: There is a description of pulsar processing in the Aug 1998 AIPS++ Newsletter. Pulsar phase bins and gating (PULSAR_BIN and PULSAR_GATE_ID columns) are built into the measurement set, see MS v2.0 description.

OL–8.4–R1 The Package shall support the processing of time-series data by folding into multiple phase bins which are synchronous with a user-specified pulsar period.

  Priority: 3

  Grade: N Severity: low

  Summary: This option is not available in most tools. It might be possible at the glish level, though there is no demonstration whether it works in practice.

OL–8.4–R2 The Package shall support at least 16 phase-bins.

  Priority: 3

  Grade: A

  Summary: There does not seem to be a limit on number of bins.

OL–8.4–R3 Interpolation onto the phase bins from the correlator output (integration times) shall be available.

  Priority: 3

  Grade: N Severity: low

  Summary: This does not appear to be available.

3 Summary

The breakdown of the grading at each of the levels of priority (with sub-requirements and sub-sub-requirements treated equally to top-level requirements) is given in Table 3.

Note that approximately 61% of the grades for all requirements, and 67% for the Priority 1 requirements, fall into the acceptable or acceptable with enhancements desired categories. This percentage does not change significantly if the sub-requirements and sub-sub-requirements are downweighted. The high-severity defects make up only 6% of all levels, or 9% of the Priority 1 items. Approximately 29% of all Priority 1 grades fall into the inadequate or not available category — these are the natural targets for high priority development by AIPS++.

The most notable problem areas are in Calibration and Editing, where 63% of the Priority 1 requirements receive grades of I or N, and Visualization where 50% of Priority 1 requirements receive I or N. These sections have numerous requirements that have millimeter-wave specific features (such as atmospheric modeling) and should get special attention by the ALMA SSR and Computing groups during AIPS++ development in these areas. In addition, the Calibration and Editing and Imaging sections contain requirements for which further scientific input or significant algorithmic development is needed. These are areas in which the ALMA Science IPT should play a role. There are also significant unsatisfied requirements in the Interface section, particularly regarding the GUIs, which cross-cut all the areas impacting usability. Note that no significant benchmarking was carried out during this auditing procedure, and performance issues (which are perceived to be a possible problem with AIPS++) will likely be a driver in the next auditing cycle once these metrics are available.

Note that 60% of all requirements are classified as Priority 1, which is too high for purposes of sensibly determining priorities. We recommend on later auditing and testing cycles that priorities be re-evaluated with the goal of having a more rational distribution of priorities. In addition to priorities of features, later revisions of the requirements should include timescales for deployment of features that are tied to Level 2 and Level 3 milestones for the ALMA project, for example. Finally, performance benchmarking and detailed testing should be folded into the next auditing cycle.

Table 1: Grade Summary

Priority	Number	A	A/E	I/N low	I/N med	I/N high	U
Overall							
1	293	60%	7%	5%	15%	9%	4%
2	135	49%	12%	23%	7%	2%	7%
3	61	23%	13%	49%	3%	2%	10%
all	489	52%	9%	16%	11%	6%	6%
1 General Requirements							
1	12	42%	8%	0%	8%	8%	33%
2	9	67%	0%	11%	0%	0%	22%
3	2	50%	0%	0%	50%	0%	0%
all	23	52%	4%	4%	9%	4%	26%
2 Interface							
1	26	65%	4%	8%	12%	8%	4%
2	22	55%	23%	14%	0%	5%	5%
3	5	40%	0%	60%	0%	0%	0%
all	53	58%	11%	15%	6%	6%	4%
3 Data Handling							
1	91	86%	1%	1%	3%	8%	1%
2	24	88%	0%	4%	0%	0%	8%
3	12	33%	0%	67%	0%	0%	0%
all	127	81%	1%	8%	2%	6%	2%
4 Calibration and Editing							
1	51	27%	4%	10%	35%	18%	6%
2	15	20%	0%	33%	27%	7%	13%
3	10	0%	10%	60%	0%	0%	30%
all	76	22%	4%	21%	29%	13%	11%
5 Imaging							
1	24	38%	29%	0%	4%	17%	13%
2	13	46%	15%	8%	23%	8%	0%
3	1	0%	0%	100%	0%	0%	0%
all	38	39%	24%	5%	11%	13%	8%
6 Data Analysis							
1	45	73%	9%	0%	13%	2%	2%
2	29	45%	28%	24%	3%	0%	0%
3	17	29%	29%	41%	0%	0%	0%
all	91	56%	19%	15%	8%	1%	1%
7 Visualization							
1	36	39%	11%	14%	28%	8%	0%
2	11	27%	9%	36%	9%	0%	18%
3	9	11%	22%	11%	11%	11%	33%
all	56	32%	13%	18%	21%	7%	9%
8 Special Features							
1	8	63%	0%	25%	13%	0%	0%
2	12	17%	0%	75%	0%	0%	8%
3	5	20%	0%	80%	0%	0%	0%
all	25	32%	0%	60%	4%	0%	4%