

Internal Memo

Title: DFO Data Association I. General rules and concepts

From: Reinhard Hanuschik
To: DFO team, DFS team
Date: 2003-07-21
Purpose: Description of general association rules

0	Applicable documents.....	2
1	Acronyms and dictionary.....	2
2	Introduction.....	2
3	Goal.....	3
4	Rules.....	3
5	DFO association rules.....	4
6	Basic concepts.....	4
6.1	Basic data sets.....	4
6.2	Complete vs. incomplete basic data sets.....	6
6.3	Time match rules.....	6
6.4	Coherent calibrations.....	7
6.5	Calibration cascade.....	7
6.6	From Reduction Blocks to Association Blocks.....	7
6.7	Primary raw files.....	7
6.8	Match keys.....	8
6.9	Chained and virtual calibration products.....	8
7	Association maps.....	9
8	Rules: common-property vs. different-property.....	11
9	Lifetime of Association Blocks.....	11
10	Results.....	11
	Appendix.....	13

Change record:

version 1.0	2003-01-23
version 1.1	2003-01-31: lifetime of ABs added plus other dfo comments
version 1.2	2003-02-03: dfo comments from meeting 2003-02-03 included
version 1.3	2003-07-21: time matching rules, single-property rule added

0 Applicable documents

- [1] DFO Data Association. II. Association Blocks (R. Hanuschik): Internal Memo to DFO group (version 1.2, 2003-07-21);
[2] DFO Data Association. III. General tools (R. Hanuschik): Internal Memo to DFO group (2003-02-12)
[3] Breakpoint editor (R. Hanuschik): Internal Memo to DFS (2003-01-23)

1 Acronyms and dictionary

AB	see: Association Block
Association Block	Contains the <i>complete association information</i> about a set of input frames.
Association map	The high-level description of the association rules. They provide an intuitive access to the most fundamental components.
DataOrganizer	classical tool for data organization and association
DO	see DataOrganizer
Basic data set	Set of input data that is complete, independent, and minimal (can be processed without knowledge about other data sets). See Sect. 6.1
Breakpoint	incidence of a major instrument change; associations are not allowed to cross breakpoints (see [3])
Cascade	Within a basic data set, calibration data are organized in a calibration cascade. The cascade is a two-dimensional scheme describing all calibration raw types and their mutual relation. For each raw type, there is one column. Rows describe relations between products and raw data. See Sect. 6.5
Coherent calibrations	All calibrations taken under the same instrument conditions can be considered as coherent. Broken by breakpoints. See Sect. 6.4
Match key	The set of instrumental parameters that defines a basic data set. See Sect. 6.8.
Primary raw file	Leaves its name to an AB. See Sect. 6.7.
RB	see: Reduction Block
Reduction Block	An ASCII file with names of raw files and associated calibration products.
Virtual calibration product	exist, at the time of AB creation, as names only

2 Introduction

Some of the core functions of DFO are data processing and data distribution. Both tasks require data classification and data association. Classification and association are tightly linked.

Data association means finding all data which are related to each other by type, by strategies for acquisition and reduction, and by time. All associated raw data form a logical data stream that could not be subdivided further without loss of information, but can be processed ignoring other data streams.

Data association is a core issue in many DFO applications. To mention just a few, there is:

- data organization, with the aim to prepare Reduction Blocks for pipeline processing
- preparing data packages for a specific run
- managing the calibration solutions.

Data association is also relevant for many applications beyond DFO, e.g. for the archive to find calibration data associated with science data; for the end user to navigate through his data package etc. Data association is also dealt with on Paranal, albeit in a very basic way since a continuous data stream is analyzed sequentially with little knowledge about previous data, and no knowledge about future data.

The core of data association is the complete knowledge, the formulation, and the proper application of the *association rules*. Once the rules are known and properly applied, the rest of the task is straightforward. All it needs then is a complete source of information about the data to be associated, which, depending on the application, may be e.g. the raw data archive database or a local repository hosting file lists.

The hitherto existing association tools use different and incomplete association rules, which may even be wrong if applied in other cases. They use different syntaxes and have different historical contexts. Such tools are:

- Data Organizer
- cdPacker
- DFO scripts.

Implicitly the concept of data association is also included in databases like 'observations', 'idc' or 'qc1' since the proper definition of tables and their keys requires some knowledge about data types and their association.

A complete set of association rules should combine all relevant rules about internal logical connection between data of a specific instrument. This set must reflect the calibration plan, the pipeline architecture and the structure of the instrument.

As a step towards a general-purpose set of association rules, an evaluation of the requirements of DFO about association has been undertaken. The purpose of this review is to collect all aspects of data association which are relevant from the DFO perspective. Because of the manifold and complex tasks of DFO it is likely (though not guaranteed), that the DFO view is complete and can be regarded as a superset for other applications dealing with association.

This report deals with the high-level concepts for a general association tool. A separate document [1] describes the Association Block which has emerged from these concepts. A third document [2] describes a prototype association tool.

3 Goal

Our goal is to obtain a formulation of association rules which is generally usable, i.e. across all VLT instruments. This includes a formulation of rules and properties *at high level*. People think about complex rules, like association, in a graphical way. Therefore a high-level description of rules should use flow charts. These can then be transformed into more technical configuration files or database tables. Any tool which has to provide associations will then read and interpret these files or databases. If the implementation of the rules is done properly, this last step is almost trivial. The main focus has to be on the formulation of the rules.

4 Rules

The search for a complete set of rules has been stimulated by the fact that each DFO scientist has an 'intuitive' picture of such rules. There is also the common view that such rules are complete since otherwise we would be unable to provide data packages.

It is worth mentioning that the assumption that *coded rules* for association do exist is fundamentally important and non-trivial. The fact that rules exist is a consequence of the VLT

standards. Non rule-based approaches have been, and still are, widespread at other observatories. E.g., a classical approach for associating all data for a package is to collect all data from an instrument of a certain night and ship them to the visitor. There is only implicit knowledge that these data are coherent, but no explicit signature like e.g. fits headers.

We can formulate the **first principle of association**:

FOLLOW RULES AND PROPERTIES

This means: do not associate using names, nightlog entries, phone call instructions. The result of an association should be reproducible by a different person two years later with the same results (provided there is the same input data set).

5 DFO association rules

The following steps in the DFO workflow deal with association and are evaluated here:

- data organization
- data packing
- managing qc1 database content
- calibration product renaming

An interesting result of this review is that each of these processes uses a different subset of the general rule set. There is no association process which explores all aspects of association. This is why finding a common and complete set of rules is not trivial, and this is why even within the DFO scripts association rules are coded several times in several ways.

To find a proper general formulation of the association rules, it is useful to describe the association process in an abstract way:

Association is selecting OBJECTS which have common PROPERTIES and follow common RULES.

Most important OBJECTS in the DFO workflow are raw and product frames, and product tables. A more extensive list of objects is found in Table 1.

Objects can have different PROPERTIES or attributes. E.g., raw frames used for calibration come in a number of types. Calibration products may, or may not, be used for science reduction; they may, or may not, be used in a calibration cascade (see below) etc. A list of properties is given in Table 2.

Finally, the association of objects with properties follows rules. E.g., in a calibration cascade the raw calibration frames must be processed in a certain well-defined order, to make their products available to the next higher step. The matching of raw frames with other raw frames follows certain match keys, which in general are different from match keys between raw and product files. Rules are listed in Table 3.

6 Basic concepts

6.1 Basic data sets

The **second principle of association** is:

WORK ON A DATA SET.

Since association is about connecting frames, we need a pool of frames. Within the two main workflows of DFO, these pools are (per instrument)

- a night, or

- a complete run.

Ideally a night is complete which means all calibration data for the science data of a night are found in that data pool. If a night is incomplete, it needs to be completed. Often this can be accomplished by shifting the boundary between nights (which is artificial anyway) from 12:00 to 18:00. Otherwise other nights could be added to the pool.

Since a general association tool always needs a memory about calibration data, which is deeper than a single night [1] [3], completion of a night is not a particularly strong requirement. An incomplete night would be completed by calibration data taken in earlier or later nights.

Typically such a pool has many kinds of data sets which are independent of each other. E.g., FORS1 LSS frames can be processed disregarding IMG frames. But even within an instrument mode there are independent data sets. All data from grism 600V are completely independent of those from grism 150I. Obviously the FORS1 data pool splits into IMG, LSS, IPOL and MOS frames; each of these further splits into independent data sets.

Association has a lot to do with finding these independent data sets which do not split further. These will be called *basic data sets* in the following. A basic data set is one which is **complete, independent, and minimal**.

Completeness. A basic data set is complete if there are no other data needed for association from the same data stream ("no missing skyflats for V_BESSELL").

Independence. It is independent if neither data from other basic data sets are required, nor data from that data set are needed elsewhere. In practice, basic data sets may partially overlap, so independence is a weaker requirement than the other criteria.

Minimal. A basic data set is defined by the smallest set of instrument parameters. It does not make sense to look for more parameters than needed to make a data set basic. E.g., if the grism selected makes the data set already basic, there is no additional information needed about the order separation filter.

Examples. The already mentioned FORS1 LSS data set is basic if the grisms are taken into account. There are 7 grisms.

The VIMOS IMG data stream splits into 20 basic data sets (5 standard filter and 4 detectors).

UVES ECHELLE has 2x13 basic data sets (2 detector modes and 13 grating modes).

GIRAFFE has 5x30 basic data sets (30 wavelength settings and 5 fibre-slit systems).

VIMOS MOS has an infinite number of basic data sets, namely one per mask and detector.

Simple vs. complex modes. Now it should be clear that the complexity of the association task depends on the following components:

- number of basic data streams
- number of raw frame types
- internal complexity of calibration cascade.

Basic data sets are a key concept for association since day-by-day association takes place only within basic data sets. The association task splits into defining the basic data sets, and then applying the association rules for each of them.

Internal complexity of a cascade may be the result of the instrument design, but also of the pipeline design.

There is no relation between basic data sets. However, the concept of basic data sets has to

be relaxed a bit. For instance, most flats need bias frames which have no filter information. Therefore a basic data set for FORS2 IMG needs some data (the bias frames) which are also needed for other data sets. So in general basic data sets will partly overlap (but never completely).

6.2 Complete vs. incomplete basic data sets

Dealing with a basic data set raises the issues of *completeness* and *redundancy*. A basic data set is complete if all calibration data are available to populate the cascade (see Sect. 6.5). If data are missing, an association tool must have a rule about handling such situation.

Likewise there must be rules what to do in case of redundant calibrations. This problem seems to be more widespread than the missing-calibrations problem. Generally, associations between frames are governed, apart from matching instrument parameters, by the **third principle of association**:

FIND THE NEAREST-IN-TIME INSTANCE.

This principle is not sufficient. If there are multiple instances of a file in the data pool (those which are identical in the relevant parameters), additional rules are needed to avoid inconsistencies. It is also true that the nearest-in-time principle is not always appropriate (see next section).

Possible strategies are:

- take multiple instances;
- delete all multiple instances but the latest;
- delete all multiple instances but the first.

Deletion of multiples is not just an option but often a requirement, if inconsistencies in the calibration cascade have to be avoided. For instance, an identical set of biases has been measured in the afternoon and in the next morning. If both are used as master biases in the cascade, it may happen that the first set is associated for science reduction, while the second set is used to create master skyflats. Both choices are the result of the closest-in-time rule. A packing tool will then find both sets of biases later on which may cause confusion and overpopulation of packages.

Examples of useful processing of all instances are flux or telluric standard stars, or wavelength calibrations. These are useful for measuring more rapidly variable parameters.

The nearest-in-time rule is modulated by the requirement to pack all frames of a certain type (e.g. standard stars) from a night.

Another modulation are *breakpoints*. They are introduced by major interventions [3]. The complete formulation of the above rule is: find the nearest-in-time match without crossing a breakpoint. This is implicitly assumed in the following.

6.3 Time match rules

As already mentioned, the closest-in-time principle is not as fundamental as the other principles. It is modulated by breakpoints, by packing requirements (pack all frames of a night) and also by the need to select a different selection strategy.

A common strategy for Service Mode nights is to take attached night calibrations. These are user-defined calibration taken immediately after the SCIENCE data, as part of the same OB (hence *attached*). Of course these calibrations should overrun the daytime calibrations if matching calibrations are needed.

There is also a matching rule that always the *next* calibration has to associated with SCIENCE data. For instance, VIMOS has nighttime calibrations taken by the Observatory

right after each SCIENCE OB, to control flexure. No other calibration can be used for association but the next one since only this calibration has been taken in exactly the same instrument position as the SCIENCE data.

To conclude, there is the need for three time matching rules:

- CLOSEST: take the calibration having the smallest difference in time
- ATTACHED: find a calibration taken with the same OB_ID
- NEXT: take the calibration having the smallest difference in time and being taken *after* the SCIENCE data.

6.4 Coherent calibrations

An important assumption about calibrations taken in sequential order is that the instrument status does not change during the time required for taking calibrations. This is why the frequency of calibrations should be determined by the instrumental stability. Critical, unstable components need to be monitored more frequently than stable ones. Usually it is assumed (and has to be proven from time to time!) that instrumental conditions are similar (at a stated precision level) between night and day so that daytime calibrations become feasible. In cases when this is not true, night calibrations have to be scheduled.

All calibrations taken under the same instrument conditions can be considered as coherent. Coherence is broken by sudden changes which may be scheduled (filter exchange, mirror recoating) or unforeseen (earthquake). Whenever a sudden change happens, this change breaks the third law of association (nearest in time). This is why breakpoints are needed [3].

6.5 Calibration cascade

Within a basic data set, calibration data are organized in a calibration cascade. The cascade is a two-dimensional scheme describing all calibration raw types and their mutual relation. For each raw type, there is one column. Rows describe relations between products and raw data. E.g., a master bias, generated by the first step in the FORS1 IMG cascade, is needed further downwards in the cascade to process a master skyflat. A calibration cascade is visualized by an association map (Sect. 7).

6.6 From Reduction Blocks to Association Blocks

A *Reduction Block* (RB) is an ASCII file with names of raw files and associated calibration products. It is used to feed a pipeline with a reduction job. It can be considered a specialized set of association information.

This concept can be generalized to the concept of *Association Blocks* (AB). The AB contains the *complete association information* about a set of input frames. It is more extended than the RB (it may have product file information, QC1 parameters etc). It is also more general since it can be defined and created in an identical way for any kind of data dealt with by DFO, both from pipeline-supported and unsupported modes.

An AB may link to other ABs within the cascade. Association Blocks are created at the data organization step. They are updated at later steps of the DFO workflow. They are finally evaluated to extract all information needed for packing.

ABs have to have such a structure that it is always possible to extract part of their information into an RB.

The concept of the Association Block is fundamental for association. Association Blocks are described in [2].

6.7 Primary raw files

A primary raw file is one which leaves its name to an AB. Depending on the association rules, this can be a single raw file, the first one in a template, or the first one in a night. This is a generalization of the RB concept which also has RB names derived from primary raw files.

6.8 Match keys

The set of parameters which defines a basic data set is called a *match key*. All raw frames in a basic data set share the same match key. Parameters correspond to FITS keywords. Often they are instrument parameters, but they may also come e.g. as PROG.ID or OBS.ID.

More generally, match keys are needed in four varieties:

- match keys between raw files, used to define a basic data set;
- match keys between raw files and product files, used to associate raw files and product files within a basic data set;
- match keys between product files, used to define validity chains;
- match keys used for packing.

These various kinds of match keys are often, but not always, identical within a specific calibration cascade.

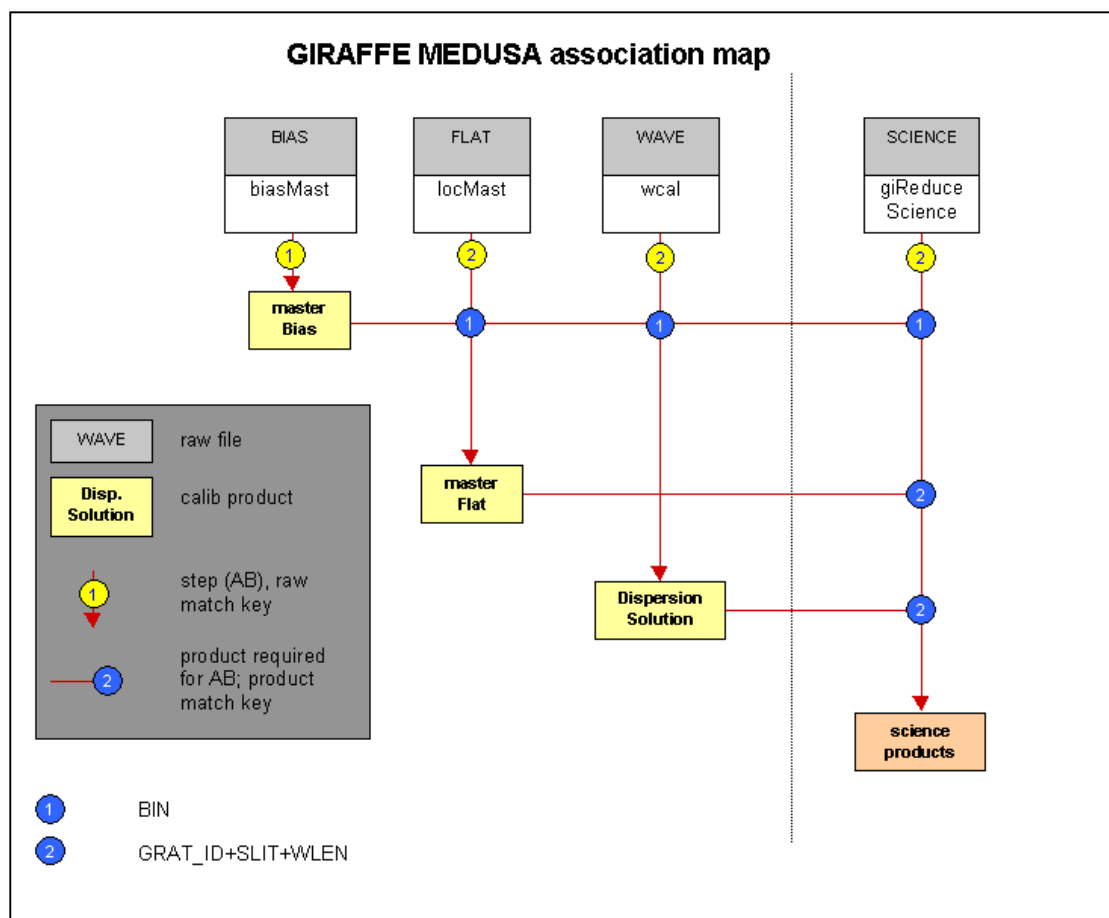


Figure 1. The GIRAFFE calibration scheme (MEDUSA mode), as an example of a simple association map.

6.9 Chained and virtual calibration products

Usually the creation of calibration or science products requires calibration products from an earlier step in the cascade (*chained calibration products*). At the time of creating the ABs, such products will generally not yet be processed and available. Their names, however, can be *predicted* on the basis of the association rules. Thereby, these names are made available for use in follow-up ABs which require those products as input. Since they exist, at the time of AB creation, as names only, they can be called *virtual calibration products*.

If processed in the proper order, the ABs produce, step by step, all required products, turning virtual into real calibration products.

Of course this concept works fine only if no RB with chained calibrations fails. If it does, all following RBs from the same basic data set will also fail.

7 Association maps

Association maps are the high-level description of the association rules. They do **not** completely describe all association information, but provide an intuitive access to the most fundamental components.

A simple association map is shown in Figure 1. A more complex scheme is presented in Figure 2.

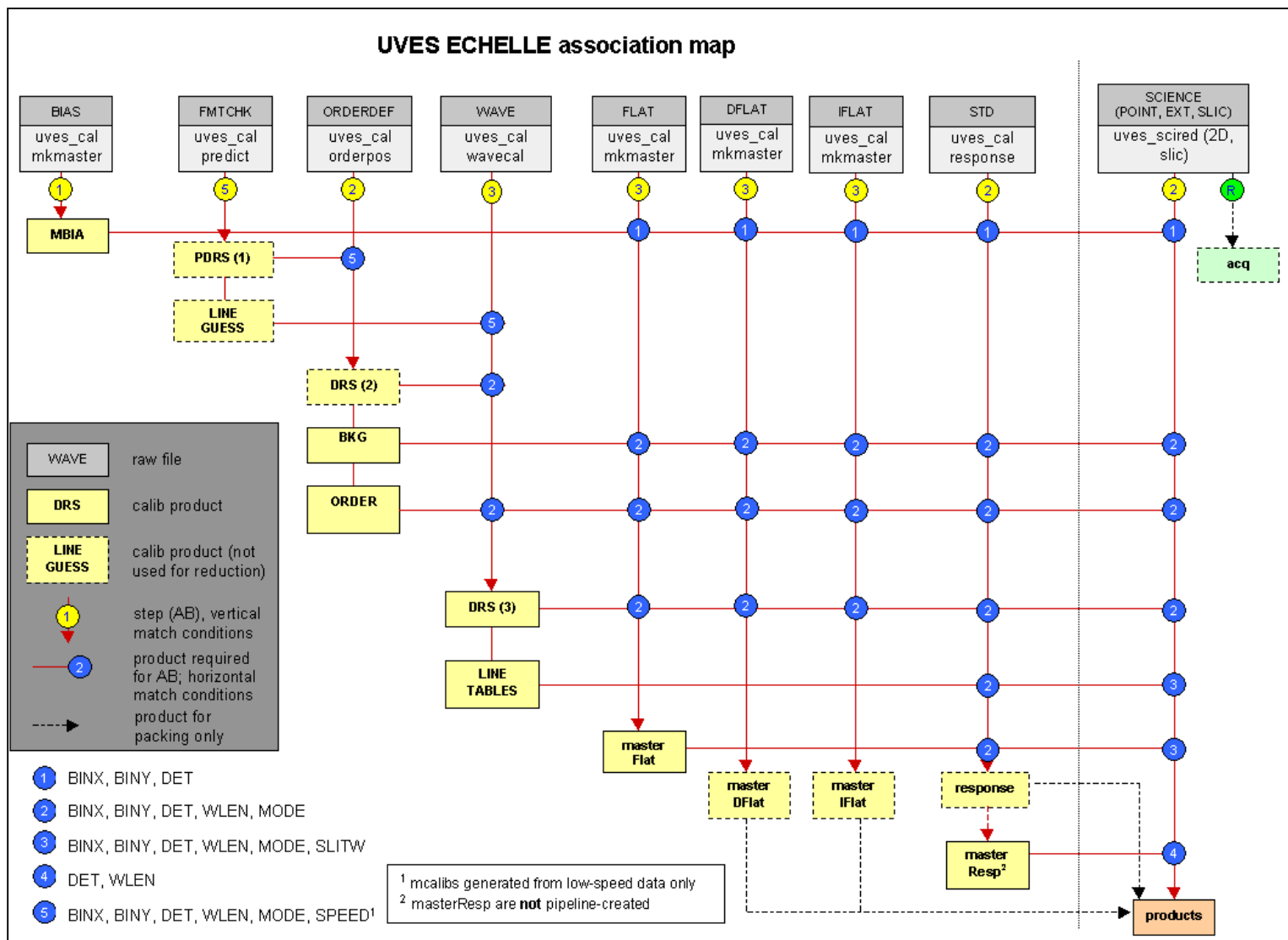


Figure 2. The UVES calibration scheme (echelle mode), as an example of a complex calibration map.

8 Rules: common-property vs. different-property

It turns out that the first principle of association is not sufficient to describe rules properly. There are rules which can be matched by a general, instrument-independent DFO association tool, and there are others which cannot.

The simplest rules are *common-property* rules. That is, match [object1 having property1] with [object2 having the same property].

Different-property rules are e.g. match [object1 having property1] with [object2 having property2].

A variation of this type of rule is the *fixed-property* rule: match [object1 having property1] with [object2 having always property2]. For instance, match SCIENCE data, having whatever value of slit width, with STD calibrations taken at 10" slit width.

There can also be *conditional* match rules, like: match [object1 having property1] with [object2 having property2], if property1 has value 1; do something else if property1 has a different value.

The **fourth principle of association** now is:

USE COMMON-PROPERTY OR FIXED-PROPERTY RULES ONLY.

Anything else is not unlogical but simply too hard to match by a common tool (as is also illustrated by the fact that none of the existing DFS tools can cope with them). Such cases should be resolved into common-property (or fixed-property) rules to fit into the scheme, e.g. by creating new data types, or find different match keys.

9 Lifetime of Association Blocks

Within the DFO workflows mentioned here, ABs are created and updated. Once they have been used for packing, they are not used anymore.

However, in a more global framework it seems useful to store all ABs in a database (AB database). This database could be used by DFO when ABs are updated, and when they are read for preparing data packages. If inserted into, or linked to, the archive database, their association information could also be used by archive tools.

In that sense, an Association Block becomes a set of information associated to a raw file (or a set of raw files), in the same way as e.g. the QC1 parameters. This is both useful and logical. Implicitly association information is also quality information. We have

- used selection criteria to associate calibration data with science data, and
- collected quality statements by checking if raw files have got products.

10 Results

1. Description of association rules in a general way is possible if the four general laws of association are respected:
 - follow rules and properties
 - work on a data set
 - evaluate a time-matching rule to find the association partner
 - use common-property or fixed-property rules only.
2. This description is complete for all VLT-compliant instruments so far investigated.
3. This description is complete for all DFO-relevant workflow steps.

4. The description of the most important association rules can be done in a graphical way, the association map.

Appendix

Table 1: Association objects

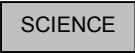

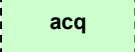


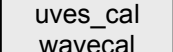
OBJECTS	varieties	signature in association map
raw frames	science	
	calibration	
	others (acquisition, test)	
product frames and tables	calibration products	
	reduced science	
log information	rblogs	
plot information	ps, gif etc.	
other information	e.g. PAF, INI	
association information	AssocBlock	
	ReductionBlock	
recipe names	e.g. uves_cal_mkmaster	
database tables	e.g. fors1_wave in qc1 database	

Table 2. Properties of association objects

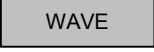


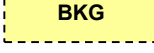

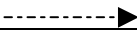


PROPERTIES	varieties	signature in association map
file types	raw file types	
	calibration product files	
calibration product ...	used for science reduction	
	not used for science reduction	
raw file ...	used for processing and packing	
	only used for packing	

Table 3. Rules for association objects

RULES	varieties	signature in association map
cascading sequence		(left to right)
match keys	raw match keys: raw to raw	
	product match keys: raw to product	
	primary match keys: product to product	
	packing match key: RUN_ID	
deletion rules	use for associations / hide	