JRA4-TRE-2000-0001

Revision : 2.0

Date : 18/11/2003

JRA4

SOFTWARE PRELIMINARY CONCEPT DESCRIPTION

Gilles Duvert (Gilles.Duvert@obs.ujf-grenoble.fr)
JMMC/LAOG

Author : Gilles Duvert Institute : JMMC/LAOG	Signature : Date :
Approved by : Institute :	Signature : Date :
Released by : Institute :	Signature : Date :

CHANGE RECORD

REVISION	DATE	AUTHOR	SECTIONS/PAGES AFFECTED
	REMARKS		
1.0	17/07/2003	Gilles Duvert	All
	First Draft		
2.0	28/11/2003	Gilles Duvert	All
	Comply with JRA4 document template		

TABLE OF CONTENTS

1 Introduction

The main objective of the JRA4 JRP is to develop the capabilities of the new large optical interferometers, with a special interest towards the VLTI. It is expected that the scientific productivity of the VLTI depends strongly on the availability of adequate algorithms and tools for the analysis and interpretation of the data. The WP2 of JRA4's JRP focusses on the software developments needed to fulfill this objective. This document tentatively adresses the requirements and general structure of the off-line data reduction, exploration and analysis proposed in WP2.

2 Requirements

a) The first requirement is obviously the existence of a User Requirement Document, to be written during the 1st semester of activity, with inputs of all the JRA4 members.

b) Forthcoming this document, a number of requirements can be formulated to introduce a discussion:

- The software developments endorsed by JRA4 for WP2 should take into account the fact that data quality for VLTI instrument is and stays the responsibility first of the consortia building the instruments, then ESO, and that each ESO instrument produces debiassed and reduced data in form of (today) FITS tables of a common accepted format.
- Similarly, it should leave the responsibility and maintenance of specialized pieces of code involving some kind of expertise in the hands of individual or groups that are best suited to improve and maintain them.
- It should involve techniques as close as possible to the ones contemplated in other current developments. We cite two of them, ALMA and the IVOA:
  - *ALMA*: this future radio-millimeter interferometer is the next big instrument operated by ESO and is currently driving ESO towards new software solutions (both for system operation and data reduction)
  - *the International Virtual Observatory Alliance*: optical interferometry should(modestly) participate in the developments of, and comply to, future tools aimed at interoperablility, data exchange, data mining and more generally the use of astronomical archives as an integrated and interoperating virtual observatory.
- The software must comply with a number of directives shared by other,independent, efforts of the astronomical community: We list here a slightly amended excerpt of the OPTICON proposal for future software environments (activity number 10, which in retrospect appears to lag a bit behind our expectations).We note that these topics are for the most part driven by the need to provide tools that can be used by non-specialists of the field (non-specialists but still trained scientists).
  - Data quality: *Characterization of data quality must be done to allow comparison and joining of different data sets. This is all the more critical in a world of interacting, distributed data holdings.(link with the "naiveté" of the user)*

- Propagation of errors: *Many new results are based on features detected at a marginal level in data sets. In order to quantify the significance of such claims, it is essential to have a firm trace of errors associated to individual values of the original raw data to uncertainties related to the final derived quantities.(idem)*
- Statistical analysis: *Even with reliable errors known for a data set, one requires access to elaborate statistical tools to obtain safe estimates of the significance of the hypotheses being tested.(ibidem)*
- Modeling: *For the physical interpretation of observational data,it is often necessary to compare them with theoretical models, either analytically or numerically, to understand the relative importance of different physical parameters. One needs to integrate such theoretical models into the data analysis.*
- Access to databases: *To compare a given data set with other similar ones it is often necessary to obtain data from remote databases.*
- Composite data types: *Most data have relations in the sense that they qualify other items (e.g. errors related to measured quantities, point spread function related to a spectrum). This can be a complex relation, but can often be mapped in a hierarchical structure.*
- Extraction of quantities: *After calibrated data have been obtained, the major task is to extract physically meaningful parameters.*
- Interface to other packages/openness: *It is neither possible nor reasonable that any single system can provide all features needed for the analysis of science data.*

3 Towards A Comprehensive Reduction, Exploration and Analysis tool for Optical Interferometry

a) To answer at least partially the abovementioned requirements, it is proposed that the software should incorporate:

- User Interface(s)*: these should run in client mode on any kind of OS/hardware. They should exist in at least two flavors, one for the general astronomer, one for the optical interferometry specialist. They should be kept as much as possible independent from the underlying data structure and data and method access.*
- Distributed modules*: these will be developed and maintained by subgroups, consortia.It is expected that most of the necessary expertise lies within the EII groups and most of the relevant code is already written and could be standardized. Reuse of old proven code and recipes is recommended.*
- Common data access*: We need to exchange structured data between the various packages involved. The software will support OI FITS tables (as defined by IAU) at beginning, however we shall consider using rapidly the VOTable as a more extendable and 'self explanatory' format, made for interoperability. It is a role of the JRA4 WP2 members to participate to the IVOA groups developing those formats.*

- Common interfaces*: we need a common descriptor for methods invoked in the processing of the data. One should consider XML and Corba, possibly using a superset of these (look at ESO's ACS developed for ALMA?)*
- A software bus*: to access distributed methods and link to the User interfaces.*
- *The user will need to replay its actions on the same data, and apply this scenario to other data. This implies that a scripting language is used at a lower level.*

  Developers and JRA4 working groups will want to use a scripting language also to quickly implement a new algorithm or test a new idea.

  This scripting language should thus be present behind the scenes: we should consider using Python for scripting, since it has been largely adopted by the community (noticeably used in already 1 ESO instrument and proposed for ALMA).

b) Sketch of the proposed solution
The following figure shows the structure of the software system proposed, with a color code corresponding to the various responsibilites encountered in the different modules: Red for ESO and Instrument Consortia (independent of EII although ESO is part of EII), Blue for JRA4 WorkPackage 2, Green for JRA4 special interest groups (with the example of the PRIMA astrometry consortium).

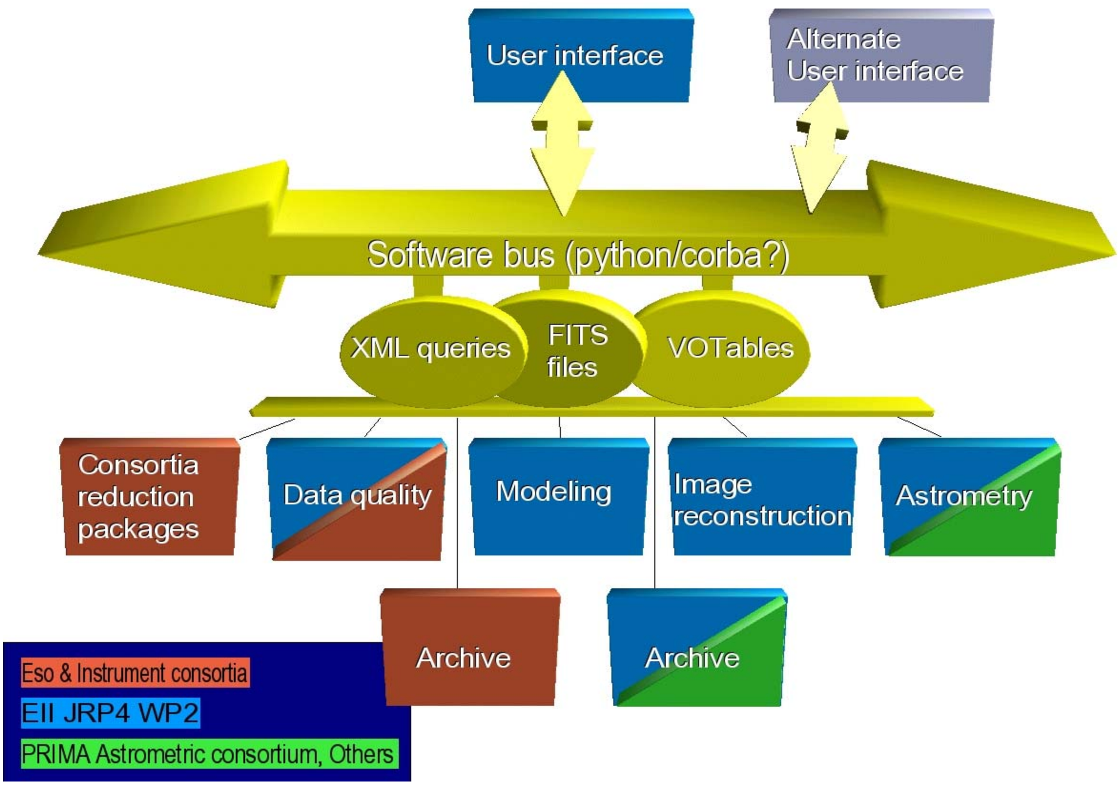

Figure 1 : Sketch of the proposed architecture for JRA4's Data Reduction, Exploration and Analysis Package

c) List of recommendations

- keep the expertise where it belongs. Assume that expert people, like the instrument consortia, know what they do. Conversely, insure that the data produced by the experts has all the 'good' interchangeabilty properties by defining properly the exchange data format. Example: data should always be transmitted with all error estimates, covariance matrix etc...
- Divide the problem into easly maintained (by individual, group or expertise centers) parts.
- Use the object-oriented and distributed software techniques to glue all the necessary components
- Use a portable and evolutive user interface for the end user
- Do not impose methods on the end user, he may be more clever than the specialist or the programmer!