26th of September Poznan

PSNC and JIVE Meeting

EXPReS project
(FABRIC)

Meeting minutes

Table of Content

Table of Figures

Document History

File name	Date	Remarks
psnc_jive_meeting_notes_26_09_2006_v_0.1.doc□	27.09.2006	Draft
psnc_jive_meeting_notes_26_09_2006_v_1.0-final.doc	18.10.2006	Final

1 Meeting information

1.1 Location and duration

The meeting was hosted by PSNC, Poznan, Poland 26th of September 2006.

1.2 Participants

The following table summarizes the list of participants.

JIVE		
Name	**Position**	**Email**
Ruud Oerlemans	Software developer	oerlemans@jive.nl

PSNC		
Marcin Okoń	System analyst and developer	hawky@man.poznan.pl
Dominik Stokłosa	System analyst and developer	osa@man.poznan.pl

2 Software Correlator (WP2.2)

This section discusses various design issues and design choices. Also a first version software correlator is described.

2.1 General application characteristics

- Object oriented using C++. Advantages: code reuse and easier adaptable code, many libraries with useful functionality available, fast processing
- Using FFTW library for Fast Fourier Transform. Proven, fast, constantly upgraded and free
- Way of execution controlled by control file with keywords and values. This file is processed by a parser

2.2 Functionality

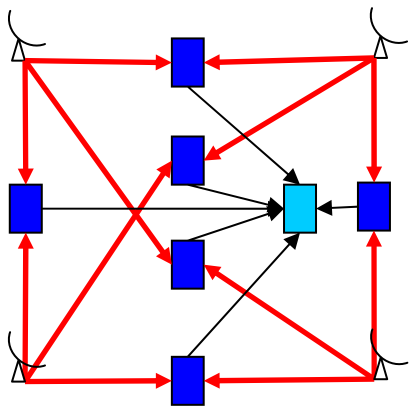
The processing of the recorded data is divided into:

- Station based **pre-correlation**: data extraction, optional filtering, delay correction
- Base-line **correlation.** Auto and cross correlation

2.3 Data distribution

Available for correlation: clusters and multiprocessor computers on various grid nodes in different physical locations. How are we going to distribute the correlation over these grid nodes and their processing cores? There are two levels of data distribution: over the grid nodes and over the cores in a cluster or multiprocessor machine. Both levels will be discussed. The software architecture and implementation will depend upon the distribution choices at both levels. Firstly, we will discuss data distribution over grid sites /nodes. This is high level distribution. Please note that the blue rectangles represent grid nodes/sites in all of the figures below.

2.3.1 Baseline slicing



- Further slicing done at grid node

Pros

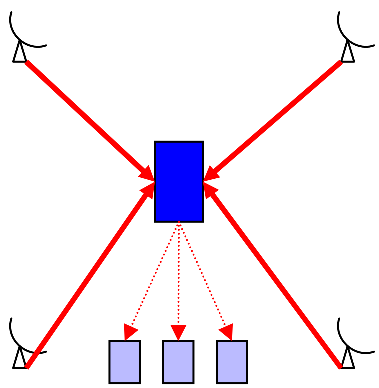
- Small nodes
- Simple implementation at node

Cons

- Multiplication of large data rates, especially when number of baselines is large
- Data logistics complex
- Scalability complex

Figure 1. Baseline slicing

2.3.2 All data to one grid site



- Data slicing at the grid site: time and channel slicing possible

Pros

- Simple data logistics
- Central processing
- Live processing easy
- Dealing with only one site

Cons

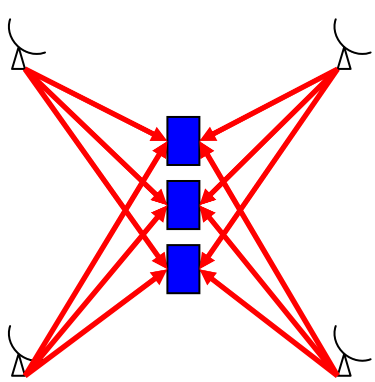
- Powerful central processing site required

Alternative

- All data to central storage and then distributed to compute nodes

Figure 2. Data transfer to one grid node

2.3.3 All data to different sites



- Data slicing at the grid sites: time and channel slicing possible

Pros

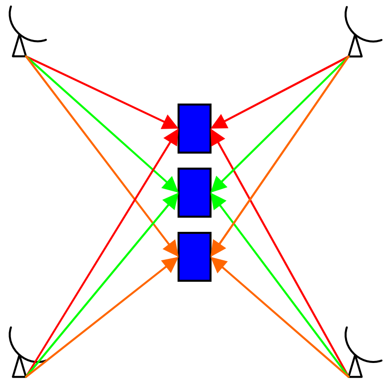
- Small nodes
- Live processing possible

Cons

- Multiplication of large data rates
- Simultaneous availability of sites necessary when processing live

Figure 3. Data transfer to different sites

2.3.4 Time slicing at telescopes



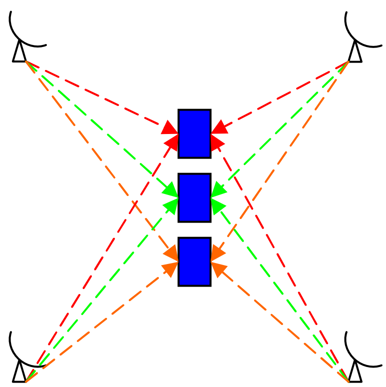
- Large time slices to different nodes, further slicing at grid nodes

Pros
- Small nodes
- Smaller data rates compared to 2 and 3
- Easy scalable
- No data multiplication

Cons
- More complex data logistics after correlation
- Live correlation more complex

Figure 4. Time slicing at telescope side

2.3.5 Channel slicing at the telescopes



- Each channel to a different node, further slicing at grid node.

Pros
- Small nodes
- Live processing per channel
- Easy scalable
- No data multiplication

Cons
- Channel extraction at telescope necessary. This increases data rate because of lower information density
- One node per channel

Figure 5. Channel slicing

2.4 Data distribution over cores in a cluster

Low level distribution.
This can hardly be regarded separately from the distribution of the functionality. See further paragraph 2.5.

2.5 Distribution of functionality

2.5.1 Distributed functionality

The data recorded at the telescopes have to pass different processing steps before the end product has been obtained, see paragraph 2.2. Each of these steps can be implemented in a different application. This enables distributing the different applications over different cores in a cluster. Intermediate results have to be passed to the next application. However

it is not always possible to have as many cores as there are instances required of the various applications. Especially when the number of stations and channels increases a shortage of cores will occur.
Suppose we have 7 telescopes each recording at 4 channels. We need at least: 7 cores for reading the Mk4 file, 7 * 4=28 cores for the station based processing and 4 cores for the actual correlation. Together that makes 39 cores.
A different number of stations and channels results into a completely different number of cores. Scalability is therefore rather complex for this type of distribution.

2.5.2 Centralized functionality

All processing steps are implemented in one application.
Time slicing the data from one channel. The degree of time slicing of the incoming data depends upon the available cores in the cluster. So this application is easy scalable according to the number of available cores in the cluster. Other channels can be processed either on a separate clusters simultaneously or can be processed consecutively on one cluster. Scalability is very simple and data handling also
Channel slicing and no time slicing. All channels can be processed simultaneously as long as the number of channels is not larger than the number of available cores. When the number of channels is larger than the numbers of cores they have to be processed later.
Choice: centralized functionality and time slicing.

2.6 Off-line versus real-time processing

The long term FABRIC goal (see 4.3) is real-time correlation of a small astronomical experiment using a software correlator running on grid nodes. However off-line processing is much less complicated than real-time processing. Therefore a first step is to develop a software correlator for off-line processing (see paragraph 2.7). This correlator will be run on various grid nodes using various input sets to gain experience with a software correlator on the grid. Also benchmark tests have to be done to see what processing power is needed. These experience will be used in the next step, the development of a real time software correlator.

2.6.1 Off-line processing

Processing is done well after the conclusion off the astronomical observations. Data are available on hard disks and can be processed on the grid nodes when they are available. Data processing does not interfere with the astronomical observations and the processing capacity does not have to keep pace with the incoming data flow of the observations.

2.6.2 Real-time processing

Grid nodes have to be available during the astronomical observations and they have to be able to process large very data flows. Real time processing also requires more reliable processing hardware and software.

2.7 Correlator architecture for off-line processing

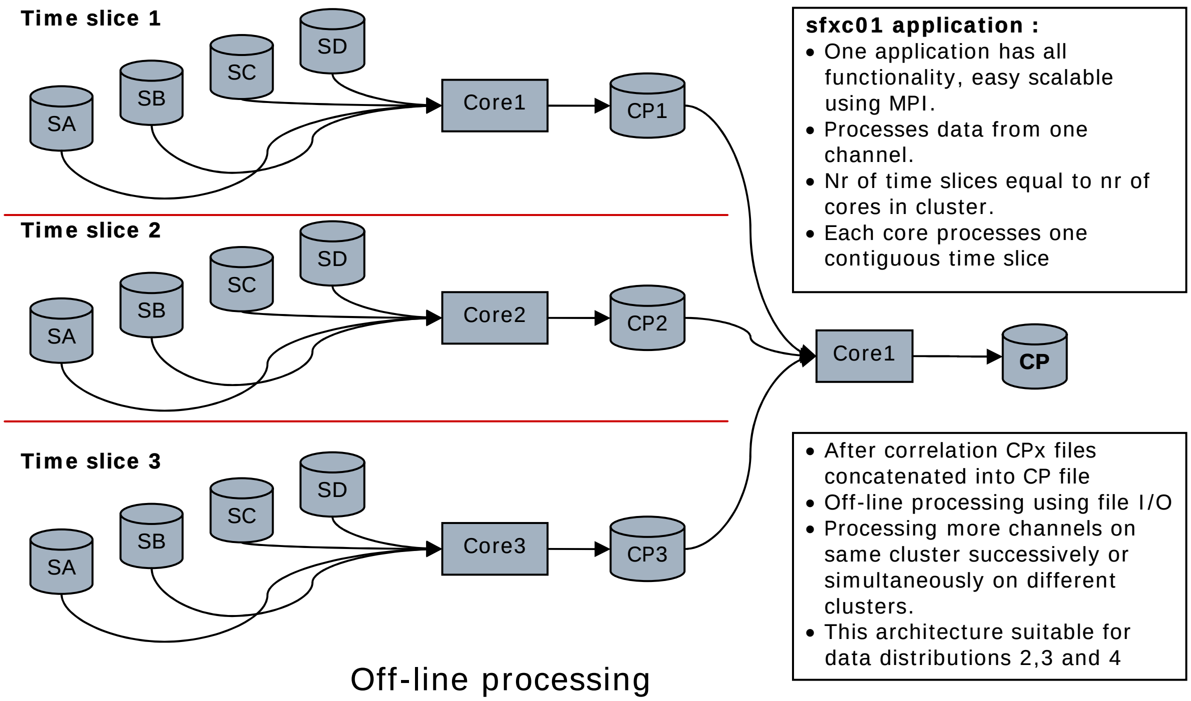


Figure 6. Off-line processing

Characteristics of the first sw correlator version: sfxc01.

- Suppose we have a 15 min observation and 3 cores. The processing is than split into 3 slices of 5 minutes which are distributed over 3 cores. Each core processes data from all involved stations.
- Uses data files as input. Data file format Mk4 for Mk5 disks. Files have to be available, therefore the application can only run after completion of astronomical experiment
- Uses a correlator control file (CCF) to describe the processing settings and I/O data (not shown in the diagram)
- This correlator is being implemented now and is scheduled for delivery to PSNC by the end of October 2006. The first purpose is to have a software correlator putting a workload on the grid nodes. In a later stage the contents will be looked at and benchmark test will be used to optimize the source.

Actions in sfxc01

- Add a header class describing the CP files

2.8 Correlator architecture - real-time processing

For real-time processing also time slicing is done, but now much smaller slices. Data is now arriving in continuous streams rather than files.

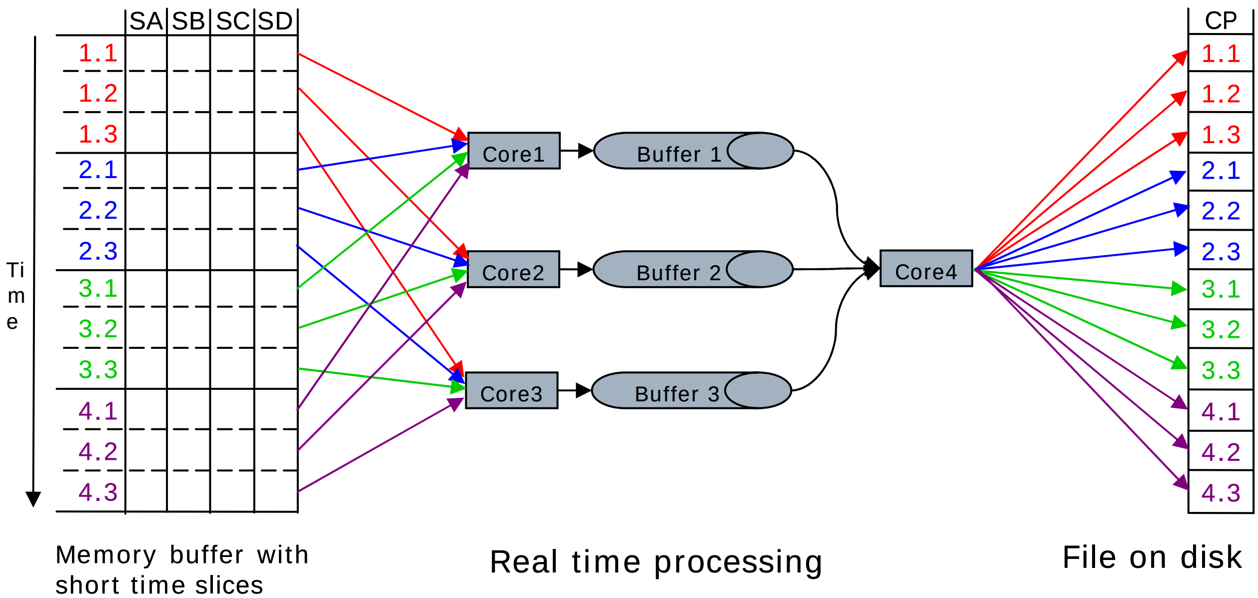


Figure 7. Real time processing

Suppose we have a cluster somewhere on the grid with 4 nodes and 4 radiotelescopes (SA, SB, SC, SD) sending data continuously to the cluster. A memory buffer is filled with short time slices (1.1, 1.2, 1.3). Each core processes a single slice and puts it in a memory buffer. Once all slices in the memory buffer are processed the results are concatenated and saved in a file on disk by a separate core. In the mean time the memory buffer gets filled with the next short time slices (2.1, 2.2, 2.3). Not shown in the previous diagram is how the memory buffer is filled with short time slices. The processing capacity should match the incoming data flow. The previous diagram shows the processing of only one channel, other channels can be processed simultaneously at other grid nodes.
The real-time software correlator running on the grid is a long term goal (see section 4.3). A more detailed design is necessary.

2.9 Correlator Control File

The correlator control file describes: processing settings, input and output streams or files and some parameters of the astronomical experiment. The complete astronomical experiment is described in a vex formatted file generated by the SCHED program. The relevant parameters are extracted from the vex file and put in the CCF. Other relevant correlation parameters are put in manually by the central operator (see chapter 3).
The structure and parameters in a CCF are determined by: the type of data distribution, real-time or off-line processing and the distribution of the functionality. In this section only some general CCF design issues will be addressed. More specific issues depend

strongly on the application details and are therefore discussed together with the application details.

- The CCF is an ASCII file with keywords and values. It is read by a parser
- The markup language XML is a possible future candidate for a CCF. Parsers and tools are available for XML type files.
- Data distribution. In the section on data distribution two levels of data distribution are distinguished: at grid node level and at cluster level. At grid node level e.g. channel slicing can be used and at cluster level time slicing of the data can be applied. High level control parameters should indicate the distribution of the data
- Application level
- File names: CO not interested in physical file names/location. CO only interested in logical file names. For the time being physical file names are used in the CCF for sfxc01

3 Workflow Manager (WP2.1)

This chapter discusses different design issues and problems connected with the prototype version of Grid Workflow Manager (WFM).

The figure below shows the overall design of eVLBI system and role of WFM module.

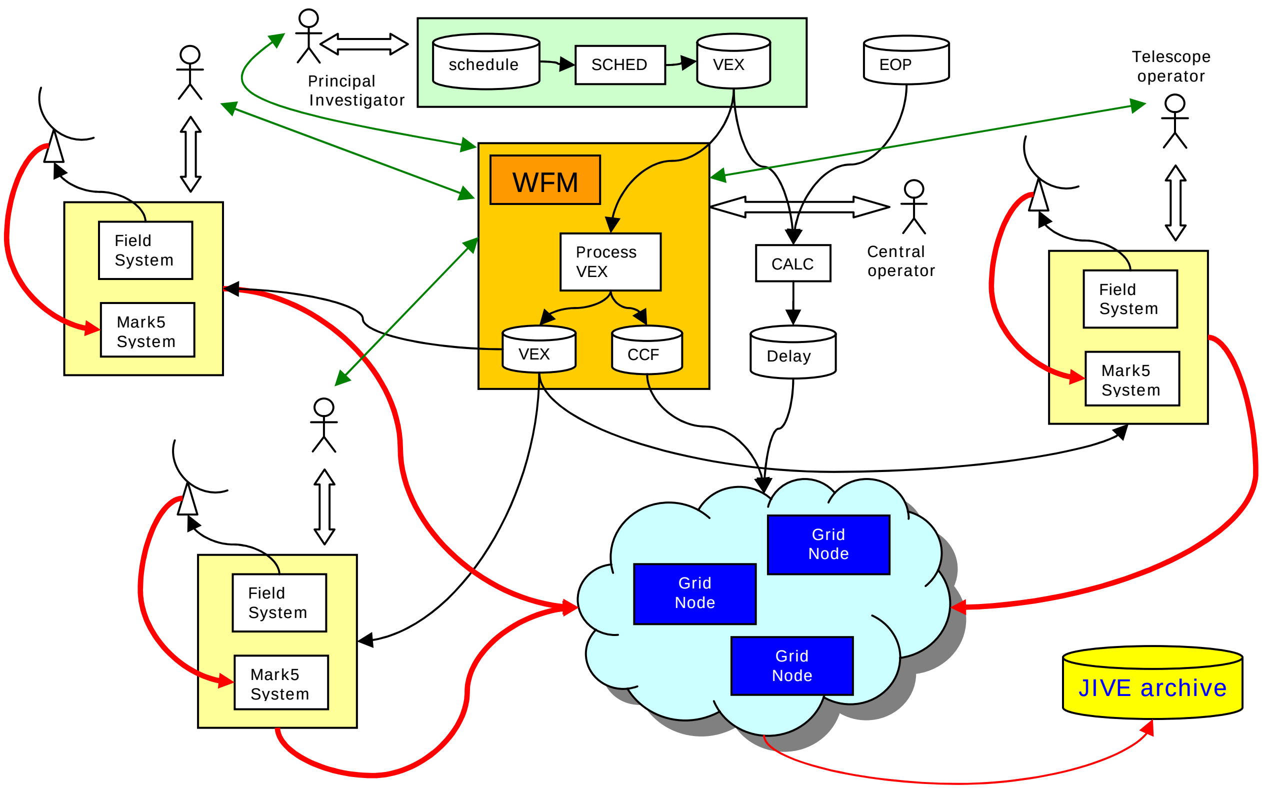


Figure 8. System architecture

The table below describes the meaning of all the arrows used in the system design diagram.

Arrow color	Description
green	communication
red	data flows
black	control information
white	person communicating through interface

eVLBI use case

- PI creates a schedule file using a text editor. Schedule file describes the experiment
- SCHED generates a VEX formatted file

- PI uploads the file to the VLBI database. The CO is automatically informed of the new experiment
- WFM is the central point of control and information for the CO. It acts as a shell around the applications on the grid nodes and provides tools to verify and modify the experiment parameters and to visually define the eVLBI data flow
- CO gets VEX file, displays it in the graphical form using WFM, verifies the parameters and provides additional ones, needed in the correlation process. In the next step, CO makes logical connections between radio telescopes and file servers, and constructs the workflow for data correlation
- During the observations data is recorded by the Mk5 system or transferred directly to the grid. TO's, CO and PI have the possibility to launch the WFM in the read-only mode to monitor current eVLBI state
- The CO creates the CCF and Delay file. The CCF describes how and where data is correlated
- The CO is responsible for starting the correlation process, or it can be started automatically after the data acquisition is complete
- The correlated data are saved in a central archive
- PI and CO are notified by the system that experiment has been processed

Remarks

- For the time being CALC and the creation of the appropriate delay files is outside the WFM

3.1 Graphical user interface of the WFM application

During the meeting we have discussed different approaches on how the WFM application can look like and how it can be used by the eVLBI users. The figure below shows the possible (prototype) design of WFM application.

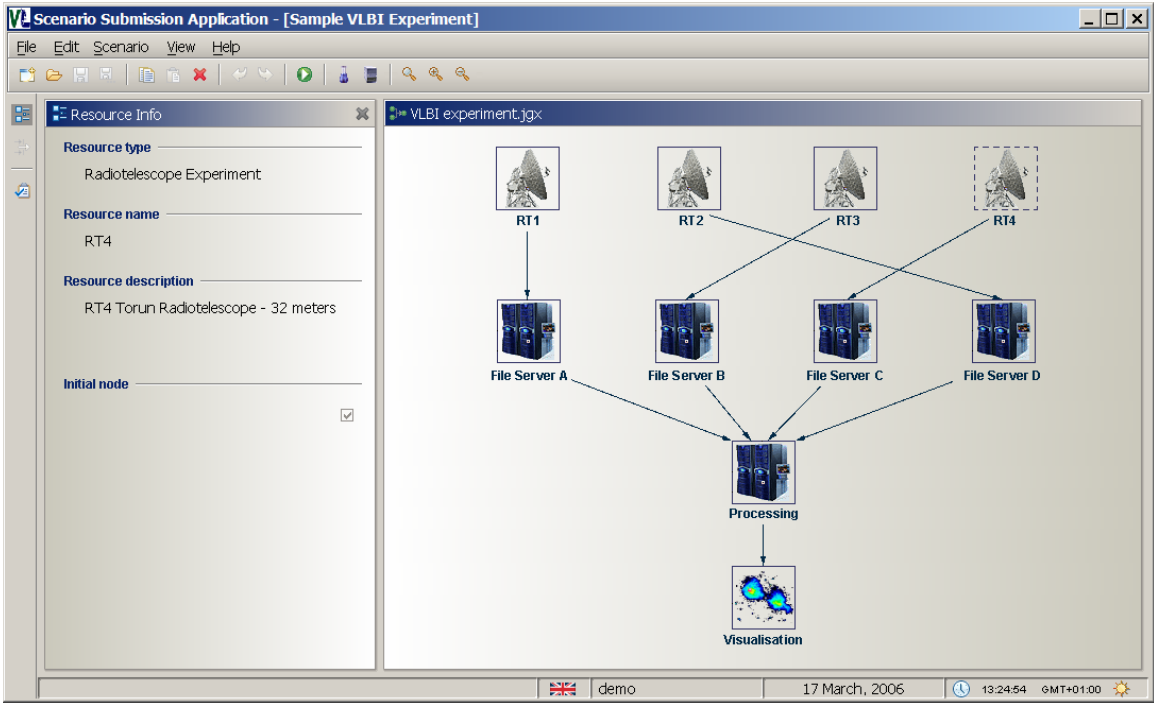


Figure 9. Possible WFM GUI

The items and issues discussed:

- ***Use case scenarios***
  It was agreed that it may be desirable to prepare use case scenarios for the WFM application. We have to decide what role in the overall system design should WFM play. Should it be only a tool used by Central Operator (see figure 13) to submit VLBI experiment to the GRID or maybe we should treat it as a central point application which allows experiment submission, as well as the management at the different levels and different users.
- ***Roles in WFM***
  Possibility of adding roles to the WFM, so it can be used by different users i.e. *Principal Investigator*, *Telescope Operator* or *Central Operator*. This needs to be resolved by JIVE. It was proposed to prepare a use case study which will help us to determine the scope of the application.
- ***Demo version***
  PSNC will prepare a working demo of WFM or interactive presentation of the graphical user interface. It will be decided later which approach will be preferable. The demo will be presented to the VLBI users. We are hoping to gather some remarks which help us to prepare better version of the user interface. The prototype demo version will not contain any functionality.
- ***Monitoring***
  Advanced monitoring issues have been also discussed. Different monitoring information i.e. from the telescopes could be visualized in the WFM application. The user would be able to check current status of his VLBI experiment, notifications send by Grid environment could be also presented to the user. This is stated here just to make sure that the issue will not be forgotten. We will decide later on, whether we

will be able to produce such a functionality within a scope of the FABRIC activity.

- ***Validation***
  Since the WFM application will be presenting VEX file and will allow setting various parameters it is desirable to validate a user input, check constraints, etc. We have discussed two possible solutions: the vex parser and validator will be integrated in the application itself or it will be created as a remote service with WebService technology. This issue requires discussions in the future.

4 Goals, planning and action items

4.1 Short term goals

JIVE		
Nr	**Title**	**Description**
1	vex to CCF	Mapping of vex formatted file parameters to correlator control file parameters. PSNC needs this to parse the vex file and create the CCF
2	WFM functionality	RO discusses with HJvL the required functionality (use cases). Who should have access to the WFM: Central Operator, Principal Investigator, Telescope Operator. What functionality should they have access to. PSNC needs this to create screen shots of the GUI or to create a GUI prototype without functionality
3	Off-line SW correlator	RO delivers by the end of October 2006 a first version of the sw correlator working with file I/O and running off-line with the observations
4	Correlator design specification	WP 2.2.1, Correlator algorithm design. DJ 1.4, Correlator design specifications (RO,HJvL) A formal deliverable to the EU

PSNC		
Nr	**Title**	**Description**
1	GRID environment	Prepare GRID environment for the software correlator, so it can be installed in the GRID
2	Correlator tests	As soon as the first version of the correlator will be delivered by JIVE, PSNC can start work on deploying it on the GRID and make some tests.
3	WFM GUI	Prepare a prototype version or presentation of WFM graphical user interface
4	WFM	Design a prototype version of the Workflow Manager Application
5	Deliverable DJ 1.6	Finish and send to JIVE for a review document "eVLBI – Grid Design Document"

4.2 Mid term goals

JIVE		
Nr	**Title**	**Description**
1	CCF Validator	If possible the WFM GUI will check on the fly the CCF parameters. Parameters that are too complex and cannot be checked by the WFM will be verified by an external CCF validator.
2	CP Convertor	Converts the correlation product from the software correlator into FITS formatted file. FITS is the data format readable by astronomical data processing packages.
3	Mk5 to net	How to get the data from the Mk5 computer/disk to the network or linux type disk.
4	CCF in XML	Investigate the possibility to use XML for CCF and implement if useful

		PSNC
Nr	**Title**	**Description**
1	Data transfer in GRID	We have to decide how we are going to manage huge files. One possible solution is to use Data Management System (DMS). The other is to use solution incorporated in Globus Toolkit (GSIFTP)
2	WFM	Develop first beta release of the Workflow Manager Application
3	Correlator benchmarks	Prepare performance benchmarks of the Software Correlator
4	Grid Broker	A first, limited version of Grid Broker will be created (adopted or modified) and integrated with WFM

4.3 Long term goals

JIVE		
Nr	**Title**	**Description**
1	Real-time SW correlator	Month 14, (May 2007)

		PSNC
Nr	**Title**	**Description**
1	eVLBI experiment	Conduct eVLBI experiment using developed design (4 radio telescopes, 2 – 4 hours with data rate at 128 Mb/s

Definitions, abbreviations, acronyms

CALC	–	Set of programs for analyzing very long baseline interferometry observations made under astrometric and geodetic programs
CO		Central Operator
FABRIC	–	Future Arrays of Broadband Radio-telescopes on Internet Computing, http://www.jive.nl/dokuwiki/doku.php?id=expres:fabric
FFTW	–	Fast Fourier Transform C++ library
GRID	–	Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed "autonomous" resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements.
GUI	–	Graphical user interface
JIVE	–	Joint Institute for Very Long Baseline Interferometry in Europe, www.jive.nl
PI		Principal Investigator
PSNC	–	Poznan Supercomputing and Networking Center, www.psnc.pl
SCHED	–	A program for planning and scheduling VLBI observations
TO		Telescope Operator
VEX	–	VEX = 'VLBI Experiment. It has been invented to prescribe a complete description of a VLBI experiment, including scheduling, data-taking and correlation
VLBI	–	Very Long Baseline Interferometry
WFM	–	Workflow Manager, application developed in the WP2.1 (FABRIC)
XML	–	Extensible Markup Language, http://www.w3.org/XML/