Phoeßus

A fully integrated workflow execution framework

Jean-Noel HOURCASTAGNOU[1],
CNES, Toulouse, France

Pascal BRANET[2]
THALES, Toulouse, France

Phoebus is a fully integrated workflow execution framework that is designed as an off the shelf product. Phoebus consists in workflow management software that provides users with an easy, flexible and powerful way to describe and orchestrate complex processing of data. It provides monitoring and control functions, which are executed inside a workflow. Process execution is done by using and monitoring computer networks. Phoebus offers the following main functions: Workflow management; Resource management; process monitoring and controling; easy integration of new processing chain. It is highly configurable and scalable to fulfill operational characteristics of data processing: processing tasks schedule, level of processing parallelization, hardware resource management ….. For example, in Pleiades context, it supports by configuration the heterogeneity of the production centers both the hardware size (from 7 up to 50 servers) and the target production levels.

This infrastructure is well adapted to use the innovative and incremental development methods like RAD, AGILE. The flexibility and the potential for upgrade of this architecture are demonstrated by the deployments in Pleiades, GAIA and Sentinel 2 ground segments.

The article presents the main features in terms of architecture, technology and standards. It also presents the benefits of reuse and the perspectives planned for Phoebus.

I. Introduction

A. Phoeßus History

Before describing the Phoeßus architecture and concepts in the part II of this article, it is useful to describe briefly its genesis. A Processing Orchestration Engine was initially developed for the Pleiades Earth Observation Image Processing Unit in order to schedule the execution of the image algorithm tasks in an efficient and optimized way. The aim was to highly parallelize the processing in order to provide products in requested delay with respect to hardware processing performance.
In Pleiades context, it has to generate systematically L0 products and on demand the high level corrected image products. It has been designed to be configurable and flexible in order to easily be deployed in the target centres whatever the production characteristics of the target centre: hardware architecture and supported production level. The biggest Pléiades Centre is the Spot-Image one in Toulouse (France), which is able to generate up to 500 products per day with 52 servers (416 tasks executed in parallel).

[1] Head of Office "Scientific Data and Image Processing Systems" DCT/PS/TIS CNES, 18 av. Edouard Belin, 31401 Toulouse Cedex9,jean-noel.hourcastagnou @cnes.fr
[2] IT Architect, Thales DSC/CIS/SSE, 3 av. de l'Europe 31400 Toulouse, pascal.branet@thalesgroup.com

The Processing Orchestration Engine has been designed in a generic and low cost reused approach in order to be reused to implement another Image Processing Facility.

The first reuse is the implementation of the CNES-GAIA Data Processing Center (ESA project) which has to perform systematic processing of 100 billion of GAIA objects (spectrum,…) every 6 months. The hardware target platform is about 150 nodes (~1200 tasks executed in parallel).

Furthermore, the CNES has noted that the development from scratch of new data processing systems (eg SMOS level 3 and 4 data processing center) was very costly in terms of validation. As a consequence and to prepare the next generation of Payload data Ground segment (MERLIN, CFOSAT or CSO), CNES had performed studies in order to define the minimal functional perimeter of a “generic” Data Processing Centre.

After the study of various existing solutions in CNES or our partners, CNES has decided to transform the component issued from the Pleiades project which already answered some of this functional scope in a reusable software product called Phoeßus (Processing High Level Orchestration Engine & Business User Services).

Phoeßus is a software framework to build by configuration and by integration (software processing add-on) a Data Processing Centre either fully autonomous (ie: GAI-DPC) or integrated in a Ground Segment Monitoring&Control infrastructure (ie: Pléiades, Sentinel 2).

The first version of Phoeßus was delivered in 2011 and integrated in GAIA-DPC.

The second reuse is the implementation of the Sentinel-2 Core payload Data Ground segment which is currently under development by THALES for ESA.

II. Functional Description

A. Phoeßus Principles

The main principles that drive the achievement of Phoeßus are the following ones:

1) maximum **automation of processing**: the system is working on 24 H/7days, minimum human intervention are required: recovering failure, Quality Control operation, configuration update procedures.
2) **modularity**: Phoeßus is designed to be deployed in many different centres as it is done for Pleiades.
3) **scalability**: the solution is independent of the hardware processing architecture (heterogeneous hardware resources with heterogeneous Linux OS versions are supported).
4) **reusability**: the framework software architecture has been chosen to ease the system reusability. The framework approach is based on the 2 following concepts: the generic behaviors are integrated in the framework and the behaviors which are intended to be adapted according to project features are specialized/overridden via software replacement mechanisms (i.e. plug-in, add-on).
5) **interoperability**: the service-oriented architecture was implemented by web services. The web services are a standard interface independent of OS, programming language and localisation of the service –internet-. This approach facilitates the integration of local or remote new services in processing chains.
6) maximum off-the shelf **open sources SW integration**: every time it was possible the solutions based on open sources are preferred.

B. Phoeßus Orchestration Processing Concept

Phoeßus Orchestration Processing approach is in total accordance with the concepts promoted by the WfMC[3] (See also Ref 1 and Ref 2).

« A workflow model is a description of a business process in sufficient detail that it is able to be directly executed by a workflow management system. A workflow model is composed of a number of activities which are connected in the form of a directed graph. An executing instance of a workflow model is called a case or process instance. There may be multiple cases of a particular workflow model running simultaneously, however each of these is assumed to have

[3] *WfMC: Founded in 1993, the Workflow Management Coalition (WfMC) is a global organization of adopters, developers, consultants, analysts, as well as university and research groups engaged in workflow and BPM. The WfMC creates and contributes to process related standards, educates the market on related issues, and is the only standards organization that concentrates purely on process.*

an independent existence and they typically execute without reference to each other. …. » 2010 © Workflow Patterns Initiative.

As illustrated in Fig 1, the processing concept of Phoeßus is based on the strict separation between:

- the **business process** that is the specification of a processing chain to generate a specific product (i.e.: "sensor" level product, orthorectified products, mosaics, …),
- the **model** that is the description in a software language of the business process. In Phoeßus, the model is described in XML language; the XSD derivates from the XPDL (XML Process Definition Language is the standard format of WfMC).
- **instances** of the model that are created for each product to generate. In Phoeßus, the instances are created and managed in a PostgreSQL database.

The workflow instance execution is based on the following 3 perspectives:

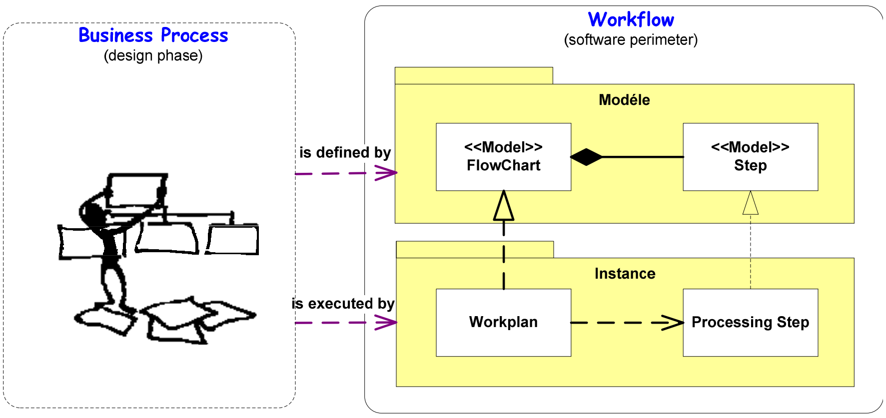

Figure 1. Phoeßus processing Concepts.

- the **control flow** is related to the schedule between the processing steps involved to realize the workflow,
- the **data flow** deals with the exchange of data between the processing steps,
- the **resource allocation** is related to the description of the resources (software-hardware) required to execute the processing steps.

Phoeßus implements control flow in accordance with the activities execution behaviors standardized by the WfMC as a Controlflow patterns (http://www.workflowpatterns.com/patterns/control/): sequence, synchronization, parallel split, exclusive choice (conditional execution), multiple instance (parallelisation management), etc, Phoeßus manages the main ones, usually involved to design Payload Data processing.

Phoeßus implements data flow perspective thanks to the ***Global data store*** paradigm: this data sharing approach assumes an a priori knowledge of the location of data and deals with potential concurrency between several jobs to access to the same data file.

Phoeßus implements resource allocation perspective thanks to a DRM (Distributed Resource Management) system in compliance with the DRMAA v1 standard API (Open Grid Forum). Phoeßus submits the processing tasks to the DRM providing the resources required for execution: number of core, size of RAM, level of priority…

The figure 2 illustrates the complex schedule of activities managed thanks to Phoeßus (Sentinel 2 L2B processing chain).

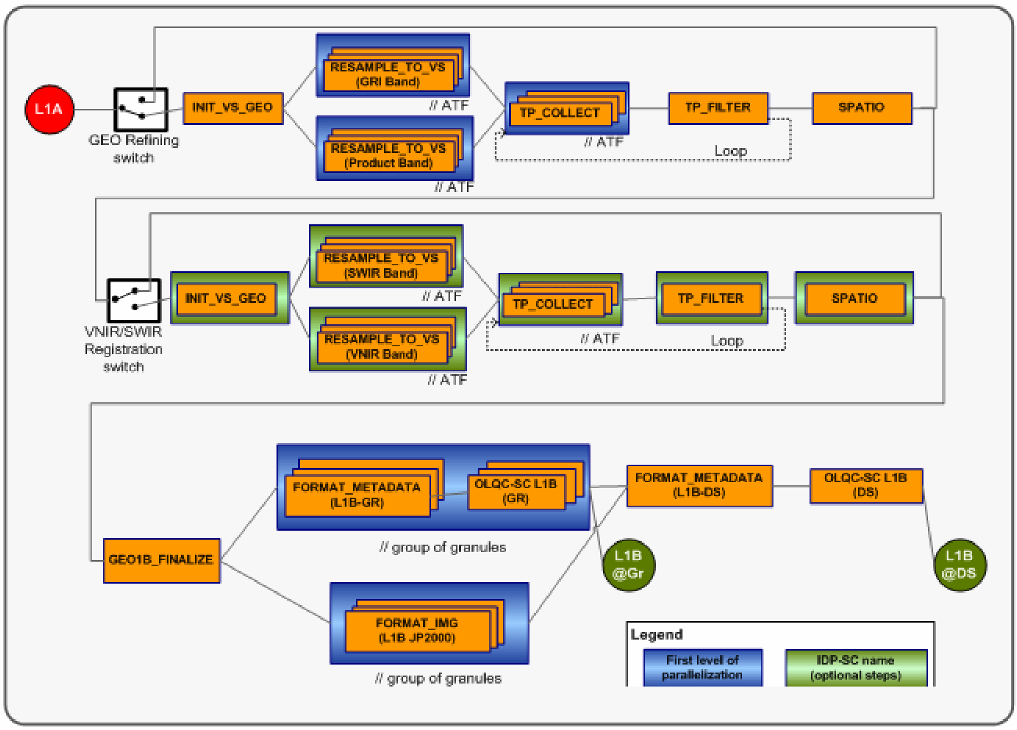

Figure 2. Sentinel 2 processing chain model managed by Phoebus.

C. Phoeßus Activity Execution Concept

Phoeßus is compliant with the SOA approach promoted by the WfMC, that recommends to monitor and control the execution of activities thanks to web services.

But, most of the times, processing tasks are implemented by software executables. So, Phoeßus provides an Execution Layer that translates web service to software executable commands. This "generic" layer is able to monitor and control both the CNES-LAI and ESA-Job interfaces.

Furthermore, the Execution Layer is designed to manage several models of sequence of software executables.

At first level, as shown in Fig 3 the executable can be merged in group; the execution of the groups can be performed either in sequence or in parallel (nota: the same group can be instantiated several times).

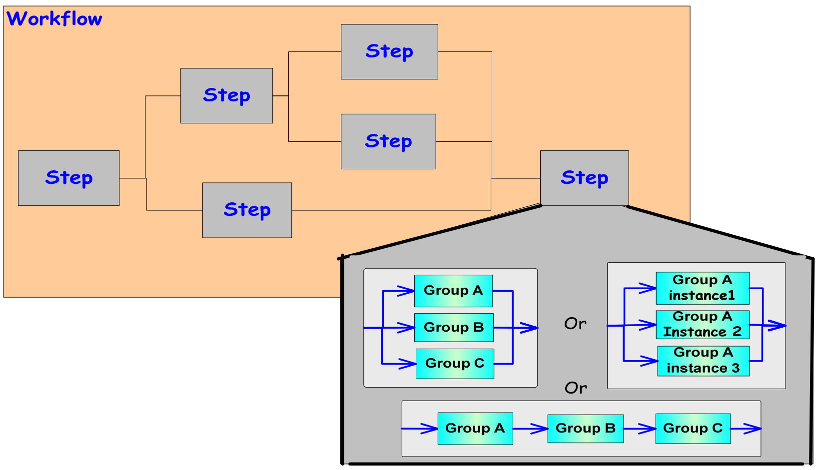

Figure 3. Models of group execution managed by Phoeßus.

AS shown if Fig 4, each group is implemented by a set of executables either launched in sequence or in parallel (nota: the same executable can be launched several times).

The group and process model is a well-adapted breakdown level to define complex sequences of tasks.The multiple instance execution models allow to launch several instances of the same executable in parallel; each one processing a sub part of the input dataset. This mechanism is widely used in Pleiades Ground Segment to reduce the duration of the product generation. The level of parallelisation can be very large. For example, to process mosaic image, the number of instance which are managed in parallel is about 800. If required, it is possible to mix both multithreading and multiprocessing parallelisations.

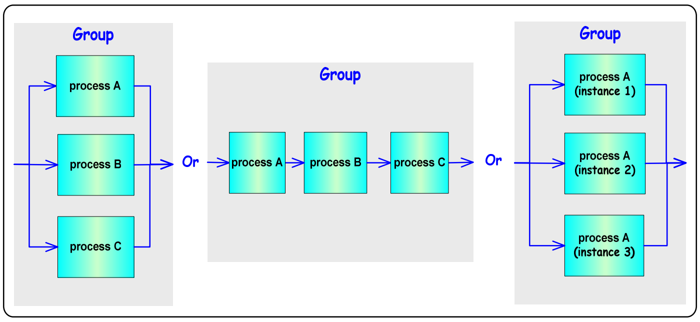

Figure 4. : Models of schedule of processus execution

D. Phoeßus Main functions

WfMC has specified an architectural model for workflow systems, called "WfMC Workflow Reference Model". identifying the characteristics, functions and interfaces of such architecture (see Fig 5). Phoeßus is designed in accordance with this functional model. Phoeßus is made up of 4 software components:

Phoeßus-Core is the central component that is in charge of monitoring and controlling the Data Processing Center (DPC) and manages the workflow execution according to execution events and operator orders,

Phoeßus-HMI is the graphical user interface that provides to the operators/administrators graphical means to monitor and control the execution of the workflows,

Phoeßus-Execution Layer is in charge of monitoring and controling the execution of the software executables in respect with the schedule and the parallelization level computed in run-time. Notice: the Phoeßus -Execution Layer is not mandatory in case of activities directly implemented by web service that can be directly invoked by Phoeßus Core.

Phoeßus-JobManagement is in charge of dispatching the software executables in respect with the resource allocation in order to optimize the overall resource usage and reduce the global product generation delay.

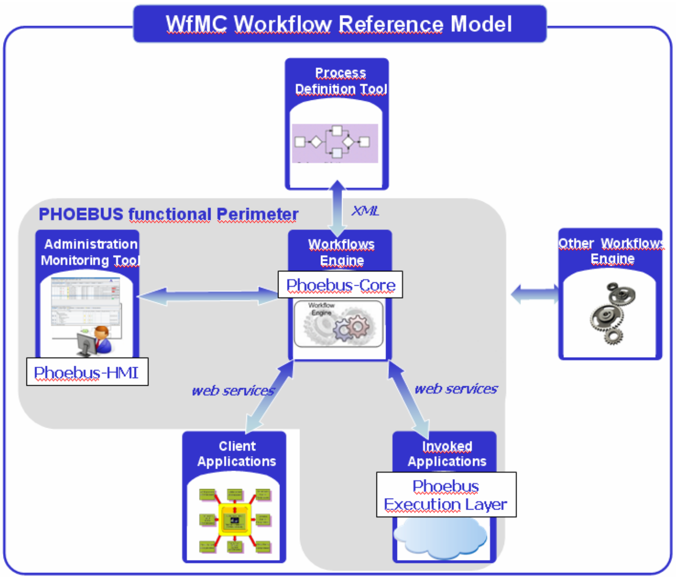

Figure 5. Phoeßus perimeter vs WfMC Workflow Reference Model

Phoeßus-Core implements the following functions:

The first one is the Administration&Supervision function wich is in charge of monitoring and controling of the Data Production Centre (startup/shutdown, ...), but also the monitoring of the hardware components status (nodes of the cluster …).In addition this function manages several sub-functions directly dedicated to the operation survey of the center like : the centralized logbook ;the security functions (access authorization – LDAP interface-), the alarms and operators actions, control of the automatic backup procedure (backup tool is outside the Phoeßus perimeter), On Line Help of the Data Production Centre.

The second one is the Event Manager which detects processing events described in an XML configuration file (data events, time events, stream events, web services events, user events).For complex events (such as combination of input data), Phoeßus provides a plug-in interface.

The last one is Orchestration Engine which

- generates workplan from the model and manages the workplan startup either automatically (immediate, delayed) or manually (via Phoeßus HMI) according to description provided in the XML execution configuration file.
- orchestrates & parallelizes the execution of the activities of the workplans according to execution events (errors, crash, …) and operator orders (stop, resume, cancel, change priority, replan) either provided locally (Phoeßus HMI) or remotely (web service).
- manages debug mode execution: the execution of a workplan can be put anytime in debug mode. In debug mode, the execution is totally under operator control: add breakpoint/resume point on any processing step, execute workplan step by step, from a selected resume point, to next breakpoint.
- at any moment, on operator request, saves a complete workplan execution context for investigation (maintenance, algorithm expertise, …).

Phoeßus-Execution Layer manages the execution of steps implemented by software executable:

- processes the parallelization level on run-time,
- manages software executable monitoring and control interface. Phoebus supports both CNES-LAI and ESA-JobOrder interfaces,
- monitors and controls the execution of the software executable delegated to Phoebus-JobManagement component.

Phoeßus-JobManagement distributes the software executable onto the cluster of processing servers on an optimized way. By configuration, Phoebus is able to manage 3 different types of intensive distributed applications (Job Management System): the proprietary one developed for Pleiades needs, any DRM tool compliant with DRMAA v1 API (like Torque/Maui, slurm, OpenPBS, Condor,…) and Cascading Hadoop.

Phoeßus-MMI provides to the Data Production Centre exploitation team (administrator and operator) a friendly and complete graphical interface (Fig 6) to administrate & supervise the Data Production Centre (startup, shutdown, logbook editor, ..), but also to monitor the workplans execution in nominal and debug mode

To fufill, program specific HMI needs, Phoeßus HMI provides a plug-in interface to add new HMI panels.

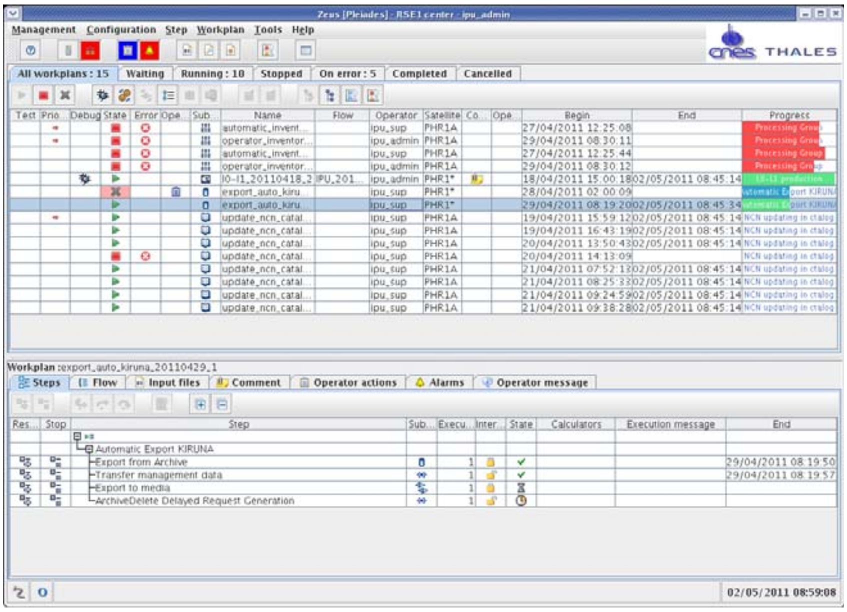

Figure 6. Phoeßus MMI main Panel

III. Technical Solution

A. General Principles

Phoeßus is designed in consistency with the state of the art of software architecture engineering:

- **SOA** (Services Oriented Architectures): the external interfaces of Phoeßus-Core are mainly implemented as web services in order to ensure flexibility, upgradeability and LAN accessibility. The WAO (an extension of SOA that implements the service communication thanks to the HTTP standard protocol) has been chosen. The message exchanges are based onto the SOAP standard protocol.

- **OSS integration approach**: in order to control development cost and delay, as far as possible Open Source Software providing high level of confidence are used (Apache/Tomcat/Axis, Joram, Quartz…).

- the architecture of Phoebus is based on **n-tier architecture** that improves maintainability since each tier part (presentation tier, a business tier , data access tier and middleware tier) is managed separately.

- **usage of standards**: as far as possible the interfaces are standardized in order increase interoperability with external tools (WPS –*OGC*-, XPDL –*WfMC*-, DRMAA –*OGF*-, XML/WSDL –*W3C*-, posix –*IEEE*-, ….),

- **hardware virtualization**: Core, Execution Layer and IHM can be deployed and executed into a virtual architecture.

- Phoeßus has been designed in order to fulfill the **robustness** and **high availability** (24/7 availability, 99,2% uptime) required for Pleiades Ground Segment,

- **MOM** (Message Oriented Architecture) solution: internal exchanges of Phoeßus are managed thanks to a MOM that is a powerful and secured infrastructure supporting sending and receiving messages between distributed systems over heterogeneous platforms. This infrastructure is known to reduce the complexity of developing applications and increase interoperability, portability and flexibility of the applications. The Joram implementation has been chosen for Phoeßus.

- **full Java** development approach: this programming language is selected regardless of the abundance of Java development environment: standard, API, libraries, OSS IDE, framework, continuous integration, ….

B. Software Architecture

The complete software architecture of Phoeßus is presented in Figure 7. It can be noticed that each software component in dash line is optional (configuration); they are executed if the functionality is required for the target DPC.

This architecture eases the integration of functional evolution in reduced delay and cost. For example, the integration of Hadoop/cascading is performed to fulfill GAIA-DPC needs and the WAN/LAN M&C component is integrated for CSO needs. But each new component is developed in a "generic" approach in order to be reusable in future DPC.

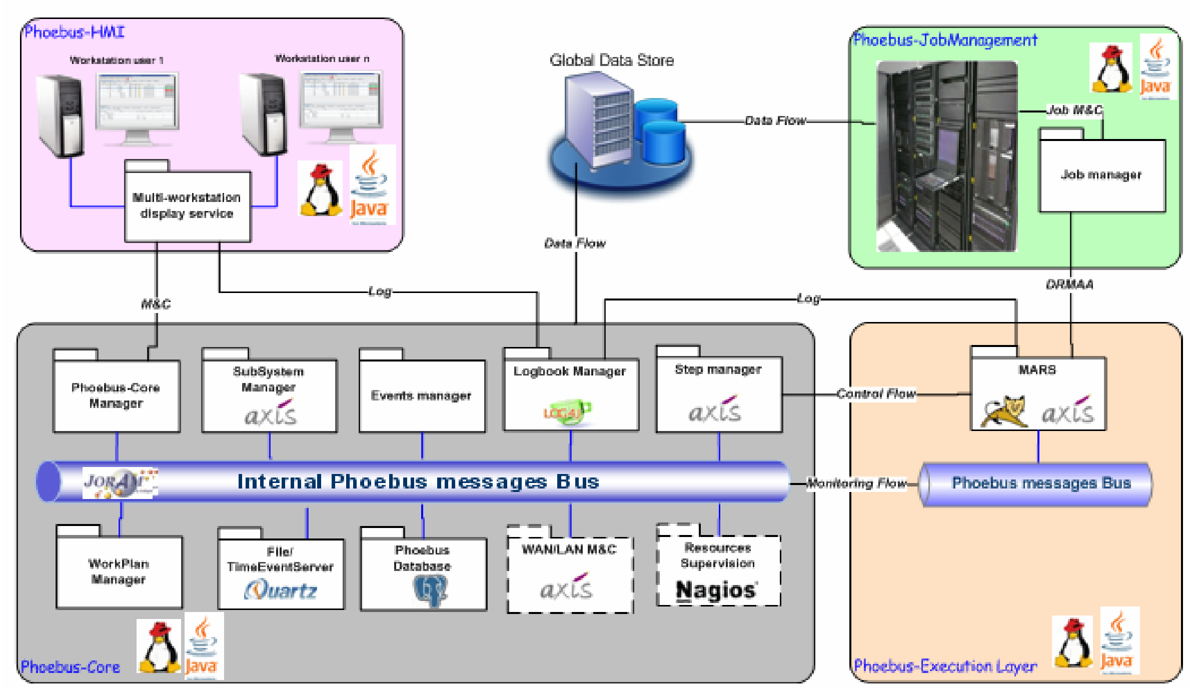

Figure 7. Phoeßus software architecture..

C. Phoeßus Hardware

Phoeßus architecture is designed to support the deployment of the Phoeßus components on one or several servers in order to support the different level of workload intended for the target Data Production Centre.

It is independent of the processing hardware architecture (Linux OS family) and supports both SAN and NAS implementation of the Global Data Store.

The hardware configuration hosting the IPU-Pleiades built from Phoeßus which has been set up in the Spot Images premises is shown in the Fig 8.

D. Phoeßus Performances Overview

The following performances are reached into the main IPU-Pleiades Center (hosted in Spot image premises):

- 500 images/day generated thanks to cluster of 52 servers (octo-core)

- Raw images in case of urgent Defense needs) are delivered in less than 7mn,

- High level corrected images are delivered in less than 25mn

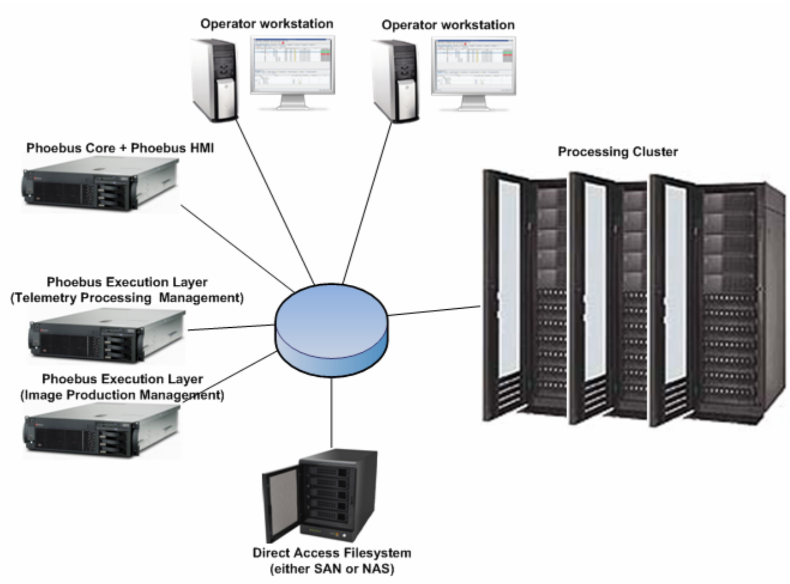

Figure 8. IPU Pleiades hardware architecture

E. Integration & Debug Approach

The integration of Phoeßus to build a DPC is based on the following schedule:

1. Define the target hardware architecture,

2. Install and configure Phoebus to fulfill the DPC operational mission,

3. Design the workflow,

4. Simulate the workflow execution in order to size the target platform thanks to simulators delivered with Phoebus,

5. Integrate and test the processing software executable as soon as available, principle based on continuous integration.

6. Debug the workflow.

To perform this schedule, the following tasks are required:

- define the processing chains in XML files,

- define workplan start up events in XML file,

- define HMI functions accessibility according to profile in XML file,

- develop plugin to process events,

- develop processing software executable,

- install and configure target Job Management System,

- develop an installation kit to deploy all the previous elements in the target repository of Phoeßus.

F. Planned Evolutions

The main evolutions planned to be integrated in Phoeßus are:

- a web HMI to monitor and control the workplans execution from the LAN,

- a generic catalog (DBMS based) to record the location and characteristics of all the data produced during the execution of the workplans; in order to ease the search and sort of the data stored in the Global Data Store,

- a light local archive to store the auxiliary data (ie : GIPP, DEM, …) in order to avoid to retrieve systematically this data files from the external archive.

IV. Perspectives

In the Future space agencies will have to meet a double challenge:

Develop and implement processing centers increasingly complex, having to deal with data volumes growing wider with more complex algorithm pipelines and therefore, mostly based on distributed calculating means and distributed storage.

Reduce widely the cost of these processing centers and as for this reason use existing tools and technologies which simplifies the development and the integration of the processing software itself.

For these reasons, opportunities opening up for the reuse of Phoebus in the data processing centers could be possibly very large. On the condition that the product can evolve to adapt to the constraints of future missions.

Priority should be placed on the functional aspects, in order to meet operational needs. Indeed, data processing centers have to manage more and more frequently heavy-overtime operations of reprocessing . This requires to preparing and following several thousand of jobs whose execution will take several months. Phoebus must be adapted to propose a fully automatic mode of these reprocessing operations.

V. Conclusion

The reuse of Phoebus with new missions as CFOSAT and CSO is scheduled in the next two years. If, as we believe, Phoebus achieves a significant reduction in costs of development and validation of these data processing centers, PHOEBUS could become an essential component in the architecture of most data processing centers. It could be reused also in other business area (e.g. video, medical, …) , where the management of intensive processing of scientific or sensor data is a key issue.

Appendix A

Acronym List

API	Application Programming Interface
CNES	Centre National d'Etudes Spatiales (French Space Agency)
CFOsat	Chinese-French Oceanic SATellite
CSO	Composante Spatiale Optique
DBMS	DataBase Management System

DEM	Digital Elevation Model
DPC	Data Processing Centre
DRM	Distributed Resource Manager
DRMAA	Distributed Resource Management Application API
ESA	European Space Agency
GIPP	Ground Image Processing Parameters
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineer
IPU	Image Processing Unit
LAI	Logiciel Algorithmique Image
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MOM	Message Oriented Middleware
NAS	Network Attached Storage
OGC	Open Geospatial Consortium
OGF	Open Grid Forum
OS	Operating System
OSS	Open Source Software
PDGS	Payload Data Ground Segment
PHOEBUS	Processing High level Orchestration Engine and BUsiness Service
RAD	Rapid Application Development
SAN	Storage Area Network
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
W3C	World Wide Web Consortium
WfMC	Workflow Management Coalition
WOA	Web Oriented Architecture
WPS	Web Processing Service
WSDL	Web Services Description Language
XML	EXtensible Markup Language
XPDL	XML Process Definition Language
XSD	XML Schema Definition

Appendix B
Glossary

Acknowledgments

Authors would like to tank all the team involved in the development of Phoeßus for their confidence, support and motivated contribution: CNES (IPU, GAIA, CSO and CFOsat teams) and THALES company in charge of the development of Phoeßus and the realization of the Sentinel 2 –PDGS.

References

[1]: WfMC : Terminology & Glossary [WFMC-TC-1011 Issue 3.0 Aug 2008] URL: http://www.wmfc.org,
[2]: WfMC : Deborah Jouin : Terminology and Glossary French, [WFMC-TC00-103 - Aug 2008]