DataGrid

EO USE CASES IN HEPCAL FORMAT

ANNEX TO
EO REQUIREMENTS SPECIFICATIONS
AND
HEPCAL DOCUMENT

Document identifier:	
Date:	
Work package:	**WP09.4**
Partner(s):	**ESA, IPSL, KNMI**
Lead Partner:	**KNMI**
Document status	**DRAFT**
Deliverable identifier:	

Abstract: In annex to DataGrid WP9 Deliverable Document D9.1 Earth Observation application requirements specification, this document gives a detailed description of one EO Use Case using the HEPCAL usecase modelling.

Delivery Slip

	Name	Partner	Date	Signature
From	Cathy Boonne, W.J. Som de Cerff	KNMI		
Verified by				
Approved by	F. Gagliardi	CERN		

Document Log

Issue	Date	Comment	Author
0.0	07/05/2003	First draft	Cathy Boonne, W.J. Som de Cerff

Document Change Record

Issue	Item	Reason for Change
0.0	Document	First draft

Files

Software Products	User files
Word	

CONTENT

1. INTRODUCTION 4
1.1. OBJECTIVES OF THIS DOCUMENT 4
1.2. APPLICATION AREA 4
1.3. APPLICABLE DOCUMENTS AND REFERENCE DOCUMENTS 4
1.4. DOCUMENT AMENDMENT PROCEDURE 5
1.5. TERMINOLOGY 5
2. EXECUTIVE SUMMARY 7
3. EO USECASE IN HEPCAL FORMAT 8
3.1. EXTENSION TO HEPCAL BASIC CONCEPTS 8
3.2. BASIC ASSUMPTIONS 8
3.3. USE CASE DESCRIPTIONS 8

INTRODUCTION

This document describes Earth Observations use cases, modelled using the HEPCAL modelling [A8].

The document consists of the following sections

- **Section 1.** The current section, provides a basic introduction about the document content
- **Section 2.** Provides an executive summary of the document
- **Section 3.** Provides the EO use cases

It is assumed that the reader is already familiar the main WP9 Use Cases document [A4], the WP9.4 usecase document [A9], the EO Requirements document [A3] and the HEPCAL common usecase document [A8].

1.1. OBJECTIVES OF THIS DOCUMENT

Input for the AWG to get a set of common use cases which should lead to a set of common services.

1.2. APPLICATION AREA

The prime focus of this document will be on the refinement of the GOME profile retrieval use case, as specified in the WP9 Use Cases document [A4] in HEPCAL format [A8].

1.3. APPLICABLE DOCUMENTS AND REFERENCE DOCUMENTS

Applicable documents

[A1] DataGrid Project Quality Plan [DataGrid-12-PQP-0101-1_0]

[A2] DataGrid Project Programme Annex 1 “Description of Work”

[A3] Requirements specification: EO application requirements for GRID [DataGrid-09-TED-0101-1_0-Requirements]

[A4] Earth Observation Use Cases for GRID [DataGrid-09-TED-0102-1_0-UseCases] [DataGrid-09-TED-0120-1_0-KNMI-UseCases]

[A5] Structuring Use Cases With Goals, Alistair Cockburn et al., journal of Object Oriented Programming, issue Sept-Oct 1997 and Nov-Dec. 1997

[A6] The DataGrid Architecture Version 2, June 6, 2001 (Pre-release)

[A7] Proposal for Standard File Output of GOME O3 WG, Y. Meijer, J.c. Lambert, S. Casadio and C. Zehner.

[A8] LHC Grid Computing Project, Common Use cases for a HEP common application layer (HEPCAL), draft v1.6

[A9] WP9.4 use case, annex to Requirement Specification: EO application requirements for GRID

Reference documents

1.4. DOCUMENT AMENDMENT PROCEDURE

This document is under the responsibility of KNMI, ESA/ESRIN and IPSL.

Amendments, comments and suggestions should be sent to the person in charge of WP9

The document control and change procedure is detailed in the DataGrid Project Quality Plan [A1]

1.5. TERMINOLOGY

Definitions

ATF	DataGrid Architectural Task Force
AMS	Archive Management System
CE	Computing Element
COTS	Commercial Off-The-Shelf
ENVISAT	ENVIronment SATellite http://envisat.esa.int/
EO	Earth Observation
ES	Earth Science
ESRD	Earth Science Requirement document
GIS	Geographical Information System
HDF	Hierarchical Data Format http://hdf.ncsa.uiuc.edu/
IDL	Interactive Data Language http://www.rsinc.com/idl
MSS	Mass Storage System
PDS	Payload Data Segment
PTB	DataGrid Project Technical Board
QoS	Quality of Service
RB	Resource Broker
JDL	Job Description Language
IDL	Interactive Data language

SCIAMACHY	SCanning Imaging Absorption SpectroMeter for Atmospheric CHartographY
TBC	To Be Confirmed
TBD	To Be Determined
UI	User Interface
WP	Work Package

Glossary

Note: the terms in this Glossary may conflict with the terms used in the HEPCAL document. To exclude confusion, 'EO' will be put in front of conflicting terms, e.g. EO-catalogue is a catalogue as described in this glossary.

Ancillary data	Data from the space craft (e.g. attitude, orbit)
Auxiliary data	Data from other source than the instrument or space craft (e.g. wind data from ECMWF)
Catalogue	Container of product descriptors and product collections
Metadata	Additional information about EO product data and geodata ▪ quality information ▪ quick browse image ▪ user help, keywords ▪ processing parameters, e.g. related to image calibration ▪ algorithm information, e.g. programs, interactive services, documentation
On-line	Data stored on a network accessible disk
Near-line	Data stored in MSS accessible over a network
Off-line	Data that are available only after human operator intervention
Near-real-time	Data required within a defined period after the data was collected
Close-real-time	Data required before future granules arrive and cause backlog problems

2. EXECUTIVE SUMMARY

This document describes Earth Observations use cases, modelled using the HEPCAL modelling [A8]. It is an addition to the HEPCAL document, specifying only those use cases, which are not in the HEPCAL document.

The document is input for the Applications Working group.

3. EO USECASE IN HEPCAL FORMAT

3.1. EXTENSION TO HEPCAL BASIC CONCEPTS

The basic concepts are described very precise in the HEPCAL document [A8, chapter 3]. Therefore this is not repeated in this document, but extended on.

ACTORS:
EO can map the actors described in the HEPCAL document to the EO domain. In the EO use cases the 'consumer' actor is added:
A consumer is a user or program, which wants to be notified when a DS becomes available.
It will be investigated if there are more actors in the EO domain.

FILES, DATASETS and CATALOGUES:
The concepts described in the HEPCAL document are well defined and have a high abstraction level. Using 'EO' prefix if an EO term is needed will prevent possible clash of terms in the usecase definitions (e.g. EO-Catalogue).

JOBS:
For EO there are three types of jobs, which might not match on the defined job types:
·Near Real Time jobs: Jobs, which need to be processed within a certain time after the input data becomes available (e.g. processing of cloud images measured by satellites). The output is often used by Time constrain jobs (models) or used in warning systems.
·Time constrained jobs: Jobs whose output needs to be processed before a certain time (e.g. Ozone concentration predictions)
·Parallel Jobs: In contrary to the job splitting, where the job is split into independent parallel jobs, EO has applications where the split jobs have dependency on each other (e.g. climate models).

3.2. BASIC ASSUMPTIONS

The included use cases are in the HEPCAL document [A8].

This chapter will need more input.

3.3. USE CASE DESCRIPTIONS

The use cases mentioned here are not a complete overview of all possible use cases. The EO domain is very broad and more discussion within the EO domain is needed to complete this effort.
The use cases specified in HEPCAL [A8] can be mapped and used for many EO use cases. Only if the EO use case cannot be mapped upon a HEPCAL use case or if the mapping is not clear, the use case is specified here.

EO USE CASE: EO NRT JOB

Identifier	*UC#EONRTJob*
Goals in Context	*Send a job to Grid computing resources, specifying the maximum time the job can take to produce the output.*
Actors	*Program*
Triggers	*Availability of the input data needed*
Included Use Cases	- *Job submission (UC#jobsubmit)* - *Job control (UC#jobcont)* - *Job resource estimation (UC#jobrestest)* - *Specify program* - *Dataset Access*
Specialised Use Cases	*When the job output becomes available, the consumer of the output must be notified the data is available.(UC#EONotifyDSAvailable)*
Pre-conditions	- *Program logged into the Grid* - *Dataset becomes available*
Post-conditions	- *Program is run. A consumer of the dataset is notified the dataset is created and available*
Basic Flow	1. *Program specifies job information;* a. *Environment needed* b. *Any grid input data needed* c. *Any local data needed* d. *Program to be executed* e. *Output files which could be added to a DS* f. *Job attributes* g. *Time contraint* h. *Consumers of the data who should be notified when the job output data become available* 2. *EXTENSION POINTS (steer submission), (resource estimation), (environment modifications)* 3. *Program submits description to job submission command* 4. *Job catalogue is updated* 5. *Job executes* 6. *Output is stored on the Grid with metadata parameters* 7. *On completion, consumers are notified the data is available*
Devious Flow(s)	*See included use cases*
Importance and	*High frequency and importance*

Frequency	
Additional Requirements	*Notification to DS consumers when output data is available*

EO USE CASE: EO NOTIFY DS IS AVAILABLE

Identifier	*UC#EONotifyDSAvailable*
Goals in Context	*Notify a User that a DS has been produced or uploaded on the Grid*
Actors	*User*
Triggers	*Notify to consumers of a DS when it becomes available*
Included Use Cases	
Specialised Use Cases	
Pre-conditions	*Actor has access to the Grid with corresponding rights to perform this action*
Post-conditions	
Basic Flow	1. *Actor specifies:* a. *Who should be notified* b. *Trigger Event (DS is available)* c. *Event thrower (job which produces the data* 2. *Actor submits its request* 3. *JMS returns notification that the request is in place* 4. *Upon completion of the specified job the specified consumers are notified*
Devious Flow(s)	*Invalid consumers*
Importance and Frequency	*High importance and frequency*
Additional Requirements	

EO USE CASE: EO TIME CONSTRAINED JOB

Identifier	*UC#EOTimeConstr*
Goals in Context	*Produce a DS within a predefined time.*
Actors	*User or Program*
Triggers	*Produce a DS within a specified timeframe*
Included Use Cases	- *Job submission (UC#jobsubmit)* - *Job control (UC#jobcont)* - *Job resource estimation (UC#jobrestest)* - *Specify program* - *Dataset Access* - EO notify DS is available *(UC#EONotifyDSAvailable)*
Specialised Use Cases	
Pre-conditions	- *User or program logged into the Grid* - *Preconditions for UC#jobsubmit*
Post-conditions	- *DS is produced within the specified time constrain* - *DS is made available on the GridConsumers are notified*
Basic Flow	1. *User specifies job information:* a) *Environment needed (hardware and software, can be any);* b) *Any Grid input dataset needed;* c) *Any local input files needed;* d) *The program to be executed;* e) *Any output files which should not be deleted;* f) *Optionally job attributes in the form of* key=value *pairs to be set in the job catalogue;* g) *Time constraints* 2. *EXTENSION POINTS: (steer submission), (resource estimation), (environment modification);* 3. *User submits description to job submission command;* 4. *The job catalogue is updated* 5. *Job executes;*
Devious Flow(s)	*No resources available to process the Job within the specified time*
Importance and Frequency	*High importance, high frequency*
Additional Requirements	

EO USE CASE: PARALLEL JOB

Identifier	*UC#PJ*
Goals in Context	*Send parallel job to Grid computing resources*
Actors	*User*
Triggers	*Decision to submit a parallel job*
Included Use Cases	*Job submission*
Specialised Use Cases	
Pre-conditions	*User Logged into the Grid* *Needed data available*
Post-conditions	*Program is run. Any files specified as "valuable output" are available for further use or removal.*
Basic Flow	1. *User specifies job information:* a. *Environment needed (hardware and software, can be any);* b. *Any Grid input dataset needed*[1]*;* c. *Any local input files needed;* d. *The program to be executed;* e. *Any output files which should not be deleted;* f. *Optionally job attributes in the form of* key=value *pairs to be set in the job catalogue;* 2. *EXTENSION POINTS: (steer submission), (resource estimation), (environment modification), (check pointing);* 3. *User submits description to job submission command;* 4. *The job catalogue is updated* 5. *Job executes;*
Devious Flow(s)	*See included use cases*
Importance and Frequency	*High frequency, high importance*
Additional Requirements	

[1] We should have a mapping: {LFN}=>{local file name} so that a program could open files using a standard naming.

EO USE CASE: PRODUCTION OF A NEW DATASET USING AN EXISTING DATASET

Identifier	*UC#EODataSetProduction*
Goals in Context	*Process one dataset into a new dataset using a specific algorithm* *This usecase is similar to the UC#dstran*
Actors	*Production manager*
Triggers	*A new dataset, which can be used for scientific analysis. This dataset is accessible using a EO-metadata catalogue (user defined catalogue)*
Included Use Cases	*Grid login* *Job submission* *Job monitoring* *Job output access or retrieval* *Job control* *DS upload* *DS metadata access* *User defined catalogue creation* *Error recovery for failed production jobs* *Data Transformation (UC#dstran)*
Specialised Use Cases	*EO parallel job, EO NRT job, EO notify*
Pre-conditions	- *Availability of software and environment files on the Grid* - *Availability of public/ private database on the Grid* - *Availability of the input dataset on at least on Grid storage element*
Post-conditions	- *Output data registered on the Grid* - *Metadata registered on the Grid*
Basic Flow	1. *Production manager specifies job information* a. *Environment needed* b. *Local input files needed* c. *Workflow chain* d. *Input DS (can be more than one DS if ancillary data and auxiliary data are needed)* e. *Metadata DS* f. *Output DS* g. *Program (algorithm) to be run*
Devious Flow(s)	*See included use cases*
Importance and	*High importance. Frequency is dependent on algorithm development, but will*

Frequency	*be at least once per year per data set, per VO.*
Additional Requirements	*Version control is important. The DS metadata will contain all necessary metadata needed to be able to reprocess the new dataset (version of input DS, version of algorithm, ancillary and auxiliary data used)*

EO USE CASE: EO DATA VALIDATION

Identifier	*UC#EOdatavalidation*
Goals in Context	*Data validation processing performed on GRID CE nodes*
Actors	*User through a web interface or a JDL script*
Triggers	*Validation result files, which can be visualised or analysed by scientists.*
Included Use Cases	- *Grid login* - *Job submission* - *Job monitoring* - *Specify program* - *Metadata access* - *Dataset transfer to Grid storage* - *Dataset replication* - *Job output access*
Specialised Use Cases	
Pre-conditions	- *Availability of the input dataset on the Grid* - *Availability of IDL software on the Grid*
Post-conditions	- *Validation output data files are created and returned to the UI/user* - *Validation output data files registered on the Grid (optional)*
Basic Flow	*1. User specifies job information (script shell which produces the appropriate JDL validation job) :* *a) Environment needed* *b) Local dataset needed* *c) Grid dataset needed* *d) Programs to be executed on the Grid* *e) Selection criteria for validation* *f) License file to be able to use IDL* *2. User submits job* *3. Job execution* *4. Validation output data files retrieval* *5. Results files are visualised or analysed*
Devious Flow(s)	*Grid input files not found or not produced;* *IDL software not available;* *No matching selection criteria for validation;* *No output files after program execution*
Importance and	*High frequency and importance*

Frequency	
Additional Requirements	*Validation data input files copy and register on a SE*

EO USE CASE: EO UPDATE AND REPLICATION OF VALIDATION DATA ON THE GRID

Identifier	*UC#Eouprepvaldata*
Goals in Context	*Update and replication of validation data on the Grid*
Actors	*Script*
Triggers	*Access to validation data*
Included Use Cases	- *Grid login* - *Dataset transfer to Grid storage* - *Dataset replication* - *Job output status*
Specialised Use Cases	
Pre-conditions	*Availability of the current validation data*
Post-conditions	*Availability of new validation data on a SE*
Basic Flow	1. *Data files are copied and registered on a GRID SE* 2. *Data files are replicated on the GRID SE* 3. *The system confirms success and reports the LDN of the registered data files* 4. *The system update the DS catalogue with the new physical location*
Devious Flow(s)	*DS access fails* *File transfer fails*
Importance and Frequency	*High frequency and importance*
Additional Requirements	

EO USE CASE: EO DATA FILE PACKAGING

Identifier	*UC#Eodatafilepackaging*
Goals in Context	*Evaluation of the efficiency/usefulness of the packaging method*
Actors	*Data administrator through JDL script*
Triggers	*A new data file packaging production to reduce the time of the validation processing*
Included Use Cases	- *Grid login* - *Job submission* - *Job monitoring* - *Specify program* - *DS access* - *Dataset transfer to Grid storage* - *Job output access*
Specialised Use Cases	
Pre-conditions	*Availability of the input dataset on the Grid SE*
Post-conditions	*Output data files registered on the Grid SE*
Basic Flow	*1. Data administrator specifies job information :* *a) Environment needed* *b) Grid dataset needed* *c) Packaging program to be executed on the Grid* *d) Output dataset registration on the GRID SE* *e) Output job status retrieval* *3. Jobs execution* *4. Output job status analysed*
Devious Flow(s)	*Grid input files not found or not produced;* *No output files after program execution*
Importance and Frequency	*High frequency and importance*
Additional Requirements	

EO USE CASE: CREATE LOGICAL COLLECTION

Identifier	*UC#CreateLC*
Goals in Context	*Create a Logical Collection*
Actors	*User, job*
Triggers	*New Logical Collection is created*
Included Use Cases	*UC#setaccessLC*
Specialised Use Cases	*Very similar to the Data Set registration use case UC#dsreg*
Pre-conditions	*VOMS proxy created, files to be collected are registered.*
Post-conditions	*Usable Logical Collection with the proper (default) ACL*
Basic Flow	1. *Actor specifies:* a. *The Logical Collection name* b. *List of logical filenames (that can be on different SEs via different protocols);* c. *optional metadata;* 2. *The Logical Collection is registered on the Grid* 3. *A new entry (with key the current LCN) is created*
Devious Flow(s)	
Importance and Frequency	
Additional Requirements	

EO USE CASE: CREATE LOGICAL COLLECTION

Identifier	*UC#CreateLC*
Goals in Context	*Create a Logical Collection*
Actors	*User, job*
Triggers	*New Logical Collection is created*
Included Use Cases	*UC#setaccessLC*
Specialised Use Cases	*Very similar to the Data Set registration use case UC#dsreg*
Pre-conditions	*VOMS proxy created, files to be collected are registered.*
Post-conditions	*Usable Logical Collection with the proper (default) ACL*
Basic Flow	4. *Actor specifies:* a. *The Logical Collection name* b. *List of logical filenames (that can be on different SEs via different protocols);* c. *optional metadata;* 5. *The Logical Collection is registered on the Grid* 6. *A new entry (with key the current LCN) is created*
Devious Flow(s)	
Importance and Frequency	
Additional Requirements	

EO USE CASE: SET ACCESS TO LOGICAL COLLECTION

Identifier	*UC#setaccessLC*
Goals in Context	*Set the access permissions of a Logical Collection, similar to Unix functionality, like chmod, chown*
Actors	*User, job*
Triggers	*Creation or modification of Logical Collection*
Included Use Cases	*Logical Collection creation*
Specialised Use Cases	
Pre-conditions	*Logical collection defined, VOMS proxy created, Access right to the Logical Collection*
Post-conditions	*ACL created or modified attached to the Logical Collection*
Basic Flow	*User issues a middleware command on an existing Logical Collection to change the ACL. User specifies:* *1. Use VOMS information or own ACL* *2. Logical Collection* *Command is run* *Result is replicated over all existing replicas of the Logical Collection*
Devious Flow(s)	*Authorisation fails, invalid permissions*
Importance and Frequency	*High importance and frequency*
Additional Requirements	