GOG Software Requirements Specification

prepared by: Y. Isasi, P. Sartoretti, C. Babusiaux, X.L̃uri, E. Masana
approved by: CU2 CCB
reference: GAIA-C2-SP-UB-YI-002-2
issue: 2
revision: 1
date: 2009-03-31
status: issued

Abstract

This is the GOG Software Requirements Specification Document.

Document History

Issue	Revision	Date	Author	Comment
2	1	2009-03-31	XL	Version for CDR (very minor changes)
1	4	20-MAY-2008	YI	Modifications from SRR RID actions
1	3	15-MAY-2008	YI	Version for DPACE review. Non-Funcional requirements added
1	3	28-OCT-2007	YI	Comments from DPACE taken into account
1	3	28-OCT-2007	YI	Comments from DPACE taken into account
1	2	8-OCT-2007	XL	Version for DPACE review
1	1	28-SEP-2007	XL	Version for internal review
D	3	27-SEP-2007	YI	First complete version
D	2	02-JUL-2007	YI	SRS Review Issued
D	1	20-OCT-2006	YI	Initial Issued Version

Contents

1 Introduction

This is the Software Requirement Specification for GOG, which describes in detail the objectives of the software simulator and the requirements to be considered on its design and construction.The document is organized in a hierarchical way going from a higer level perspective to a more detailed view on GOG operational flows.

1.1 Objectives

GOG is a tool to directly get Catalogue, Main Database (MDB) data and final (in the sense of statistically equivalent to the final mission data) data without the use GASS telemetry, GIBIS or the main ESAC database The Gaia Object Generator (GOG) is expected to simulate catalog data, auxiliary data and intermediate main database data (including mission final data) for all the Gaia instruments. It will use error models to provide rapid simulations of large amount of data. The realism of the simulations depends on the realism of the available error models. The schema shown in figure 1 depicts from a high level perspective where GOG will produce its data on the overall current data flow for Gaia:

GOG is based on the Universe and spacecraft & payload models of the GaiaSimu library (GAIA-C2-TN-OPM-FC-001) and uses error models to provide rapid simulations of large amount of data. The realism of the simulations depends on the realism of the available error models. The implementation of the instrumental error models is planned to be a gradual process that will be based on the study of the detailed pixel-level simulated data produced by GIBIS. The simulation of processed data requires models of the calibration errors that will be provided by the CUs in charge of the calibration algorithms development. In order to simulate data at a precise step of the processing, the addition of the calibration errors is planned to be user-selectable. The output format of GOG simulated data will be according to the Interface Control Document for the Gaia Main database (GAIA-C3-SP-ESAC-JH-002 in preparation). Note that some of the steps and options described here will have to be revised when the actual ICD will be available. GOG is expected to be in continuous evolution depending on the availability of the error models. To allow satisfying all the needs to access specific data, GOG will be entirely modular and configurable through an input configuration file. For the source generation, 2 main options will be available:

1. Generation of the data from the Universe model of the GaiaSimu library: the user will specify a sky region (e.g. an HTM, square, circle). This sky region can also be defined as the region scanned by Gaia during a given time interval.
2. Generation of the data for an input object catalog: this catalog is generated by the user and provides all the necessary information of the objects to simulate (with or without spectra as input).

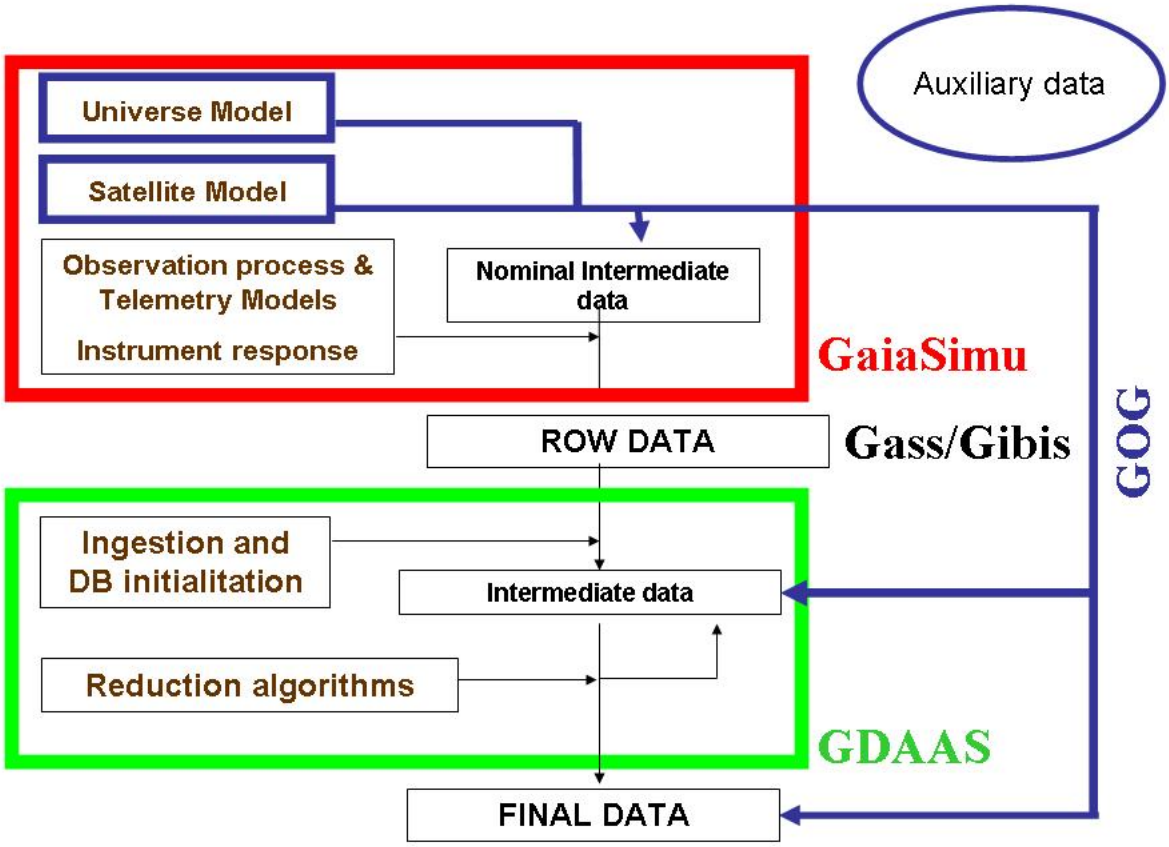

FIGURE 1: GOG global overview diagram

Note that the relative positions of the sources with each other will not be used by GOG to add overlapping effects, GOG only works source by source. Only the density at the source position could be used to adjust the error models (if they are provided as a function of density). However, we consider the possibly, for a GOG version close to launch, to work also per sky regions (in addition of working source by source), and to accurately model the overlapping effects (TBC).

Three main outputs will be available:

- universe model catalog data, according to the universe model ICD.
- main data base data epoch data: this is the data as observed and reduced for each transit of the source. The user can ask for only one typical epoch data or for all the transits of the source. Those outputs contain the field angles, epoch radial velocity, G magnitude, spectra, etc.
- main data base data combined data: this is the output produced by combining all the available epoch data, e.g. the derived parallax, proper motions, mean radial velocity, G magnitude, spectra, etc. The combined data can be queried for a given reduction stage (6 months, 12 months, etc. from the mission start) or for all transits (thereby providing the end-of-mission data). There is also the possibility, especially used when the sources are generated from an input object catalog with no information on the object position implying that the number of transits can not be derived, to specify the number of transits (for end-of-mission data) instead of the reduction stage.

For the noise generation, the noise effects will be user-selectable: they can be added or not and their values may also be given by the user (depending on the noise). Two main types of noises will be simulated:

1. Instrumental noise: noise that will come from the observation process, e.g. on-board detection and tracking or observation dead-time.
2. Reduction noise: Noise that will be added to the data when trying to remove the instrumental noise (see above) and extract the scientific content, e.g. residuals from cosmic rays removal, background subtraction, subtraction of close sources or second FoV contamination.

This will depend on the availability of the models.

Note that GOG aim will not be to produce raw data but pre-processed data so that GOG will put more effort on simulating how the instrumental noise will be removed (or not) than on

simulating the instrumental noise itself. The user interested on the simulation of the details of the instrumental noise (including crowding issues) should use GIBIS instead. Note that spectra can be considered as close to the instrumental noise as the PSF in particular will never be removed from the raw spectra.

1.2 Scope

The present document is address to all the Cu's members that require an understanding of the software requirements,restriccions and assumptions considered to design and build GOG. To help on this, several SADT diagrams of the main and alternate execution flows are provided.

1.3 Assumptions

The following requirements have been capture thinking on a GOG generic operational behaviour. At this stage some spectations for GOG might changed.The main hurdle GOG has, is that the contents of the main database are not fully completed. For this reason the initial versions of Gog will consider the current specifications and will be based on the restrictions described in section 11.

1.4 Applicable Documents

JH-001	Interface Control Document for the Gaia Main Database
CR-001	Gaia Universe Model Interface Control Document
TL-001	DPAC Product Assurance Plan
XL-005	Software Development Plan for CU2

1.5 References

Hernandez, J., 2008, *Main Database Interface Control Document*,
GAIA-C1-SP-ESAC-JH-001,
URL http://www.rssd.esa.int/llink/livelink/open/2786145

Levoir, T., Damery, J., Hoar, J., et al., 2007, *DPAC Software Product Assurance Plan*,
GAIA-C1-PL-CNES-TL-001,
URL http://www.rssd.esa.int/llink/livelink/open/2439085

Luri, X., Babusiaux, C., Wallut, J., 2009, *CU2 Software development plan*,
GAIA-C2-PL-UB-XL-005,
URL http://www.rssd.esa.int/llink/livelink/open/2785388

Reyle, C., Robin, A., Arenou, F., et al., 2008, *Universe model ICD*,
GAIA-C2-SP-LAOB-CR-001,
URL http://www.rssd.esa.int/llink/livelink/open/2732832

1.6 Definitions, acronyms, and abbreviations

The requirements set out in this SRS follow the labeling scheme:

CU2-*GOG-module-X-SCOPE-xxx*

where *GOG* is the Software Product Label , *module* identifies the GOG module. *X* is either S (for **S**cientific), T (for **T**echnical), Q (for **Q**uality Assurance), or M (for **M**anagement). *SCOPE* is a three or four letter scope specification of the requirement following the identified list of possible values of TL-001 whenever feasible. *xxx* is a monotonically increasing counter starting at 20 for every unique combination of *module-SCOPE*.

Each requirement is presented with its unique label and a number of attributes in accordance with TL-001 in the following form:

GOG-module-X-SCOPE-**020**	$C.v$	Verific.	Status
Description			
Parent: Parent			

with (see TL-001 for lists of allowed values, meanings and valid ranges):

GOG-module-X-SCOPE-xxx
The unique identifier of the requirement (see above)

$C.v$	Version number of the requirement composed of major (C) and minor (v) part
Priority	Priority of the requirement
Verification	Envisaged validation method of requirement - this will be either AUT for automated or MAN for Manual. The Software Test Plan defines precisely how the requirement is verified.
Status	Status identifier
Parent	Higher level requirement or requirements, comma separated list

SCOPE is a three character scope specification of the requirement one of :

REQ: concerning requirements,

SPC: concerning specification activities,
DES: concerning design activities,
IMP: concerning implementation activities,
INT: concerning integration activities,
VAL: concerning validation activities,
ACP: concerning acceptance activities,
MNT: concerning maintenance activities,
OPS: concerning operation activities,
FUN: concerning functional requirements,
PLN: concerning organisational requirements (including planning, phases, reviews, etc.),
SUP: concerning support requirements,
PRF: concerning performance requirements (including dimensioning, resource estimation, etc.),
RSK: concerning risk management requirements,
CTL: concerning verification and control activities,
CNF: concerning configuration management activities,
DIM: concerning dimensioning activities.

The following is a complete list of acronyms used in this document.

List of acronyms used in this document:

Acronym	Description
CU	Coordination Unit (in DPAC)
DPAC	Data Processing and Analysis Consortium
DU	Development Unit (in DPAC)
GASS	GAia System-level Simulator
GIBIS	Gaia Instrument and Basic Image Simulator
GOG	Gaia Object Generator
I/O	Input/Output
QA	Quality Assurance
SADT	Structured Analysis and Design Technique
SRS	System Requirement Specification
UM	Universe Model
WP	Work Package

2 General description and requirements

2.1 Context

On this overview we see GOG as a unique software product. From the high level perspective it can be decomposed on the SADT diagram as follows:

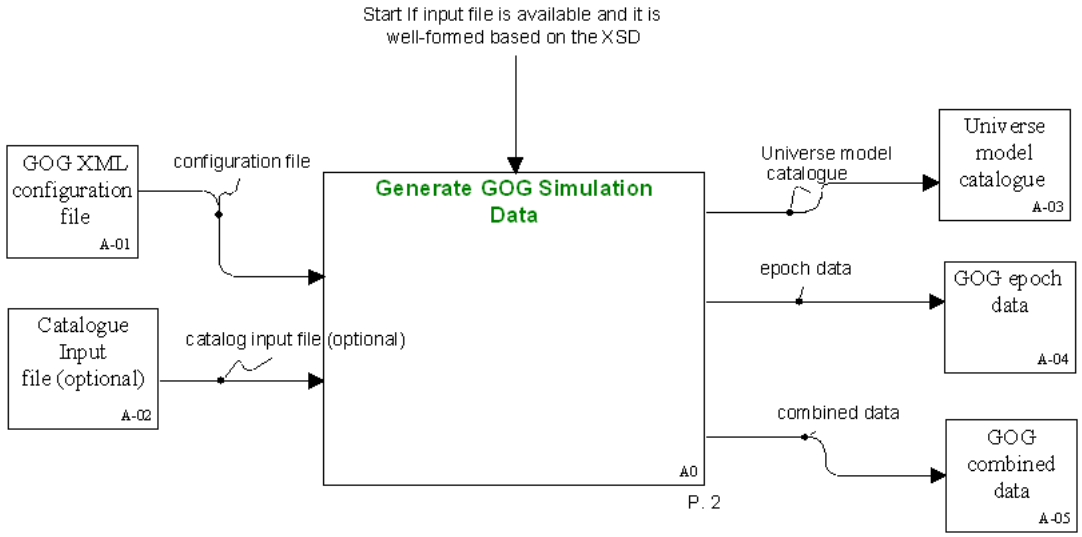

FIGURE 2: GOG general i/o context diagram

Exploding the central box, looking one level deeper, we could distinguish the following main blocks of execution. The following pictures shows how the problem is depicted.

2.2 Decomposition

According to TL-001 a 'Software Product' is composed of several 'Modules'. In this section we describe how the system is logically decomposed and mention the modules which will appear from Sect. 3 onward and how they interact.

GOG as a software product can be decomposed in software modules. We have identified the following main conceptual modules

- Initialisation Module
- Epoch parameters Module
- Combined parameters Module
- Spectrum simulation Module
- I/O Module

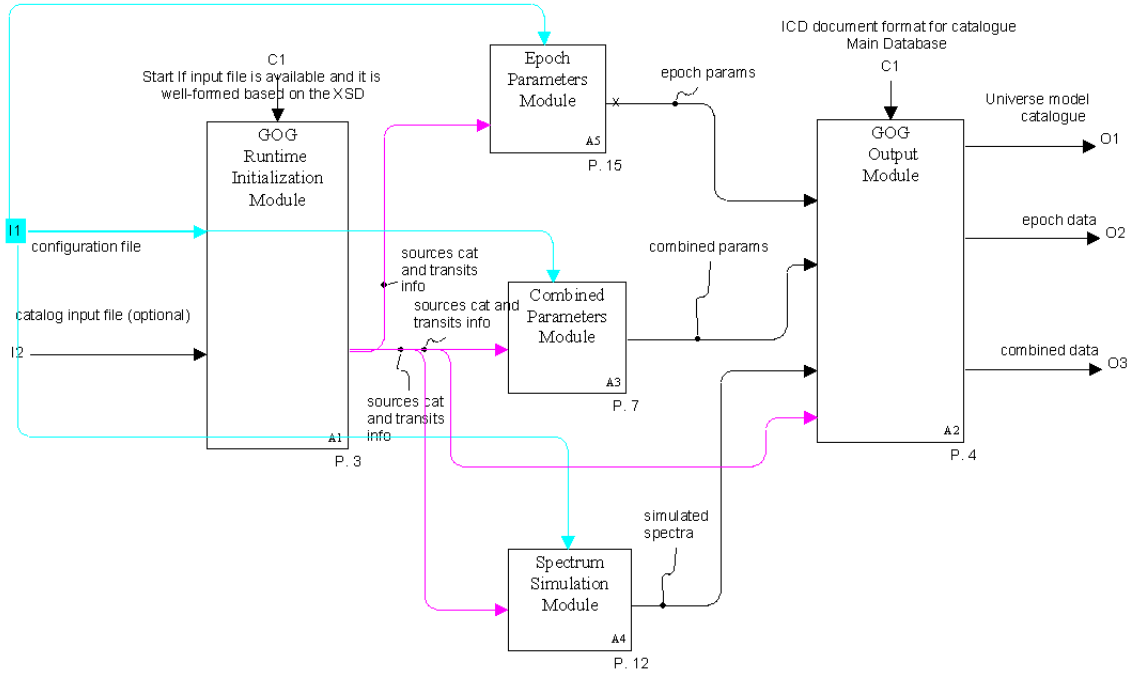

FIGURE 3: GOG data simulation general view

2.2.1 Initialisation module

Figure 4 shows how the initialization module is decomposed

2.2.2 Epoch Parameters Module

If epoch data is requested a transit calculation takes place. The transit (epoch) parameters are described in figure 5.

The epoch astrometry is decomposed as shown in figure 6.

2.2.3 Combined Parameters Module

If combined data are requested the combined parameters are computed for every requested mode (see figure: 7).

The combined astrometry parameters computation flow is different from the epoch one as shown in figure 8

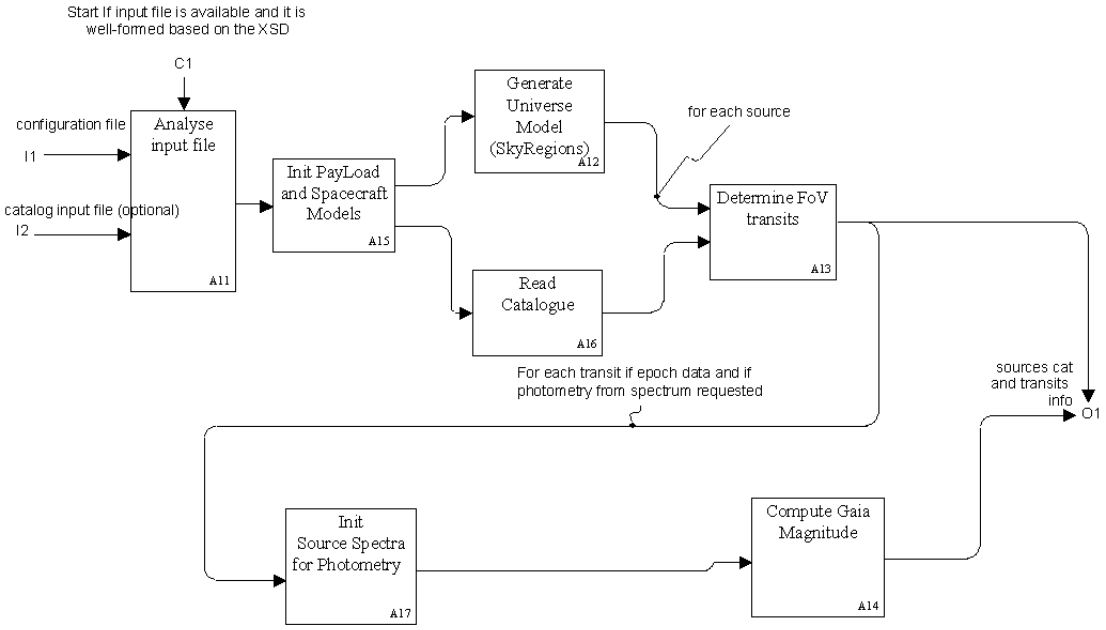

FIGURE 4: GOG initialisation module general view.

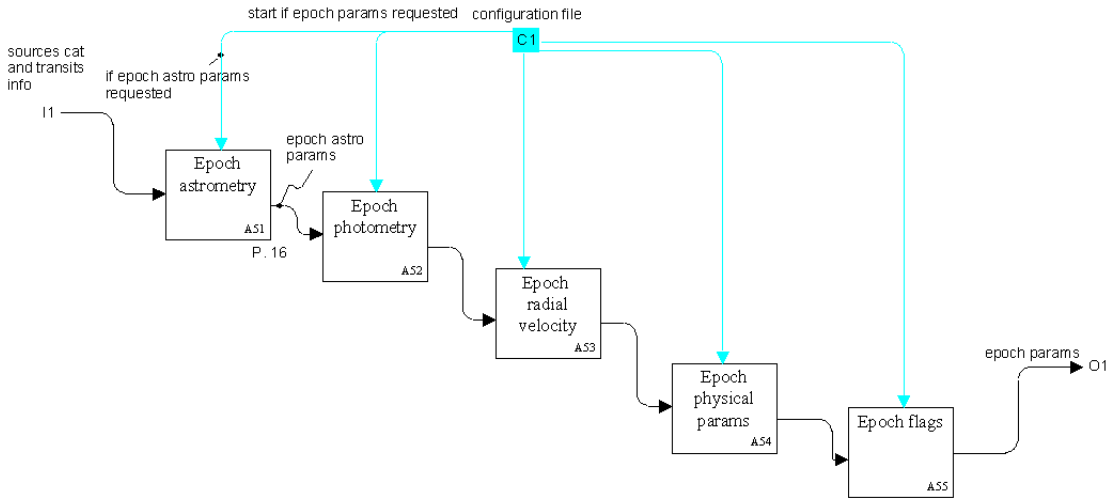

FIGURE 5: GOG epoch data computation flow.

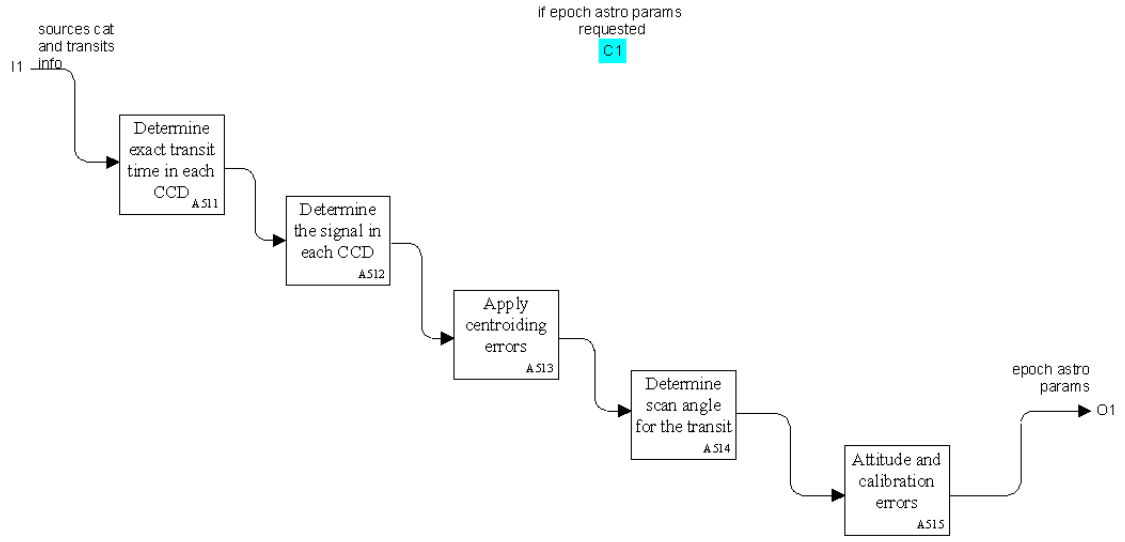

FIGURE 6: GOG astrometric epoch data generation flow.

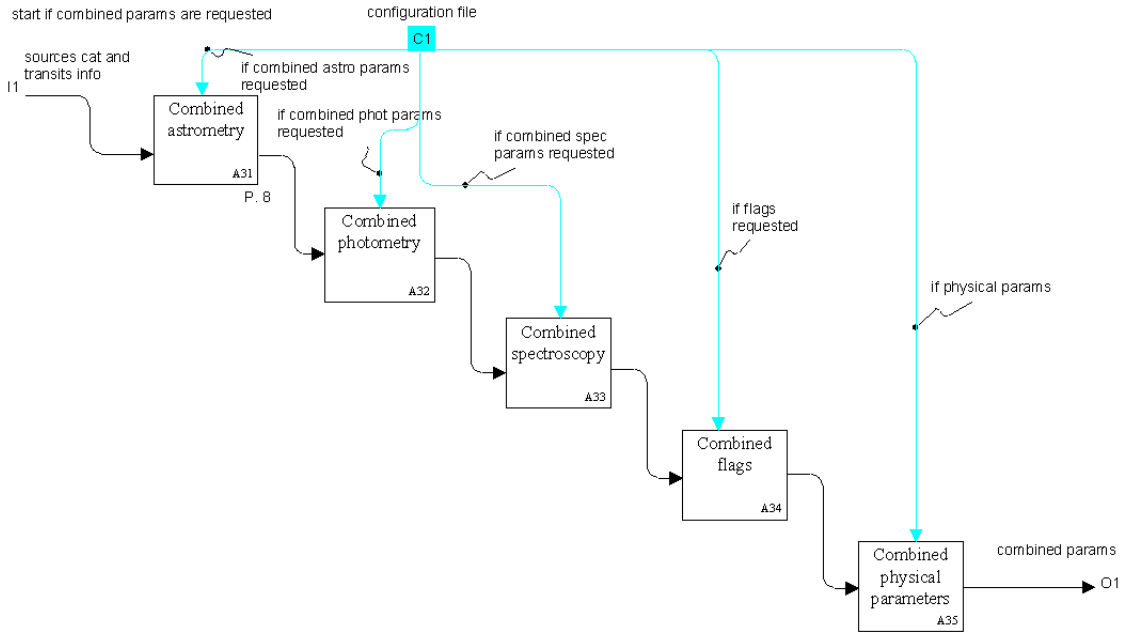

FIGURE 7: GOG combined data computation flow.

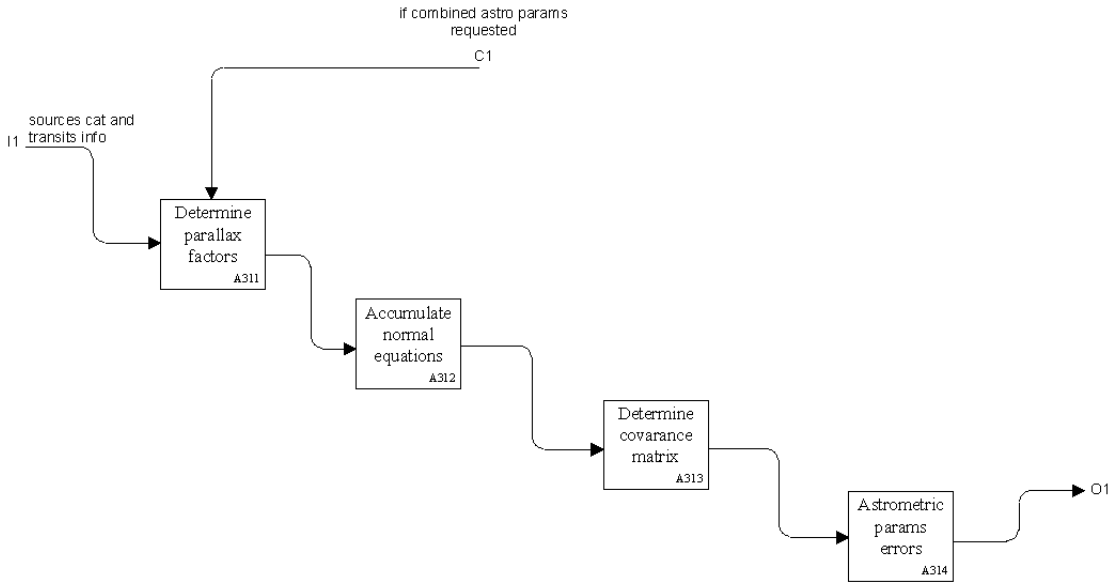

FIGURE 8: GOG astrometric combined data flow.

2.2.4 Spectrum simulation module

Figure 9 depicts the alternate simulation flow for spectra processing.

2.2.5 Output module

GOG output module has the responsability to write the final results. Figure 10 shown its decomposition.

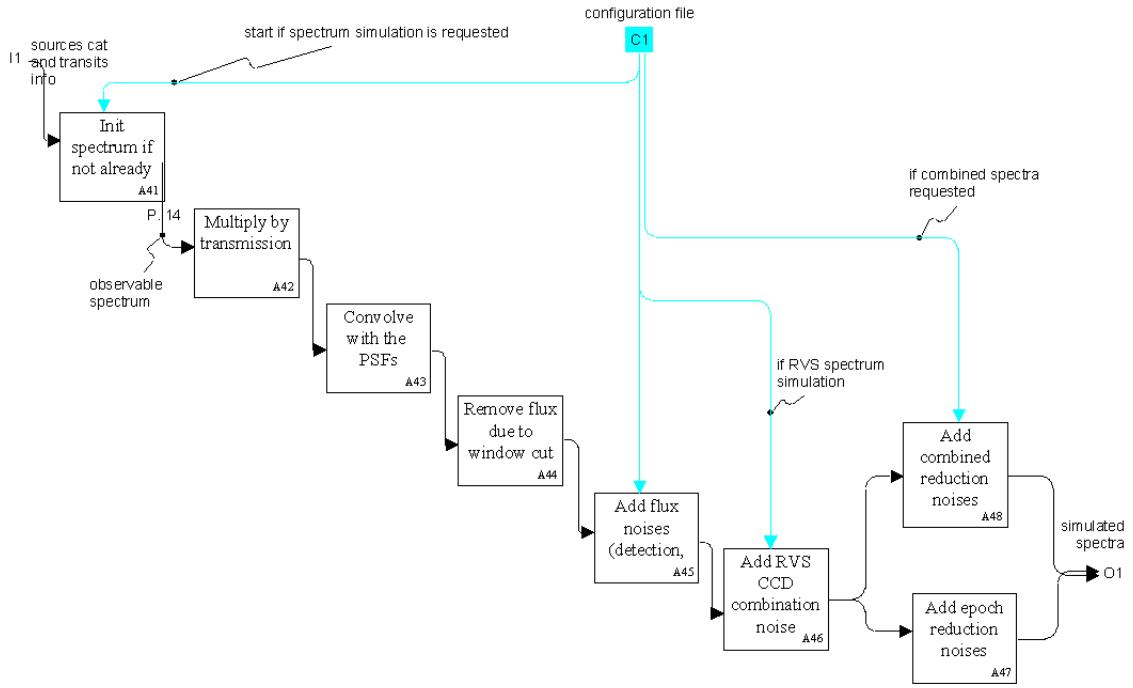

FIGURE 9: GOG spectra simulation flow.

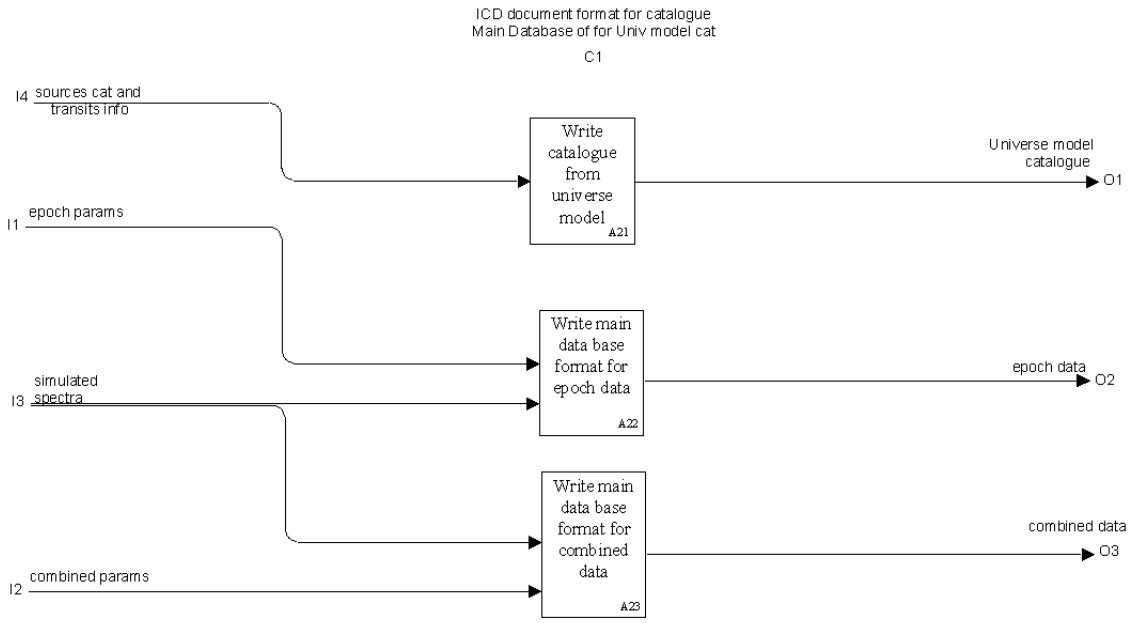

FIGURE 10: GOG output generation flow.

2.3 Requirements

This is for high level requirements which apply to all modules. These requirements may be removed from this document and appear in a higher level requirements document for CU2or at the DPC level. If this section is empty mention the higher documents - they should be in the applicable document list.

GOG-GEN-S-FUN-020	1.1	AUTO	Issued
CU2 will be responsible for the design and developing of GOG that will provide Catalogue, Main Database (MDB) and final data.(in the sense of statistically equivalent to the final mission data) data without the use GASS telemetry, GIBIS or the main ESAC database			
Parent: DPAC-CU2-S-FUN-320			

GOG-GEN-S-FUN-040	1.1	MAN	Issued
GOG will be fully integrated in the Gaia Simulator project. That is, it will use as other the other simulators the Gaia Simu library and the Gaia Tools library.			

GOG-GEN-S-FUN-060	1.1	MAN	Issued
For source generation GOG has 2 main options available: Universe Model from GaiaSimu or a Object Input Catalogue			

GOG-GEN-S-FUN-080	1.1	MAN	Issued
GOG has a source by source processing mechanism			

GOG-GEN-S-FUN-100	1.1	MAN	Issued
GOG must consider noise effects. For noise generation those effects should be user-selectable (they can be added or not and the values should be given by users).We classify noises in two groups: 1. Instrumental noises: noises that come from observation process such as:on board detection and tracking or observation dead-time 2. reduction noise: noise added to the data when removing instrumental noise			

GOG-GEN-S-FUN-120	1.1	MAN	Issue
GOG I/O Catalogue format is driven by [AD3] . Main Data base format is defined in [AD1]			

GOG-GEN-S-FUN-140	1.1	MAN	Issued
GOG must give the same result for identical input configuration parameters			

GOG-GEN-T-NFUN-020	1.1	MAN	Issued
GOG must provide a way to be run from a command line or a shell script.			

GOG-GEN-T-NFUN-040	1.1	MAN	Issued
GOG should be able to run in a cluster or grid environtments			

GOG-GEN-T-NFUN-060	1.1	MAN	Issued
GOG is built as a standalone java aplication that could be invoked from a command line and deployed as a jar file.			

GOG-GEN-T-NFUN-080	1.1	MAN	Issued
GOG is designed to run on either in a single thread mode on grid environments.			

GOG-GEN-T-NFUN-100	1.1	MAN	Issued
GOG runs on LINUX , UNIX and MAC computers (minimum).GOG results are platform independent			

GOG-GEN-T-NFUN-120	1.1	MAN	Issued
GOG will have to generate data of a considerable amount of astrosources whose number could be a significant part of the expected final catalogue (20%).			

GOG-GEN-T-NFUN-140	1.1	MAN	Issued
GOG must be able to process one complete source processing on 2 CPU seconds.			

3 Initialisation Module

3.1 Description and Objectives

The initialisation modules is in charge of loading and storing the read configuration in a proper way. T he module has to offer a common interface to get values from the configuration in an easy manner.

3.2 Requirements

List here all of the requirements on the module including functional and non-functional requirements

GOG-CON-T-FUN-020	1.1	MAN	Issued
GOG initialisation Module should be able to read information of the input sources. Options are to use the Universe Model or a provided user sources			
Parent: DPAC-CU2-S-FUN-380			

GOG-CON-T-FUN-040	1.1	MAN	Issued
The GOG catalogue module must include information on the universe model: • Galaxy model • SkyBackGroung model • Extinction Law model • DMS model			
Parent: DPAC-CU2-S-FUN-380			

GOG-CON-T-FUN-060	1.1	MAN	Issued
GOG initialisation module must include information on the instrument model: • The selected scanning law • Dead time. • Noise models (including detection noise)			
Parent: DPAC-CU2-S-FUN-380			

GOG-CON-T-FUN-080	1.1	MAN	Issued
The GOG catalogue module must include information on the reduction model: • Instrument effects substraction node bias substraction noises PPE residuals noise • Scientific content extraction noise substraction of close sources second FoV contamination noise backgound substraction.			
Parent: DPAC-CU2-S-FUN-380			

GOG-CON-T-FUN-100	1.1	MAN	Issued
GOG initialisation module will define the output specification output parameters (spectrum RPBP/RVS, astrometry, Vr, physical parameters and flags) epoch and/or combined number of epochs			
Parent: DPAC-CU2-S-FUN-380			

.

3.3 Input / Output

Input: an XML configuration file

Output: a memory properties-like structure loaded ready to be used

For further descrption on the inputs refer to GOG ICD input parameters section

4 Epoch Parameters module

4.1 Description and Objectives

The epoch parameters module is in charge of all the epoch parameters settings and calculations of the GOG simulation

4.2 Requirements

List here all of the requirements on the module including functional. This includes non-functional requirements

GOG-EPCH-S-FUN-020	1.1	MAN	Issued
Determine for epoch data the astrometric parameter (with instrumental and reduction errors): transit times and field angles in each CCD mean transit time and field angle for the transit.Gate activation and signal loss due to window cutoff must be taken into account.			

GOG-EPCH-S-FUN-040	1.1	MAN	Issued
Determine the epoch data photometry (with instrumental and reduction errors): G mag (for each CCD and average for the transit), GRP, GBP, GRVS . Gate activation and signal loss due to window cutoff must be taken into account.			

GOG-EPCH-S-FUN-060	1.1	MAN	Issued
Determine the epoch radial velocity (with instrumental and reduction errors)			

GOG-EPCH-S-FUN-080	1.1	MAN	Issued
Determine the epoch flags			

GOG-EPCH-S-FUN-100	1.1	MAN	Issued
Determine the epoch physical parameters (Av, Teff, logg, FeH)			

4.3 Input / Output

The input is the source itself that has a particular transit

The output is the epoch data bean

5 Combined Parameters module

5.1 Description and Objectives

The combined parameters module is in charge of all the combined parameters settings and calculations of the GOG simulation

5.2 Requirements

GOG-CMB-S-FUN-020	1.1	MAN	Issued
• Compute the astrometric combined data using all previous transits: • Compute parallax and proper motion errors • Determine de covariance matrix • Determine estimated errors for the astrometric parameters at the mean epoch			

GOG-CMB-S-FUN-040	1.1	MAN	Issued
Determine the combined data photometry: mean G, GRP, GBP, GRVS			

GOG-CMB-S-FUN-060	1.1	MAN	Issued
Determine the mean radial velocity			

GOG-CMB-S-FUN-080	1.1	MAN	Issued
Determine the final flags (variability, non-single star.)			

GOG-CMB-S-FUN-100	1.1	MAN	Issued
Determine the mean physical parameters (classification, Av, Teff , logg, FeH)			

5.3 Input / Output

The input is the source itself

The output is the combined data bean

6 Spectrum Simulation Module

6.1 Description and Objectives

The spectrum module is in charge of all the spectrum settings and calculations of the GOG simulation. This module covers both RP/BP and RVS spectrum simulation.

6.2 Requirements

GOG-SPCT-S-FUN-020	1.1	MAN	Issued

Noise consideration on:

- Flux noise (Poisson noise, total detection noise,...)
- CCD combination noise for (RVS)
- Wavelength calibration
- Multi-transit combination noise (only for combined data)should be added

6.3 Input / Output

The input is the source itself.The output is the epoch and or combined data bean

7 Output Module

7.1 Description and Objectives

The output module comunicates GOG flow execution with the filesystem. When the flow request a file it obtains it from the system and also it writes the files requested as GOG outputs

7.2 Requirements

GOG-IO-T-FUN-020	1.1	MAN	Issued
The universe catalog output data of the simulation has to follow the universe model ICD document.Note: GaiaSimu library will allow to write and read this format.			

GOG-IO-T-FUN-040	1.1	MAN	Issued
GOG output should be driven by the MDB data model.The output will be used to distribute extracts to the DPC's			
Parent: DPAC-CU2-S-FUN-340,DPAC-CU2-S-FUN-360			

7.3 Input / Output

The output data of the simulation has to follow the universe model ICD document.

A Traceability to upper level requirements

The traceability between this SRS and CU or DPC SRS or the SIRD is given here.

Parent Requirement	Requirements in this document
DPAC-CU2-S-FUN-320	CU2-GOG-GEN-S-FUN-20
DPAC-CU2-S-FUN-340	CU2-GOG-IO-T-FUN-40
DPAC-CU2-S-FUN-360	CU2-GOG-IO-T-FUN-40
DPAC-CU2-S-FUN-380	CU2-GOG-CON-T-FUN-20, CU2-GOG-CON-T-FUN-40, CU2-GOG-CON-T-FUN-60, CU2-GOG-CON-T-FUN-80, CU2-GOG-CON-T-FUN-100