

JASMINE simulator

Seiji Ueda¹, Yoshiyuki Yamada², Takashi Kuwabara³,
Naoteru Gouda⁴, Takuji Tsujimoto⁴, Yukiyasu Kobayashi⁴,
Tadashi Nakajima⁴, Hideo Matsuhara⁵, Taihei Yano⁴,
Masahiro Suganuma⁴, and the JASMINE Working Group

¹The Graduate University for Advanced Studies, Mitaka, Tokyo, 181-8588, Japan
email: seiji.ueda@nao.ac.jp

²Department of Physics, Kyoto University, Kyoto, 606-8502, Japan
email: yamada@amesh.org

³Department of Physics, Niigata University, Niigata, 950-2181, Japan
email: takashi@astro.sc.niigata-u.ac.jp

⁴National Astronomical Observatory Japan, National Institutes of Natural Sciences, Mitaka,
Tokyo 181-8588 Japan
email: naoteru.gouda@nao.ac.jp, taku.tsujimoto@nao.ac.jp, yuki@merope.mtk.nao.ac.jp,
tadashi@dodgers.mtk.nao.ac.jp, t.yano@nao.ac.jp, suganuma@merope.mtk.nao.ac.jp

⁵Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency,
Sagamihara, Kanagawa, 229-8510 Japan
email: maruma@ir.isas.ac.jp

Abstract. We explain simulation tools in the JASMINE project (JASMINE simulator). The JASMINE project stands at the stage where its basic design will be determined in a few years. Therefore it is very important to simulate the data stream generated by astrometric fields at JASMINE in order to support investigations of error budgets, sampling strategy, data compression, data analysis, scientific performances, etc. We find that new software technologies, such as Object Oriented (OO) methodologies, are ideal tools for the simulation system of JASMINE (the JASMINE simulator). In this article, we explain the framework of the JASMINE simulator.

1. Introduction

The JASMINE project stands at the stage where its basic design will be determined in a few years. Thus it is very important to simulate the data stream generated by astrometric fields at JASMINE in order to support investigations of error budgets, sampling strategy, scientific performances, etc. The simulation system should include all components in JASMINE as “objects” in order to resolve all issues which can be expected beforehand and make it easy to cope with some unexpected problems which might occur during the mission of JASMINE. Components includes not only models of concrete materials such as scientific instruments and satellite bus system, but also abstract ones such as observation methods, orbits, data transfer, algorithms of data analysis, etc. Furthermore, many researchers in various fields, including engineering of the bus system, should participate in developing the simulator. Then the simulator includes many objects and furthermore, flexibility and interaction are needed to cope with the objects. On the other hand, calibrations must be protected from unintentional modification for each model of the objects. Thus we conclude that the Object Oriented methodologies are ideal for the demands described above in the simulator. High maintainability and reusability

of the object oriented approach is very useful in making the simulator system. Thus we are now constructing the JASMINE simulator using techniques developed in the field of information science.

In §2, requirements of JASMINE simulator are discussed; §3 contains some examples of JASMINE system investigations with the simulator software; §4 is devoted to the future schedule of software development.

In this paper we will make reference to terms with specific meanings within a object-oriented programming context. These are:

- class:** encapsulates common behaviour of a group of objects;
- attributes:** data member of a class;
- methods:** functions which may be performed on instances of a class;
- object:** an instance of a class;
- abstract class:** a class from which no instances may be created;
- inheritance:** a way to form new classes or objects using predefined objects or classes where new ones simply take over old ones' implementations and characteristics.

2. Specification requirement of framework

Implementation of the simulator is carried out in two steps. In the first step, we construct or choose the framework of the software. The second step is building software which are JASMINE specific under the framework built in the first step. In this section, we discuss the framework of the JASMINE simulator.

2.1. Implementation of proto-model

JASMINE Simulator will be mainly used for assessing the scientific goals of the mission, instrument design investigation, data treatment preparation, and algorithm study such as stellar image detection or evaluating astrometric parameters from raw data. The need for a clear, modular and easily communicable structure for the simulator led us to choose an object-oriented language for its implementation.

In the simulator, each component of JASMINE is treated as an “object” which has specifications as attributes. For example, an object which represents a CCD detector has attributes such as pixel size, number of pixels, quantum efficiency, flatness, charge transfer rate, etc. Such attributes affect final scientific outputs.

Within the context of system investigations, some attributes of some components may depend on other attributes of other components. We can draw diagrams which represents dependencies of each component (object) by analyzing the dependencies. This is called a “Data Flow Diagram”. The framework is expected to provide the function of creating and editing such “data flow” diagrams.

The “data flow” during JASMINE observation begins with astronomical objects, is related to an optical system, detectors, and data-processing units, and ends as a communication system. One simple implementation is the specifying of a relation of abstract classes which represent such components beforehand. Then the framework can deal with detailed investigation easily by making concrete classes that are inherited from the abstract classes. The following calculation is performed in the flow of the JASMINE specification study,

$$f = \frac{N_{\text{PSF}} w D}{\lambda},$$

where f , w , D , λ are total focal length of the optical system, pixel size of the CCD, diameter of the primary mirror, and wavelength, respectively, and N_{PSF} is a constant. An

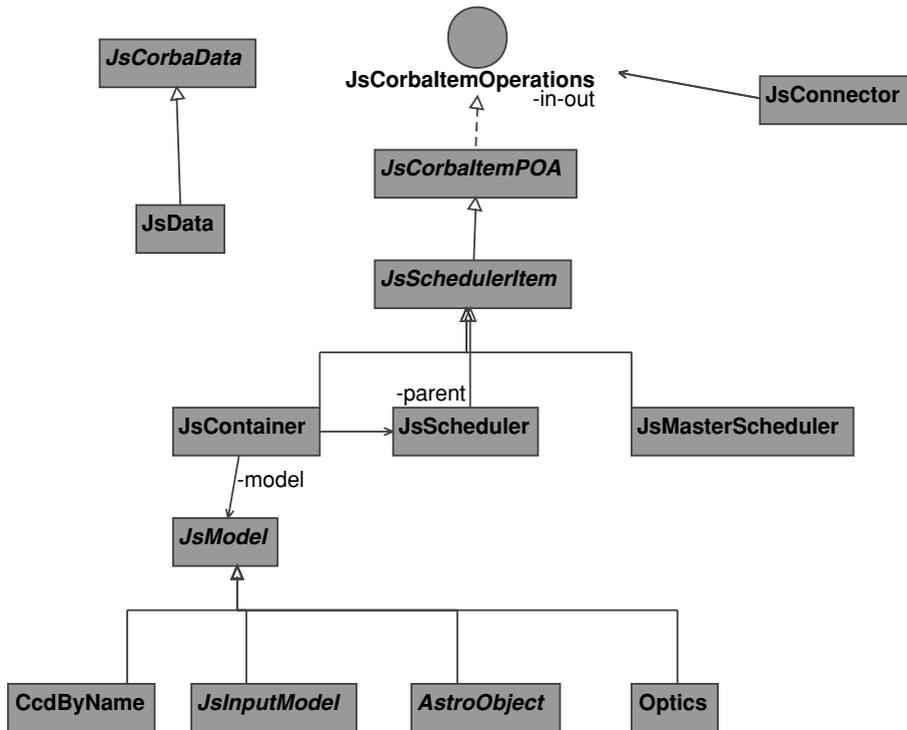


Figure 1. UML diagram of the framework of prototype simulator code. Names of classes JsModel, JsContainer, JsConnector, and JsData correspond to “model”, “container”, “connector”, and “data” of the text.

attribute of the optical system, f , depends on an attribute of the detector, w . These dependencies (“data flow”) have opposite direction of a real “data flow” in observation. One of requirements of our simulator is that we can specify direction of the “data flow” in our diagram as both the same and counter direction as the “data flow” of real observations.

From above considerations, the framework of the JASMINE simulator may have functions for managing DAG (directed acyclic graph). To summarize the requirements,

- computational modules are extendible,
- data types are extendible,
- open source (if using existing software),
- some simple graphical tools are provided,
- platform independent,
- availability of treating distributed computation via network.

Requirements (a)–(c) are critical and (d)–(f) are optional.

A schematic UML diagram of our prototype simulator is shown in Fig. 1. In our implementation of the prototype simulator, each of the components, such as optical systems or detectors are inherited from an abstract class “model”. A class “container” which manages the dependency graph has the class “model”. The structure ensures not also extendibility of modules but the arbitrariness of the direction of dependency. Each directed arc is represented as a class “connector”, which encapsulates flowing “data” between two “models”. The abstraction “data” ensures extendibility of data. The model can be easily extendible by making subclass of “model”.

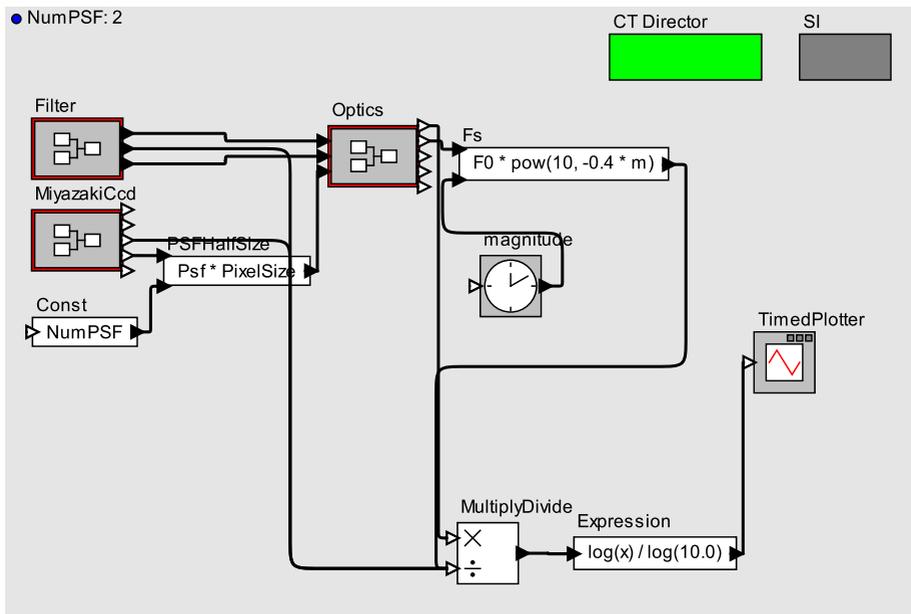


Figure 2. Framework of the prototype. Filter, MiyazakiCcd, and Optics modules are implemented as composite actors.

2.2. Ptolemy II for framework

Many implementations of such simple data flow simulators exist. Good examples are the visualization tools of numerical simulations. The methods of extending a calculation model are also provided in some tools. However, faults of such visualization tools are difficulties of data type extension. We concluded that building a JASMINE simulator as a module of AVS or OpenDX was not feasible.

The Ptolemy II project (<http://ptolemy.eecs.berkeley.edu/ptolemyII/>) carried out at UC Berkeley provides the framework which fills all of our above demands. Ptolemy II is designed as concurrent modelling and design tools under the fairly liberal UC Berkeley copyright (requirements a–d). It is implemented by Java language (requirement e). It also provides a “JXTA” class loader, which is JAVA PtoP implementation (requirement f). Thus, Ptolemy II is adopted as the framework of our simulator. A sample graph for the specification study implemented on Ptolemy II is shown in Fig. 2.

3. Contents of our simulator

In this section, we show implemented results and implementation plans of JASMINE specific components. We have already performed several items:

- observational method and accuracy,
- data reduction scheme,
- stellar image detection,
- data compression scheme,
- star number counts,
- optical system performance,
- satellite attitude and accuracy,
- orbits.

In this section, we show two topics.

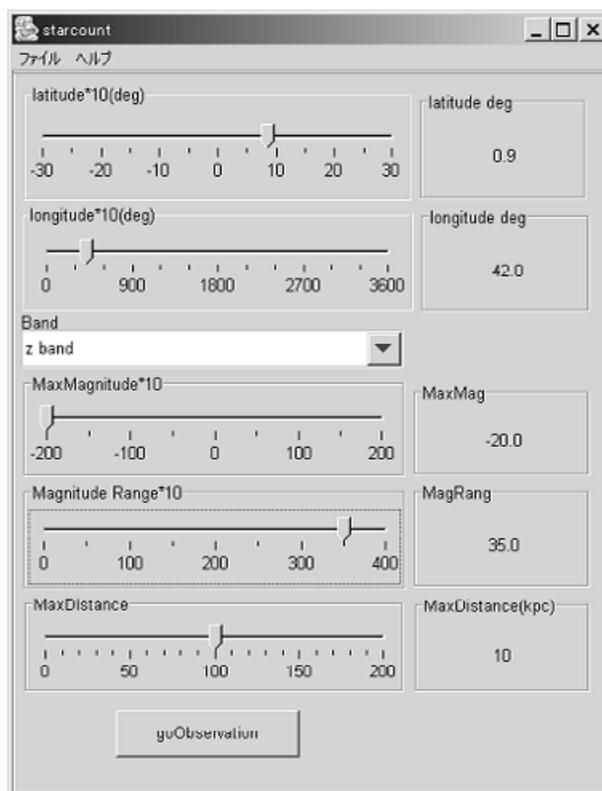


Figure 3. StarCount Simulator

3.1. Galaxy model and stellar number counts

The main targets of the JASMINE project are stars in the Galactic disk and bulge. There is no complete observation which shows number density of stars brighter than specified magnitude in the z-band and contained in the Galactic disk and bulge. Thus it is important to forecast the number of stars, data rate, etc. using a model Galaxy which is well-confirmed from observations. We use Wainscoat *et al.* (1992) and Cohen (1994) for the model of the stellar number distribution in the galaxy. This model reproduces well 2MASS results. We also take the motion of stars from the Evans model (1994), and the distribution of dust from COBE DIRBE data into account. An outline of the model is shown in appendix.

We implemented Star Count Simulator (SCS) first as Fig. 3. This is the tool for calculating THE number of stars with arbitrary direction, arbitrary distance, and arbitrary spectral types. This enables us evaluating the number of stars observed at every moment. We also implemented a three-dimensional viewer of the galaxy model (Fig. 4).

Fig. 5 shows the time variation of the number of stars observed by JASMINE. We cannot estimate the amount of processing by average stellar density because the variability of numbers is very large. Such information is very important for designing the data processing unit and data recorders.

3.2. Modelling of CCD and focal plane simulation

We have also started implementing the CCD model. The CCD can be modelled as the module which as many accumulators (pixels), data transfer mechanisms, and reading mechanisms. Efficiency of accumulation (quantum efficiency) is about 0.9 and depends

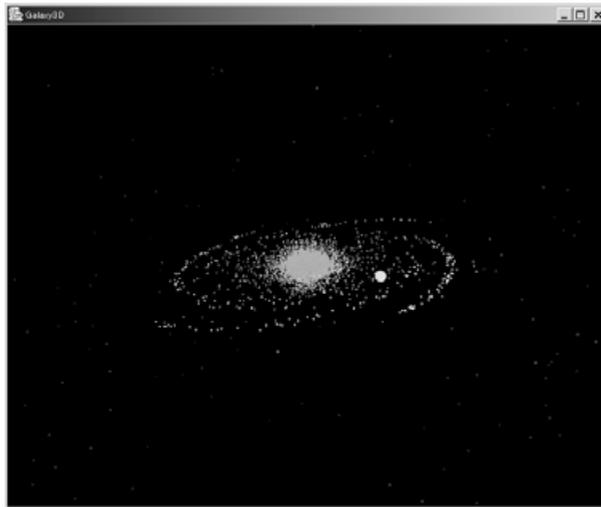


Figure 4. 3D Galaxy view.

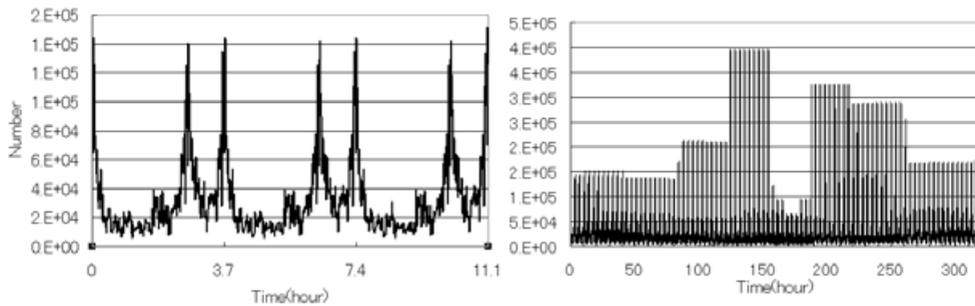


Figure 5. Time variation of Star Number.

on the wavelength. Data (charge) transfer efficiency is extremely close to unity, but is not unity. Readout of the data has also some errors. These effects are implemented as a probability process on the CCD model. Satellite environmental effects such as cosmic rays may affect such probabilities.

Using our galaxy model and satellite attitude model, a list of stars contained in two fields of view are generated. According to attributes of each star such as distance, spectral types, etc., photons are generated using a probability process. The astrometric optical system has a very long focal length. Point spread functions are broad compared with pixel size. To generate images on the focal plane, it is also required to use a probability process. From the above, several probability process (times, locations of detector plane) pairs belonging to each photon are generated. Each “photon” may be the trigger of the CCD model. By each trigger, the CCD model accumulates “charge” in the appropriate “pixel”. Continuing pseudo-exposure during an appropriate interval, we can get a pseudo-image of a JASMINE observation.

4. Conclusions

We chose Ptolemy II as a framework of the JASMINE simulator. We have already preformed several investigations on the JASMINE system design using Fortran, C++

and JAVA codes. We will reconstruct such codes in a form suitable for the Ptolemy II framework.

Appendix A. Milky Way model

In this model, 87 spectral types of stars are considered. We can estimate the number of stars classified by types independently. The original Cohen (1994) model provides information on the number of stars within seven bands, *B, V, H, I, J, 12 μm, 25 μm*. We extend it to *z* (by inserting Cohen data), *L, M* (see Tokunaga 2000) bands.

We consider five components of the galaxy: disk, bulge, halo, spiral arm, molecular ring. The number density $\rho(S)$ with given spectral type *S* and distance *r* is denoted as

$$\rho(S) = \rho(S)_{\text{disk}} + \rho(S)_{\text{bulge}} + \rho(S)_{\text{halo}} + \rho(S)_{\text{arm}} + \rho(S)_{\text{ring}}. \tag{A 1}$$

Number densities of each component are normalized by number density of *S* type stars near solar systems, $\rho_0(S)$. Density distributions of each component are as follows.

(a) exponential disk

$$\rho(S)_{\text{disk}} = \rho_0(S) f_D(S) \exp \left[-\frac{r - r_0}{h} - \frac{|z|}{h_z(S)} \right] \tag{A 2}$$

where $r_0, h, h_z(S)$ are the distance to the Galactic center (8 kpc), the radial scale length (3.5 kpc), and scale height for each *S* type star, respectively,

(b) axisymmetric bulge component

$$\rho(S)_{\text{bulge}} = \rho_0(S) f_B(S) g_B x^{-1.8} e^{-x^3} \tag{A 3}$$

where $x = \sqrt{r^2 + k_1^2 z^2} / r_1$ (k_1 : the bulge axis ratio = 1.6; r_1 : the bulge radius = 2 kpc). g_B : the ratio of densities of bulge at the point $x^{-1.8} e^{-x^3} = 1$ and disks near solar system = 3.6,

(c) Halo with $r^{1/4}$ law

$$\rho(S)_{\text{halo}} = \rho_0(S) f_H(S) g_H \exp \left[-7.669 \left(\frac{r}{r_e} \right)^{1/4} \right] \frac{r^{-7/8}}{r_e} \tag{A 4}$$

where r_e : the effective radius of the halo = 2.83 kpc, $g_H = 0.002$,

(d) spiral arm pair which shown in table A,

$$\theta(r) = a \log \left(\frac{r}{r_{\text{min}}} \right) + \theta_{\text{min}} \tag{A 5}$$

where

$$\rho(S)_{\text{arm}} = \rho_0(S) f_A(S) g_A \exp \left[-\frac{(r - r_0)}{h} - \frac{|z|}{h_z(S)} \right] \tag{A 6}$$

where g_A : ratio of the total area of disk and the area occupied by arm (=5),

(e) and molecular ring at $r \sim 0.45r_0$

$$\rho(S)_{\text{ring}} = \rho_0(S) f_r(S) g_r \exp \left[\frac{-(r - r_r)^2}{2\sigma_r^2} - \frac{|z|}{h_z(S)} \right] \tag{A 7}$$

where r_r : the ring radius = $0.45r_0$, $\sigma_r = 0.064r_0$, g_r : ratio of the total area of disk and ring (=25).

We use dust absorption with the result from the DIRBE/IRAS dust Map 9

Table 1. parameter of spiral arm

Arm	a	r_{\min} (kpc)	θ_{\min} (radian)	Extent (radian)	width (pc)
1	4.25	3.48	0.000	6.0	750
1'	4.25	3.48	3.141	6.0	750
2	4.89	4.90	2.525	6.0	750
2'	4.89	4.90	5.666	6.0	750
L	4.57	8.10	5.847	0.55	300
L'	4.57	7.59	5.847	0.55	300

(<http://astron.berkeley.edu/dust/>). To convert from the integration value provided by DIRBE to the real distribution, we assumed an exponential distribution of dust. The absorption δA_λ within the distance r and $r + \delta r$ is written as

$$\delta A_\lambda = A_{\lambda 0} \exp \left[-\frac{r - r_0}{h_{da}} - \frac{|z|}{h_{za}} \right] \delta r. \quad (\text{A } 8)$$

where $A_{\lambda 0}$: absorption near solar system, h_{da} : the radial scale length of the absorption = 3.5 kpc, h_{za} : the scale height of the absorption = 100 pc. The value of $A_{\lambda 0}$ at each (ℓ, b) is determined by

$$A_\lambda(\text{DIRBE}) = A_{\lambda 0} \int_0^\infty \exp \left[-\frac{r - r_0}{h_a} - \frac{|z|}{h_{za}} \right] dr. \quad (\text{A } 9)$$

The simulation of astrometric observations such as JASMINE also requires a model of stellar motion. The axisymmetric Evans (1994) model is simple for discussing the motion. The potential in the Evans model is as follows,

$$\Psi = \frac{\Psi_a R_c^\beta}{(R_c^2 + R^2 + z^2 q^{-2})^{\beta/2}}, \quad \beta \neq 0, \quad (\text{A } 10)$$

where q , Ψ_a , R_c are axis ratio of equipotential, central potential, core radius respectively. We can get mass density from Poisson's equation.

$$\rho = \frac{v_a^2 R_c^\beta}{4\pi G q^2} \frac{R_c^2(1 + 2q^2) + R^2(1 - \beta q^2) + z^2[2 - q^{-2}(1 + \beta)]}{(R_c^2 + R^2 + z^2 q^{-2})^{(\beta+4)/2}} \quad (\text{A } 11)$$

Using these models, we can compute the motion of stars in each component.

Acknowledgements

We would like to acknowledge Y. Kawakatsu, A. Noda, A. Tsuiki, M. Utashima, A. Ogawa, N. Sakou, H. Ueda for their collaboration in the investigations on the JASMINE spacecraft system. Furthermore, we would like to thank Miyashita Hisashi at IBM Tokyo Research for useful advice on general code construction technique. This work has been supported in part by the Grant-in-Aid for the Scientific Research Funds (15340066) and Toray Science Foundation.

References

- Cohen, M. 1994, *ApJ* **107**, 582
 Evans, N.W. 1994, *MNRAS* **267**, 333
 Tokunaga, A.T. 2000 *in Allen's Astrophysical Quantities, 4th edition*, ed. A.N. Cox, Springer-Verlag (New York), p. 143
 Wainscoat, R. J., Cohen, M., Volk, K., Walker, H. J., Schwartz, D.E. 1992, *ApJS* **83**, 111