Astronomical Data Analysis Software and Systems XXVI
ASP Conference Series, Vol. 521
Marco Molinaro, Keith Shortridge, and Fabio Pasian, eds.
© 2019 Astronomical Society of the Pacific

Easylife: a Conceptual Framework for Semi-automatic Survey Management

Paolo Luigi Scala,[1] Paolo Franzetti,[1] Marco Fumana,[1] Bianca Garilli,[1] and Marco Scodeggio[1]

[1]*INAF, IASF-MI, Milano, Italy;* `pls@lambrate.inaf.it`

Abstract. Easylife is a conceptual framework aimed at the semi-automatic management of spectroscopic surveys.

Conceived in 2012 (Garilli et al. 2012) as a tool to manage the VIPERS spectroscopic survey (Guzzo et al. 2014) and based on a prototype version of the FASE framework (Grosböl et al. 2012), it evolved into a survey-independent generalised framework following the MVC (Model-View-Controller) paradigm.

Easylife has been deeply modified to exploit PNGS (Pandora Next Generation Software) APIs (Fumana et al. 2019) and FASE stable version, and is currently used to manage the ongoing VANDELS ESO public spectroscopic survey[1] carried out using the VIMOS@VLT spectrograph.

1. The complexity of large-scale surveys management

Large-scale spectroscopic surveys represent nowadays a consolidated industry: this has been made possible by the scientific and technological advancements in the field of astronomical instrumentation.

The availability of instruments such as MOONS (Cirasuolo et al. 2014), a large field (500 square arcminutes), multi-object (500 object + 500 sky fibres) spectrograph proposed for the VLT Nasmyth focus, will allow to gather huge amounts of data that need to be efficiently and reliably managed in order to exploit its full scientific potential.

Managing a large-scale survey means also to coordinate data distribution, both among survey team members and to the broad scientific community, and keeping track of the survey processes, usually done by the project coordinator or by a restricted group of people.

Easylife has been conceived as a general framework which provides a coherent set of tools that allow a semi-automatic management of spectroscopic surveys, supporting astronomers in charge of managing this kind of projects during their whole lifecycle, from raw data gathering to the release of the resulting scientific products. It is general because it is not tied to a specific survey; it is a framework because it offers generic tools whose behaviour can be modified by writing additional code responding to users specific needs.

The tools offered by Easylife consist of:

[1]http://vandels.inaf.it/

- a graphical user interface to organise, reduce and automatically classify spectroscopic data offered by *pandora.vipgi* application (Fumana et al. 2019)
- a Web-based interface for: a) survey status monitoring in terms of targeted, observed, reduced and measured *elements*, where elements are survey-specific entities such as galaxies (VANDELS) or pointings (VIPERS); b) management of the redshift measurement and quality assessment workload of team members
- a set of algorithms for spectroscopic data reduction
- a set of customisable MySQL database tables collecting scientific and organisational data

These tools are realised through the interaction between different software objects, described in section 3.

2. From VIPERS to VANDELS

The first version of Easylife has been developed as an aid to the management of the VIPERS survey.

VIPERS was an ESO Large Programme aimed at measuring redshifts for $\sim 10^5$ galaxies at redshift $0.5 \leq z \leq 1.2$, to measure clustering, the growth of structure and galaxy properties at an epoch when the universe was about half its current age.

VANDELS is an ongoing ESO public spectroscopic survey which will observe around 2500 high redshift galaxies over 3 years spanning three samples of 20, 40 and 80 hours.

VIPERS and VANDELS differ in the nature of targeted objects, and this results in two radically different survey strategies: the former observed a large number of bright objects over a large sky area with a relatively low exposure time using a single mask, while the latter will observe few faint objects over a small area with a long exposure time using up to four masks.

Even though modularity has been taken into consideration in designing the first version of Easylife, it was still heavily tailored on the VIPERS survey. It was thus difficult to extend and modify the system to manage a very different survey such as VANDELS.

Instead of developing a new set of specialised tools, which will become obsolete as soon as the VANDELS survey will end, we decided to re-design Easylife following the MVC paradigm (see next section) and implementing it using the PNGS APIs, to take advantage of well-tested, already existing modular code. This allows to obtain a more flexible system, realised through adaptable, reusable components that is open for extension and, in case of future surveys, can be tailored to meet the specific needs that will arise.

3. Easylife framework architecture and functionalities

The Easylife framework follows the *model-view-controller* (MVC) architectural pattern, which conceptually subdivides the elements constituting a software product into three distinct and decoupled classes, or layers: model, view and controller.

It is based on the idea of separating the presentation layer (*view*) from the data model (*model*) and the application logic (*controller*), and defines the way elements in a given layer interact with elements of other layers: this distinction makes software products simpler to re-use and maintain.

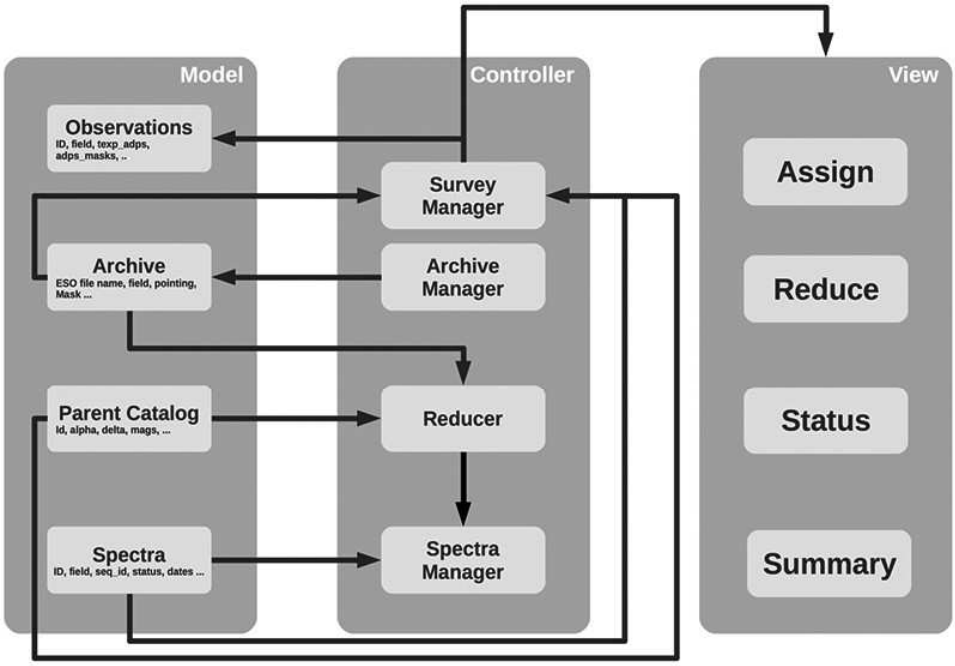

Figure 1. The MVC architecure of Easylife

In Figure 1 are depicted the various objects constituting the tools offered by Easylife, the role of each object according to the MVC pattern and the interactions between them; specifically, for each layer:

- **model:** objects in this layer encapsulate the application data (defined in the MySQL database tables) and implement basic operations on it: the *Archive* object realises the application-specific representation of the raw spectroscopic data, for example coming from the ESO archive (filename, seeing, airmass...); the *Observations* object represents information about a single astronomical object observations (requested exposure time, actual object exposure time); the *Parent catalog* object represents the survey input photometric catalog (galaxy identifiers, astronomical coordinates, magnitudes...); the *Spectra* object stores single spectra information (galaxy identifier, total exposure time, redshift value...)
- **view:** view objects represent the graphical user interface by means of which users can interact with the application: the *Assign* object renders the Web interface used by the P.I. to assign packets of raw data to scientists to be elaborated and keep track of the ongoing actions; the *Reduce* object is an ad-hoc configured version of *pandora.vipgi* and is the main interface for spectroscopic data management and classification; the *Status* object allows to graphically visualise the number of already performed observations against the total of them; the *Summary* object shows the number of objects that have been targeted, observed, reduced and

measured. The coupling of the *Status* and the *Summary* view objects allows an immediate perception of the advancement of the survey.

- **controller:** controller objects implement the application logic, activated by view objects and interacting with model objects (that is, objects that implement operations on data activated through the user interface): the *Survey manager* object implements the logic for automatically retrieving and deriving the data rendered by the *Summary* and the *Status* views objects; the *Archive manager* object implements archive-specific operations to communicate with the provider of the raw spectroscopic data (e.g. the ESO archive); the *Reducer* object is a collection of reduction algorithms which can be invoked directly from the command line or through the *pandora.vipgi* GUI; the *Spectra Manager* object implements the logic for assigning spectra to be measured and analysed to a scientist and keeping track of these assignments

4. Conclusions

The first version of Easylife has been successfully employed in the management of the VIPERS survey and it offered a solid starting point for the current Easylife implementation, which is being used to manage the VANDELS survey.

By adopting the MVC architectural pattern for the design of the new system and the PNGS APIs for its implementation, we have been able to obtain a more maintainable and extensible software platform for the support of survey management activities.

References

Cirasuolo, M., et al. 2014, in Ground-based and Airborne Instrumentation for Astronomy V, vol. 9147 of Proc. SPIE, 91470N

Fumana, M., et al. 2019, in ADASS XXVI, edited by M. Molinaro, K. Shortridge, & F. Pasian (San Francisco: ASP), vol. 521 of ASP Conf. Ser., 511

Garilli, B., et al. 2012, PASP, 124, 1232. 1210.4737

Grosböl, P., et al. 2012, in ADASS XXI, edited by P. Ballester, D. Egret, & N. P. F. Lorente, vol. 461 of ASP Conf. Ser., 619

Guzzo, L., et al. 2014, A&A, 566, A108. 1303.2623