AstroShelf: Understanding the Universe Through Scalable Navigation of a Galaxy of Annotations [1]

Panayiotis Neophytou, Roxana Gheorghiu, Rebecca Hachey, Timothy Luciani, Di Bao,
Alexandros Labrinidis, G. Elisabeta Marai and Panos K. Chrysanthis
Department of Computer Science, University of Pittsburgh
{panickos, roxana, reh59, tbl8, di, labrinid, marai, panos}@cs.pitt.edu

ABSTRACT

This demo presents AstroShelf, our on-going effort to enable astrophysicists to collaboratively investigate celestial objects using data originating from multiple sky surveys, hosted at different sites. The AstroShelf platform combines database and data stream, workflow and visualization technologies to provide a means for querying and displaying telescope images (in a Google Sky manner), visualizations of spectrum data, and for managing annotations. In addition to the user interface, AstroShelf supports a programmatic interface (available as a web service), which allows astrophysicists to incorporate functionality from AstroShelf in their own programs. A key feature is Live Annotations which is the detection and delivery of events or annotations to users in real-time, based on their profiles. We demonstrate the capabilities of AstroShelf through real end-user exploration scenarios (with participation from "stargazers" in the audience), in the presence of simulated annotation workloads executed through web services.

Categories and Subject Descriptors

H.2.8 [**DATABASE MANAGEMENT**]: Database Applications—*Scientific databases*; H.3.5 [**INFORMATION STORAGE AND RETRIEVAL**]: Online Information Services—*Data Sharing, Web-Based Services*

Keywords

Big Data, Scientific Databases, Continuous Workflows, Annotations, Visualization, Astronomy

1. INTRODUCTION

The growing onslaught of astronomical data available presents a great challenge. Astronomy lacks an easy-to-use and scalable way to collect and distribute expert information about objects from datasets of tens of thousands to billions of individual events and objects. Over the next decade, the amount of information available to the typical astronomer will grow by two orders of magnitude both in raw data size and in the number of objects.

We have been working with a group of astrophysicists to develop *AstroShelf* which will enable them to collaboratively annotate sky objects and phenomena, as well as visualize parts of the sky using different algorithms.

AstroShelf pursues two research directions, each of which has the potential to transform how astronomers interface with large datasets: (1) a *scalable annotation framework* to enable linking of observations to specific experiments, models, or other observations; and (2) a *continuous workflow enactment system* that would support automated reasoning in the presence of uncertainty.

In this demo we will demonstrate the design and usage of our AstroShelf prototype system, and show how it is being used in one timely scientific program: methods to incorporate images and catalogs from disparate datasets, using multi-wavelength data from two surveys, allowing astronomers to study how the demographics of galaxies have changed over the last 8 billion years.

This project enables astrophysicists to share information and expert opinions quickly and widely, about each new observation or event, fundamentally changing our ability to learn about the Universe; such functionality can also be applied in support of other scientific domains. A similar effort has been demonstrated in [1], where the system was displaying earthquake information. That system, though, did not address collaboration through annotations, nor did it provide a notification mechanism (such as the one in AstroShelf).

We will demonstrate the following features of AstroShelf:

- The AstroShelf's User Interface that supports: (1) search and visualization of sky objects, their data and meta-data, (2) annotations management, and (3) specification of interest and *Live Annotations*.
- The use of CONFLuEnCE [6] to support the monitoring of data streams which are either entered into the system as new celestial events (e.g., new supernova detected) or are created within the system as new annotations.

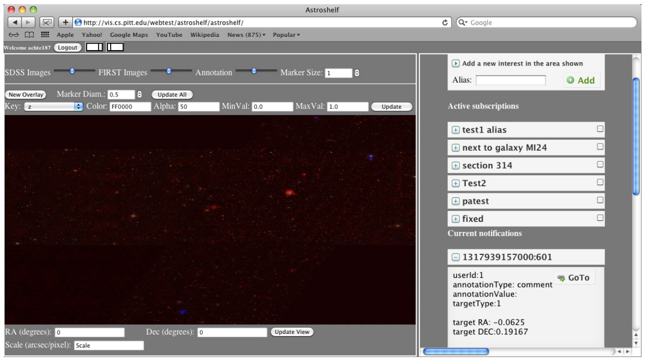
The current version of AstroShelf is being packaged to be used as an educational tool, allowing astronomy students to collaboratively explore the sky.

2. ASTROSHELF - USER VIEW

AstroShelf's User Interface (UI), depicted in Figure 1 provides the user with a *SkyView* component, which superimposes sky images from different surveys using multiple *overlays*. Meta-data regarding celestial objects are also displayed, as annotations, in other overlays. The UI also facilitates searching of objects and annotations on objects or areas, as well as the ability to add new an-

[1] This research was supported in part by NSF career awards IIS-0746696 and IIS-0952720 and NSF grant OIA-1028162.

SIGMOD '12, May 20–24, 2012, Scottsdale, Arizona, USA.
Copyright 2012 ACM 978-1-4503-1247-9/12/05 ...$10.00.



Welcome achte187

SDSS Images FIRST Images Annotation Marker Size: 1

Marker Diam.: 0.5

Key: z Color: FF0000 Alpha: 50 MinVal: 0.0 MaxVal: 1.0

RA (degrees): 0 Dec (degrees): 0

Scale (arcsec/pixel): Scale

Add a new interest in the area shown

Alias:

Active subscriptions

- test1 alias
- next to galaxy MI24
- section 314
- Test2
- patest
- fixed

Current notifications

1317939157000:601

userId:1
annotationType: comment
annotationValue:
targetType:1

target RA: -0.0625
target DEC:0.19167

Figure 1: Screenshot from AstroShelf's User Interface, currently showing the SkyView (left) and the Live Annotations View (right).

notations. As part of the visualization of meta-data the interface allows the user to generate *trend images* on objects retrieved using the search mechanism. Lastly, a *Live Annotations* module provides real-time updates on new annotations, or new transient events. We elaborate on these user interface components next.

2.1 Objects, Catalogs, and Annotations

The AstroShelf system provides a single dashboard to seamlessly access images and meta-data that are either stored locally (for performance reasons or because it is user-contributed content) or are stored remotely, originating from different servers, and are accessed via an API.

One of the main challenges in putting together the AstroShelf system had to do with the ability to properly integrate data from different surveys. This materializes in two dimensions. First, the ability to "link" meta-data (which we will refer to as *catalog data* from now on) for the same object, coming from different sky survey databases. The challenges are primarily domain-specific, in that there are multiple coordinate systems in use and also given the inherent differences of the surveys (i.e., different target wavelengths), the same object may or may not have the same "footprint" across different surveys. Second, the ability to "stitch" images together, given the above challenges.

We distinguish between four types of data in our system:

- catalog data, stored remotely
- annotation data, stored locally
- image data, stored remotely
- image data, stored locally (for performance reasons)

An annotation is defined as any piece of information that is added to the database in order to enhance and complement the existing information of celestial objects. In our system, any annotation is represented as a pair <*annotation type, annotation value*>. Because of that, our system supports any type of annotation one may want to add (e.g., a tag, a text comment, link to a file with experimental results, image, link to a paper, etc.) Annotations can be made on three different levels of granularity:

- Annotations on a *celestial object* or on a *Right Ascension (RA), Declination (Dec) coordinate*, using a unique identifier for that object (i.e., object name and survey). We are exploring the option to store part of catalog data locally from external sources, for improved performance.
- Annotations on a *set of objects* can be made by specifying a set of objects having some specific properties (based on a user defined query or view). Moreover, the system allows the definition of annotations on specific *(RA, Dec)* ranges. This can be seen as an annotation that is added on an area of the sky instead of particular objects, and any objects which may be added later in that area are also annotated with the area's annotation in a similar fashion as in view-based annotations [3].
- *Annotations on annotations*, can be made, for example, to represent revisions on annotation values or discussions among users.

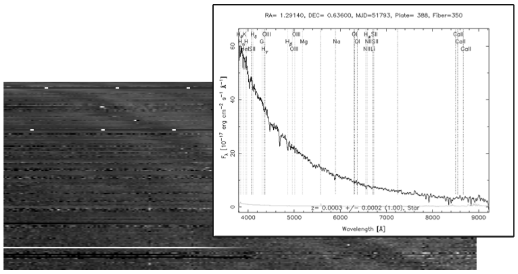


Figure 2: Trend Image for 139 Type Ia supernovae.

2.2 SkyView

The SkyView in our system is an interactive, 2D HTML5 canvas, implemented in webGL. Within this canvas the user can interact with data and analyze the output of other components of our platform. The images of the sky are provided by astronomical surveys such as SDSS (www.sdss.org) and FIRST (sundog.stsci.edu) which are displayed using different levels of opacity. The SkyView also allows zooming and panning. Finally, the SkyView uses pixel-based annotations to view data provided by our Annotations Management System as well as survey databases. Pixel-based annotations allow the user to visualize thousands of objects in a single image, possibly spotting trends in the data otherwise impossible to perceive (e.g., markers for heavily annotated objects appear brighter than those of lightly annotated objects). Since these annotations are pixel-sized, numerous queries can be viewed at once without hindering the overall view, having each query in a separate overlay.

2.3 Trend Images

Trend images are used to visualize overall patterns in a data set. A trend image consists of a set of pixels, each mapped to a specific data point in such a way that patterns in the data become apparent from patterns in the resulting image.

An example of how a trend image could be used is to visualize patterns in spectroscopic data from astronomy observations. Spectroscopic data involves measuring the flux or intensity of light at different wavelengths for objects of interest. The trend image in Figure 2 consists of spectroscopic data for 139 Type Ia supernovae from Balland et al. [2]. Each horizontal row corresponds to the spectra for a supernova, and they are ordered from top to bottom in increasing redshift. Along the x-axis the data points are ordered in increasing wavelength from left to right. The gray-scale pixel values correspond to the flux of the object at that wavelength. The general broad features of the Type Ia supernovae can be seen in the dark and light bands that represent characteristic features of explosions. These smoothly trace back over redshift indicating that these are a consistent class of an object.

AstroShelf supports generating a trend image for any number and type of objects specified by the user. In the current demo, the objects are from the specObj view of the seventh data release of SDSS but, in general, AstroShelf can support any catalog of celes-

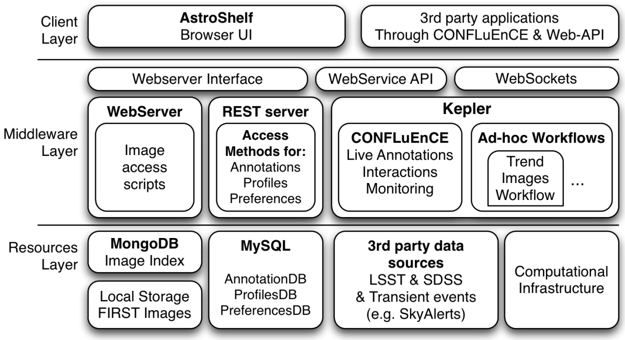


Figure 3: High-level design of the AstroShelf collaboration platform.

tial data. Closer inspection of the spectra of outliers is supported by the clickable image map which links a graph of the spectral data of a single object to the horizontal line of pixels it corresponds to.

2.4 Live Annotations View

The Live Annotations module of the user interface displays, in real-time, any new notifications regarding areas and types of objects which are of interest to the current user. The user may register multiple *interests* with the system and the updates will reach her by the preferred means (e.g., browser UI when logged in, email when not, notifications on mobile phone). The user may filter the notifications by interest, and then by clicking on them she will be navigated to the area of the sky where the annotation was made (through the SkyView).

The Live Annotations view not only displays the new annotations made in specific areas and specific objects, but it also provides the user with popular areas or objects which are relevant to the user's interests. This proves to be particularly useful when the number of notifications is either high, or the user is registered to areas of the sky where there is not too much activity, while in other areas (unbeknownst to the user), there has been a new discovery that may be of interest to the user.

2.5 Search

AstroShelf supports searching of catalogs of celestial objects or of annotations stored in our system. Users can browse and search the database using the UI, through scripts (using our API) or by creating a workflow. When the result of a search is too large, the system will filter the information using each user's preferences.

Users can express their preferences using the AstroShelf interface. Users are allowed to submit preferences on both annotations, and/or objects and meta-data associated with each object for a particular survey. When submitting preferences over annotations, users can define their own preferences over different characteristics of an annotation (e.g., annotation type, annotation value, annotations inserted by a particular user, annotations related to a particular object or to a particular survey). When defining preferences over objects, users are presented with a list of catalogs and for each catalog they can define the preferred attributes of that particular catalog.

In addition to defining a preference, the system allows uses to label each preference with an intensity value between 0 and 1 (1 for most preferred, 0 for *negative* preference). This weight expresses the intensity of interest in a particular annotation, object or metadata related to an object.

Data Source	Ad-hoc Queries	Continuous Queries
Catalog Data	REST API though 3rd party API	Events Notification
Annotations Data	REST API combined with preferences	Live Annotations

Table 1: Query Types

3. ASTROSHELF - SYSTEM PERSPECTIVE

In this section, we first describe the architecture design of the AstroShelf platform, as depicted in Figure 3, and then describe the most important components which will be demonstrated in this demo.

The top layer (Client Layer) is where the user interface is running, as well as third party applications or user scripts which can use our publicly available web service (REST) API. The AstroShelf UI interacts with the middleware layer using multiple channels depending on the purpose. It uses the web server interface to initialize and retrieve sky images. It connects through the web services interface for searching, inserting, and any other database related functionality related to Annotations or Catalog data as well as profile and preferences data. The AstroShelf UI connects through web sockets to the Kepler/CONFLuEnCE module for receiving Live Annotations, notifications, and other workflow-related functions. We found that web sockets provide the fastest and most efficient way to deliver real-time updates to users.

The Middleware Layer is responsible bridging and interfacing with all the components in the resources layer. The web server is used for loading the AstroShelf UI on the user's browser and also includes server side scripts for accessing the image files needed to render the SkyView. The image files are indexed using a Geohash index which is implemented inside MongoDB. We require a geospatial index in order to store the sky images' bounding box coordinates (using RA and Dec). The client layer will request images for the current bounding box of the viewport inside the SkyView. The index will return all the images that fall into that bounding box and then the server script will put them together and return them to the SkyView for rendering.

The REST server implements all the necessary access methods for remotely searching catalog data for celestial objects and inserting, searching, deleting, and modifying celestial annotations, users' profiles and preferences. We decided to use *GlassFish* as our REST server because it was designed to easily interface with relational databases and automatically generate the necessary Java classes for our schema. The REST access methods are used by the workflow engine, the user interface and 3rd party applications to manipulate and access the data. This provides for a uniform access method to the data thus guaranteeing consistency throughout the system. The REST server also provides search facilities for the 3rd party data sources which expose their own API. One of the challenges we faced in this task was to integrate the heterogeneous semistructured data stored in these 3rd party data sources and enable searching and annotating these data in a structured way.

Finally, the Kepler workflow engine is used to facilitate two types of functions. Firstly, monitoring functionality (like the one required for Live Annotations) is facilitated using CONFLuEnCE (Continuous Workflow Execution Engine)[6]. Kepler also provides ad-hoc workflow execution functionality for batch data processing, like calculating the trend images, or for enabling running simulations by astronomers.

The types of queries and the way they are handled by our system are summarized in Table 1. Below, we describe the most important individual components of the architecture.

3.1 Annotations Service

The Annotations Service is the system component which handles annotations on celestial objects, sky coordinates, or other annotations in varying granularity levels. The service also works as a notification service; it sends notifications of changes in annotations to registered entities. The service exposes a RESTful API which can be used by third parties to insert, query or update annotations, given the right credentials.

3.2 Personalization

The AstroShelf can support personalization in the form of explicit preferences, that the user marks and are being used to filter the results of a query. Preferences are being stored as general predicates on any attribute. For example, $redshift \in [0.1, 0.3]$ would indicate a user preference for objects with a redshift value between 0.1 and 0.3. When a query is submitted, the preference predicates are incorporated in the query and pushed down close to the data sources when the query is optimized.

3.3 CONFLuEnCE

CONFLuEnCE, short for CONtinuous workFLow ExeCution Engine, is the implementation of our Continuous Workflow (CWf) model [7, 5]. It was built on top of Kepler [4], and can facilitate reactive, monitoring, and collaborative applications from both the scientific and the business domain. CONFLuEnCE was a natural choice for AstroShelf because of its ability to apply window semantics on streams of events while employing the workflow model.

In the context of AstroShelf, we have implemented a continuous workflow to monitor the activity of inserting, updating or deleting annotations as well as integrating the detection of transient events from various sky surveys that are of interest to the users, all in real-time. The users express their interest in areas of the sky and/or keywords on annotations. They are then notified if an annotation or an event appears in an area they have expressed interest in and which contains the provided keywords (if any). The workflow keeps an R-Tree spatial index of all the subscriptions to efficiently match annotations to subscriptions, since the areas are a combination of (RA, Dec) pairs and the events and annotations are spatially tagged. Once notified of something, the users can then provide feedback in the form of annotations. By interacting with the workflow, the users may refine the annotations, iterating over them until they reach a consensus (e.g., as part of collaborative classification of transient events).

As part of the monitoring functionality, CONFLuEnCE groups the events and annotations by area and measures the temporal density of events per area (how many events per unit of time), and notifies users about popular areas in the sky (i.e., areas that are of high interest to users at the time, since a lot of annotations are happening, or a lot of transient events are appearing.)

4. DEMONSTRATION

In our demonstration we would like our audience to interact with the AstroShelf system as users of our platform. They will be able to use multiple displays to enter new annotations, express interest in areas in the sky and then get notified by various ways (e.g., on screen, email etc.) when new events of their interest are detected. In order to make it interactive we assume all annotations are public and every user can see any annotation. In this way, a user can define an area of interest and any annotation added to that area will be pushed, as a notification, to their account.

Another aspect in our project is the visualization tool. One can see the image of any portion of the sky, zoom in for a closer look of celestial objects and surrounding area and make annotations. In the visualization environment users can play with different ways of sky visualization. They can see only the redshift, (e.g., when light seen coming from an object that is moving away is proportionally increased in wavelength, or shifted to the red end of the spectrum) only the image of the sky as found in different astronomy surveys or both of them combined. Moreover, the interface allows users to see a mark on the sky next to objects that have been annotated already. Each mark has a different color intensity depending on how many annotations are defined for that particular patch of sky.

Finally, we will demonstrate to our audience how the AstroShelf system can be used through an API. By eliminating the need of a user interface, we allow our users to write and retrieve annotations in a fast way. We will show how easy it is to submit multiple annotations in the same time and how annotations can be fetched from our system. This will be demonstrated using automated scripts, which will also help in demonstrating the notifications functionality with sufficient annotation traffic generated in the background.

In the end, we would like to demonstrate that our system addresses the following problems:

- *Annotations*: We have two different ways of adding annotations (i.e., using the visualization tool or programmatically, using the annotation service). In this way, we address two different problems: a) an interface that may become a bottleneck in the process of adding annotations by hand and b) a system that can become too technical for someone who only wants to visualize the sky and add various notes on things he/she discovers.
- *Continuous Workflows*: Our system provides an interactive way of staying connected with the community through a continuous workflow execution engine. Annotations relevant to users preferences are pushed continuously and can be accessed on different platforms.
- *Visualization*: AstroShelf visualization system is a great way to look at sky images and celestial object data. Using the various layers the AstroShelf system offers (i.e., SkyView, Redshift slider, Trend images) astrophysicists can interact directly with the sky images and discover interesting areas or objects on the sky faster.

Acknowledgments: We thank Boyu Sun from the ADMT Lab and our astronomy collaborators: Brian Cherinca, Arthur Kosowsky, Jeffrey Newman, Anja Weyant, and Michael Wood-Vasey.

5. REFERENCES

[1] M. Ali, B. Chandramouli, J. Fay, C. Wong, S. Drucker, and B. S. Raman. Online visualization of geospatial stream data using the worldwide telescope. In *PVLDB*, vol. 4, pp. 1379–1382, 2011.

[2] C. Balland et al. The ESO/VLT 3rd year Type Ia supernova data set from the supernova legacy survey. In *A&A*, vol. 507, pp. 85–103, 2009.

[3] Q. Li, A. Labrinidis, and P. K. Chrysanthis. Vip: A user-centric view-based annotation framework for scientific data. In *Proceedings of SSDBM*, pp. 295–312, 2008.

[4] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065, 2006.

[5] P. Neophytou, P. K. Chrysanthis, and A. Labrinidis. Towards continuous workflow enactment systems. In *Proceedings of CollaborateCom*, pp. 162–178, 2008.

[6] P. Neophytou, P. K. Chrysanthis, and A. Labrinidis. Confluence: Continuous workflow execution engine. In *Proceedings of SIGMOD*, pp. 1311–1314, 2011.

[7] P. Neophytou, P. K. Chrysanthis, and A. Labrinidis. Confluence: Implementation and application design. In *Proceedings of CollaborateCom*, pp. 181–190, 2011.