To link to this article: DOI:10.1016/j.ascom.2017.05.002
URL:http://dx.doi.org/10.1016/j.ascom.2017.05.002

To cite this version:

Louge, Thierry and Karray, Mohamed Hedi and Archimède, Bernard and Knödlesder, Jurgen *CASAS: A tool for composing automatically and semantically astrophysical services.* (2017) Astronomy and Computing, vol. 20. pp. 34-43. ISSN 2213-1337

CASAS: A tool for composing automatically and semantically astrophysical services

T. Louge [a,*], M.H. Karray [b], B. Archimède [b], J. Knödlseder [c]

[a] Université de Toulouse - UPS, Institut de Recherche en Astrophysique et Planétologie, 57 Avenue d'Azereix, 65000 Tarbes, France
[b] Université de Toulouse, Ecole Nationale d'Ingénieurs de Tarbes ENIT, 47 avenue d'Azereix, BP1629, 65016 Tarbes Cedex, France
[c] Université de Toulouse - UPS, Institut de Recherche en Astrophysique et Planétologie, 9, avenue du Colonel Roche BP 44346 - 31028 Toulouse Cedex 4, France

Keywords:
Services composition
Astrophysics
Semantic web
Ontologies

ABSTRACT

Multiple astronomical datasets are available through internet and the astrophysical Distributed Computing Infrastructure (DCI) called Virtual Observatory (VO). Some scientific workflow technologies exist for retrieving and combining data from those sources. However selection of relevant services, automation of the workflows composition and the lack of user-friendly platforms remain a concern. This paper presents CASAS, a tool for semantic web services composition in astrophysics. This tool proposes automatic composition of astrophysical web services and brings a semantics-based, automatic composition of workflows. It widens the services choice and eases the use of heterogeneous services. Semantic web services composition relies on ontologies for elaborating the services composition; this work is based on Astrophysical Services ONtology (ASON). ASON had its structure mostly inherited from the VO services capacities. Nevertheless, our approach is not limited to the VO and brings VO plus non-VO services together without the need for premade recipes. CASAS is available for use through a simple web interface.

1. Introduction

The use of modern astrophysical instruments creates the production of massive datasets. Modern astrophysical instruments are conceived from the ground up integrating science gateways in their design. Those science gateways (SGWs) are composed of data, processing pipelines and applications dedicated to the instrument. They consist of software packages designed for the specificities of the said instrument. Those packages generally include web and desktop environment for online and offline use. They are composed of scientific libraries, grid and cloud computing, as well as workflow managers.

Use of astrophysical instruments produces data that are made public, either after scientific publication or after the proprietary period has passed. The VO has been designed with the goal of allowing the access and enhancing the interoperability of this huge amount of data. The VO is an astronomy-specific Distributed Computing Infrastructure (DCI) offering standards for services query, data access and data models. Around this DCI, different applications (called "VO-compliant" applications) have been developed either for specific needs like the CASSIS (Centre d'Analyse Scientifique de Spectres Instrumentaux et Synthétiques)[1] for spectral analysis, or more generic like Aladin[2] (Boch et al., 2011) that is probably the most well-known VO-compliant tool.

Despite the VO efforts, bringing together a number of different tasks in order to produce a scientific result may involve different tools and services. Those steps may not be easy to assemble, share and modify outside of a dedicated workflow manager. To overcome this problem, Ruiz et al. (2014) propose AstroTaverna, which is a plugin for Taverna workflow manager that brings VO services.

Meanwhile, many web services are available for astrophysics, both inside and outside the Virtual Observatory architecture and specific instruments pipelines. Exploring the capacities of services (the information they may provide and the input parameters they need to be used) regardless of their specific DCI or outside from any DCI remains a concern.

Web services discovery and composition is at the core of many research work in computer science addressing exactly this kind of concerns. A recent survey (Lemos et al., 2016) establishes a framework for the description and comparison of the different approaches, based on the analysis of 12 platforms. Among the 12 platforms analyzed, only one platform (Taverna) was addressing scientific workflows, while 10 out of 12 were dealing with business

---

* Corresponding author.
E-mail address: tlouge@irap.omp.eu (T. Louge).

[1] http://cassis.irap.omp.eu/?page=cassis.
[2] http://cdsads.u-strasbg.fr/abs/2011ascl.soft12019C.

processes and 1 with mashups (Yahoo! Pipes) but targeting end-users programmers audience rather than domain experts.

While Taverna addresses scientific workflows, the composition itself is not automatic and needs technical background. It also needs to be aware of the content available inside the thousands and thousands of services available. Even after services discovery process, dozens of services may be candidates for providing the same output. In that sense, Taverna (and as a result, this goes for AstroTaverna) is oriented towards end-user programmers (Lemos et al., 2016), even if the composed workflows are available online through recipes reusable by domain experts without technical background.

In order to propose an automatic composition of workflows for end-users without technical background knowledge, this work presents a tool called CASAS. This tool aims at providing a semantic composition of web services independent from their original DCI, specific language, protocol or format. It provides this composition on-the-fly, from domain-expert description without the need to conform to a specific description format. This composition needs no action from the user; the selection of relevant services among different candidates is made by analyzing the previous workflows results. This behavior is modifiable by the user, leading to another services election.

Related works on services selection and workflow composition in astrophysics, and the motivations behind the design of CASAS will be exposed in Section 2. Section 3 will present the design of the application and Section 4 will give an overview of its use. Section 5 will discuss the results obtained and the limitations of our approach. Conclusions and future works will be raised in Section 6.

2. Related works

2.1. Web services selection in astrophysics

The widest source of web services for obtaining astrophysical information is the Virtual Observatory. The VO is unified under an association, International Virtual Observatory Alliance (IVOA)[3] that defines standards and protocols for describing and sharing astrophysical data. While IVOA is the central actor, the VO is not monolithic and several initiatives coexist addressing the same goal: Providing a unified, efficient and simple way to access data from various sources. Among those initiatives, one can find software development for specific tasks or scientific use-cases, federation of instruments accessible under a common interface like USVAO Data Discovery Tool,[4] or IVOA standards adaptation to specific astrophysical fields like astrochemistry in VAMDC (Dubernet et al., 2016).

IVOA offers the use of registries (Schaaff, 2007) to select relevant web services for a given request. Those registries may either be harvested by keywords or by a more advanced search taking several elements into account (services descriptions, subjects etc.). When queried by the use of keywords, every keyword entered is to be found alone or in combination with the other (AND/OR toggle in the interface). Results from the registries include a brief description of the services with their URL. An XML file is also returned, containing all the information available through the use of the said services. The following steps consisting of deciding whether or not a service is suitable among every service returned and querying the selected services are up to the user.

Another way to find astrophysical web services inside the VO is using VO applications, such as Aladin and Topcat Taylor (2011). Those applications offer a simple way to find services. Aladin offers to search VO content for images, spectra or catalogs based on the coordinates or the name of an object. Aladin displays only a short description of the services, and determining which service is the more suitable for a specific use-case is up to the user.

Topcat uses a keyword system similar to the one used in registries. Topcat allows browsing different registries, accessing services content and displaying the results. However, the queries to the services have to be done one by one until a service provides useful data.

2.2. Workflow composition in astrophysics

Workflow composition in astrophysics is mainly used inside the instruments science gateways. As an example, the INAF (Instituto Nazionale Di Astrofisica) Cherenkov Telescope Array (CTA) Science Gateway (Costa et al., 2015) designed for the upcoming CTA instrument uses WS-PGRADE/gUse framework for its workflow management. Two main components are found in this framework (Kacsuk et al., 2012): gUse handles the access to multiple DCIs and WS-PGRADE is the workflow manager. WS-PGRADE offers a user interface and embeds a workflow engine called "Zen". WS-PGRADE uses its own XML-based language for workflows descriptions, that are directed, acyclic graphs.

In their work for AstroTaverna, Ruiz et al. (2014) list the most important approaches for workflow management in astrophysics outside the scope of SGWs. Their conclusion is still valid, that those approaches solve very specific scientific cases with specialized data reduction. The "Virtual Observatory oriented SPECtrum workflow" (VOSPECFLOW) (Lèbre et al., 2012) application is an example of using the VO protocols and formats for a given scientific use-case embedded in a specific software. Databases are queried through VO protocols and data retrieved are described using VO formats. However, the specificities of the spectra used in this workflow prevented the use of any pre-existing VO-compliant software. AstroTaverna brings the astrophysical services composition into Taverna, thus offering a reliable, reusable and modifiable services composition embedded inside dedicated software.

AstroTaverna has been reused inside a more generic workflow manager in a work dedicated to bring VO-compliant workflows inside SGWs (Castelli et al., 2015). This work embeds AstroTaverna workflows inside a global manager using the "SHaring Interoperable Workflows for large-scale scientific simulations on Available DCIs" (SHIWA)[5] Simulation Platform (SSP). SHIWA is based on WS-PGRADE/gUse framework and allows frameworks from different languages and managers to be incorporated as modules.

2.3. Motivations

The Virtual Observatory is not the only way to find astrophysical data, and several databases outside the VO can be found on the web offering specific query interfaces and content like HyperLeda (Makarov et al., 2014) for galaxies. There is no standardized way of finding resources outside of the VO in astrophysics. Besides, integrating such services in existing workflows managers needs consequent technical background. Interfaces need to be designed, and automation of the queries ensured.

Besides, the counterpart of VO registries for VO applications is at its very beginnings.[6] As a consequence, VO-compliant software like Aladin offers possibilities of analysis, display and data manipulation that are not registered in the VO DCI. VOSPECFLOW (Lèbre et al., 2012) is another good example of scientific application that is not described inside the VO.

---

[3] http://ivoa.net/.

[4] http://usvao.org/index.html%3Fpage_id=344.html.

[5] http://www.shiwa-workflow.eu.

[6] http://voar.jmmc.fr/index.html.

Furthermore, existing workflows approaches in astrophysics rely on recipes. AstroTaverna shares its recipes through the MyExperiment website,[7] where they can be downloaded. Modifying those recipes or creating new recipes from scratch needs both technical background and a-priori knowledge on the services available. Even if AstroTaverna allows searching for services, those services need to be selected or rejected by the end-user. This is not an easy task, because there are more than 10 000 services registered in the VO only for ConeSearch protocol. Other protocols will offer supplementary services, and many dozens of services may fulfill the user requirements. Moreover, the content of services is tied to astrophysical objects. There is no limit to the number of services that may provide given information, but only a few of them may provide it for a given object. Thus, the user is constrained, while elaborating a workflow either to fulfill a very precise scientific case (for example, only dealing with specific objects) or finding as many services as possible that may generalize the use of the said workflow. Elaborating a recipe with a workflow manager or modifying an existing recipe is not an easy task.

The work exposed in this paper aims at automating the composition of astrophysical workflows. This composition should integrate any service described inside ASON, regardless of the nature of the said service (data provider or analytical service) and original DCI. The goal of the composition should be expressed in natural language, so that the user does not have to conform to predefined keywords. The composition should not require any technical skill. Lastly, CASAS should propose alternative composition of services for the cases where services do not contain the required information for a given context.

3. The CASAS design

CASAS explores the solutions that semantic web composition of services can bring concerning those motivations. The user of CASAS only needs to express what result the system may provide. The workflows generated by the system incorporate VO and non VO data services. The services are elected following criteria taking into account the history of the previous generated workflows. The weight of this history into the workflow generation is controlled by the user, may be modified and alternative workflows generated. Every workflow generation is done automatically, and no specific keywords are necessary (details are provided in Section 3.2).

CASAS is designed to be a complementary tool for existing workflows managers. This complementarity comes from two main features:

- The fact that CASAS discovery of services removes the need for predefined keywords. This allows a more efficient matching of user needs with actual services content.
- The CASAS ability to automatically evaluate more services during the composition than a user may do on his own.

As a consequence, CASAS compositions may serve as a basis for other workflows managers involving more specialized services or analysis whose description is not available in ASON.

CASAS is also a testing semantic infrastructure for bringing VO services, non-VO services and analytical tools altogether.

3.1. Overall architecture

Following the principles of the semantic web, CASAS transfers the knowledge of the services capacities from the user to ontology. Choice of relevant services is based on generic user criteria defined on the web interface. The composed workflows are graphically displayed on this interface, and results come with a description of the services used.

The architecture of CASAS follows a preliminary design exposed in Louge et al. (2015), centered on ontology for astrophysical services. The generated workflows come from automatic reasoning using semantic representation of knowledge based on ASON (Louge et al., 2017). Fig. 1 describes CASAS final design that is exposed hereafter. In the following description, the term "service" will be used without distinction for data providers services and analytical services, registered in the VO, any other DCI or outside of any DCI.

The first component of the architecture is not accessible for the end-user. It is used for integrating new services inside the ontology, and is called the feeding layer. The feeding layer consists of procedures and methods allowing the description of a service inside ASON. Registering a service in ASON requires an XML file and a piece of software code. The XML file contains a general description of the service, a description of every output and input information and the units associated, if any. Every description is expressed in natural language, and automatically associated with the more relevant element inside ASON. The last two information in this XML file are either:

- the URL of the service and the protocol, if any existing protocol inside ASON fits the service (e.g. a new IVOA SSAP service)
- or the name of the software code that takes the inputs and produces the outputs for the service.

Services registered in ASON and consequently available in CASAS have been registered through automatic use of the feeding layer. Their descriptions mainly come from the automatic query of VO-Paris registry.[8] Non-VO services in CASAS have been registered through manual use of the feeding layer, by generating an XML file and passing it to the feeding layer interface.

The listing 1 exposes an example of such an XML file tailored for the feeding layer.

Every tag from this file represents some metadata underlying the semantic description of the service. The metadata specifically describing services capacities are the content of "<DESCRIPTION>" tags. Every element from those tags is mapped with ASON content to find the best individual or concept in the ontology for expressing its semantics. The mapping method itself is non trivial, and beyond the scope of this paper. For a better understanding, we can precise that it relies on syntactic and semantic matching between ASON annotations and the content of XML tags. The ontology ASON has been designed by using every description found in services from VOParis registry. The taxonomic structure and the annotations of concepts and individuals in the ontology have been extracted from those descriptions. Every VO service may be expressed in an XML file similar to the one exposed in listing 1, that is quite close to the XML definition found in registries. Consequently, more than 11130 VO services are available inside CASAS for workflow composition.

Finding a good services' composition (a composition that matches the users' requirements) is the role the reasoning layer. This layer is in charge of selecting the relevant services to produce the outputs of the workflow. It uses the information given as input by the user and ensures every services interoperability in-between. Every useful service is selected in a first run of the composition algorithm, and among those services only one per information is elected. This step of electing a service among many candidates takes history of the previous composed workflows into account, as described in 3.3. The reasoning layer uses a Knowledge Based (KB)

---

[7] http://www.myexperiment.org.

[8] http://voparis-srv.obspm.fr/portal/vo.php.

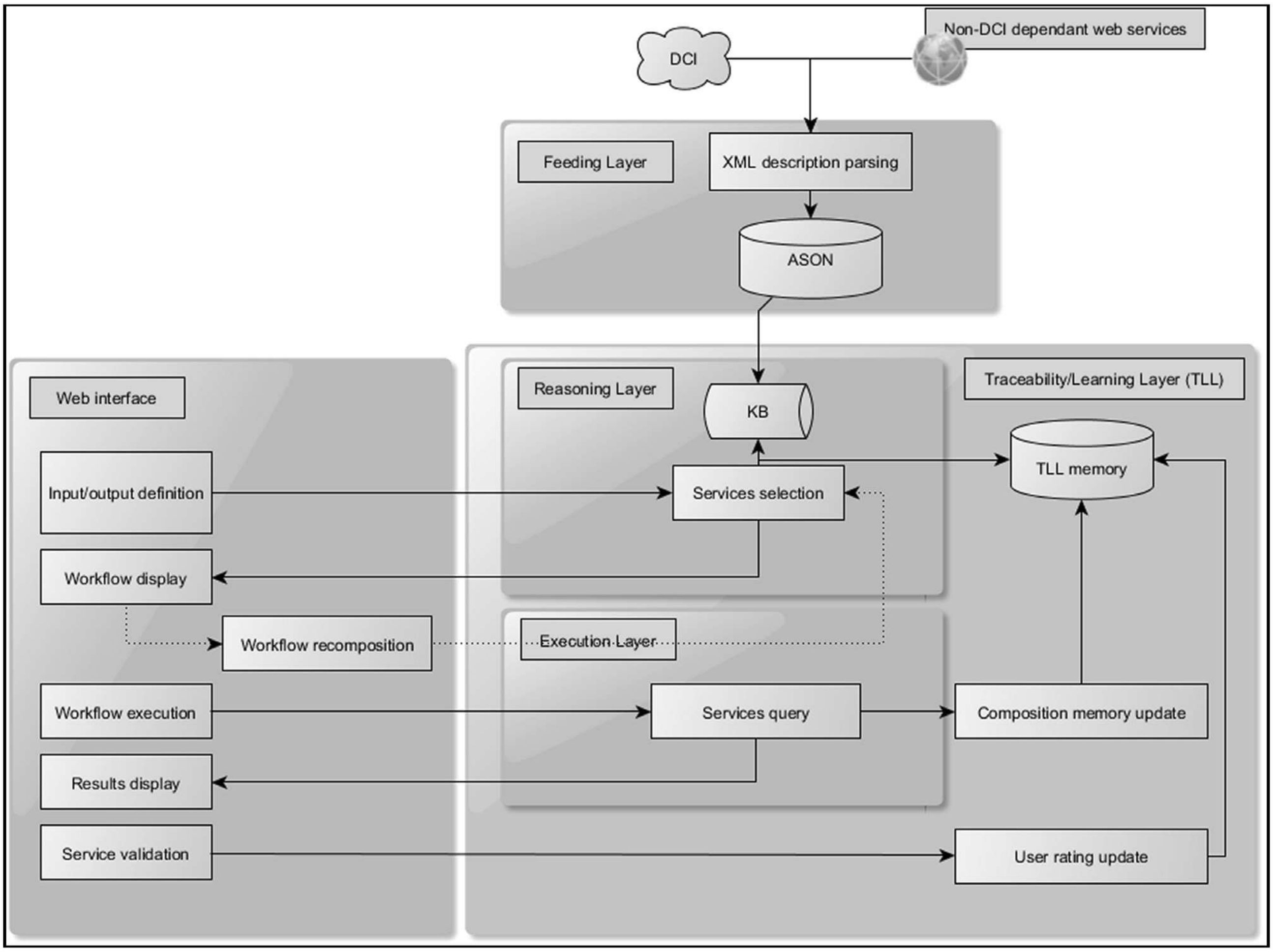


Fig. 1. CASAS architecture.

copied from ASON without embedding human-related resources like labels and descriptions.

Once the workflow has been composed, it is necessary to run the services, which comprises the orchestration phase (determining the order in which services may be run) and the execution phase (running the services) (Puttonen et al., 2013) . The execution layer is in charge of those phases. Taking the input information provided by the user, it runs the corresponding services, producing the necessary information for the following step and goes on until all of the outputs are produced. The execution layer is very content-sensitive, when it comes to the actual query of services. In most cases, a given service may be registered as a provider for a given information, but will not contain the given information for the current context. As an example, the service "j_apj_693_1084" is registered in ASON as a provider for "radial velocity" measurement, but does not contain this information for the specific galaxy "MRK1224". These "execution holes" are prevented by the traceability/learning layer, which stores the combination of services that worked well together, and the contexts where a service queries were successful. Besides this layer, by using the web interface the user may:

- Redefine the weights given to a specific history aspect.
- Run every service available, so that every solution provided by CASAS is explored ("Deep Workflow").

The traceability/learning layer (TLL) is the last component of CASAS. It provides the reasoning layer with information about the previous results associated with a given service. When the composition algorithm encounters a service, it queries the TLL about:

- Whether or not this service has been previously used in the same context (with the same input values).
- The number of services candidates in the current workflow that have been previously used in conjunction with the said service in previous workflows that provided information.
- The quality of the service for the information requested (the "specialization" parameter in the interface).

The TLL stores:

- The input parameters of a service when this service provided the information requested for a previous query.
- The composition of the workflows that successfully provided the information they were designed to provide.
- The user rating of a service for a given output information.

3.2. Services selection

The services selection in CASAS is based on advanced syntactic matching between a user description of outputs and the semantic content available in CASAS. This method resolves the problems of misunderstanding keywords ("color index" and "color index" will match together), but this is not its main benefit.

The main advantage of this syntactic matching is to be able to describe the outputs of the workflow without changing the user

```
<SERVICE NAME>j_aj_133_1528 </SERVICE NAME>
<SERVICE DESCRIPTION>Our K-band observations of the reflection nebulae were conducted with QUIRC, the 1024x1024 HgCdTe facility
near-IR camera on the University of Hawaii 2.2m telescope on Mauna Kea. All observations were made through the Mauna Kea Observatories
(MKO) Data were taken from 2003 January to September.IVOA ID = ivo://CDS.VizieR/J/AJ/133/1528</SERVICE DESCRIPTION>
<PROTOCOL>cs:ConeSearch</PROTOCOL>
<ACCESSURL> http://vizier.u-strasbg.fr/viz-bin/votable/-A?-out.all+-source=J%2FAJ%2F133%2F1528%2Fzeropt+</ACCESSURL>
<DESCRIPTION> Record number assigned by the VizieR team. Should Not be used for identification.</DESCRIPTION><UCD>
meta.record</UCD><UNIT>UNITLESS></UNIT>
<DESCRIPTION> Target name (HHMMm+DDMM)</DESCRIPTION><UCD> meta.id;meta.main</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION>  Zero-point K magnitude</DESCRIPTION><UCD> phot.mag;em.ir.k</UCD><UNIT>mag</UNIT>
<DESCRIPTION>  Right ascension (J2000) </DESCRIPTION><UCD> pos.eq.ra;meta.main</UCD><UNIT>h_m_s</UNIT>
<DESCRIPTION>  Declination (J2000) </DESCRIPTION><UCD> pos.eq.dec;meta.main</UCD><UNIT>d_m_s</UNIT>
<DESCRIPTION> Note</DESCRIPTION><UCD> meta.note</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> Record number assigned by the VizieR team. Should Not be used for identification.</DESCRIPTION><UCD>
meta.record</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> IRAS identification (HHMMm+DDMMA) \linkRole{K-band photometry calibration }</DESCRIPTION><UCD>
meta.id;meta.main</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> Associations identification(s)</DESCRIPTION><UCD> meta.id.assoc</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION>  Estimated distance</DESCRIPTION><UCD> pos.distance;pos.heliocentric</UCD><UNIT>pc</UNIT>
<DESCRIPTION>  Reference for Dist </DESCRIPTION><UCD> meta.ref;pos.frame</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION>  Bolometric luminosity</DESCRIPTION><UCD> phys.luminosity</UCD><UNIT>solLum</UNIT>
<DESCRIPTION>  2MASS J band magnitude </DESCRIPTION><UCD> phot.mag;em.ir.j</UCD><UNIT>mag</UNIT>
<DESCRIPTION>  2MASS H band magnitude </DESCRIPTION><UCD> phot.mag;em.ir.h</UCD><UNIT>mag</UNIT>
<DESCRIPTION>  2MASS Ks band magnitude </DESCRIPTION><UCD> phot.mag;em.ir.k</UCD><UNIT>mag</UNIT>
<DESCRIPTION>  Source spectral index</DESCRIPTION><UCD> spect.index</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> Details from 2MASS point source catalog (II/246)</DESCRIPTION><UCD>
meta.ref.url</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION>  Square root of the area within the K=19mag/arcsec^2^ surface brightness contour</DESCRIPTION><UCD>
phys.angsize;src</UCD><UNIT>arcsec</UNIT>
<DESCRIPTION> [*] stellar image dominates </DESCRIPTION><UCD> meta.code</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> Download the FITS file</DESCRIPTION><UCD> meta.ref.url</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> Name of the FITS image file in fig subdirectory</DESCRIPTION><UCD>
meta.id;meta.fits</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> Name of the FITS second image file in fig subdirectory</DESCRIPTION><UCD>
meta.id;meta.fits</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION> Designation understandable by the {\bf\fg{FireBrick}Simbad} data-base</DESCRIPTION><UCD>
meta.id</UCD><UNIT>UNITLESS</UNIT>
<DESCRIPTION>  2MASS Hour of Right Ascension (J2000)</DESCRIPTION><UCD>
pos.eq.ra;meta.main</UCD><UNIT>h_m_s</UNIT>
<DESCRIPTION>  2MASS Degree of Declination (J2000)</DESCRIPTION><UCD> pos.eq.dec;meta.main</UCD><UNIT>d_m_s</UNIT>
```

Listing 1: XML description of IVOA service

usual behavior. Any level of detail is acceptable, and here is an example of this: "heliocentric radial velocity" will give a more generic selection of services than "heliocentric radial velocity in optics", that will itself lead to the same composition than "heliocentric radial velocity, optical measurement". All the three descriptions will be identified in CASAS and will guide the composition algorithm to the appropriate level of detail.

3.3. Workflow composition

As a modular services ontology containing an astrophysical-specific module, ASON provides us with the ability to access astrophysical web services through a description that is independent from the services original DCI. Inputs, outputs and grounding (the actual way to access a given service) are described in ASON through the two modules that compose the ontology. Fig. 2 exposes the ASON generic architecture, while Fig. 3 shows a concrete example of its use Louge et al. (2017). CASAS resolves all the constraints in one single run, so that every service available for any output requested is elected in a single call to the composition algorithm.

Once a workflow has been composed, it is likely to contain dozens of services for some very common information. The composition algorithm calculates weights for each service regarding each information the given service may provide. Those weights are not set up once and for all, they rather depend on the current context on which the service may be used. They are calculated using the information provided by the TLL described in 3.1, and the best service after calculation is the one elected for the given information in the "quick workflow".

Those weights are divided in three categories, which are:

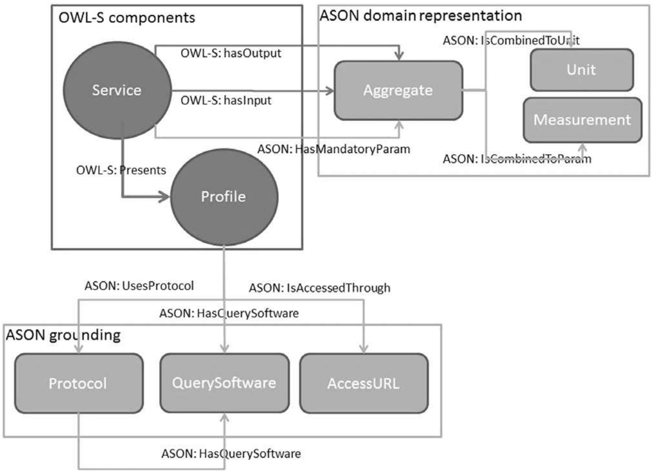


Fig. 2. ASON architecture.

- (1) The services precedent answers with the same inputs.
- (2) The service precedent use in conjunction with other services in a successful workflow.
- (3) The service specialization (user rating).

Two main aspects must be taken into account concerning the workflow composition, that are the possible bias and the specific problem of multiple names for astrophysical objects.

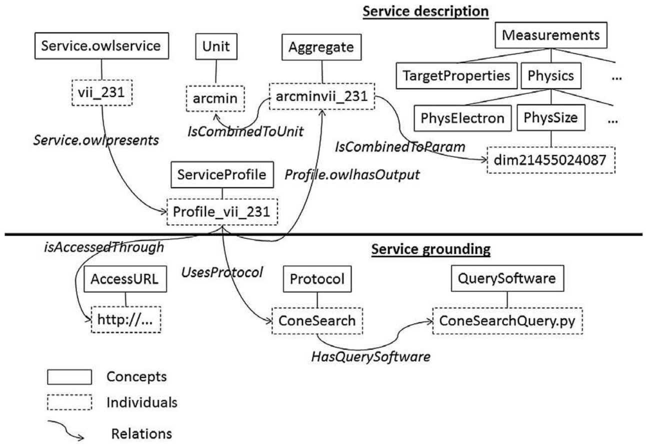


Fig. 3. ASON concrete implementation.

Concerning the first aspect, the automatic selection of services by the weight-based method proposed in CASAS may cause bias, such as very large services with many capacities participating in more workflows than smaller, more specialized ones. This could cause weights (1) and (2) to grow, and lead to large services dominance towards more specialized services that may better match an outlier user case. That kind of bias is countered by the weight (3), that enhances selection of smaller, more specialized catalogs towards large, generic catalogs. The importance of each weight may so vary from case to case, and it is possible to use the interface to adjust these parameters to the current case. This will lead to the composition of another workflow. Therefore, if the user needs to give the priority to the weight expressing the specialization of services, this can be done by increasing the role of weight (3) and decreasing the role of weights (1) and (2) in the composition. Besides, if the user wants to test every combination of services possible, this may be done by using the "Deep workflow". This "Deep workflow" queries every service available, so that it does not suffer from any bias deriving from the history of previous compositions. Different bias may occur in a human-related research through the VO and through non-VO astrophysical resources. The dominance of very large, well-known and well-described services on smaller services has been mentioned. Another kind of bias is the user knowledge that will lead to query more frequently services that the user knows a-priori, rather than services discovered after analysis of the wide range of services available. This bias is excluded from CASAS, that composes services by reasoning on every service registered in the system. Nevertheless, there may be biases that CASAS does not address, or even biases that are caused by the automatic reasoning underlying the approach.

The second aspect is that in astrophysics, an object can have multiple names or identifiers. CASAS relies on the services themselves for disambiguation of target names; that its algorithms do not provide. This problem of multiple target names is usually resolved by the services themselves, like SOPHIE database[9] or Polarbase.[10] VO protocols like ConeSearch or SSAP can be used through coordinates of objects instead of their identifiers.

4. Using CASAS

CASAS is available online[11] and its interface provides some examples of use. Several services that do not provide VO description or VO protocol have been integrated to illustrate the capacities of the approach. CASAS may use Aladin[12] for field chart generation and AladinLite for interactive sky views. Some non-VO services are available (e.g. for "magnetic field detection"). However, the very large majority of the services integrated in CASAS come from the automatic filling exposed in III.1 with VO services.

4.1. *CASAS web interface*

Using CASAS is as simple as possible: the user indicates what input he may provide and what output information he wants to obtain. In this first version, the only available input information is the name of a single object and a radius around this object. Once those input have been defined, the button "Seek services" will run the workflow composition.

Fig. 4 illustrates the input and output parameters definition of CASAS, defined from use of "Galactic example 2".

The composed workflow will appear beyond the "Seek services" button. The displayed workflow is by default the "quick workflow" composed using the TLL for electing the best services available for the current workflow. The blue squares indicate services elected in the workflow, the blue rounds information used by the services. The information with no entering edges is user-given and the green rounds indicate output information of the workflow fulfilling the outputs asked by the user. Two options, "Draw quick workflow" and "Draw deep workflow" will toggle the display between the two workflows. An example of generated workflow is presented in Fig. 5. The next, optional step consists in redefining the importance of the criteria listed in III.3 and recomposing the quick workflow accordingly. The quick workflow or the deep workflow may then be run, and the results will be displayed in tables beyond the workflow display window. For each service queried containing a relevant information for the workflow, the description of the service comes as a tooltip.

Some important information for the TLL is gathered during the run of the workflow. The first corresponds to the first category of weights in Section 3.3, when a service successfully returns an information for its input parameters it is registered into the TLL. Whether this information was of good quality or not comes later in the process. When a workflow ends, all the services successfully queried (meaning having returned a usable information) during the run are registered together into the TLL. As the success of a query greatly depends on the content of services, keeping this track makes sense. It means that in a given context, some services have been successfully used together. This corresponds to the second category of weights in Section 3.3. The button "validate this service" gives the TLL another feedback. This is the user feedback that the associated service, with its associated inputs has given useful information for the current workflow. This feedback will be used by the composition algorithm for the future compositions and corresponds to the third category of weights in Section 3.3. An example of results is exposed in Fig. 6.

4.2. *The provided examples*

CASAS web interface comes with examples of outputs and inputs for workflow composition. None of those examples provided in the top of the query interface is hardcoded or pre-tailored. They have been put together to show what possibilities the semantic web services composition approach may bring in astrophysics workflow composition. Every example is only a pre-determined filling of the interface fields. They are fully modifiable and only present a first sandbox set of possible queries.

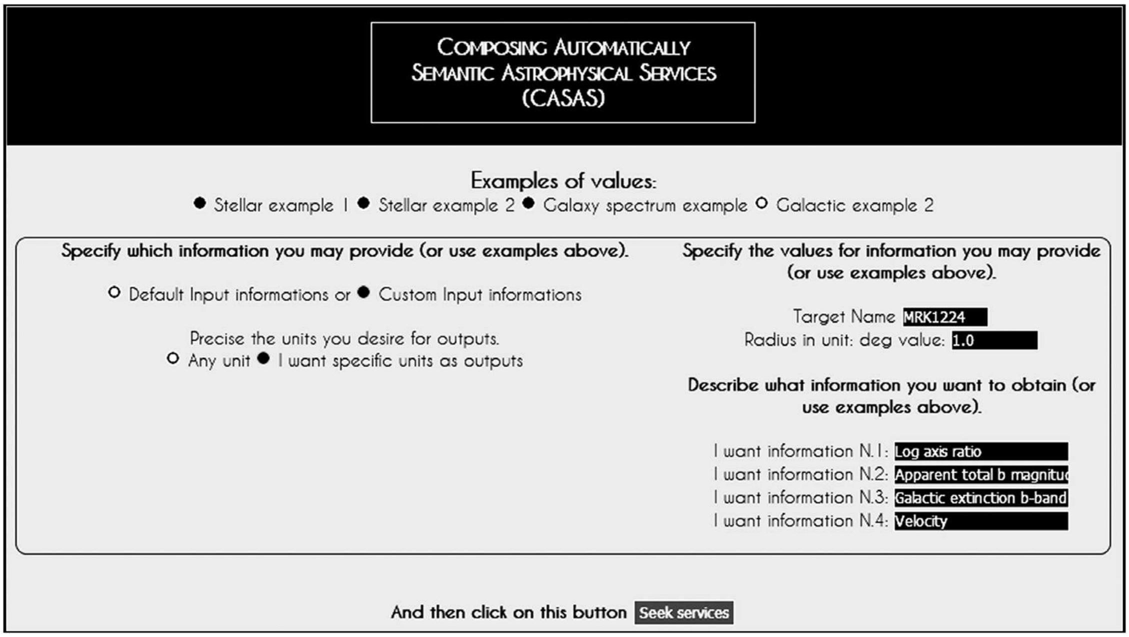
---

[9] http://atlas.obs-hp.fr/sophie/.

[10] http://polarbase.irap.omp.eu.

[11] http://cta1.bagn.obs-mip.fr.

[12] http://aladin.u-strasbg.fr/java/FAQ.htx#ToC158.


COMPOSING AUTOMATICALLY SEMANTIC ASTROPHYSICAL SERVICES (CASAS)

Examples of values:
● Stellar example 1 ● Stellar example 2 ● Galaxy spectrum example ○ Galactic example 2

Specify which information you may provide (or use examples above).
○ Default Input informations or ● Custom Input informations

Precise the units you desire for outputs.
○ Any unit ● I want specific units as outputs

Specify the values for information you may provide (or use examples above).

Target Name: MRK1224
Radius in unit: deg value: 1.0

Describe what information you want to obtain (or use examples above).

I want information N.1: Log axis ratio
I want information N.2: Apparent total b magnitud
I want information N.3: Galactic extinction b-band
I want information N.4: Velocity

And then click on this button

Fig. 4. CASAS I/O definition.

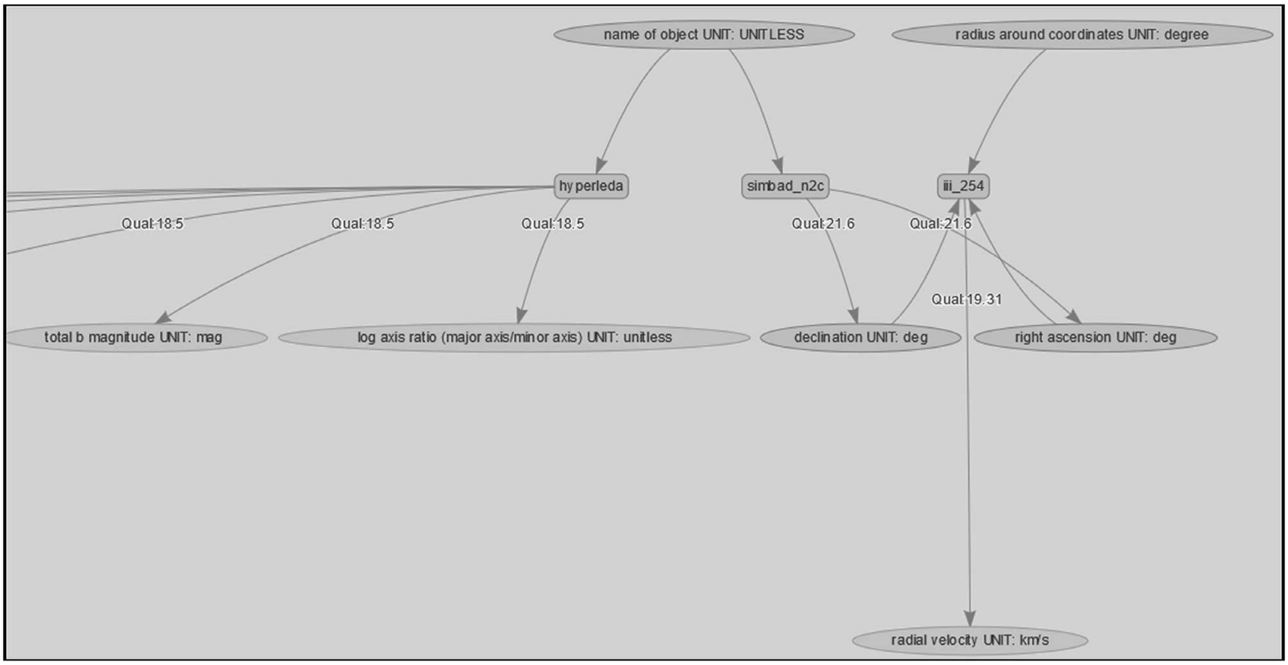


Fig. 5. Extract of a workflow displayed in CASAS.

The first available example illustrates the use of analytical services into CASAS approach by using Aladin field chart generation and interactive skyview together with the query for heliocentric radial velocity and effective temperature for the star SIRIUS.

The second one illustrates the use of non-VO data provider (for magnetic field detection) together with VO services.

The third one illustrates how spectra can be queried in CASAS, taking the wavelength into account during the services selection.

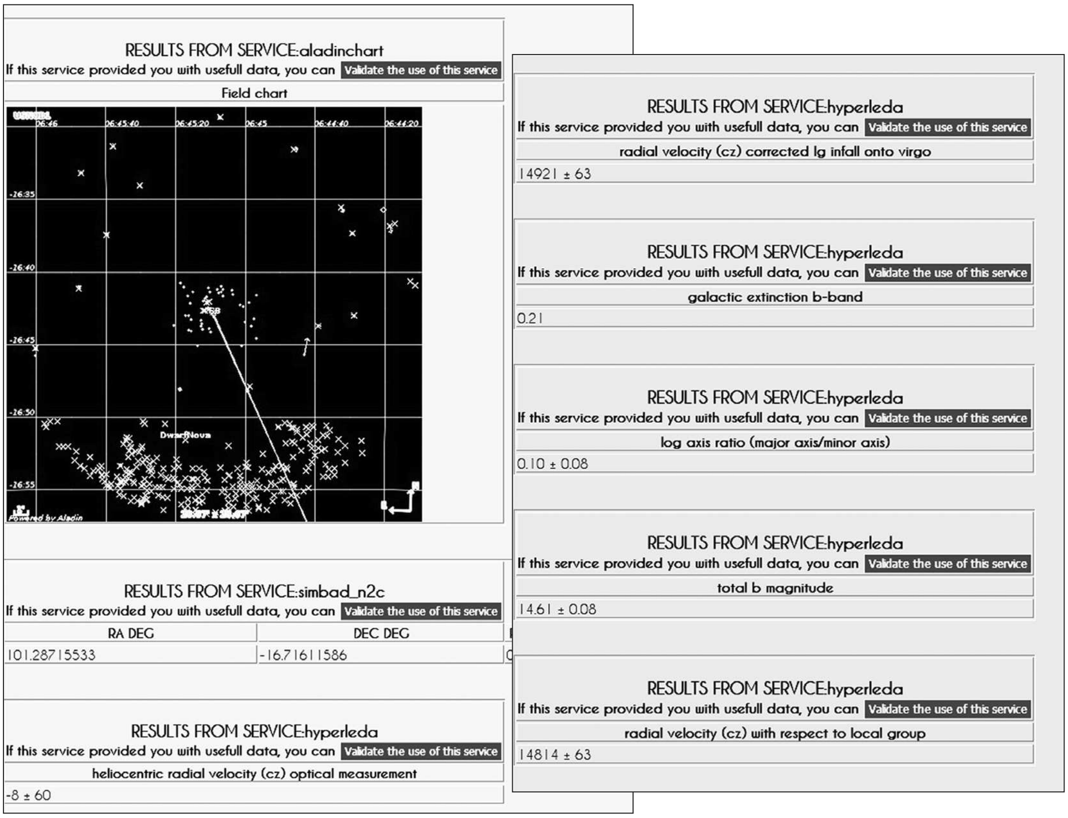
The last example, labeled "Galactic example 2" in the interface, mimics a workflow taken from Taverna.[13] The original workflow extracts five different information from HyperLeda database, given a list of galaxies. CASAS counterpart extracts four out of the original five information either from HyperLeda or other relevant services, for one galaxy at a time.

5. CASAS vs. Taverna: discussion

In their paper, Lemos et al. (2016) propose a framework for composition approaches characterization, and also state that approaches dealing with scientific workflows and approaches using semantic web services composition are rare. Taverna, which is characterized in Lemos et al. (2016) is one of the approaches dealing with scientific workflows. AstroTaverna (Ruiz et al., 2014),

---

[13] http://www.myexperiment.org/workflows/2614.html.



RESULTS FROM SERVICE:aladinchart

If this service provided you with usefull data, you can

RESULTS FROM SERVICE:simbad_n2c

If this service provided you with usefull data, you can

RA DEG	DEC DEG
101.28715533	-16.71611586

RESULTS FROM SERVICE:hyperleda

If this service provided you with usefull data, you can

heliocentric radial velocity (cz) optical measurement

-8 ± 60

RESULTS FROM SERVICE:hyperleda

If this service provided you with usefull data, you can

radial velocity (cz) corrected lg infall onto virgo

14921 ± 63

RESULTS FROM SERVICE:hyperleda

If this service provided you with usefull data, you can

galactic extinction b-band

0.21

RESULTS FROM SERVICE:hyperleda

If this service provided you with usefull data, you can

log axis ratio (major axis/minor axis)

0.10 ± 0.08

RESULTS FROM SERVICE:hyperleda

If this service provided you with usefull data, you can

total b magnitude

14.61 ± 0.08

RESULTS FROM SERVICE:hyperleda

If this service provided you with usefull data, you can

radial velocity (cz) with respect to local group

14814 ± 63

Fig. 6. Extract of results displayed in CASAS.

based on Taverna, is a widely-used tool for workflow composition in astrophysics. It allows the composition and execution of workflows for astrophysical services. This is the reason why CASAS is compared to Taverna in this section. The comparison is done using the framework proposed by Lemos et al. (2016). Characterization of CASAS is exposed in the following tables. The characterization of Taverna comes from the survey itself.

Table 1 exposes the Language and target user dimensions of the framework. The most relevant differences are that while Taverna stands for the most widely-used components (SOAP/REST, JSON/XML etc.), CASAS brings the semantic web services technologies in the field of astrophysics.

The target user is also different, as when Taverna targets the end-user programmers, CASAS is directed towards the end-user without any required technical background knowledge (“end-users app remixers”, in the language of the framework). CASAS may also describe any web service, and any non-web service properly described in ASON. As an example, the screenshots in Fig. 6 contain the results for the “Galactic example2” example that mimics Taverna workflow. It also displays an image obtained by the automatic call of Aladin routine through CASAS. This may also include non-scientific services, explaining why the “target app” is also different. Some other, less crucial differences exist especially in dataflow management. When Taverna relies on pieces of dedicated software to handle units management, in CASAS this management is done by the means of regular services. As a consequence, units conversion in CASAS will be achieved by the means of services registered inside the system and automatically included in the workflow, when needed. No service specifically tailored for units conversion is included in the first CASAS version. Nevertheless, the use of such services has been taken into account in the design and does not differ from the use of any other service in the CASAS composition approach.

The “controlled natural language“ in Table 1 means that the users’ query expressed in natural language is mapped to the ASON content. This allows selection of the most relevant information identified by the query inside the ontology. So that it is natural language, controlled because it is mapped towards an ontological description.

Table 2 presents the knowledge reuse and automation dimensions. When Taverna proposes a complete definition of the reusable workflows or fragments through the MyExperiment website,[14] CASAS does not show its internal reuse of previous compositions. In return, CASAS automates all the reuses and benefits from users feedback when elaborating the services composition.

Table 3 exposes the tool support and execution platform dimensions. Because Taverna uses Business Process Engine it natively brings multithreading, runtime monitoring and some other functionalities that the native code of CASAS does not brings. There is no need for support tools for CASAS, as its goal is to propose the simplest and most comprehensive experience as possible.

Table 4 summarizes the above by assessing AstroTaverna and CASAS on a set of features. The sign “++” designates a criterion that is a part of the platform specific goals, “+” indicates that the criterion is taken into account in the platform, and “−” indicates a criterion for which the platform does not provide any help.

As CASAS is designed to be fully automated, some interaction must be hidden. This happens when analytical services are invoked; the internal parameters are out of range for the end-user and need to be defined once and for all.

Another limitation is that for a given set of objects, a given workflow in CASAS has to be rerun for every object in the set. Besides that, as the values of the input parameters are taken at the very beginning of the services composition, the workflow has to be

---

[14] http://www.myexperiment.org.

Table 1
Language and target user dimensions.

Language							
	Component	Target app.	Notation and paradigm	Composition constructs		Crosscutting concerns	Target user
				Control flow	Data flow and data transfer		
Taverna	Composes data. JAVA, JSON and XML formats. Supports SOAP and REST protocol. Push interactions. Design time selection	Scientific workflows	Visual notation, flow diagrams. Flow-based paradigm	Simple control flow (sequence and exclusive choices)	Dataflow for data exchange, prebuilt processors for data processing. Data transformation via transformation languages	Exceptions (retry and alternative tasks)	End-user programmers (domain experts)
CASAS	Composes data. OWL-S based description. Natively supports VO protocols and REST. Supports every protocol with provided grounding code. Pull interaction. Deployment time selection.	Scientific workflows, mashups.	Visual notation, flow diagrams. Flow-based paradigm. Controlled natural language.	Simple control flow (sequence and alternative choices)	Blackboard. Data transformation services described as regular services.	Exceptions (Alternative tasks).	End-user app remixers

Table 2
The knowledge reuse and automation dimensions.

	Knowledge reuse		
	Reused artifact	Reuse technique	Automation
Taverna	Components, examples, and fragments through their encapsulation as components	Keyword search, copy/paste, repository and forum	Not addressed
CASAS	Fragments, user feedback	File description matchings +criteria selection	Embedded in the composition algorithm

Table 3
Tool support and execution platform dimensions.

	Tool support	Execution platform	
		Deployment option	Execution engine
Taverna	Versioning, manuals, tutorials, and FAQ	On premises and on cloud	Business process engine
CASAS	Not available	On premises and on cloud	Native code

Table 4
Taverna/CASAS comparison.

Criteria	AstroTaverna	CASAS
Overall ease of use	+	++
Ease of services discovery	+	++
Automation of composition	−	++
Quality evaluation during composition	−	++
Workflow documentation	++	−
Workflow reuse with different parameters	++	+
User control on the composition	++	+

re-composed for every object. Other workflow managers (namely, Taverna) authorize a list of targets to be passed and queried as a whole.

CASAS is currently hosted in a virtual machine with limited storage space (20 GB) and RAM size (4 GB) with Intel(R) Xeon(R) CPU E5-2640 0 @ 2.50 GHz. As a result, CASAS composition and execution of large workflows are slow and may last several minutes each. CASAS is currently not compatible with Safari web browser.

6. Conclusion and future work

Exploiting the full potential of astrophysical data and analytical services through services composition is an ongoing challenge. Existing platforms tend to be either very specialized (SGWs), or suffer from the difficulty in discovering relevant services for a specific goal. In all cases, sufficient technical background and a-priori knowledge about services content are necessary to ensure efficient composition.

CASAS brings the semantic web services composition to astrophysics. It proposes fully-automated composition of services based on a semantic description of their capacities and grounding. It eases the description of the workflows outputs, making the automatic selection of services easier than the keyword-based method used by existing tools.

CASAS is designed to provide a complementary tool for existing workflows managers. The semantic web composition proposed by CASAS is an approach that deserves to be tested, as it brings a new method for finding astrophysical services composition. This method embeds any DCI-compliant data provider or analytical service alike. The workflows composition will become more accurate as it is used and the Traceability/Learning Layer (TLL) is filled with user returns and history of successful combination of services.

In future works, integration of a web interface allowing the end-user to provide description of new services is planned. This will automatically bring the new service inside ASON description and allow its selection for future compositions. This future enhancement will come with a list of all services accessible through CASAS, divided into VO and non-VO services. The users will so have a better understanding of whether or not a service is available.

TLL is currently composed of individual files, storing each of the TLL criteria. The overall system will benefit from having those information either integrated back on ASON, or in a side ontology. This will be discussed and integrated into future releases of the system.

ASON ontology has been populated using the VOParis registry. In order to provide automatic access to the higher number of services possible, the next update of this ontology will come from a registry that has record for every resource known in the VO. A list of such registries can be found in the IVOA registry of registries.[15]

The last software update planned for CASAS consists in including services for units conversion in the system, so that services not sharing the same units may be used in conjunction.

The technical architecture enhancement is also on the roadmap, and along with those changes the user will gain the ability to give more inputs than the object name and a radius. Specifying the units desired for any output value, together with the ability to pass a list of objects rather than an object alone, will also appear. Bringing the results in a VO-table description is on the planning, and the architectural enhancement would also give CASAS the opportunity to return more outputs than the 4 ones currently available.

The relevancy of bringing CASAS into a more global scope in astrophysics will be investigated. This scope would consist in the composition of wide astrophysics domain ontology able to do reasoning sustained with expert knowledge. This reasoning would incorporate CASAS for finding relevant information or analysis when missing. This may set the stone for a real decision support system for astrophysicists.

References

Boch, T., Oberto, A., Fernique, P., Bonnarel, F., 2011. Aladin: An open source all-sky browser. In: Astronomical Data Analysis Software and Systems XX. ASP Conference Proceedings, vol. 442, Boston, Massachusetts, USA, 2011.

Castelli, G., Taffoni, G., Sciacca, E., Becciani, U., Costa, A., Krokos, M., Pasian, F., Vuerli, C., 2015. VO-compliant workflows and science gateways. Astron. Comput. 11, 102–108.

Costa, A., Massimino, P., Bandieramonte, M., Becciani, U., Krokos, M., Pistagna, C., Riggi, S., Sciacca, E., Vitello, F., 2015. An innovative science gateway for the cherenkov telescope array. J. Grid Comput. 13, 547–559.

Dubernet, M.-L., Antony, B.K., Ba, Y.-A., Babikov, Y.L., Bartschat, K., Boudon, V., Braams, B.J., Chung, H.-K., Daniel, F., Delahaye, F., et al., 2016. The virtual atomic and molecular data centre (VAMDC) consortium. J. Phys. B: At. Mol. Opt. Phys. 49, 074003. http://www.vamdc.eu.

Kacsuk, P., Farkas, Z., Kozlovszky, M., Hermann, G., Balasko, A., Karoczkai, K., Marton, I., 2012. WS-PGRADE/gUSE generic DCI gateway framework for a large variety of user communities. J. Grid Comput. 10, 601–630.

Lèbre, A., Palacios, A., Sanguillon, P., Maeght, M., 2012. Automatic comparison between observed and computed stellar spectra with tools and protocols from the virtual observatory. In: SF2A-2012: Proceedings of the Annual meeting of the French Society of Astronomy and Astrophysics.

Lemos, A.L., Daniel, F., Benatallah, B., 2016. Web service composition: a survey of techniques and tools. ACM Comput. Surv. 48, 33.

Louge, T., Karray, M.H., Archimède, B., Knödlseder, J., 2015. Semantic interoperability in astrophysics for workflows extraction from heterogeneous services. In: International IFIP Working Conference on Enterprise Interoperability.

Louge, T., Karray, M.H., Archimède, B., Knödlseder, J., 2017. ASON: An OWL-S based ontology for astrophysical services. Astron. Comput. (submitted for publication) December 2016, under review Ref. ASCOM_2016_77.

Makarov, D., Prugniel, P., Terekhova, N., Courtois, H., Vauglin, I., 2014. HyperLEDA. III. The catalogue of extragalactic distances. Astron. Astrophys. 570, A13.

Puttonen, J., Lobov, A, Lastra, J.L.M., 2013. Semantics-based composition of factory automation processes encapsulated by web services. IEEE Trans. Ind. Inform. 9, 2349–2359.

Ruiz, J.E., Garrido, J., Santander-Vela, J.D., Sánchez-Expósito, S., Verdes-Montenegro, L., 2014. AstroTaverna—Building workflows with virtual observatory services. Astron. Comput. 7, 3–11.

Schaaff, A., 2007. Data in astronomy: From the pipeline to the virtual observatory. In: International Conference on Web Information Systems Engineering, 2007.

Taylor, M., 2011. TOPCAT: tool for operations on catalogues and tables. Astrophys. Source Code Libr. 1.

---

[15] http://rofr.ivoa.net/.